ABOV SEMICONDUCTOR Co., Ltd.
8-BIT MICROCONTROLLERS

MC96F6432/F6332/F6232

User's Manual (Ver. 1.0)

REVISION HISTORY

VERSION 0.0 (January 14, 2011)

VERSION 1.0 (June 27, 2011) This book

Change ‘Typ/Max’ to “LVR/LVI level” in LVR/LVI electrical characteristics.

Change ‘600/1200/2000 kΩ (Min/Typ/Max)’ to “RX1” in DC electrical characteristics.

Change ‘3/6/9 kHz (Min/Typ/Max)’ to “f_{WDTRC}” in INTERNAL WATCH-DOG RC OSCILLATION characteristics.

Remove WDTRC Current Max value at INTERNAL WATCH-DOG RC OSCILLATION characteristics.

Add ‘1.0 μA (Typ)’ to “I_{WDTRC}” in INTERNAL WATCH-DOG RC OSCILLATION characteristics.

Add NOTE at ‘T4CK[3:0] bits description’.

Change ‘cleared by software(S/W)’ in TIMER 3/4 BlOCK DIAGRAM.

Deleted AN15 Function in 12-bit A/D converter.

Added VDD18 Function in 12-bit A/D converter

Change ‘2500/5000/10000 kΩ (Min/Typ/Max)’ to “RX2” in DC electrical characteristics.

Change ‘8.0/12.0 uA (Typ/Max) to “Current consumption(one of two)” in LVR/LVI characteristics.

Change ‘10.0/15.0 uA (Typ/Max) to “Current consumption(both)” in LVR/LVI characteristics.

Change ‘±3 (Max)’ to “ILE” in ADC characteristics.

EXTRF/LVRF initial value changed ‘0’ to ‘unknown’.

Add NOTE 3 at LVRF description.

Version 1.0
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

MC96F6432/F6332/F6232

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

1. Overview

1.1 Description

The MC96F6432/F6332/F6232 is advanced CMOS 8-bit microcontroller with 32k 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 : 32k bytes of FLASH, 256 bytes of IRAM, 768 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, 10-bit PWM output, watch timer, buzzer driving port, SPI, USI, 12-bit A/D converter, LCD driver, on-chip POR, LVR, LVI, on-chip oscillator and clock circuitry. The MC96F6432/F6332/F6232 also supports power saving modes to reduce power consumption.

Device Name	FLASH	XRAM	IRAM	ADC	I/O PORT	Package
MC96F6432Q	32k bytes	768 bytes	256 bytes	15 channel	42	44 MQFP
MC96F6332D				11 channel	30	32 SOP
MC96F6232M				10 channel	26	28 SOP

1.2 Features

- **CPU**
  - 8 Bit CISC Core (8051 Compatible)
- **ROM (FLASH) Capacity**
  - 32k Bytes
  - Flash with self read/write capability
  - On chip debug and In-system programming (ISP)
  - Endurance : 100,000 times
- **256 Bytes IRAM**
- **768 Bytes XRAM**
  - (27 Bytes including LCD display RAM)
- **General Purpose I/O (GPIO)**
  - Normal I/O : 9 Ports
    (P0[2:0], P5[5:0])
  - LCD shared I/O : 33 Ports
    (P0[7:3], P1, P2, P3, P4)
- **Basic Interval Timer (BIT)**
  - 8Bit × 1ch
  - **Watch Dog Timer (WDT)**
  - 8Bit × 1ch
  - 6kHz internal RC oscillator
- **Timer/ Counter**
  - 8Bit × 1ch (T0), 16Bit × 2ch (T1/T2)
  - 8Bit × 2ch (T3/T4) or 16 Bit × 1ch (T3)
- **Programmable Pulse Generation**
  - Pulse generation (by T1/T2)
  - 8Bit PWM (by T0)
  - 6-ch 10Bit PWM for Motor (by T4)
- **Watch Timer (WT)**
  - 3.91mS/0.25S/0.5S/1S/1M interval at 32.768kHz
- **Buzzer**
  - 8Bit × 1ch
- **SPI 2**
  - 8Bit × 1ch
- **USI0/1 (UART + SPI + I2C)**
  - 8Bit UART × 2ch, 8Bit SPI × 2ch and I2C × 2ch
- **12 Bit A/D Converter**
  - 15 Input channels
- **LCD Driver**
  - 21 Segments and 8 Common terminals
  - Internal or external resistor bias
  - 1/2, 1/3, 1/4, 1/5, 1/6 and 1/8 duty selectable
  - Resistor Bias and 16-step contrast control
- **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
    (EXINT0~7, EINT8, EINT10, EINT11, EINT12) (12)
  - Timer(0/1/2/3/4) (5)
  - WDT (1)
  - BIT (1)
  - WT (1)
  - SPI 2 (1)
  - USI0/1 (6)
  - ADC (1)
- **Internal RC Oscillator**
  - Inernal RC frequency: 16MHz ±1% (T_A= -25°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 ~ 10.0MHz with X-tal)
  - 3.0V ~ 5.5V (@0.4 ~ 12.0MHz with X-tal)
  - 1.8V ~ 5.5V (@0.5 ~ 16.0MHz with Internal RC)
  - Voltage dropout converter included for core
- **Minimum Instruction Execution Time**
  - 125nS (@ 16MHz main clock)
  - 61μS (@t 32.768kHz sub clock)
- **Operating Temperature:** – 40 ~ + 85℃
- **Oscillator Type**
  - 0.4-12MHz Crystal or Ceramic for main clock
  - 32.768kHz Crystal for sub clock
- **Package Type**
  - 44 MQFP-1010
  - 32 SOP
  - 28 SOP
  - Pb-free package

1.3 Ordering Information

Table 1-1 Ordering Information of MC96F6432/F6332/F6232

Device name	ROM size	IRAM size	XRAM size	Package
MC96F6432Q	32k bytes FLASH	256 bytes	768 bytes	44 MQFP
MC96F6332D				32 SOP
MC96F6232M				28 SOP

1.4 Development Tools

1.4.1 Compiler

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

The core of MC96F6432/F6332/F6232 is Mentor 8051. And, device ROM size is smaller than 32k bytes. Developer can use all kinds of third party's standard 8051 compiler.

1.4.2 OCD emulator and debugger

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

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

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

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

Connection:

- SCLK (MC96F6432/F6332/F6232 P01 port)
- SDATA (MC96F6432/F6332/F6232 P00 port)

OCD connector diagram: Connect OCD with user system

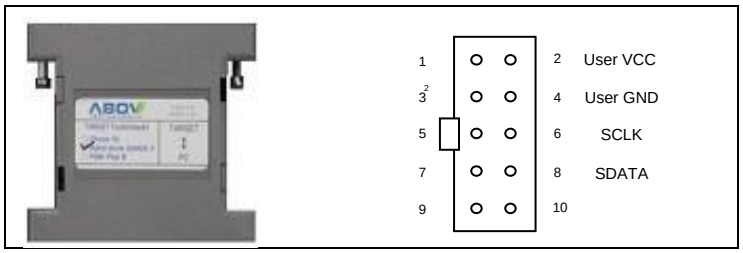


Figure 1.1 OCD Debugger and Pin Description

1.4.3 Programmer

Single programmer:

PGMplus USB: It programs MCU device directly.

Figure 1.2 PGMplusUSB (Single Writer)

StandAlone PGMplus: It programs MCU device directly.

Figure 1.3 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 1.4 StandAlone Gang8 (for Mass Production)

2. Block Diagram

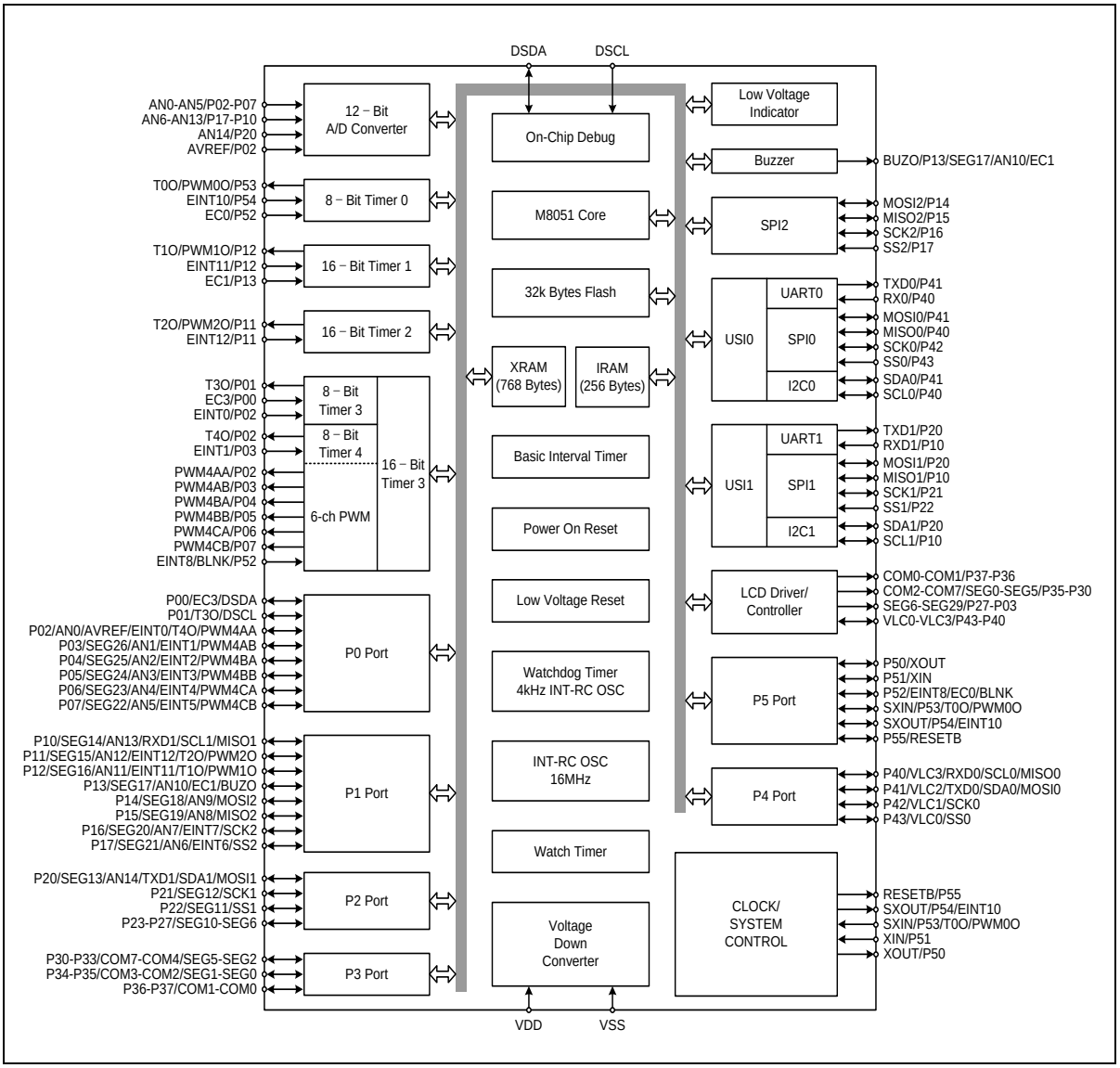


Figure 2.1 Block Diagram

NOTES) 1. The P14–P17, P23–P25, P34–P37, and P43 are not in the 32-pin package.
2. The P13–P17, P22–P27, P34–P37, and P43 are not in the 28-pin package.

3. Pin Assignment

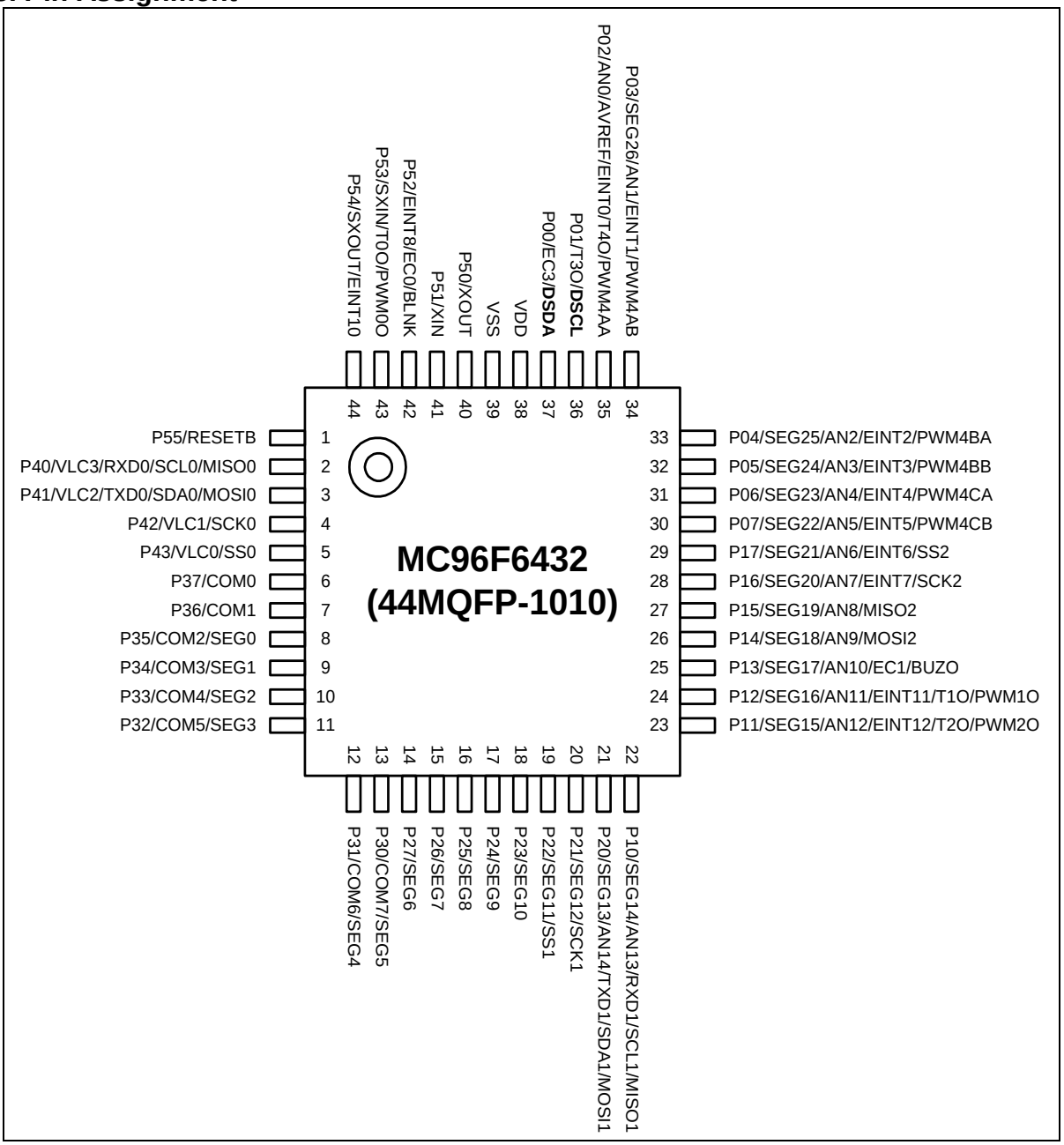


Figure 3.1 MC96F6432 44MQFP-1010 Pin Assignment

NOTE) On On-Chip Debugging, ISP uses P0[1:0] pin as DSDA, DSCL.

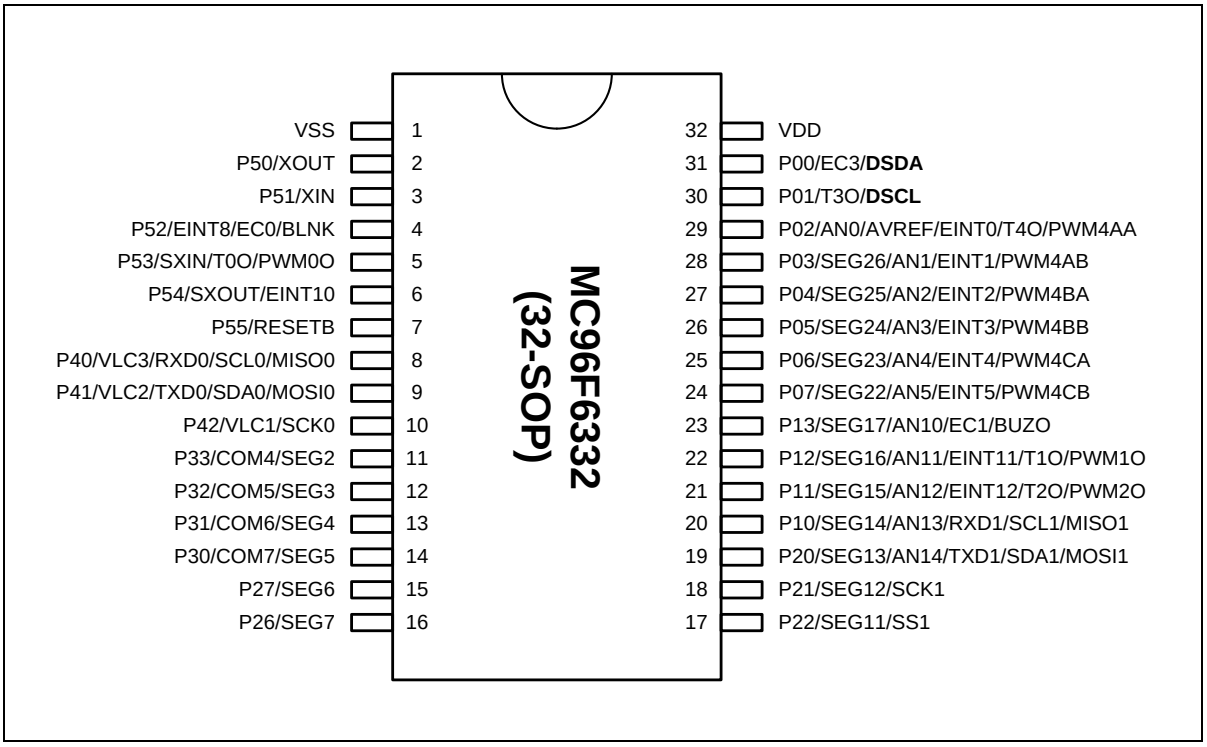


Figure 3.2 MC96F6332 32SOP Pin Assignment

NOTES) 1. On On-Chip Debugging, ISP uses P0[1:0] pin as DSDA, DSCL.
2. The P14-P17, P23-P25, P34-P37 and P43 pins should be selected as a push-pull output or an input with pull-up resistor by software control when the 32-pin package is used.

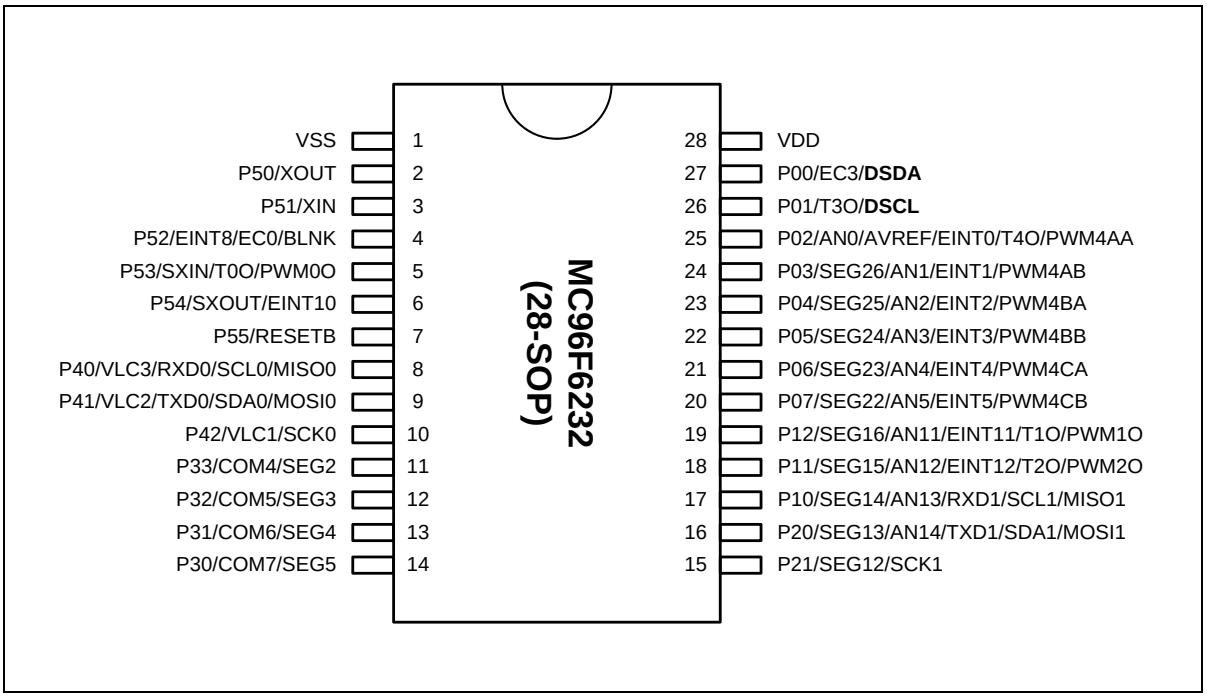


Figure 3.3 MC96F6232 28SOP Pin Assignment

NOTES) 1. On On-Chip Debugging, ISP uses P0[1:0] pin as DSDA, DSCL.
2. The P13-P17, P22-P27, P34-P37 and P43 pins should be selected as a push-pull output or an input with pull-up resistor by software control when the 32-pin package is used.

4. Package Diagram

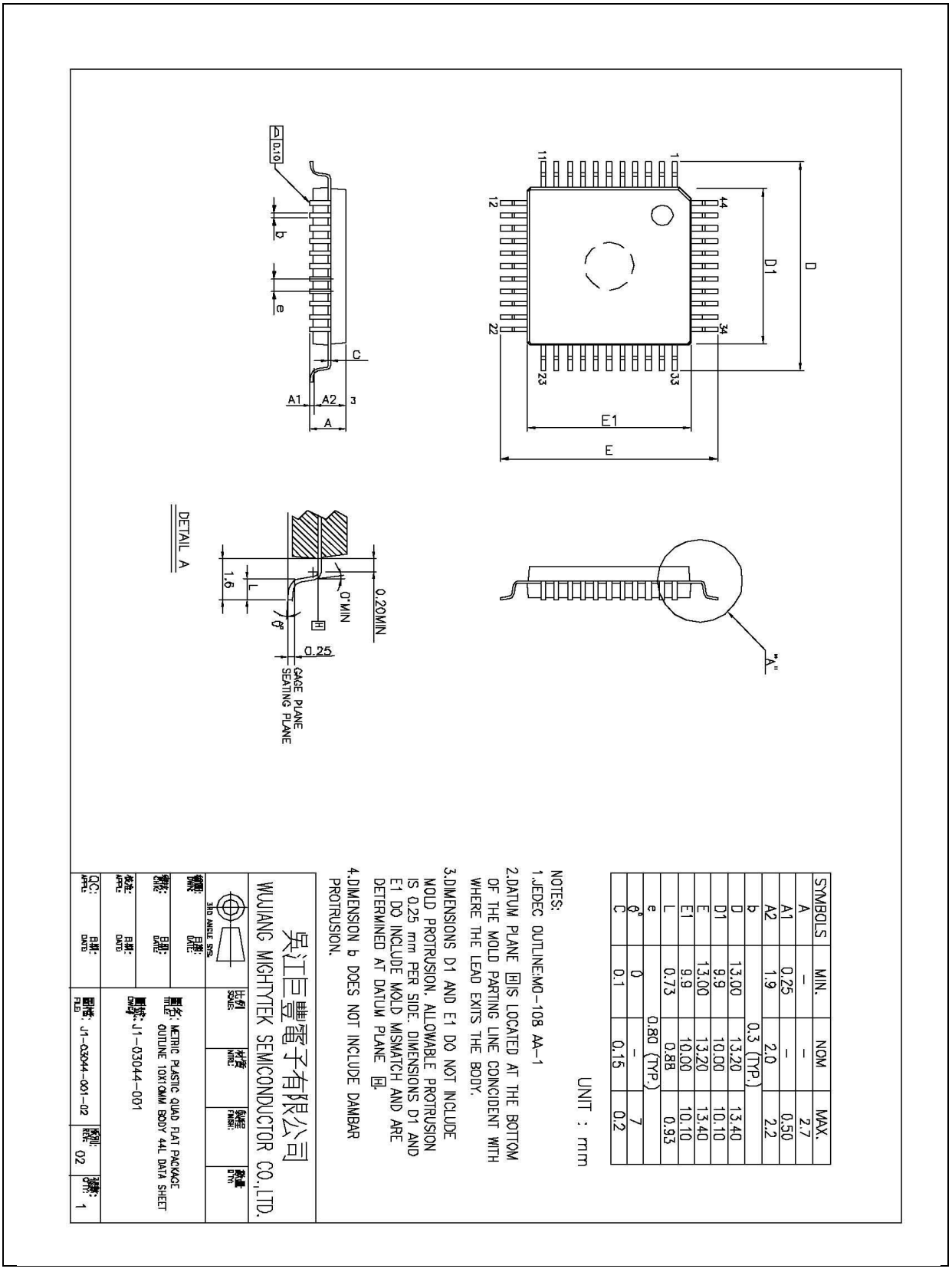


Figure 4.1 44-Pin MQFP Package

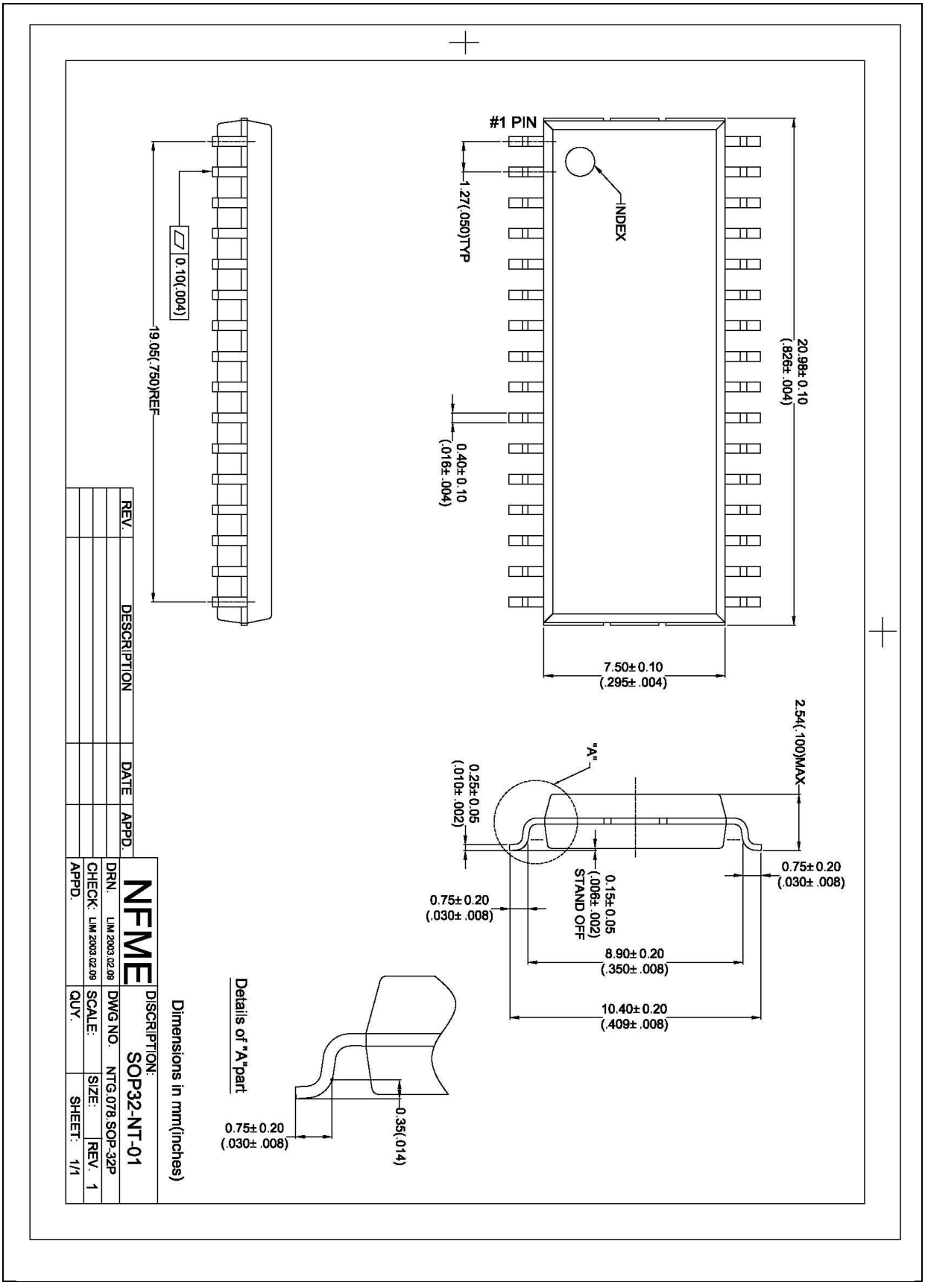


Figure 4.2 32-Pin SOP Package

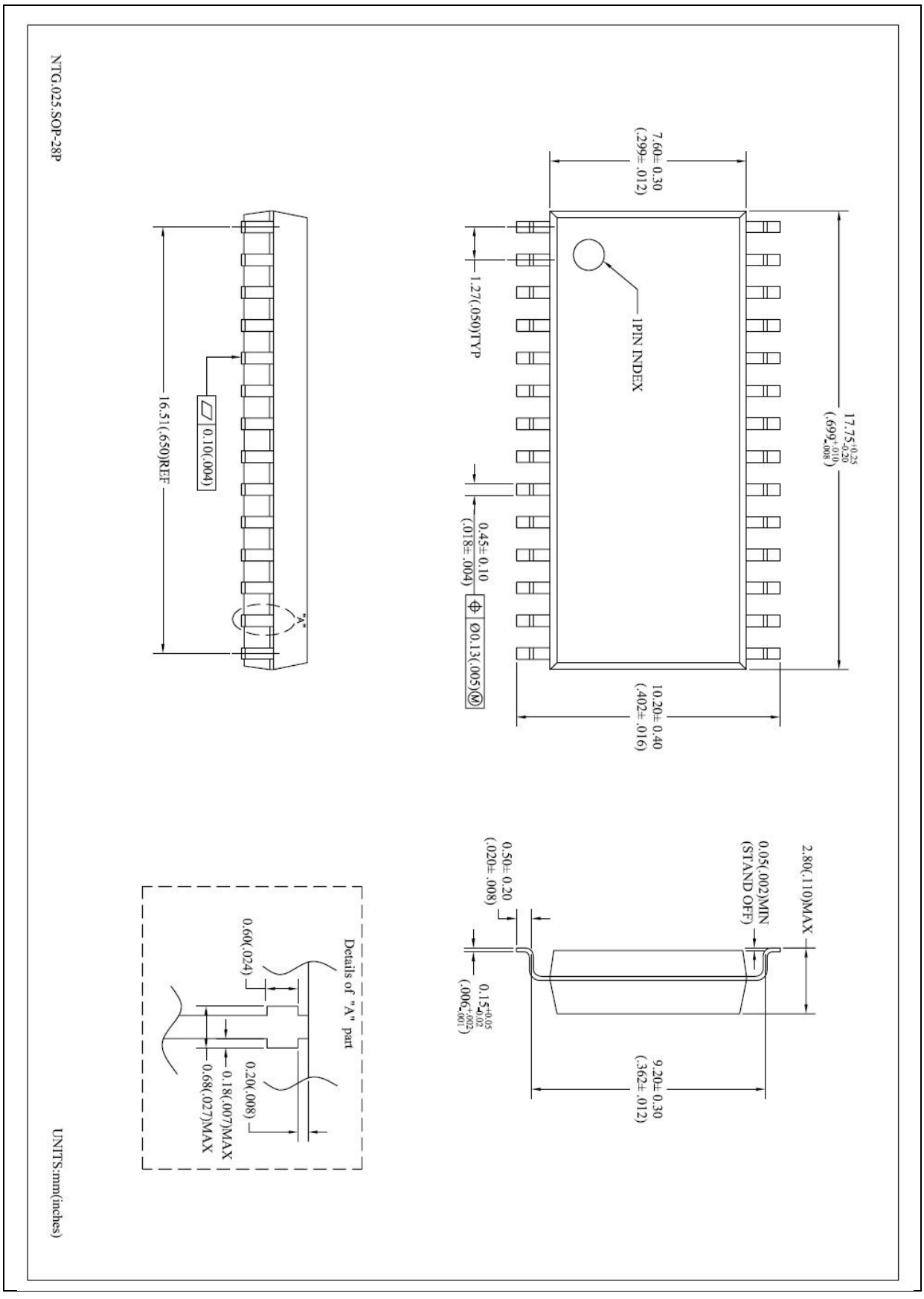


Figure 4.3 28-Pin SOP Package

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 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	EC3/DSDA
P01				T3O/DSCL
P02				AN0/AVREF/EINT0/T4O/PWM4AA
P03				SEG26/AN1/EINT1/PWM4AB
P04				SEG25/AN2/EINT2/PWM4BA
P05				SEG24/AN3/EINT3/PWM4BB
P06				SEG23/AN4/EINT4/PWM4CA
P07				SEG22/AN5/EINT5/PWM4CB
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. The P14 – P17 are not in the 32-pin package. The P13 – P17 are not in the 28-pin package.	Input	SEG14/AN13/RXD1/SCL1/MISO1
P11				SEG15/AN12/EINT12/T2O/PWM2O
P12				SEG16/AN11/EINT11/T1O/PWM1O
P13				SEG17/AN10/EC1/BUZO
P14				SEG18/AN9/MOSI2
P15				SEG19/AN8/MISO2
P16				SEG20/AN7/EINT7/SCK2
P17				SEG21/AN6/EINT6/SS2
P20	I/O	Port 2 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 P23 – P25 are not in the 32-pin package. The P22 – P27 are not in the 28-pin package.	Input	SEG13/AN14/TXD1/SDA1/MOSI1
P21				SEG12/SCK1
P22				SEG11/SS1
P23				SEG10
P24				SEG9
P25				SEG8
P26				SEG7
P27				SEG6
P30	I/O	Port 3 is a bit-programmable I/O port which can be configured as an input, a push-pull output. A pull-up resistor can be specified in 1-bit unit. The P34 – P37 are only in the 44-pin package.	Input	COM7/SEG5
P31				COM6/SEG4
P32				COM5/SEG3
P33				COM4/SEG2
P34				COM3/SEG1
P35				COM2/SEG0
P36				COM1
P37				COM0
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. The P43 is only in the 44-pin package.	Input	VLC3/RXD0/SCL0/MISO0
P41				VLC2/TXD0/SDA0/MOSI0
P42				VLC1/SCK0
P43				VLC0/SS0

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 or a push-pull output. A pull-up resistor can be specified in 1-bit unit.	Input	XOUT
P51				XIN
P52				EINT8/EC0/BLNK
P53				SXIN/T0O/PWM0O
P54				SXOUT/EINT10
P55				RESETB
EINT0	I/O	External interrupt input and Timer 3 capture input	Input	P02/AN0/AVREF/T4O/PWM4AA
EINT1	I/O	External interrupt input and Timer 4 capture input	Input	P03/SEG26/AN1/PWM4AB
EINT2	I/O	External interrupt inputs	Input	P04/SEG25/AN2/PWM4BA
EINT3				P05/SEG24/AN3/PWM4BB
EINT4				P06/SEG23/AN4/PWM4CA
EINT5				P07/SEG22/AN5/PWM4CB
EINT6				P17/SEG21/AN6/SS2
EINT7				P16/SEG20/AN7/SCK2
EINT8				P52/EC0/BLNK
EINT10	I/O	External interrupt input and Timer 0 capture input	Input	P54/SXOUT
EINT11	I/O	External interrupt input and Timer 1 capture input	Input	P12/SEG16/AN11/T1O/PWM1O
EINT12	I/O	External interrupt input and Timer 2 capture input	Input	P11/SEG15/AN12/T2O/PWM2O
T0O	I/O	Timer 0 interval output	Input	P53/SXIN/PWM0O
T1O	I/O	Timer 1 interval output	Input	P12/SEG16/AN11/EINT11/PWM1O
T2O	I/O	Timer 2 interval output	Input	P11/SEG15/AN12/EINT12/PWM2O
T3O	I/O	Timer 3 interval output	Input	P01/DSCL
T4O	I/O	Timer 4 interval output	Input	P02/AN0/AVREF/EINT0/PWM4AA
PWM0O	I/O	Timer 0 PWM output	Input	P53/SXIN/T0O
PWM1O	I/O	Timer 1 PWM output	Input	P12/SEG16/AN11/EINT11/T1O
PWM2O	I/O	Timer 2 PWM output	Input	P11/SEG15/AN12/EINT12/T2O
PWM4AA	I/O	Timer 4 PWM outputs	Input	P02/AN0/AVREF/EINT0/T4O
PWM4AB				P03/SEG26/AN1/EINT1
PWM4BA				P04/SEG25/AN2/EINT2
PWM4BB				P05/SEG24/AN3/EINT3
PWM4CA				P06/SEG23/AN4/EINT4
PWM4CB				P07/SEG22/AN5/EINT5
BLNK	I/O	External sync signal input for 6-ch PWMs	Input	P52/EINT8/EC0
EC0	I/O	Timer 0 event count input	Input	P52/EINT8/BLNK
EC1	I/O	Timer 1 event count input	Input	P13/SEG17/AN10
EC3	I/O	Timer 3 event count input	Input	P00/DSDA

Table 5-1 Normal Pin Description (Continued)

PIN Name	I/O	Function	@RESET	Shared with
BUZO	I/O	Buzzer signal output	Input	P13/SEG17/AN10/EC1
SCK0	I/O	Serial 0 clock input/output	Input	P42/VLC1
SCK1	I/O	Serial 1 clock input/output	Input	P21/SEG12
SCK2	I/O	Serial 2 clock input/output	Input	P16/SEG20/AN7/EINT7
MOSI0	I/O	SPI 0 master output, slave input	Input	P41/VLC2/TXD0/SDA0
MOSI1	I/O	SPI 1 master output, slave input	Input	P20/SEG13/AN14/TXD1/SDA1
MOSI2	I/O	SPI 2 master output, slave input	Input	P14/SEG18/AN9
MISO0	I/O	SPI 0 master input, slave output	Input	P40/VLC3/RXD0/SCL0
MISO1	I/O	SPI 1 master input, slave output	Input	P10/SEG14/AN13/RXD1/SCL1
MISO2	I/O	SPI 2 master input, slave output	Input	P15/SEG19/AN8
SS0	I/O	SPI 0 slave select input	Input	P43/VLC0
SS1	I/O	SPI 1 slave select input	Input	P22/SEG11
SS2	I/O	SPI 2 slave select input	Input	P17/SEG21/AN6/EINT6
TXD0	I/O	UART 0 data output	Input	P41/VLC2/SDA0/MOSI0
TXD1	I/O	UART 1 data output	Input	P20/SEG13/AN14/SDA1/MOSI1
RXD0	I/O	UART 0 data input	Input	P40/VLC3/SCL0/MISO0
RXD1	I/O	UART 1 data input	Input	P10/SEG14/AN13/SCL1/MISO1
SCL0	I/O	I2C 0 clock input/output	Input	P40/VLC3/RXD0/MISO0
SCL1	I/O	I2C 1 clock input/output	Input	P10/SEG14/AN13/RXD1/MISO1
SDA0	I/O	I2C 0 data input/output	Input	P41/VLC2/TXD0/MOSI0
SDA1	I/O	I2C 1 data input/output	Input	P20/SEG13/AN14/TXD1/MOSI1
AVREF	I/O	A/D converter reference voltage	Input	P02/AN0/EINT0/T4O/PWM4AA
AN0	I/O	A/D converter analog input channels	Input	P02/AVREF/EINT0/T4O/PWM4AA
AN1				P03/SEG26/EINT1/PWM4AB
AN2				P04/SEG25/EINT2/PWM4BA
AN3				P05/SEG24/EINT3/PWM4BB
AN4				P06/SEG23/EINT4/PWM4CA
AN5				P07/SEG22/EINT5/PWM4CB
AN6				P17/SEG21/EINT6/SS2
AN7				P16/SEG20/EINT7/SCK2
AN8				P15/SEG19/MISO2
AN9				P14/SEG18/MOSI2
AN10				P13/SEG17/EC1
AN11				P12/SEG16/EINT11/T1O/PWM1O
AN12				P11/SEG15/EINT12/T2O/PWM2O
AN13				P10/SEG14/RXD1/SCL1/MISO1
AN14				P20/SEG13/TXD1/SDA1/MOSI1

Table 5-1 Normal Pin Description (Continued)

PIN Name	I/O	Function	@RESET	Shared with
VLC0	I/O	LCD bias voltage pins	Input	P43/SS0
VLC1				P42/SCK0
VLC2				P41/TXD0/SDA0/MOSI0
VLC3				P40/RXD0/SCL0/MISO0
COM0–COM1	I/O	LCD common signal outputs	Input	P37–P36
COM2–COM3				P35–P34/SEG0–SEG1
COM4–COM7				P33–P30/SEG2–SEG5
SEG0–SEG1	I/O	LCD segment signal outputs	Input	P35–P34/COM2–COM3
SEG2–SEG5				P33–P30/COM4–COM7
SEG6–SEG10				P27–P23
SEG11				P22/SS1
SEG12				P21/SCK1
SEG13				P20/AN14/TXD1/SDA1/MOSI1
SEG14				P10/AN13/RXD1/SCL1/MISO1
SEG15				P11/AN12/EINT12/T2O/PWM2O
SEG16				P12/AN11/EINT11/T1O/PWM1O
SEG17				P13/AN10/EC1
SEG18				P14/AN9/MOSI2
SEG19				P15/AN8/MISO2
SEG20				P16/AN7/EINT7/SCK2
SEG21				P17/AN6/EINT6/SS2
SEG22				P07/AN5/EINT5/PWM4CB
SEG23				P06/AN4/EINT4/PWM4CA
SEG24				P05/AN3/EINT3/PWM4BB
SEG25				P04/AN2/EINT2/PWM4BA
SEG26				P03/AN1/EINT1/PWM4AB

Table 5-1 Normal Pin Description (Continued)

PIN Name	I/O	Function	@RESET	Shared with
RESETB	I/O	System reset pin with a pull-up resistor when it is selected as the RESETB by CONFIGURE OPTION	Input	P55
DSDA	I/O	On chip debugger data input/output (NOTE4,5)	Input	P00/EC3
DSCL	I/O	On chip debugger clock input (NOTE4,5)	Input	P01/T3O
XIN	I/O	Main oscillator pins	Input	P51
XOUT				P50
SXIN	I/O	Sub oscillator pins	Input	P53/T0O/PWM0O
SXOUT				P54/EINT10
VDD, VSS	–	Power input pins	–	–

NOTES) 1. The P14–P17, P23–P25, P34–P37, and P43 are not in the 32-pin package.
2. The P13–P17, P22–P27, P34–P37, and P43 are not in the 28-pin package.pin
3. The P55/RESETB pin is configured as one of the P55 and RESETB pin by the “CONFIGURE OPTION.”
4. If the P00/EC3/DSDA and P01/T3O/DSCL pins are connected to an emulator during the resetor power-on reset, the pins are automatically configured as the debugger pins.
5. The P00/EC3/DSDA and P01/T3O/DSCL pins are configured as inputs with internal pull-up resistor only during the reset or power-on reset.
6. The P50/XOUT, P51/XIN, P53/SXINT/T0O/PWM0O, and P54/SXOUT/EINT10 pins are configured as a function pin by software control.

6. Port Structures

6.1 General Purpose I/O Port

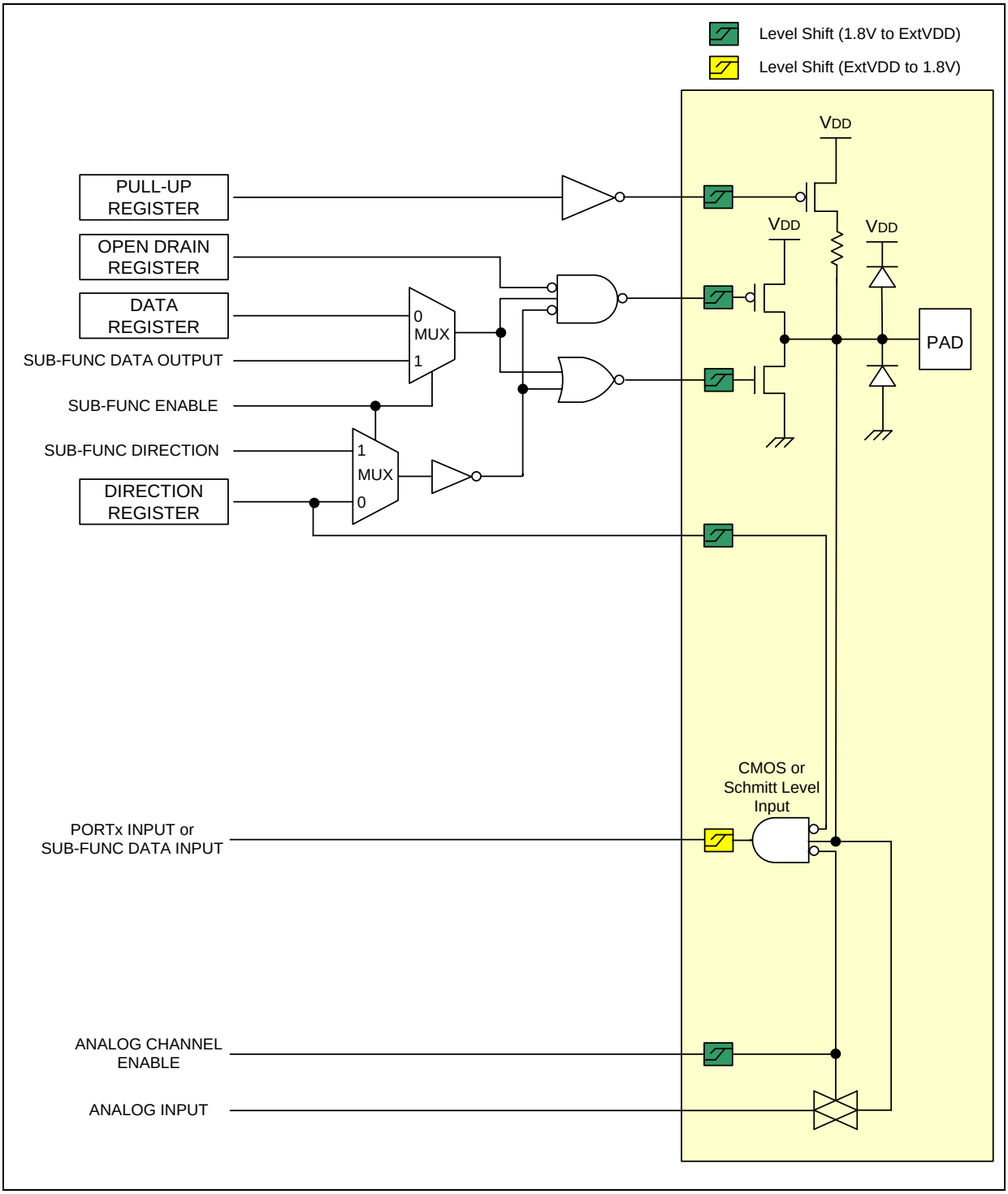


Figure 6.1 General Purpose I/O Port

6.2 External Interrupt I/O Port

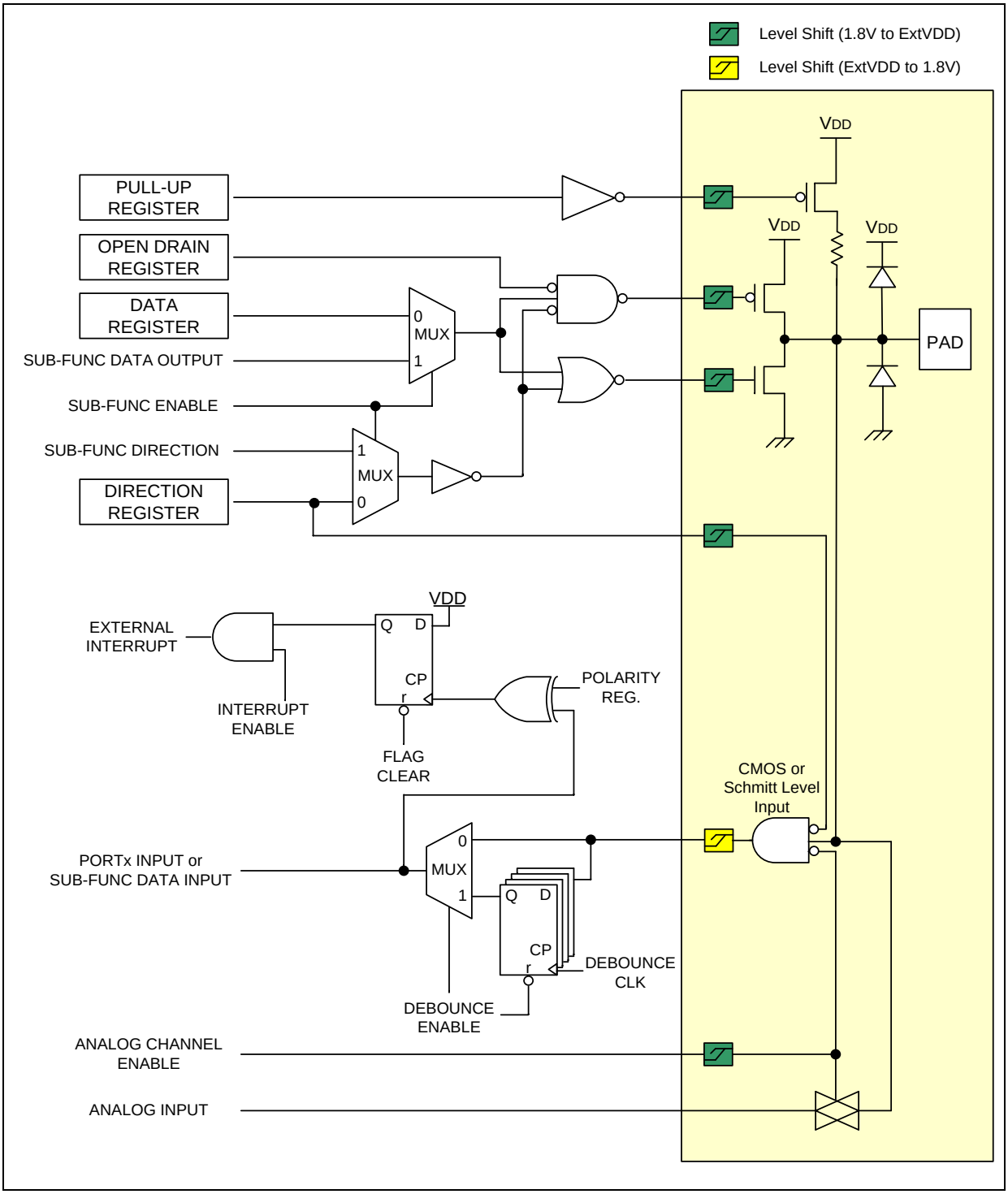


Figure 6.2 External Interrupt I/O Port

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 ~ 10.0MHz		2.7	–	5.5	
		f_X= 0.4 ~ 12.0MHz		3.0	–	5.5	
		f_X= 0.5 ~ 16.0MHz	Internal RC	1.8	–	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		–	–	±3	LSB
Differential Linearity Error	DLE			–	–	±1	
Zero Offset Error	ZOE			–	–	±3	
Full Scale Error	FSE			–	–	±3	
Conversion Time	t_{CON}	12bit resolution, 8MHz		20	–	–	µS
Analog Input Voltage	V_{AN}	–		VSS	–	AVREF	V
Analog Reference Voltage	AVREF	–		1.8	–	VDD	
VDD18	–	–		–	1.8	–	V
Analog Input Leakage Current	I_{AN}	AVREF= 5.12V		–	–	2	µA
ADC Operating Current	I_{ADC}	Enable	VDD= 5.12V	–	1	2	mA
		Disable		–	–	0.1	µA

NOTES) 1. Zero offset error is the difference between 0000000000 and the converted output for zero input voltage (VSS).

2. Full scale error is the difference between 1111111111 and the converted output for full-scale input voltage (AVREF).

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	–	–	V/mS
POR Current	I_{POR}	–	–	0.2	–	µA

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	1.75	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	–		–	10	100	mV
Minimum Pulse Width	t_{LW}	–		100	–	–	μS
LVR and LVI Current	I_{BL}	Enable (Both)	VDD= 3V	–	10.0	15.0	μA
		Enable (One of two)		–	8.0	12.0	
		Disable (Both)		–	–	0.1	

7.6 High 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}	–	–	16	–	MHz
Tolerance	–	T_A = 25°C	–	–	±1	%
		T_A = -40°C to +85°C			±3	
Clock Duty Ratio	TOD	–	40	50	60	%
Stabilization Time	T_{HFS}	–	–	–	100	μS
IRC Current	I_{IRC}	Enable	–	0.2	–	mA
		Disable	–	–	0.1	μA

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}	–	3	6	9	kHz
Stabilization Time	t_{WDTS}	–	–	–	1	mS
WDTRC Current	I_{WDTRC}	Enable	–	1	–	μA
		Disable	–	–	0.1	

7.8 LCD Voltage Characteristics

Table 7-8 LCD Voltage Characteristics

(T_A= -40°C ~ +85°C, VDD= 2.0V ~ 5.5V, VSS= 0V)

Parameter	Symbol	Conditions		MIN	TYP	MAX	Unit
LCD Voltage	V_{LC0}	LCD contrast disabled, 1/4 bias		Typx0.95	VDD	Typx1.05	V
		LCD contrast enabled, 1/4 bias, No panel load	LCDCCR=00H	Typx0.9	VDDx16/31	Typx1.1	V
			LCDCCR=01H		VDDx16/30		
			LCDCCR=02H		VDDx16/29		
			LCDCCR=03H		VDDx16/28		
			LCDCCR=04H		VDDx16/27		
			LCDCCR=05H		VDDx16/26		
			LCDCCR=06H		VDDx16/25		
			LCDCCR=07H		VDDx16/24		
			LCDCCR=08H		VDDx16/23		
			LCDCCR=09H		VDDx16/22		
			LCDCCR=0AH		VDDx16/21		
			LCDCCR=0BH		VDDx16/20		
			LCDCCR=0CH		VDDx16/19		
			LCDCCR=0DH		VDDx16/18		
			LCDCCR=0EH		VDDx16/17		
			LCDCCR=0FH		VDDx16/16		
LCD Mid Bias Voltage(note)	V_{LC1}	VDD=2.7V to 5.5V, LCD clock = 0Hz, 1/4 bias, No panel load		Typx0.9	3/4xVLC0	Typx1.1	V
	V_{LC2}			Typx0.9	2/4xVLC0	Typx1.1	
	V_{LC3}			Typx0.9	1/4xVLC0	Typx1.1	
LCD Driver Output Impedance	R_{LO}	VLCD=3V, ILOAD=±10uA		–	5	10	kΩ
LCD Bias Dividing Resistor	R_{LCD}	T_A = 25°C		40	60	80	

NOTE) It is middle output voltage when the VDD and the V_{LC0} node are connected.

7.9 DC Characteristics

Table 7-9 DC Characteristics

(T_A= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V, VSS= 0V, f_{XIN}= 12MHz)

Parameter	Symbol	Conditions		MIN	TYP	MAX	Unit
Input High Voltage	V_{IH1}	P0, P1, P5, RESETB		0.8VDD	–	VDD	V
	V_{IH2}	All input pins except V_{IH1}		0.7VDD	–	VDD	V
Input Low Voltage	V_{IL1}	P0, P1, P5, RESETB		–	–	0.2VDD	V
	V_{IL2}	All input pins except V_{IL1}		–	–	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}= 10mA; All output ports except V_{OL2}		–	–	1.0	
	V_{OL2}	VDD= 4.5V, I_{OL}= 15mA ; P1		–	–	1.0	V
Input High Leakage Current	I_{IH}	All input ports		–	–	1	µA
Input Low Leakage Current	I_{IL}	All input ports		-1	–	–	µA
Pull-Up Resistor	R_{PU}	VI=0V, T_A= 25°C All Input ports	VDD=5.0V	25	50	100	kΩ
			VDD=3.0V	50	100	200	
		VI=0V, T_A= 25°C RESETB	VDD=5.0V	150	250	400	kΩ
			VDD=3.0V	300	500	700	
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}= 12MHz)

Parameter	Symbol	Condition		MIN	TYP	MAX	Unit
Supply Current	I_{DD1} (RUN)	f_{XIN}= 12MHz, VDD= 5V±10%		–	3.0	6.0	mA
		f_{XIN}= 10MHz, VDD= 3V±10%		–	2.2	4.4	
		f_{IRC}= 16MHz, VDD= 5V±10%		–	3.0	6.0	
	I_{DD2} (IDLE)	f_{XIN}= 12MHz, VDD= 5V±10%		–	2.0	4.0	mA
		f_{XIN}= 10MHz, VDD= 3V±10%		–	1.3	2.6	
		f_{IRC}= 16MHz, VDD= 5V±10%		–	1.5	3.0	
	I_{DD3}	f_{XIN}= 32.768kHz VDD= 3V±10% T_A= 25°C	Sub RUN	–	100.0	180.0	µA
	I_{DD4}		Sub IDLE	–	8.0	16.0	µA
	I_{DD5}	STOP, VDD= 5V±10%, T_A= 25°C		–	0.5	3.0	µA

NOTES) 1. Where the f_{XIN} is an external main oscillator, 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.

7.10 AC Characteristics

Table 7-10 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_{RSL}	Input, VDD= 5V	10	–	–	μS
Interrupt input high, low width	t_{INTH}, t_{INTL}	All interrupt, VDD= 5V	200	–	–	nS
External Counter Input High, Low Pulse Width	tECWH, tECWL	ECn, VDD = 5 V (n= 0, 1, 3)	200	–	–	
External Counter Transition Time	tREC, tFEC	ECn, VDD = 5 V (n= 0, 1, 3)	20	–	–	

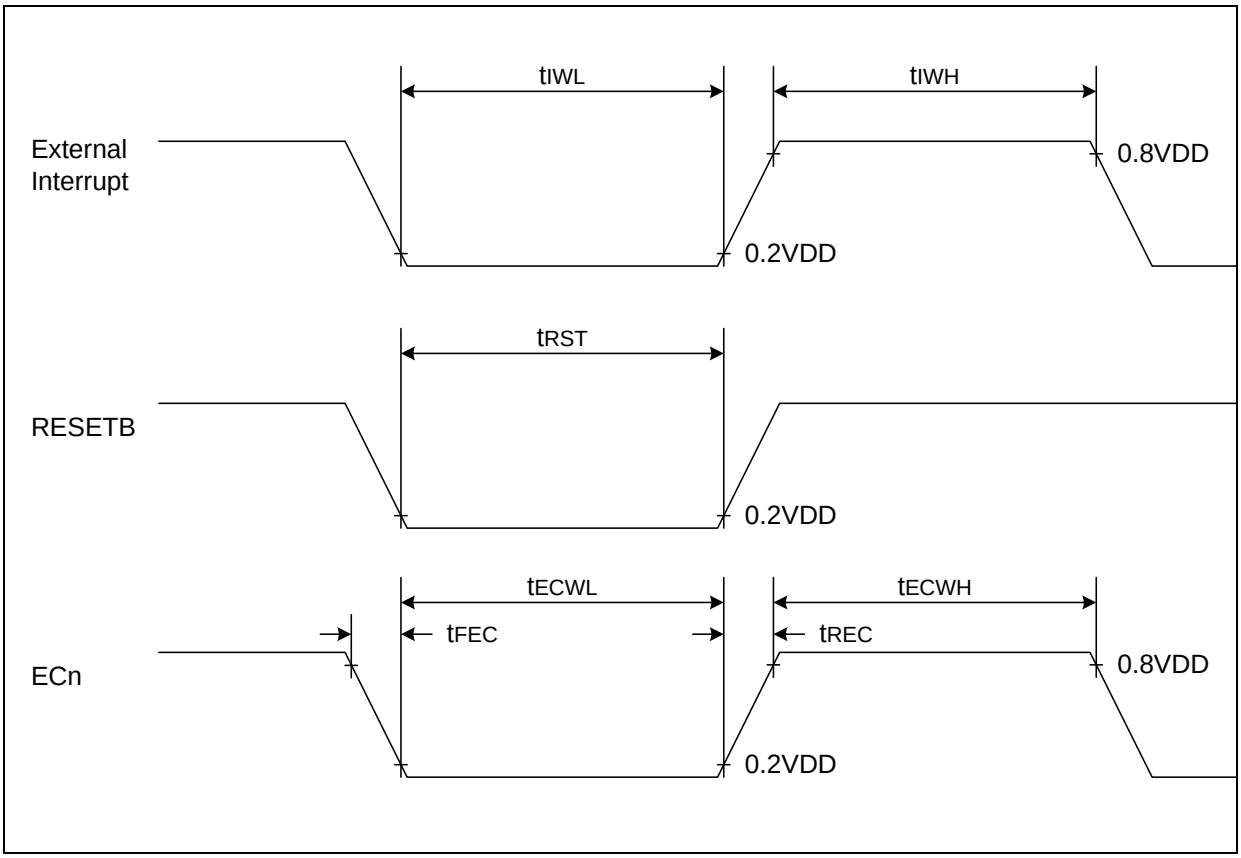


Figure 7.1 AC Timing

7.11 SPI0/1/2 Characteristics

Table 7-11 SPI0/1/2 Characteristics

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

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
Output Clock Pulse Period	tSCK	Internal SCK source	200	–	–	nS
Input Clock Pulse Period		External SCK source	200	–	–	
Output Clock High, Low Pulse Width	tSCKH, tSCKL	Internal SCK source	70	–	–	
Input Clock High, Low Pulse Width		External SCK source	70	–	–	
First Output Clock Delay Time	tFOD	Internal/External SCK source	100	–	–	
Output Clock Delay Time	tDS	–	–	–	50	
Input Setup Time	tDIS	–	100	–	–	
Input Hold Time	tDIH	–	150	–	–	

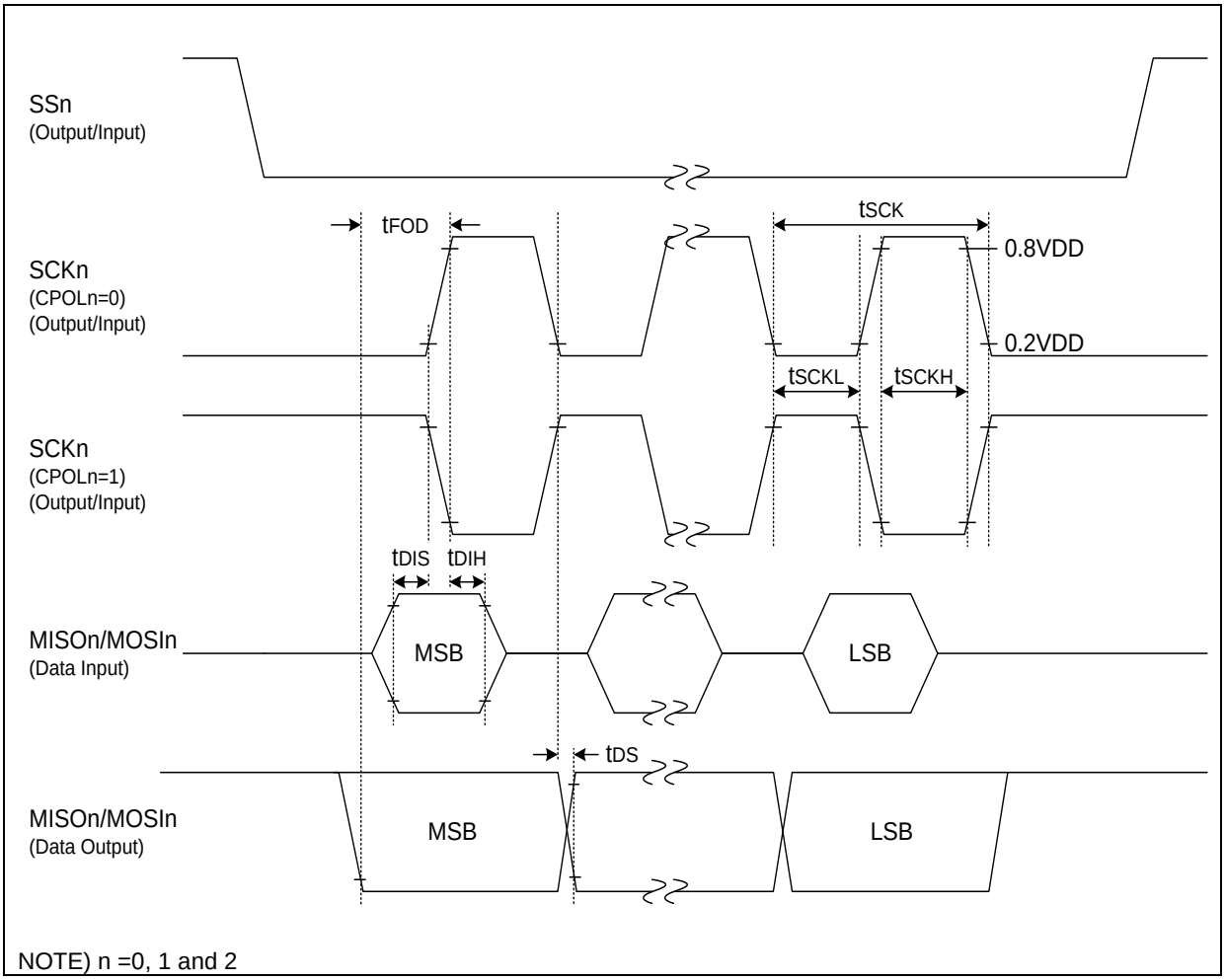


Figure 7.2 SPI0/1/2 Timing

7.12 UART0/1 Characteristics

Table 7-12 UART0/1 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

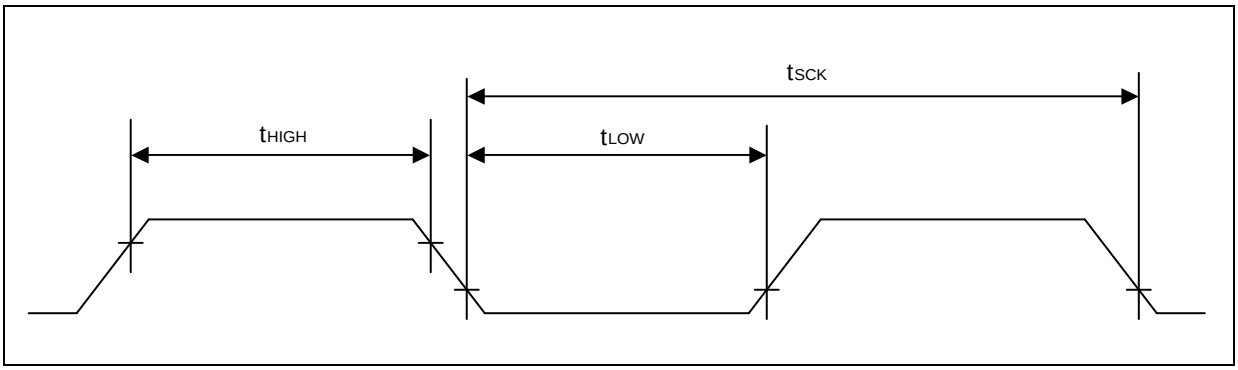


Figure 7.3 Waveform for UART0/1 Timing Characteristics

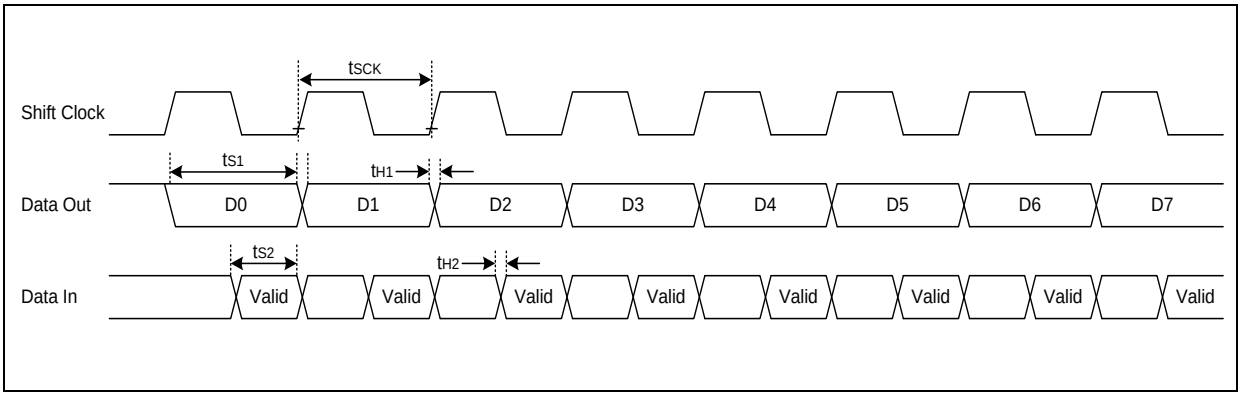


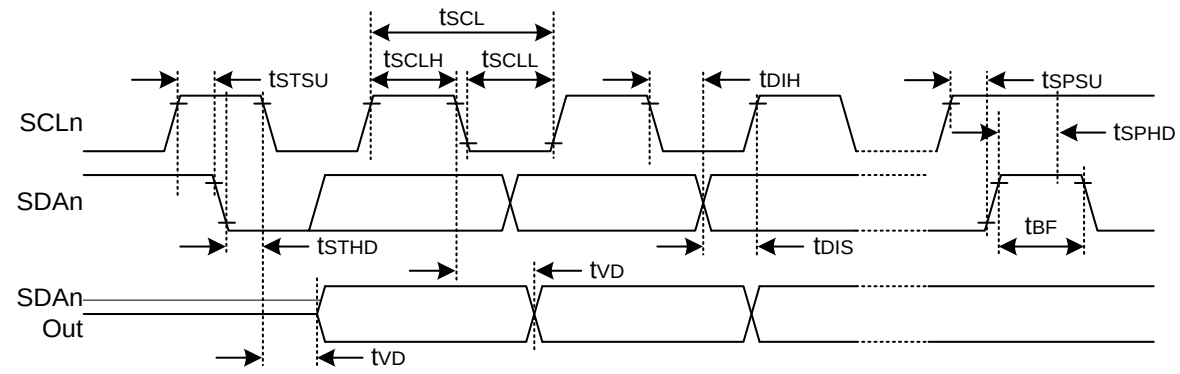
Figure 7.4 Timing Waveform for the UART0/1 Module

7.13 I2C0/1 Characteristics

Table 7-13 I2C0/1 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	tSCL	0	100	0	400	kHz
Clock High Pulse Width	tSCLH	4.0	–	0.6	–	μS
Clock Low Pulse Width	tSCLL	4.7	–	1.3	–	
Bus Free Time	tBF	4.7	–	1.3	–	
Start Condition Setup Time	tSTSU	4.7	–	0.6	–	
Start Condition Hold Time	tSTHD	4.0	–	0.6	–	
Stop Condition Setup Time	tSPSU	4.0	–	0.6	–	
Stop Condition Hold Time	tSPHD	4.0	–	0.6	–	
Output Valid from Clock	tVD	0	–	0	–	
Data Input Hold Time	tDIH	0	–	0	1.0	
Data Input Setup Time	tDIS	250	–	100	–	nS



NOTE) n= 0, and 1

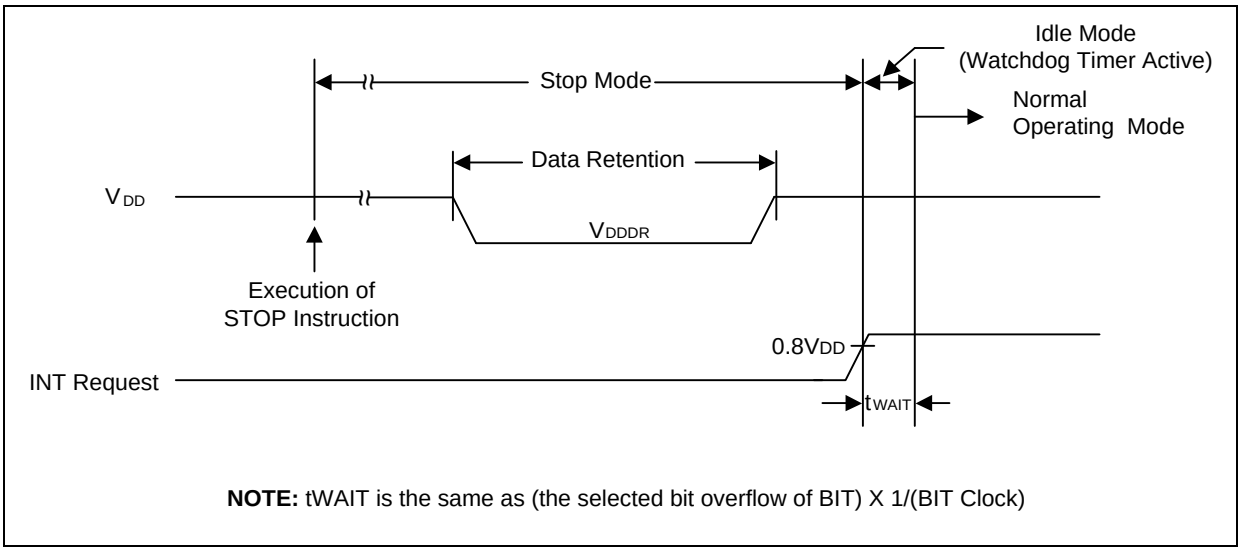
Figure 7.5 I2C0/1 Timing

7.14 Data Retention Voltage in Stop Mode

Table 7-14 Data Retention Voltage in Stop Mode

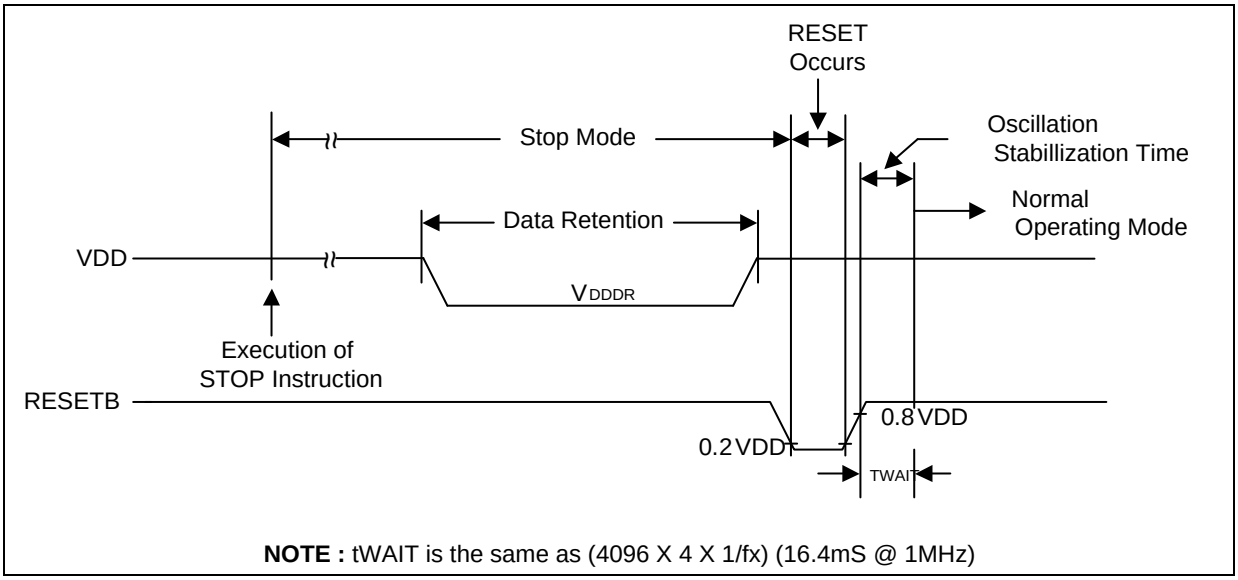
(T_A= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V)

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	μA



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

Figure 7.6 Stop Mode Release Timing when Initiated by an Interrupt



NOTE : tWAIT is the same as (4096 X 4 X 1/fx) (16.4mS @ 1MHz)

Figure 7.7 Stop Mode Release Timing when Initiated by RESETB

7.15 Internal Flash Rom Characteristics

Table 7-15 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	
Hard-Lock Time	t_{FHL}	–	–	2.5	2.7	
Page Buffer Reset Time	t_{FBR}	–	–	–	5	µS
Flash Programming Frequency	f_{PGM}	–	0.4	–	–	MHz
Endurance of Write/Erase	NF_{WE}	–	–	–	100,000	Times

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).

7.16 Input/Output Capacitance

Table 7-16 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}					

7.17 Main Clock Oscillator Characteristics

Table 7-17 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	–	10.0	
		3.0V – 5.5V	0.4	–	12.0	
Ceramic Oscillator	Main oscillation frequency	1.8V – 5.5V	0.4	–	4.2	MHz
		2.7V – 5.5V	0.4	–	10.0	
		3.0V – 5.5V	0.4	–	12.0	
External Clock	XIN input frequency	1.8V – 5.5V	0.4	–	4.2	MHz
		2.7V – 5.5V	0.4	–	10.0	
		3.0V – 5.5V	0.4	–	12.0	

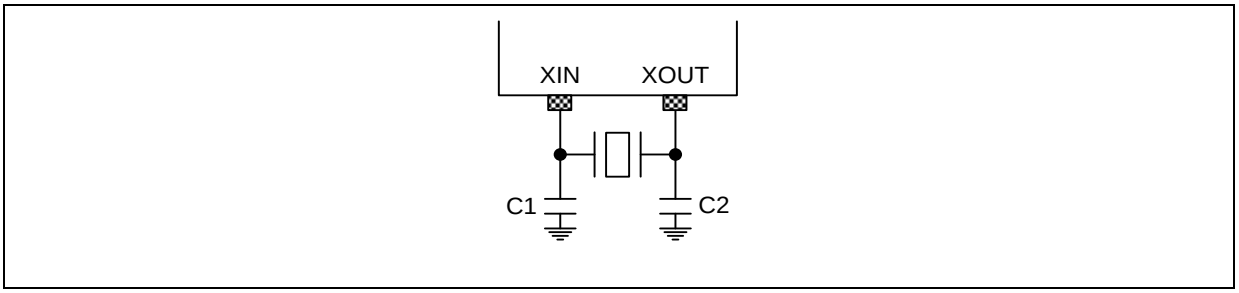


Figure 7.8 Crystal/Ceramic Oscillator

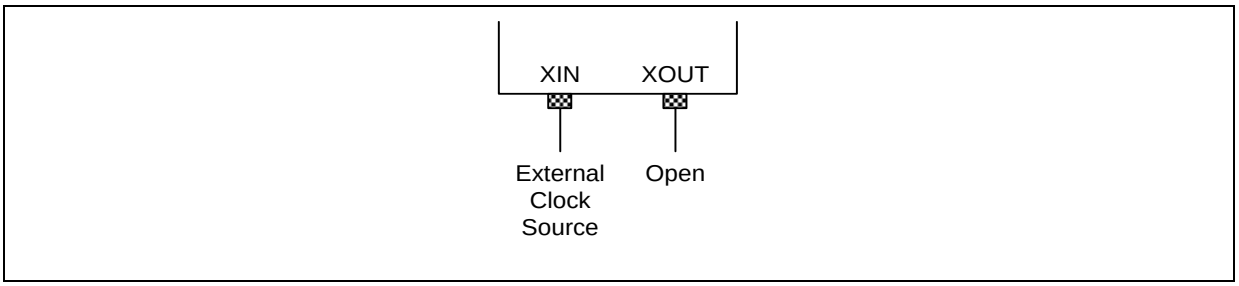


Figure 7.9 External Clock

7.18 Sub Clock Oscillator Characteristics

Table 7-18 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

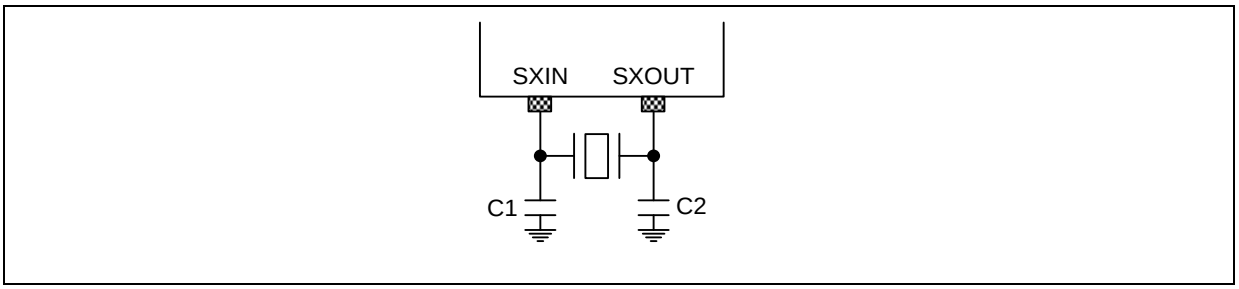


Figure 7.10 Crystal Oscillator

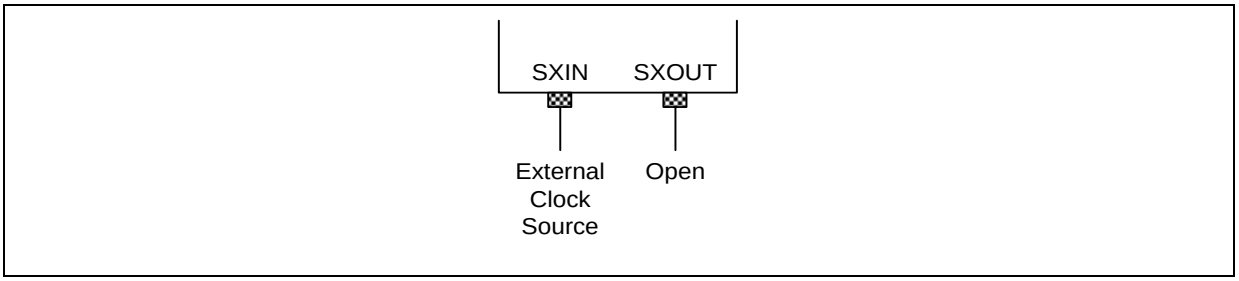


Figure 7.11 External Clock

7.19 Main Oscillation Stabilization Characteristics

Table 7-19 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 12MHz XIN input high and low width (t_{XH}, t_{XL})	42	–	1250	nS

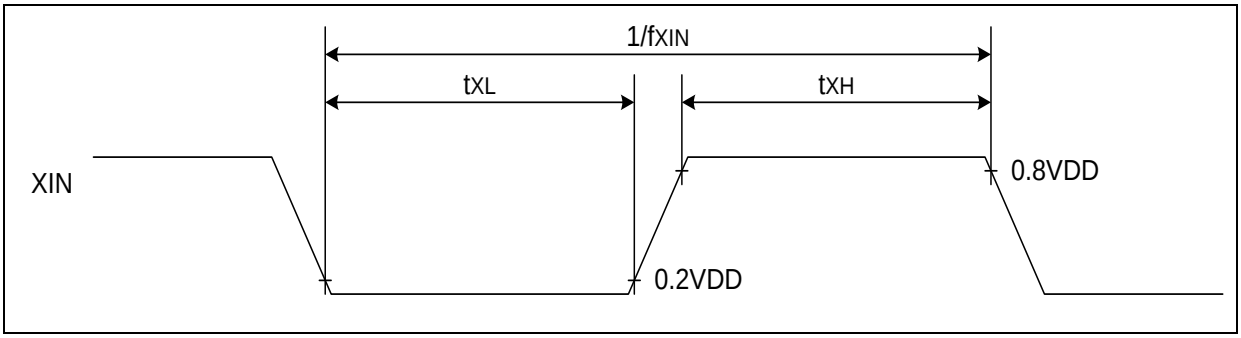


Figure 7.12 Clock Timing Measurement at XIN

7.20 Sub Oscillation Characteristics

Table 7-20 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	µS

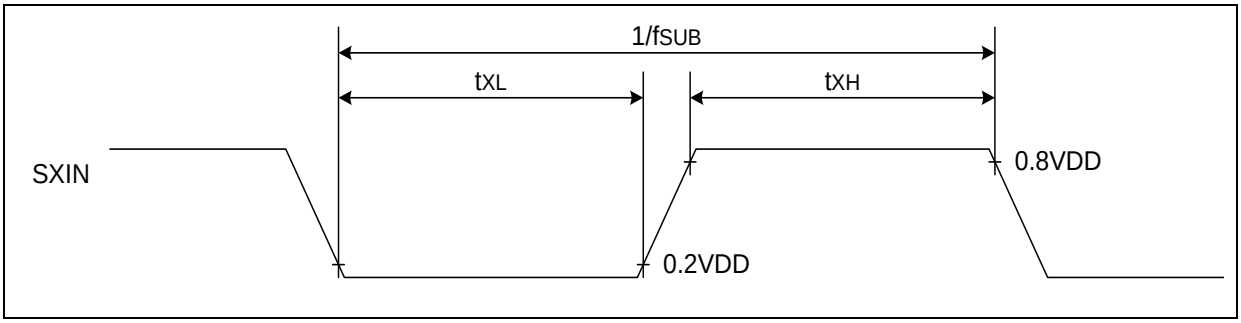


Figure 7.13 Clock Timing Measurement at SXIN

7.21 Operating Voltage Range

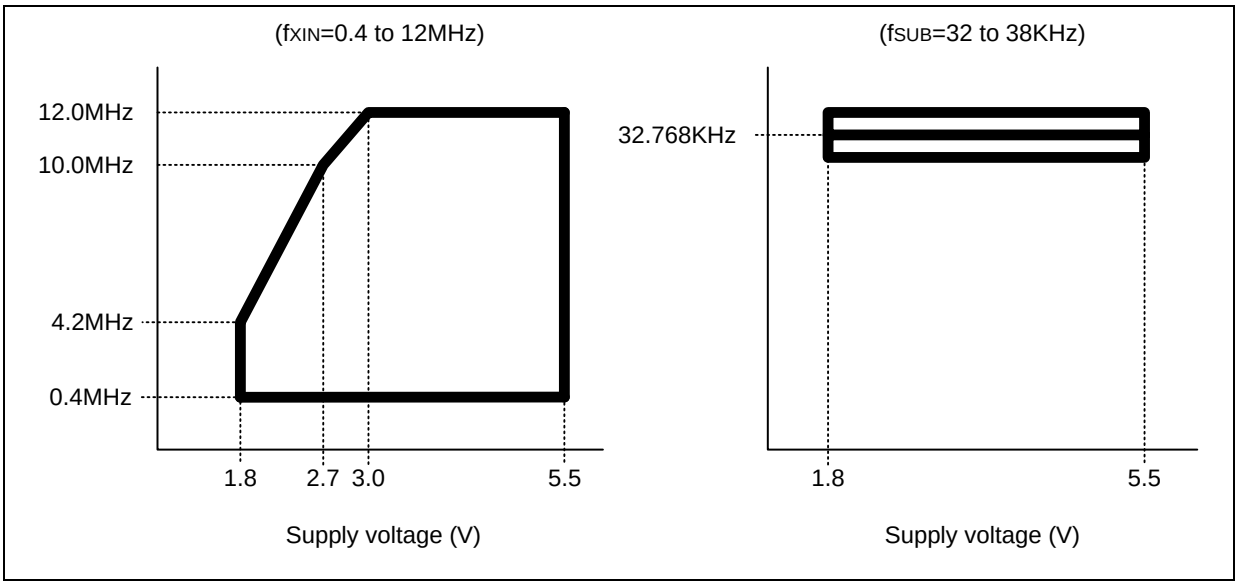


Figure 7.14 Operating Voltage Range

7.22 Recommended Circuit and Layout

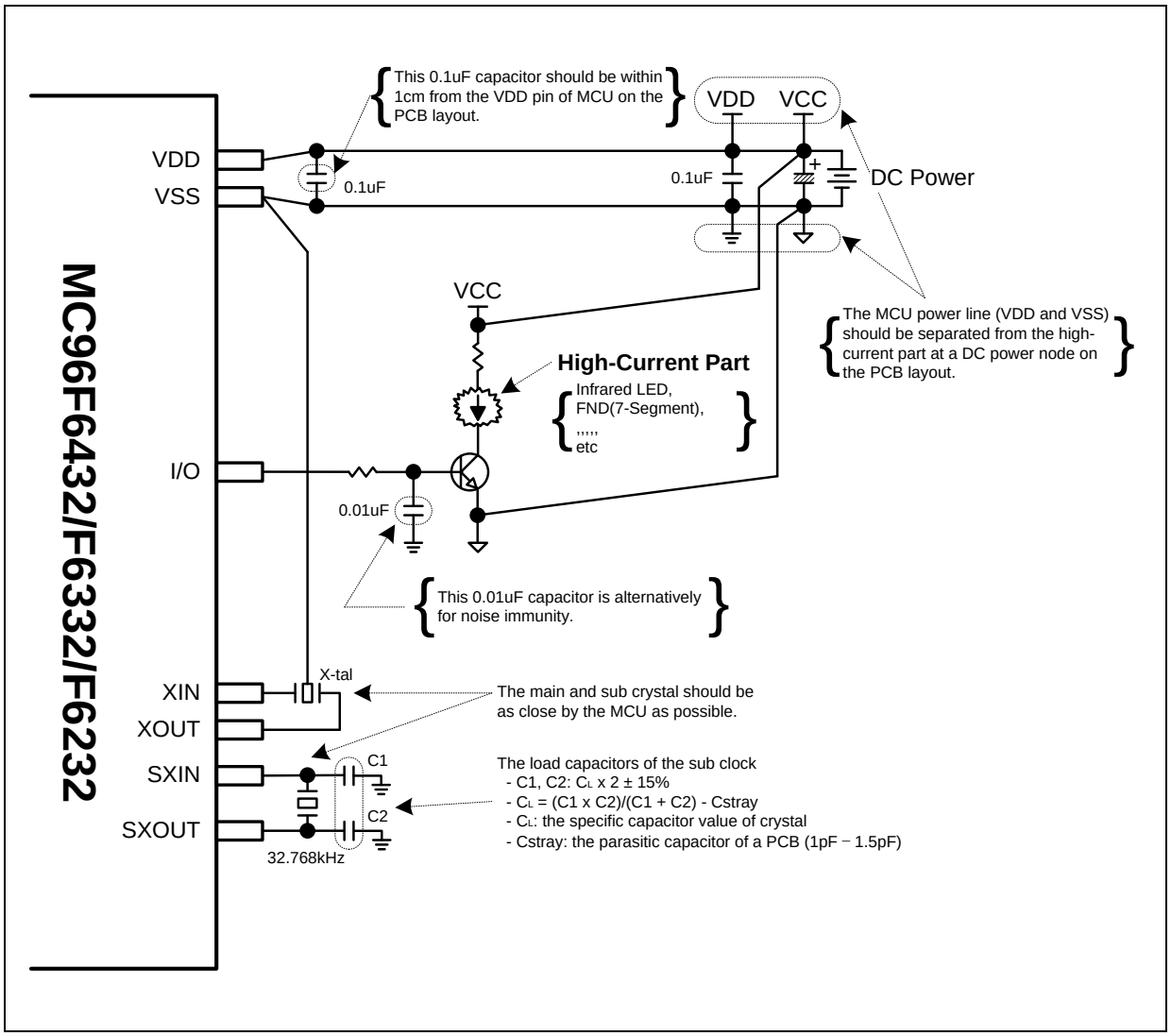


Figure 7.15 Recommended Circuit and Layout

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.

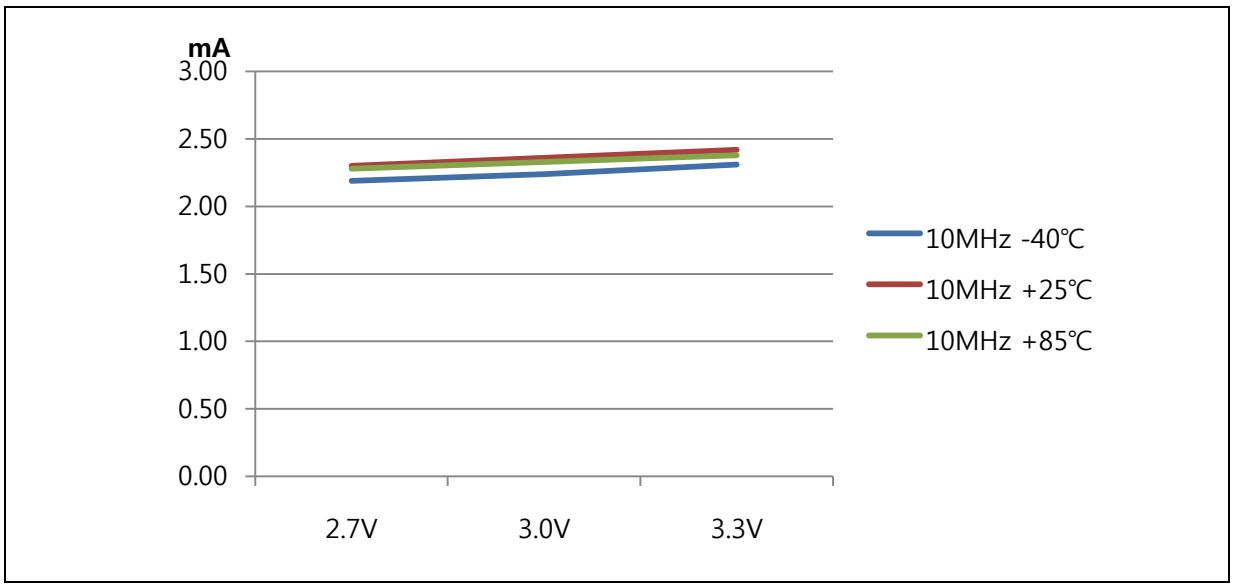


Figure 7.16 RUN (IDD1) Current

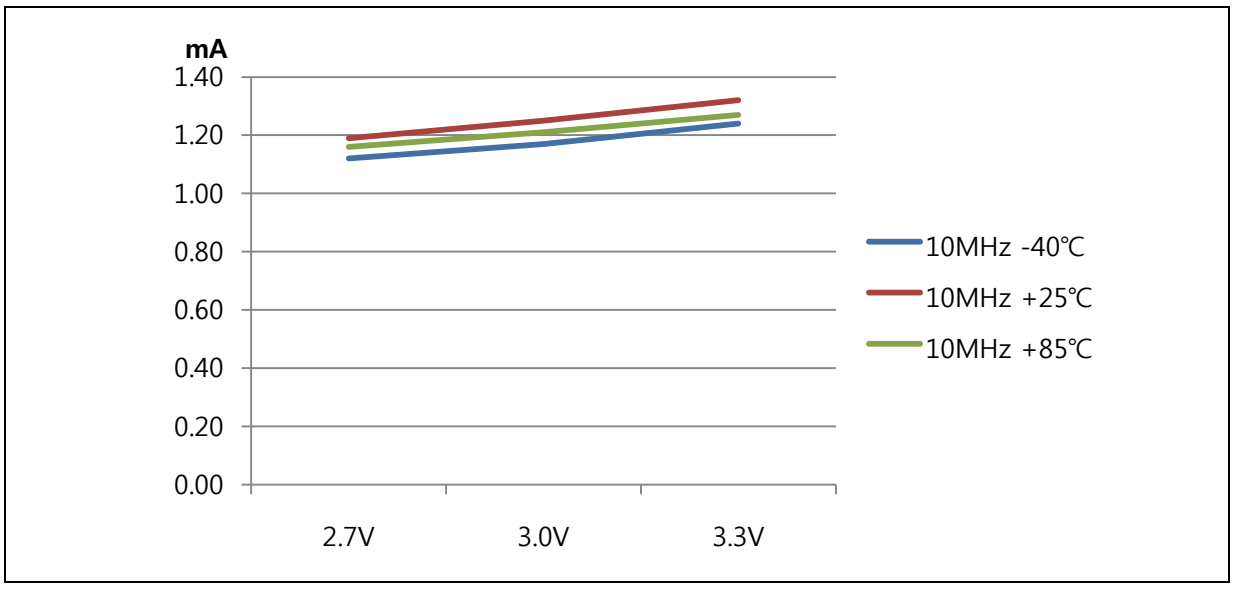


Figure 7.17 IDLE (IDD2) Current

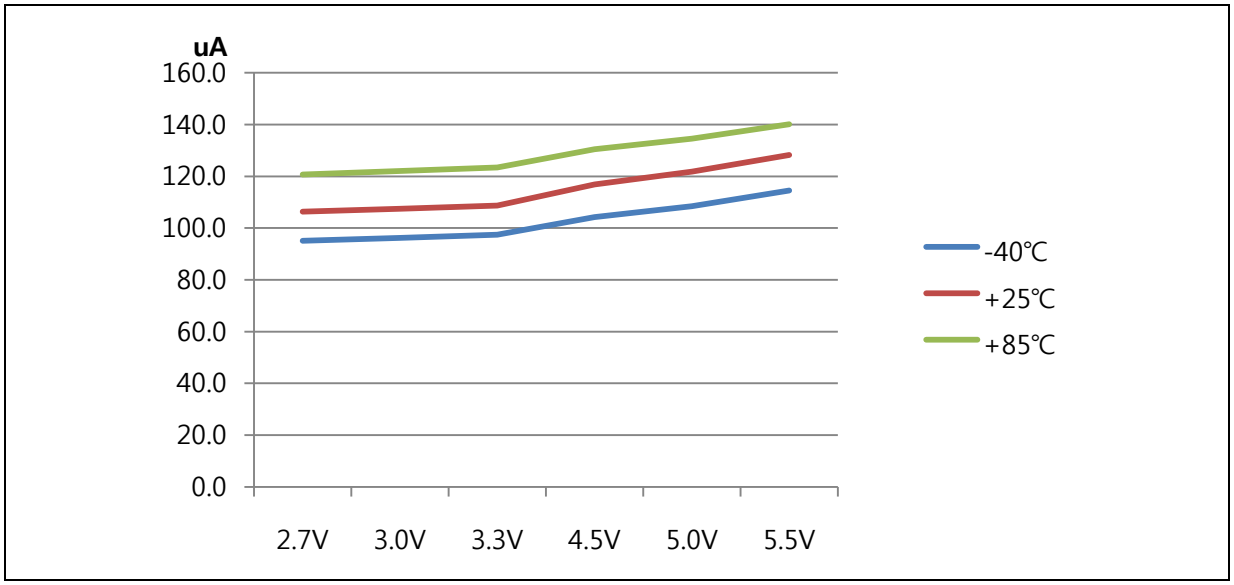


Figure 7.18 SUB RUN (IDD3) Current

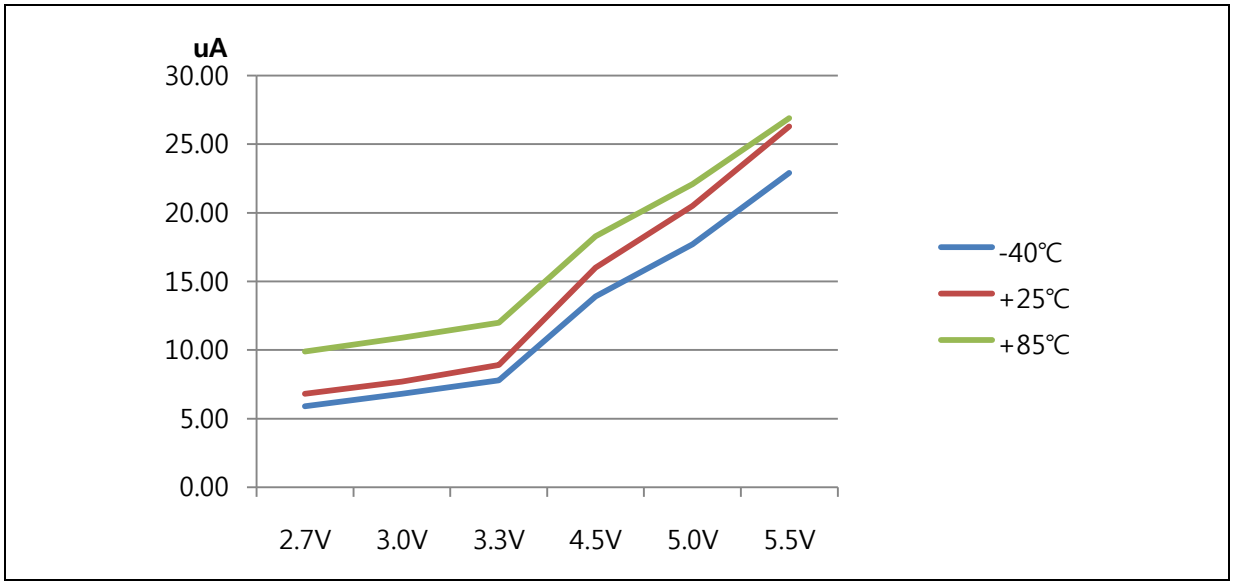


Figure 7.19 SUB IDLE (IDD4) Current

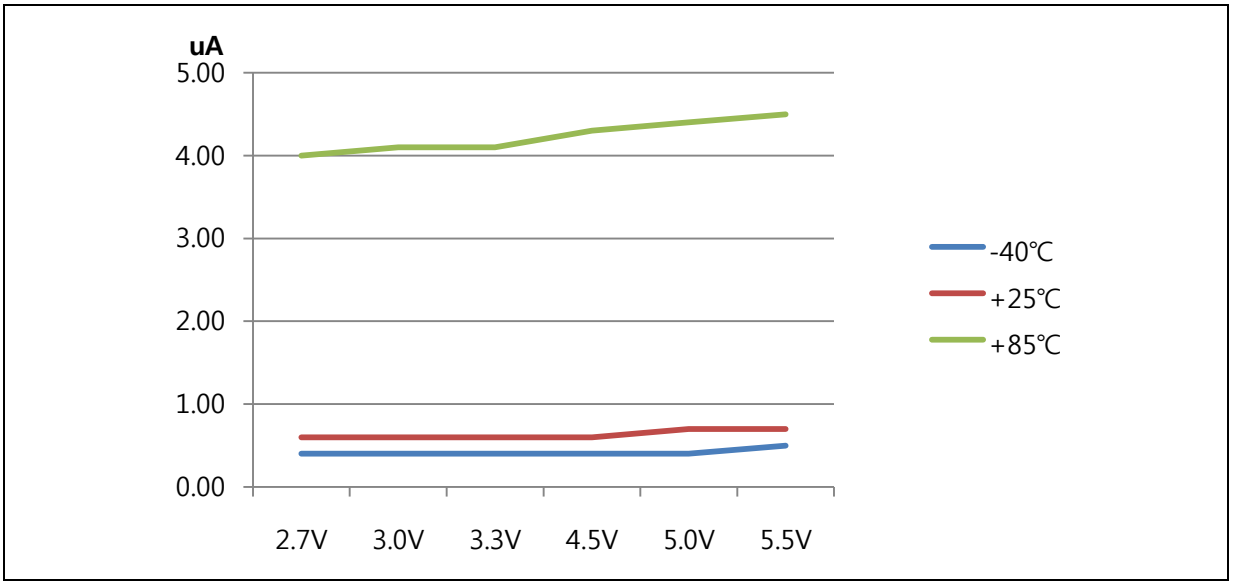


Figure 7.20 STOP (IDD5) Current

8. Memory

The MC96F6432/F6332/F6232 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.

MC96F6432/F6332/F6232 provides on-chip 32k 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 768 bytes and it includes 27 bytes of LCD display RAM.

8.1 Program Memory

A 16-bit program counter is capable of addressing up to 64k bytes, but this device has just 32k 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 11, for example, is assigned to location 000BH. If external interrupt 11 is going to be used, its service routine must begin at location 000BH. 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.

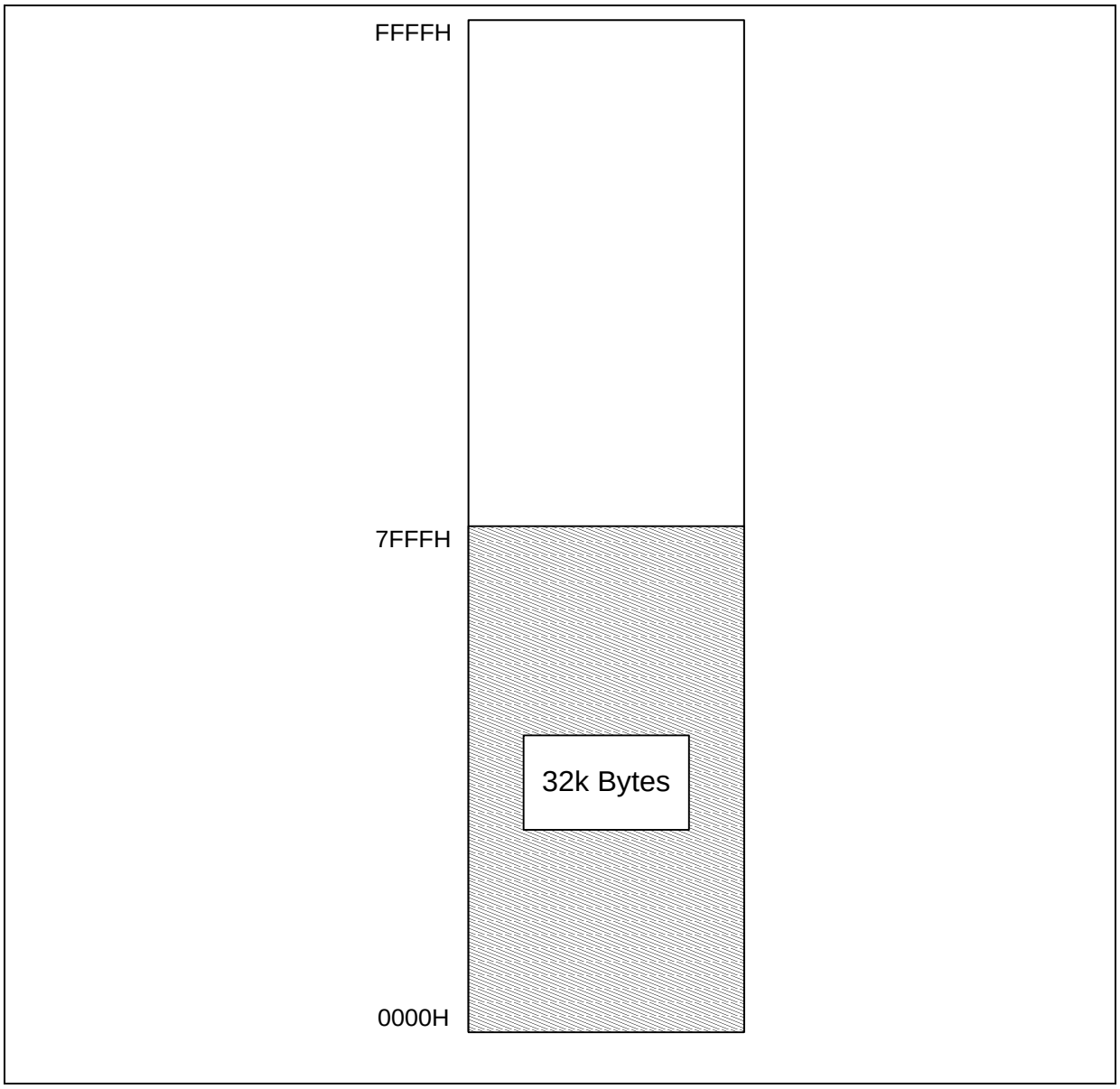


Figure 8.1 Program Memory

- 32k Bytes Including Interrupt Vector Region

8.2 Data Memory

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

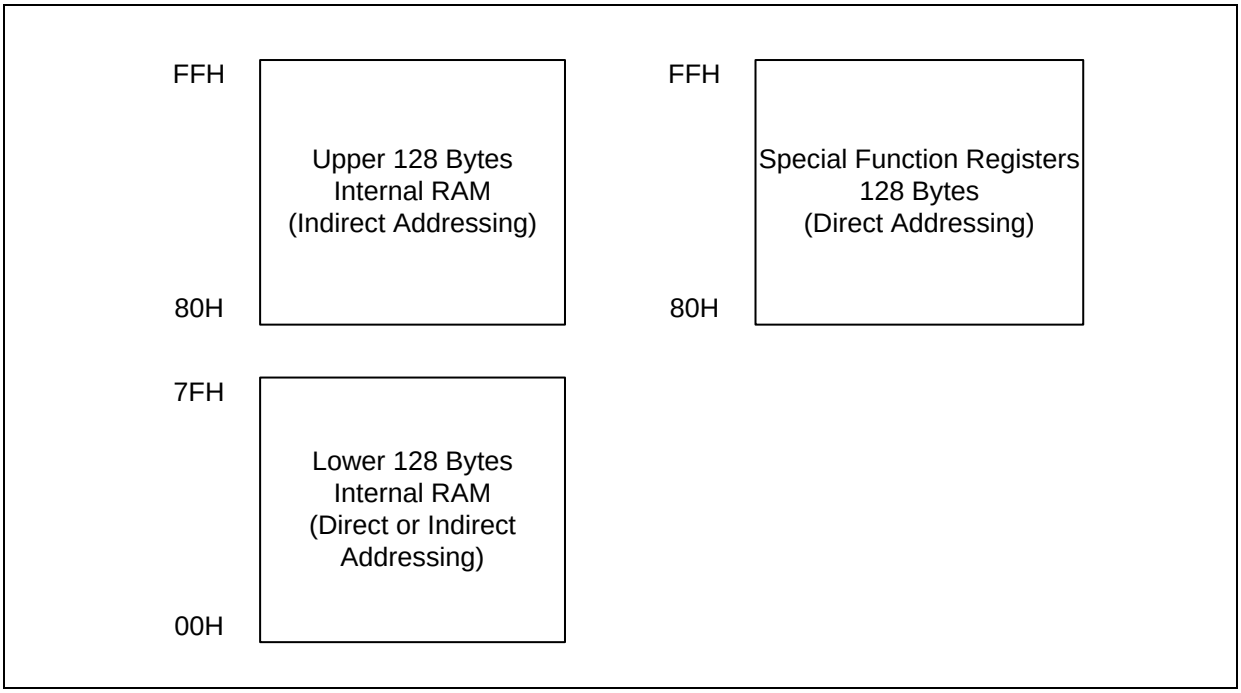


Figure 8.2 Data Memory Map

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.

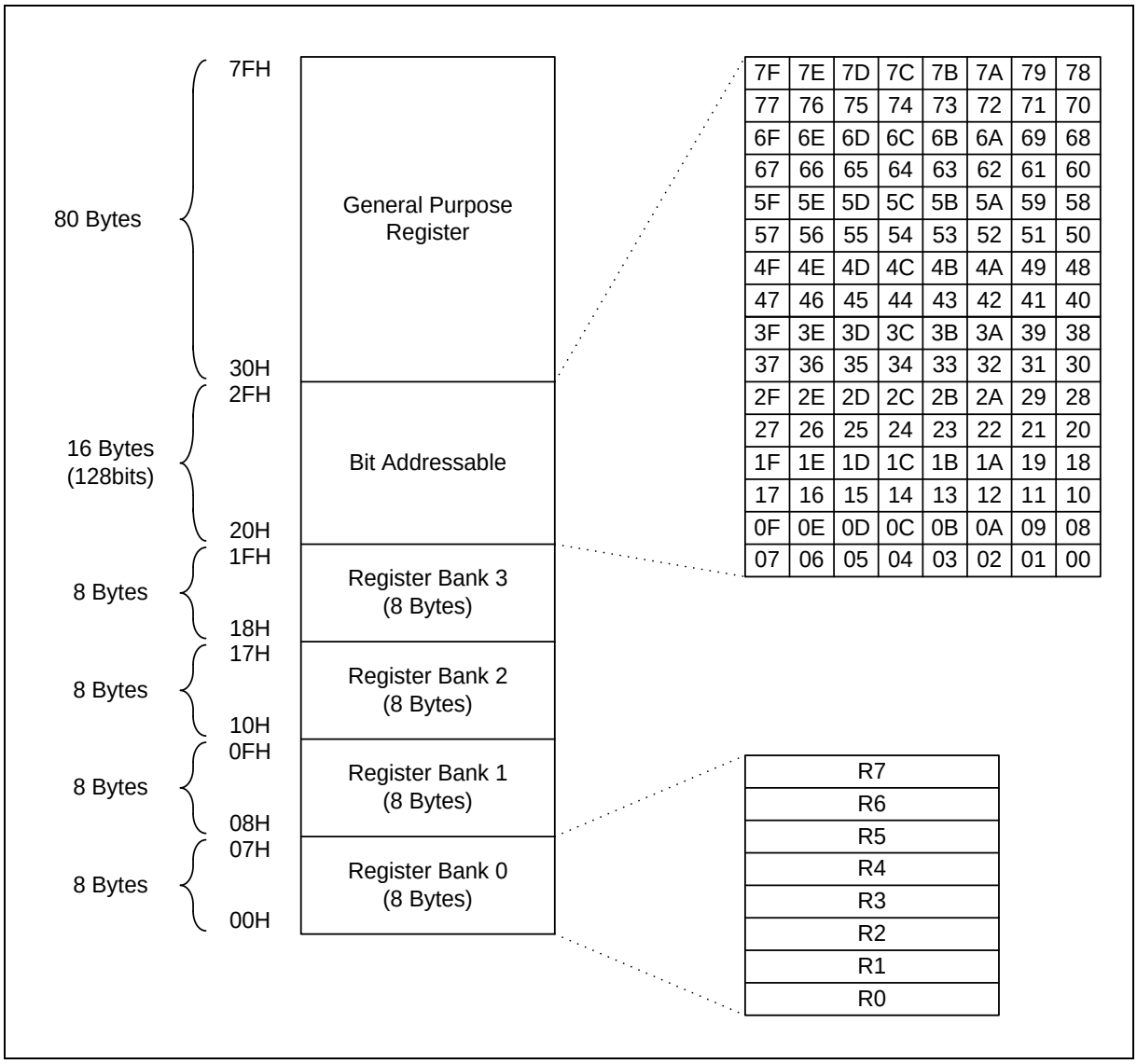


Figure 8.3 Lower 128 Bytes RAM

8.3 XRAM Memory

MC96F6432/F6332/F6232 has 768 bytes XRAM. This area has no relation with RAM/FLASH. It can be read and written to through SFR with 8-bit unit.

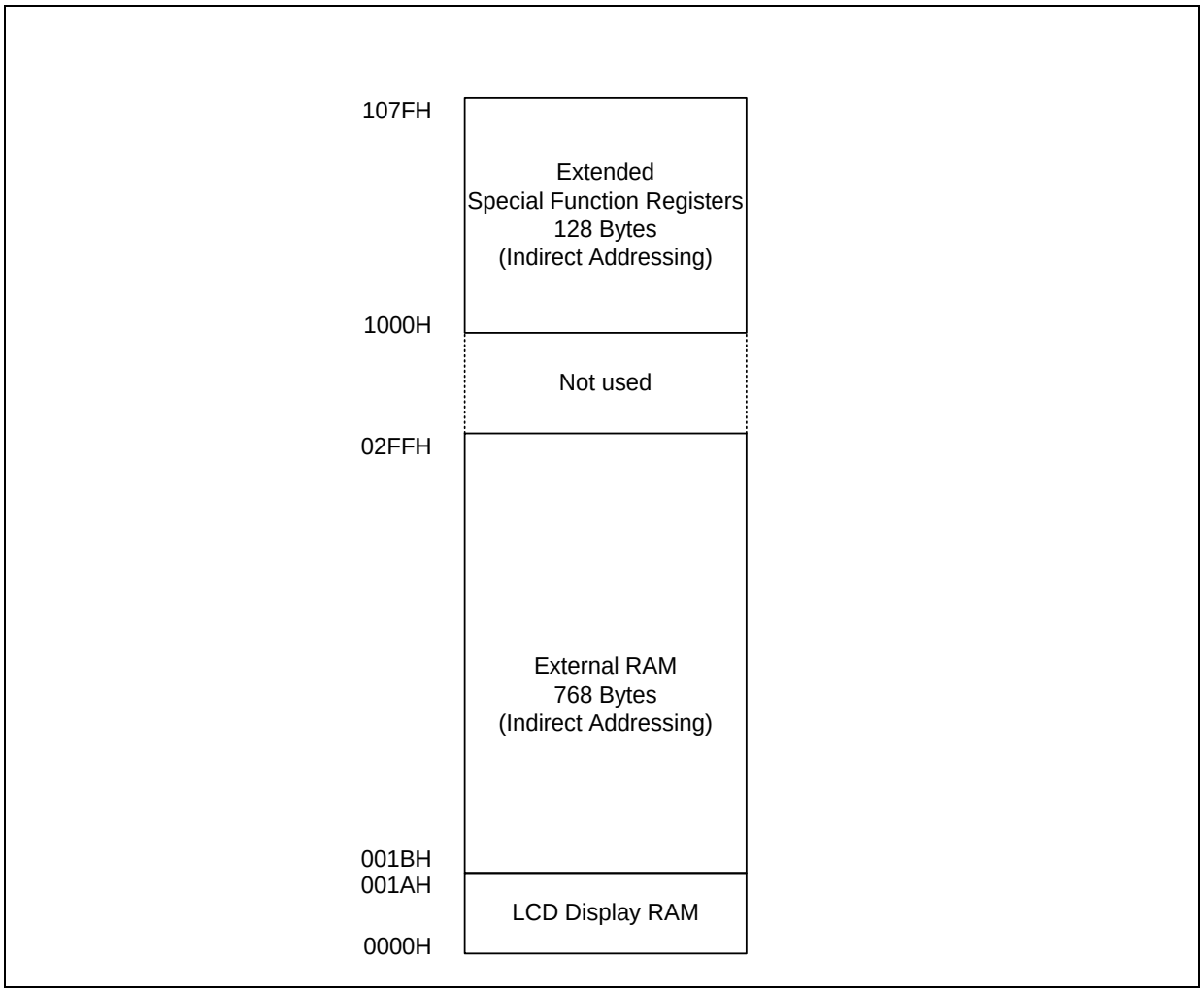


Figure 8.4 XDATA Memory Area

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	IP1	–	FSADRH	FSADRM	FSADRL	FIDR	FMCR	P5FSR
0F0H	B	USI1ST1	USI1ST2	USI1BD	USI1SDHR	USI1DR	USI1SCLR	USI1SCHR
0E8H	RSTFR	USI1CR1	USI1CR2	USI1CR3	USI1CR4	USI1SAR	P3FSR	P4FSR
0E0H	ACC	USI0ST1	USI0ST2	USI0BD	USI0SDHR	USI0DR	USI0SCLR	USI0SCHR
0D8H	LVRCR	USI0CR1	USI0CR2	USI0CR3	USI0CR4	USI0SAR	P0DB	P15DB
0D0H	PSW	P5IO	P0FSRL	P0FSRH	P1FSRL	P1FSRH	P2FSRL	P2FSRH
0C8H	OSCCR	P4IO	–	–	–	–	–	–
0C0H	EIFLAG0	P3IO	T2CRL	T2CRH	T2ADRL	T2ADRH	T2BDRL	T2BDRH
0B8H	IP	P2IO	T1CRL	T1CRH	T1ADRL	T1ADRH	T1BDRL	T1BDRH
0B0H	P5	P1IO	T0CR	T0CNT	T0DR/ T0CDR	SPICR	SPIDR	SPISR
0A8H	IE	IE1	IE2	IE3	P0PU	P1PU	P2PU	P3PU
0A0H	P4	P0IO	EO	P4PU	EIPOL0L	EIPOL0H	EIFLAG1	EIPOL1
98H	P3	LCDCRL	LCDCRH	LCDCCR	ADCCRH	ADCCRH	ADCDRL	ADCDRH
90H	P2	P0OD	P1OD	P2OD	P4OD	P5PU	WTCR	BUZCR
88H	P1	WTDR/ WTCNT	SCCR	BITCR	BITCNT	WDTCR	WDTDR/ WDTCNT	BUZDR
80H	P0	SP	DPL	DPH	DPL1	DPH1	LVICR	PCON

NOTE) These registers are bit-addressable.

Table 8-2 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	–	–	–	–	–	–	–	–
1050H	–	–	–	–	–	–	–	–
1048H	–	–	–	–	–	–	–	–
1040H	–	–	–	–	–	–	–	–
1038H	–	–	–	–	–	–	–	–
1030H	–	–	–	–	–	–	–	–
1028H	–	–	–	–	–	–	–	–
1020H	–	–	–	–	–	–	–	–
1018H	–	–	–	–	–	–	–	–
1010H	T4DLYA	T4DLYB	T4DLYC	T4DR	T4CAPR	T4CNT	–	–
1008H	T4PPRL	T4PPRH	T4ADRL	T4ADRH	T4BDRL	T4BDRH	T4CDRL	T4CDRH
100H	T3CR	T3CNT/ T3DR/ T3CAPR	T4CR	T4PCR1	T4PCR2	T4PCR3	T4ISR	T4IMSK

NOTE) These registers are bit-addressable.

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	Basic Interval Timer 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	BUZZER Data Register	BUZDR	R/W	1	1	1	1	1	1	1	1
90H	P2 Data Register	P2	R/W	0	0	0	0	0	0	0	0
91H	P0 Open-drain Selection Register	P0OD	R/W	0	0	0	0	0	0	0	0
92H	P1 Open-drain Selection Register	P1OD	R/W	0	0	0	0	0	0	0	0
93H	P2 Open-drain Selection Register	P2OD	R/W	0	0	0	0	0	0	0	0
94H	P4 Open-drain Selection Register	P4OD	R/W	–	–	–	–	0	0	0	0
95H	P5 Pull-up Resistor Selection Register	P5PU	R/W	–	–	0	0	0	0	0	0
96H	Watch Timer Control Register	WTCR	R/W	0	–	–	0	0	0	0	0
97H	BUZZER Control Register	BUZCR	R/W	–	–	–	–	–	0	0	0
98H	P3 Data Register	P3	R/W	0	0	0	0	0	0	0	0
99H	LCD Driver Control Low Register	LCDCRL	R/W	–	–	0	0	0	0	0	0
9AH	LCD Driver Control High Register	LCDCRH	R/W	–	–	–	0	–	–	0	0
9BH	LCD Contrast Control register	LCDCCR	R/W	0	–	–	–	0	0	0	0
9CH	A/D Converter Control Low Register	ADCCRL	R/W	0	0	0	0	0	0	0	0
9DH	A/D Converter Control High Register	ADCCRH	R/W	0	–	0	0	0	0	0	0
9EH	A/D Converter Data Low Register	ADCDRL	R	x	x	x	x	x	x	x	x
9FH	A/D Converter Data High Register	ADCDRH	R	x	x	x	x	x	x	x	x

Table 8-2 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
A1H	P0 Direction Register	P0IO	R/W	0	0	0	0	0	0	0	0
A2H	Extended Operation Register	EO	R/W	–	–	–	0	–	0	0	0
A3H	P4 Pull-up Resistor Selection Register	P4PU	R/W	–	–	–	–	0	0	0	0
A4H	External Interrupt Polarity 0 Low Register	EIPOL0L	R/W	0	0	0	0	0	0	0	0
A5H	External Interrupt Polarity 0 High Register	EIPOL0H	R/W	0	0	0	0	0	0	0	0
A6H	External Interrupt Flag 1 Register	EIFLAG1	R/W	0	0	0	0	0	0	0	0
A7H	External Interrupt Polarity 1 Register	EIPOL1	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	P0 Pull-up Resistor Selection Register	P0PU	R/W	0	0	0	0	0	0	0	0
ADH	P1 Pull-up Resistor Selection Register	P1PU	R/W	0	0	0	0	0	0	0	0
AEH	P2 Pull-up Resistor Selection Register	P2PU	R/W	0	0	0	0	0	0	0	0
AFH	P3 Pull-up Resistor Selection Register	P3PU	R/W	0	0	0	0	0	0	0	0
B0H	P5 Data Register	P5	R/W	–	–	0	0	0	0	0	0
B1H	P1 Direction Register	P1IO	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	SPI 2 Control Register	SPICR	R/W	0	0	0	0	0	0	0	0
B6H	SPI 2 Data Register	SPIDR	R/W	0	0	0	0	0	0	0	0
B7H	SPI 2 Status Register	SPISR	R/W	0	0	0	–	0	0	–	–
B8H	Interrupt Priority Register	IP	R/W	–	–	0	0	0	0	0	0
B9H	P2 Direction Register	P2IO	R/W	0	0	0	0	0	0	0	0
BAH	Timer 1 Control Low Register	T1CRL	R/W	0	0	0	0	–	0	0	0
BBH	Timer 1 Counter High Register	T1CRH	R/W	0	–	0	0	–	–	–	0
BCH	Timer 1 A Data Low Register	T1ADRL	R/W	1	1	1	1	1	1	1	1
BDH	Timer 1 A Data High Register	T1ADRH	R/W	1	1	1	1	1	1	1	1
BEH	Timer 1 B Data Low Register	T1BDRL	R/W	1	1	1	1	1	1	1	1
BFH	Timer 1 BData High Register	T1BDRH	R/W	1	1	1	1	1	1	1	1

Table 8-2 SFR Map (Continued)

Address	Function	Symbol	R/W	@Reset							
				7	6	5	4	3	2	1	0
C0H	External Interrupt Flag 0 Register	EIFLAG0	R/W	0	0	0	0	0	0	0	0
C1H	P3 Direction Register	P3IO	R/W	0	0	0	0	0	0	0	0
C2H	Timer 2 Control Low Register	T2CRL	R/W	0	0	0	0	–	0	–	0
C3H	Timer 2 Control High Register	T2CRH	R/W	0	–	0	0	–	–	–	0
C4H	Timer 2 A Data Low Register	T2ADRL	R/W	1	1	1	1	1	1	1	1
C5H	Timer 2 A Data High Register	T2ADRH	R/W	1	1	1	1	1	1	1	1
C6H	Timer 2 B Data Low Register	T2BDRL	R/W	1	1	1	1	1	1	1	1
C7H	Timer 2 BData High Register	T2BDRH	R/W	1	1	1	1	1	1	1	1
C8H	Oscillator Control Register	OSCCR	R/W	–	–	0	0	1	0	0	0
C9H	P4 Direction Register	P4IO	R/W	–	–	–	–	0	0	0	0
CAH	Reserved	–	–	–							
CBH	Reserved	–	–	–							
CCH	Reserved	–	–	–							
CDH	Reserved	–	–	–							
CEH	Reserved	–	–	–							
CFH	Reserved	–	–	–							
D0H	Program Status Word Register	PSW	R/W	0	0	0	0	0	0	0	0
D1H	P5 Direction Register	P5IO	R/W	–	–	0	0	0	0	0	0
D2H	P0 Function Selection Low Register	P0FSRL	R/W	–	0	0	0	0	0	0	0
D3H	P0 Function Selection High Register	P0FSRH	R/W	–	–	0	0	0	0	0	0
D4H	P1 Function Selection Low Register	P1FSRL	R/W	0	0	0	0	0	0	0	0
D5H	P1 Function Selection High Register	P1FSRH	R/W	0	0	0	0	0	0	0	0
D6H	P2 Function Selection Low Register	P2FSRL	R/W	–	–	0	0	0	0	0	0
D7H	P2 Function Selection High Register	P2FSRH	R/W	–	–	–	–	0	0	0	0
D8H	Low Voltage Reset Control Register	LVRCR	R/W	0	–	–	0	0	0	0	0
D9H	USI0 Control Register 1	USI0CR1	R/W	0	0	0	0	0	0	0	0
DAH	USI0 Control Register 2	USI0CR2	R/W	0	0	0	0	0	0	0	0
DBH	USI0 Control Register 3	USI0CR3	R/W	0	0	0	0	0	0	0	0
DCH	USI0 Control Register 4	USI0CR4	R/W	0	–	–	0	0	–	0	0
DDH	USI0 Slave Address Register	USI0SAR	R/W	0	0	0	0	0	0	0	0
DEH	P0 Debounce Enable Register	P0DB	R/W	0	0	0	0	0	0	0	0
DFH	P1/P5 Debounce Enable Register	P15DB	R/W	–	–	0	0	0	0	0	0

Table 8-2 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	USI0 Status Register 1	USI0ST1	R/W	0	0	0	0	–	0	0	0
E2H	USI0 Status Register 2	USI0ST2	R	0	0	0	0	0	0	0	0
E3H	USI0 Baud Rate Generation Register	USI0BD	R/W	1	1	1	1	1	1	1	1
E4H	USI0 SDA Hold Time Register	USI0SHDR	R/W	0	0	0	0	0	0	0	1
E5H	USI0 Data Register	USI0DR	R/W	0	0	0	0	0	0	0	0
E6H	USI0 SCL Low Period Register	USI0SCLR	R/W	0	0	1	1	1	1	1	1
E7H	USI0 SCL High Period Register	USI0SCHR	R/W	0	0	1	1	1	1	1	1
E8H	Reset Flag Register	RSTFR	R/W	1	x	0	0	x	–	–	–
E9H	USI1 Control Register 1	USI1CR1	R/W	0	0	0	0	0	0	0	0
EAH	USI1 Control Register 2	USI1CR2	R/W	0	0	0	0	0	0	0	0
EBH	USI1 Control Register 3	USI1CR3	R/W	0	0	0	0	0	0	0	0
ECH	USI1 Control Register 4	USI1CR4	R/W	0	–	–	0	0	–	0	0
EDH	USI1 Slave Address Register	USI1SAR	R/W	0	0	0	0	0	0	0	0
EEH	P3 Function Selection Register	P3FSR	R/W	0	0	0	0	0	0	0	0
EFH	P4 Function Selection Register	P4FSR	R/W	–	0	0	0	0	0	0	0
F0H	B Register	B	R/W	0	0	0	0	0	0	0	0
F1H	USI1 Status Register 1	USI1ST1	R/W	0	0	0	0	–	0	0	0
F2H	USI1 Status Register 2	USI1ST2	R	0	0	0	0	0	0	0	0
F3H	USI1 Baud Rate Generation Register	USI1BD	R/W	1	1	1	1	1	1	1	1
F4H	USI1 SDA Hold Time Register	USI1SHDR	R/W	0	0	0	0	0	0	0	1
F5H	USI1 Data Register	USI1DR	R/W	0	0	0	0	0	0	0	0
F6H	USI1 SCL Low Period Register	USI1SCLR	R/W	0	0	1	1	1	1	1	1
F7H	USI1 SCL High Period Register	USI1SCHR	R/W	0	0	1	1	1	1	1	1
F8H	Interrupt Priority Register 1	IP1	R/W	–	–	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	P5 Function Selection Register	P5FSR	R/W	–	–	0	0	0	0	0	0

Table 8-2 SFR Map (Continued)

Address	Function	Symbol	R/W	@Reset							
				7	6	5	4	3	2	1	0
1000H	Timer 3 Control Register	T3CR	R/W	0	–	0	0	0	0	0	0
1001H	Timer 3 Counter Register	T3CNT	R	0	0	0	0	0	0	0	0
	Timer 3 Data Register	T3DR	W	1	1	1	1	1	1	1	1
	Timer 3 Capture Data Register	T3CAPR	R	0	0	0	0	0	0	0	0
1002H	Timer 4 Control Register	T4CR	R/W	0	0	0	0	0	0	0	0
1003H	Timer 4 PWM Control Register 1	T4PCR1	R/W	0	0	0	0	0	0	0	0
1004H	Timer 4 PWM Control Register 2	T4PCR2	R/W	0	0	0	0	0	0	0	0
1005H	Timer 4 PWM Control Register 3	T4PCR3	R/W	–	0	0	0	–	–	–	–
1006H	Timer 4 Interrupt Status Register	T4ISR	R/W	0	0	0	0	0	–	–	–
1007H	Timer 4 Interrupt Mask Register	T4MSK	R/W	0	0	0	0	0	–	–	–
1008H	Timer 4 PWM Period Low Register	T4PPRL	R/W	1	1	1	1	1	1	1	1
1009H	Timer 4 PWM Period High Register	T4PPRH	R/W	–	–	–	–	–	–	0	0
100AH	Timer 4 PWM A Duty Low Register	T4ADRL	R/W	0	1	1	1	1	1	1	1
100BH	Timer 4 PWM A Duty High Register	T4ADRH	R/W	–	–	–	–	–	–	0	0
100CH	Timer 4 PWM B Duty Low Register	T4BDRL	R/W	0	1	1	1	1	1	1	1
100DH	Timer 4 PWM B Duty High Register	T4BDRH	R/W	–	–	–	–	–	–	0	0
100EH	Timer 4 PWM C Duty Low Register	T4CDRL	R/W	0	1	1	1	1	1	1	1
100FH	Timer 4 PWM C Duty High Register	T4CDRH	R/W	–	–	–	–	–	–	0	0
1010H	Timer 4 PWM A Delay Register	T4DLYA	R/W	0	0	0	0	0	0	0	0
1011H	Timer 4 PWM B Delay Register	T4DLYB	R/W	0	0	0	0	0	0	0	0
1012H	Timer 4 PWM C Delay Register	T4DLYC	R/W	0	0	0	0	0	0	0	0
1013H	Timer 4 Data Register	T4DR	R/W	1	1	1	1	1	1	1	1
1014H	Timer 4 Capture Data Register	T4CAPR	R	0	0	0	0	0	0	0	0
1015H	Timer 4 Counter Register	T4CNT	R	0	0	0	0	0	0	0	0
.....											
107FH	Reserved	–	–	–							

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

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 Software TRAP Instruction

1 Select MOVC @(DPTR++), A

DPSEL[2:0] Select Banked Data Pointer Register

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

9. I/O Ports

9.1 I/O Ports

The MC96F6432/F6332/F6232 has ten groups of I/O ports (P0 ~ P5). 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 P0 includes 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 ~ P4 and a bit for P5. 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)

P0[7:2], P1[2:1], P1[7:6], P52 and P54 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	A1H	R/W	00H	P0 Direction Register
P0PU	ACH	R/W	00H	P0 Pull-up Resistor Selection Register
P0OD	91H	R/W	00H	P0 Open-drain Selection Register
P0DB	DEH	R/W	00H	P0 Debounce Enable Register
P0FSRH	D3H	R/W	00H	P0 Function Selection High Register
P0FSRL	D2H	R/W	00H	P0 Function Selection Low Register
P1	88H	R/W	00H	P1 Data Register
P1IO	B1H	R/W	00H	P1 Direction Register
P1PU	ADH	R/W	00H	P1 Pull-up Resistor Selection Register
P1OD	92H	R/W	00H	P1 Open-drain Selection Register
P15DB	DFH	R/W	00H	P1/P5 Debounce Enable Register
P1FSRH	D5H	R/W	00H	P1 Function Selection High Register
P1FSRL	D4H	R/W	00H	P1 Function Selection Low Register
P2	90H	R/W	00H	P2 Data Register
P2IO	B9H	R/W	00H	P2 Direction Register
P2PU	AEH	R/W	00H	P2 Pull-up Resistor Selection Register
P2OD	93H	R/W	00H	P2 Open-drain Selection Register
P2FSRH	D7H	R/W	00H	P2 Function Selection High Register
P2FSRL	D6H	R/W	00H	P2 Function Selection Low Register
P3	98H	R/W	00H	P3 Data Register
P3IO	C1H	R/W	00H	P3 Direction Register
P3PU	AFH	R/W	00H	P3 Pull-up Resistor Selection Register
P3FSR	EEH	R/W	00H	P3 Function Selection Register
P4	A0H	R/W	00H	P4 Data Register
P4IO	C9H	R/W	00H	P4 Direction Register
P4PU	A3H	R/W	00H	P4 Pull-up Resistor Selection Register
P4OD	94H	R/W	00H	P4 Open-drain Selection Register
P4FSR	EFH	R/W	00H	P4 Function Selection Register
P5	B0H	R/W	00H	P5 Data Register
P5IO	D1H	R/W	00H	P5 Direction Register
P5PU	95H	R/W	00H	P5 Pull-up Resistor Selection Register
P5FSR	EFH	R/W	00H	P5 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), debounce enable register (P0DB), 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) : A1H

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

NOTE: EC3/EINT0 ~ EINT5 function possible when input

P0PU (P0 Pull-up Resistor Selection Register) : ACH

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) : 91H

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

P0DB (P0 Debounce Enable Register) : DEH

7	6	5	4	3	2	1	0
DBCLK1	DBCLK0	P07DB	P06DB	P05DB	P04DB	P03DB	P02DB
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

P07DB Configure Debounce of P07 Port

0	Disable
1	Enable

P06DB Configure Debounce of P06 Port

0	Disable
1	Enable

P05DB Configure Debounce of P05 Port

0	Disable
1	Enable

P04DB Configure Debounce of P04 Port

0	Disable
1	Enable

P03DB Configure Debounce of P03Port

0	Disable
1	Enable

P02DB Configure Debounce of P02 Port

0	Disable
1	Enable

NOTES) 1. If the same level is not detected on 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 clock 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.

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 (P15DB), 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) : B1H

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: EINT6/ENINT7/EINT11/EINT12/SS2/EC1 function possibl when input

P1PU (P1 Pull-up Resistor Selection Register) : ADH

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) : 92H

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 : 08H

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

0 Push-pull output

1 Open-drain output

P15DB (P1/P5 Debounce Enable Register) : DFH

7	6	5	4	3	2	1	0
–	–	P54DB	P52DB	P17DB	P16DB	P12DB	P11DB
–	–	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P54DB	Configure Debounce of P54 Port	
	0	Disable
	1	Enable
P52DB	Configure Debounce of P52 Port	
	0	Disable
	1	Enable
P17DB	Configure Debounce of P17 Port	
	0	Disable
	1	Enable
P16DB	Configure Debounce of P16 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

NOTES) 1. If the same level is not detected on 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 clock 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 0 debounce enable register (P0DB) for the debounce clock of port 1 and port 5.

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), 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) : B9H

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: SS1 function possible when input

P2PU (P2 Pull-up Resistor Selection Register) : AEH

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) : 93H

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

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) and P3 pull-up resistor selection register (P3PU). 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) : C1H

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

P3PU (P3 Pull-up Resistor Selection Register) : AFH

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

9.7 P4 Port

9.7.1 P4 Port Description

P4 is 4-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
–	–	–	–	P43	P42	P41	P40
–	–	–	–	R/W	R/W	R/W	R/W

Initial value : 00H

P4[3:0] I/O Data

P4IO (P4 Direction Register) : C9H

7	6	5	4	3	2	1	0
–	–	–	–	P43IO	P42IO	P41IO	P40IO
–	–	–	–	R/W	R/W	R/W	R/W

Initial value : 00H

P4IO[3:0] P4 Data I/O Direction

0 Input

1 Output

NOTE: SS0 function possible when input

P4PU (P4 Pull-up Resistor Selection Register) : A3H

7	6	5	4	3	2	1	0
–	–	–	–	P43PU	P42PU	P41PU	P40PU
–	–	–	–	R/W	R/W	R/W	R/W

Initial value : 00H

P4PU[3:0] Configure Pull-up Resistor of P4 Port

0 Disable

1 Enable

P4OD (P4 Open-drain Selection Register) : 94H

7	6	5	4	3	2	1	0
–	–	–	–	P43OD	P42OD	P41OD	P40OD
–	–	–	–	R/W	R/W	R/W	R/W

Initial value : 00H

P4OD[3:0] Configure Open-drain of P4 Port

0 Push-pull output

1 Open-drain output

9.8 P5 Port

9.8.1 P5 Port Description

P5 is 6-bit I/O port. P5 control registers consist of P5 data register (P5), P5 direction register (P5IO) and P5 pull-up resistor selection register (P5PU) . 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
–	–	P55	P54	P53	P52	P51	P50
–	–	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P5[5:0] I/O Data

P5IO (P5 Direction Register) : D1H

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

Initial value : 00H

P5IO[5:0] P5 Data I/O Direction

0 Input

1 Output

NOTE: EC0/EINT8/EINT10/BLNK function possible when input

P5PU (P5 Pull-up Resistor Selection Register) : 95H

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

Initial value : 00H

P5PU[5:0] Configure Pull-up Resistor of P5 Port

0 Disable

1 Enable

9.9 Port Function

9.9.1 Port Function Description

Port function control registers consist of Port function selection register 0 ~ 5. (P0FSRH/L ~ P5FSR).

9.9.2 Register description for P0FSRH/L ~ P5FSR

P0FSRH (Port 0 Function Selection High Register) : D3H

7	6	5	4	3	2	1	0
–	–	P0FSRH5	P0FSRH4	P0FSRH3	P0FSRH2	P0FSRH1	P0FSRH0
–	–	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P0FSRH[5:4] P07 Function Select

P0FSRH5	P0FSRH4	Description
0	0	I/OPort (EINT5 function possible when input)
0	1	SEG22 Function
1	0	AN5 Function
1	1	PWM4CB Function

P0FSRH[3:2] P06 Function Select

P0FSRH3	P0FSRH2	Description
0	0	I/OPort (EINT4 function possible when input)
0	1	SEG23 Function
1	0	AN4 Function
1	1	PWM4CA Function

P0FSRH[1:0] P05 Function Select

P0FSRH1	P0FSRH0	Description
0	0	I/OPort (EINT3 function possible when input)
0	1	SEG24 Function
1	0	AN3 Function
1	1	PWM4BB Function

P0FSRL (Port 0 Function Selection Low Register) : D2H

7	6	5	4	3	2	1	0
–	P0FSRL6	P0FSRL5	P0FSRL4	P0FSRL3	P0FSRL2	P0FSRL1	P0FSRL0
–	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P0FSRL[6:5] P04 Function Select

P0FSRL6	P0FSRL5	Description
0	0	I/OPort (EINT2 function possible when input)
0	1	SEG25 Function
1	0	AN2 Function
1	1	PWM4BA Function

P0FSRL[4:3] P03 Function Select

P0FSRL4	P0FSRL3	Description
0	0	I/OPort (EINT1 function possible when input)
0	1	SEG26 Function
1	0	AN1 Function
1	1	PWM4AB Function

P0FSRL[2:1] P02 Function Select

P0FSRL2	P0FSRL1	Description
0	0	I/OPort (EINT0 function possible when input)
0	1	AVREF Function
1	0	AN0 Function
1	1	T4O/PWM4A Function

P0FSRL0 P01 Function Select

0	I/OPort
1	T3O Function

P1FSRH (Port 1 Function Selection High Register) : D5H

7	6	5	4	3	2	1	0
P1FSRH7	P1FSRH6	P1FSRH5	P1FSRH4	P1FSRH3	P1FSRH2	P1FSRH1	P1FSRH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P1FSRH[7:6] P17 Function Select

P1FSRH7	P1FSRH6	Description
0	0	I/OPort (EINT6/SS2 function possible when input)
0	1	SEG21 Function
1	0	AN6 Function
1	1	Not used

P1FSRH[5:4] P16 Function Select

P1FSRH5	P1FSRH4	Description
0	0	I/OPort (EINT7 function possible when input)
0	1	SEG20 Function
1	0	AN7 Function
1	1	SCK2 Function

P1FSRH[3:2] P15 Function Select

P1FSRH3	P1FSRH2	Description
0	0	I/OPort
0	1	SEG19 Function
1	0	AN8 Function
1	1	MISO2 Function

P1FSRH[1:0] P14 Function Select

P1FSRH1	P0FSRH0	Description
0	0	I/OPort
0	1	SEG18 Function
1	0	AN9 Function
1	1	MOSI2 Function

P1FSRL (Port 1 Function Selection Low Register) : D4H

7	6	5	4	3	2	1	0
P1FSRL7	P1FSRL6	P1FSRL5	P1FSRL4	P1FSRL3	P1FSRL2	P1FSRL1	P1FSRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P1FSRL[7:6] P13 Function Select

P1FSRL7	P1FSRL6	Description
0	0	I/OPort (EC1 function possible when input)
0	1	SEG17 Function
1	0	AN10 Function
1	1	BUZO Function

P1FSRL[5:4] P12Function Select

P1FSRL5	P1FSRL4	Description
0	0	I/OPort (EINT11 function possible when input)
0	1	SEG16 Function
1	0	AN11 Function
1	1	T1O/PWM1O Function

P1FSRL[3:2] P11 Function Select

P1FSRL3	P1FSRL2	Description
0	0	I/OPort (EINT12 function possible when input)
0	1	SEG15 Function
1	0	AN12 Function
1	1	T2O/PWM2O Function

P1FSRL[1:0] P10 Function Select

P1FSRL1	P1FSRL0	Description
0	0	I/OPort
0	1	SEG14 Function
1	0	AN13 Function
1	1	RXD1/SCL1/MISO1 Function

P2FSRH (Port 2 Function Selection High Register) : D7H

7	6	5	4	3	2	1	0
–	–	–	–	P2FSRH3	P2FSRH2	P2FSRH1	P2FSRH0
–	–	–	–	R/W	R/W	R/W	R/W

Initial value : 00H

P2FSRH3 P27 Function select
- 0 I/OPort
- 1 SEG6 Function

P2FSRH2 P26 Function Select
- 0 I/OPort
- 1 SEG7 Function

P2FSRH1 P25 Function select
- 0 I/OPort
- 1 SEG8 Function

P2FSRH0 P24 Function Select
- 0 I/OPort
- 1 SEG9 Function

P2FSRL (Port 2 Function Selection Low Register) : D6H

7	6	5	4	3	2	1	0
–	–	P2FSRL5	P2FSRL4	P2FSRL3	P2FSRL2	P2FSRL1	P2FSRL0
–	–	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P2FSRL5 P23 Function Select

0	I/OPort
1	SEG10 Function

P2FSRL4 P22Function Select

0	I/OPort (SS1 function possible when input)
1	SEG11 Function

P2FSRL[3:2] P21 Function Select

P2FSRL3	P2FSRL2	Description
0	0	I/OPort
0	1	SEG12 Function
1	0	Not used
1	1	SCK1 Function

P2FSRL[1:0] P20 Function Select

P2FSRL1	P1FSRL0	Description
0	0	I/OPort
0	1	SEG13 Function
1	0	AN14 Function
1	1	TXD1/SDA1/MOSI1 Function

P3FSR (Port 3 Function Selection Register) : EEH

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 P37 Function select
- 0 I/OPort
- 1 COM0 Function

P3FSR6 P36 Function Select
- 0 I/OPort
- 1 COM1 Function

P3FSR5 P35 Function select
- 0 I/OPort
- 1 COM2/SEG0 Function

P3FSR4 P34 Function Select
- 0 I/OPort
- 1 COM3/SEG1 Function

P3FSR3 P33 Function select
- 0 I/OPort
- 1 COM4/SEG2 or COM0 Function

P3FSR2 P32 Function Select
- 0 I/OPort
- 1 COM5/SEG3 or COM1 Function

P3FSR1 P31 Function select
- 0 I/OPort
- 1 COM6/SEG4 or COM2/SEG4 Function

P3FSR0 P30 Function Select
- 0 I/OPort
- 1 COM7/SEG5 or COM3/SEG5 Function

NOTES) 1. The P30-P35 is automatically configured as common or segment signal according to the duty in the LCDCRL register when the pin is selected as the sub-function for common/segment.

2. The COM0-COM3 signals can be outputted through the P33-P30 pins. Refer to the LCD drive control high register (LCDCRH).

.

P4FSR (Port 4 Function Selection Register) : EFH

7	6	5	4	3	2	1	0
–	P4FSR6	P4FSR5	P4FSR4	P4FSR3	P4FSR2	P4FSR1	P4FSR0
–	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P4FSR6 P43 Function Select

0	I/OPort (SS0 function possible when input)
1	VLC0 Function

P4FSR[5:4] P42 Function Select

P4FSR5	P4FSR4	Description
0	0	I/OPort
0	1	VLC1 Function
1	0	SCK0 Function
1	1	Not used

P4FSR[3:2] P41 Function Select

P4FSR3	P4FSR2	Description
0	0	I/OPort
0	1	VLC2 Function
1	0	TXD0/SDA0/MOSI0 Function
1	1	Not used

P4FSR6[1:0] P40 Function Select

P4FSR1	P4FSR0	Description
0	0	I/OPort
0	1	VLC3 Function
1	0	RXD0/SCL0/MISO0 Function
1	1	Not used

P5FSR (Port 5 Function Selection Register) : FFH

7	6	5	4	3	2	1	0
–	–	P5FSR5	P5FSR4	P5FSR3	P5FSR2	P5FSR1	P5FSR0
–	–	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P5FSR5 P54 Function Select

0	I/OPort (EINT10 function possible when input)
1	SXOUT Function

P5FSR[4:3] P53 Function Select

P5FSR4	P5FSR3	Description
0	0	I/OPort
0	1	SXIN Function
1	0	T0O/PWM0O Function
1	1	Not used

P5FSR2 P51 Function Select

1	0	I/OPort
1	1	XIN Function

P5FSR[1:0] P50 Function Select

P5FSR1	P5FSR0	Description
0	0	I/OPort
0	1	XOUT Function
1	0	Not used
1	1	Not used

NOTE) Refer to the configure option for the P55/RESETB.

10. Interrupt Controller

10.1 Overview

The MC96F6432/F6332/F6232 supports up to 23 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 23 interrupt source
- 6 group 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 MC96F6432/F6332/F6232 supports a four-level priority scheme. Each maskable interrupt is individually assigned to one of four priority levels according to IP and IP1.

Default interrupt mode is level-trigger mode basically, but if needed, it is possible to change to edge-trigger mode. Table 10-1 shows the Interrupt Group Priority Level that is available for sharing interrupt priority. 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.

Table 10-1 Interrupt Group Priority Level

Interrupt Group	Highest → Lowest				
0 (Bit0)	Interrupt 0	Interrupt 6	Interrupt 12	Interrupt 18	Highest
1 (Bit1)	Interrupt 1	Interrupt 7	Interrupt 13	Interrupt 19	
2 (Bit2)	Interrupt 2	Interrupt 8	Interrupt 14	Interrupt 20	
3 (Bit3)	Interrupt 3	Interrupt 9	Interrupt 15	Interrupt 21	
4 (Bit4)	Interrupt 4	Interrupt 10	Interrupt 16	Interrupt 22	
5 (Bit5)	Interrupt 5	Interrupt 11	Interrupt 17	Interrupt 23	Lowest

10.2 External Interrupt

The external interrupt on INT0, INT1, INT5, INT6 and INT11 pins receive various interrupt request depending on the external interrupt polarity 0 high/low register (EIPOL0H/L) and external interrupt polarity 1 register (EIPOL1) as shown in Figure 10.1. Also each external interrupt source has enable/disable bits. The External interrupt flag 0 register (EIFLAG0) and external interrupt flag 1 register 1 (EIFLAG1) provides the status of external interrupts.

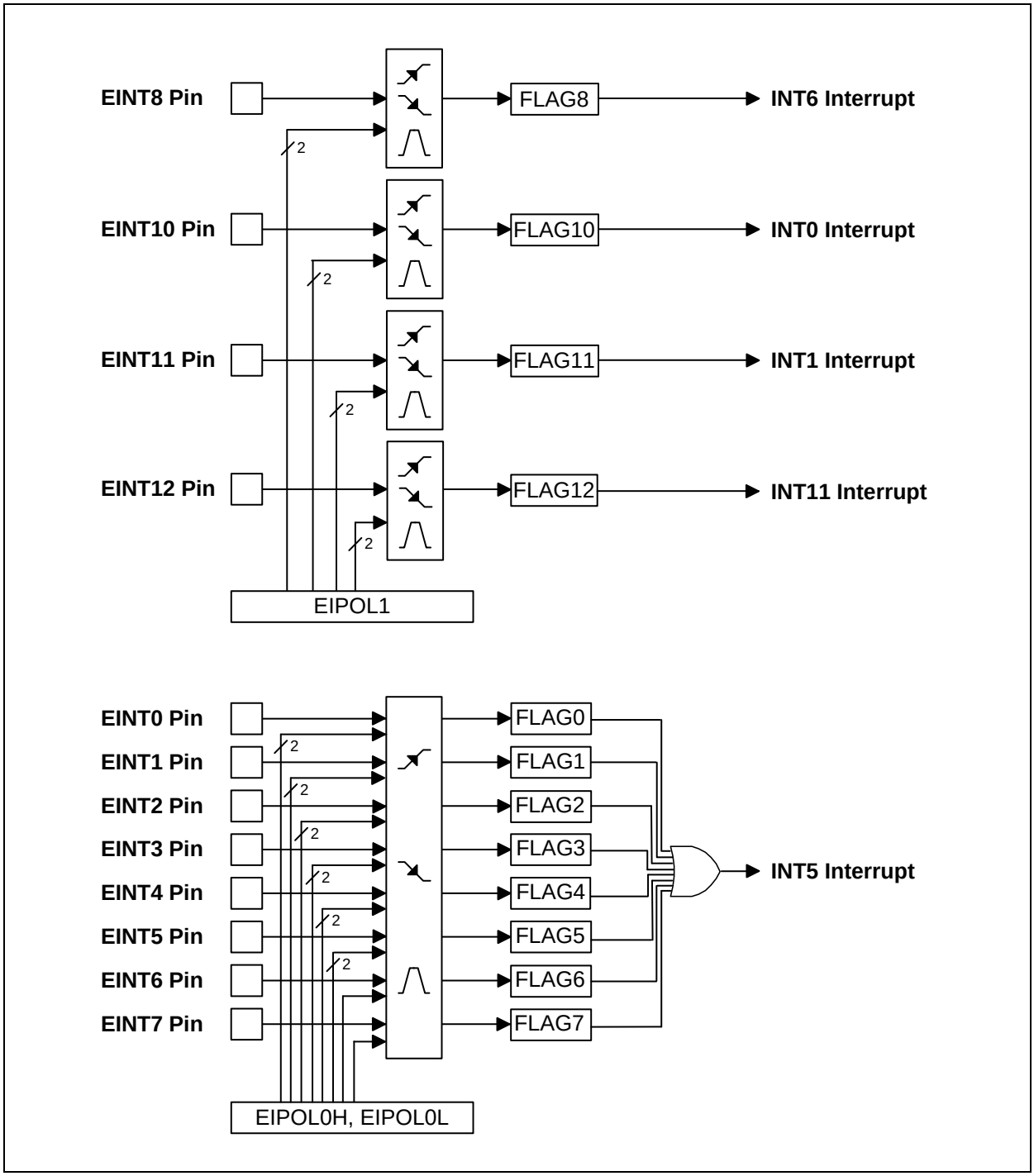


Figure 10.1 External Interrupt Description

10.3 Block Diagram

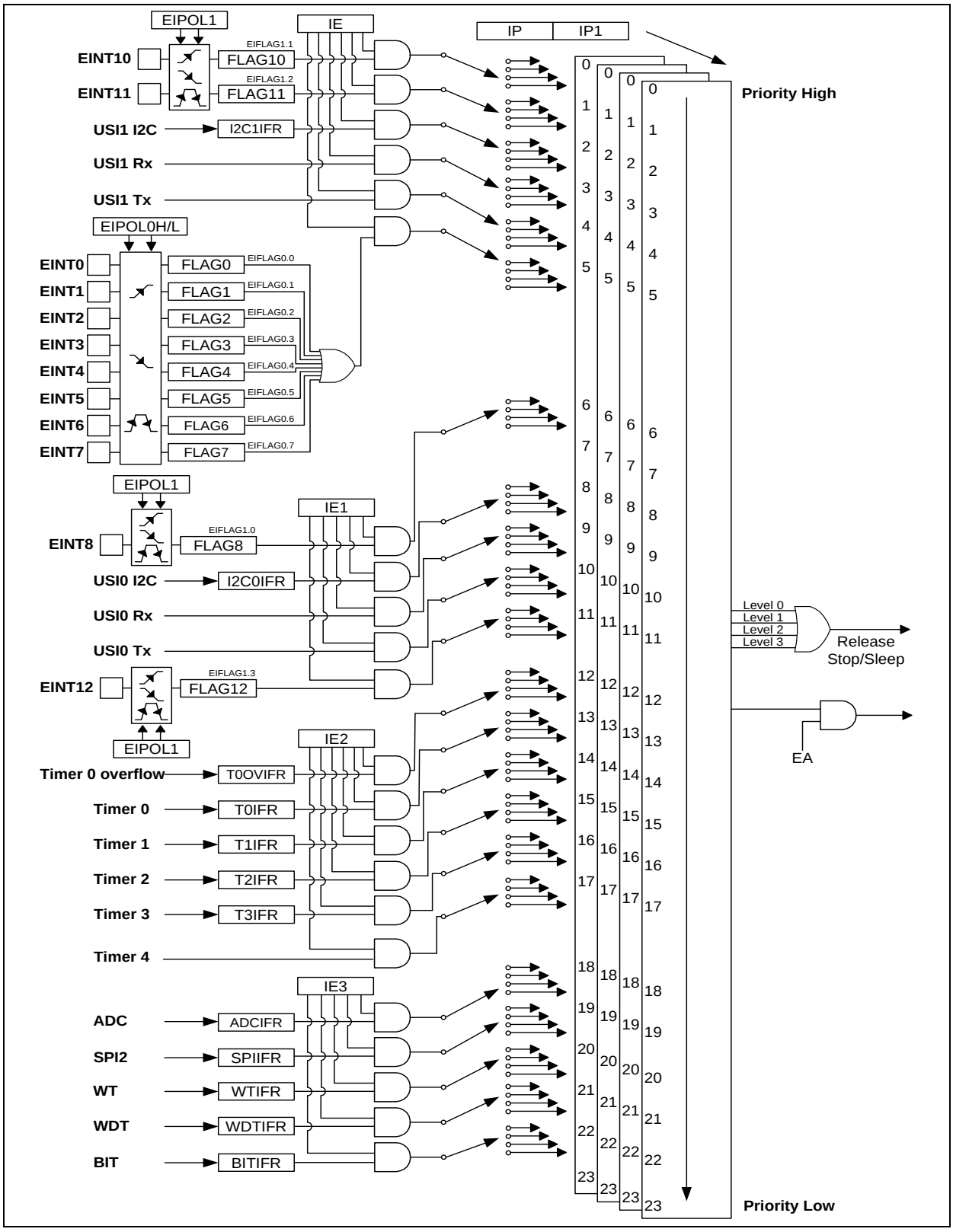


Figure 10.2 Block Diagram of Interrupt

NOTES) 1. The release signal for stop/idle mode may be generated by all interrupt sources which are enabled without reference to the priority level.

2. An interrupt request is delayed while data are written to IE, IE1, IE2, IE3, IP, IP1, 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 10-2 Interrupt Vector Address Table

Interrupt Source	Symbol	Interrupt Enable Bit	Polarity	Mask	Vector Address
Hardware Reset	RESETB	0 0	0	Non-Maskable	0000H
External Interrupt 10	INT0	IE.0	1	Maskable	0003H
External Interrupt 11	INT1	IE.1	2	Maskable	000BH
USI1 I2C Interrupt	INT2	IE.2	3	Maskable	0013H
USI1 Rx Interrupt	INT3	IE.3	4	Maskable	001BH
USI1 Tx Interrupt	INT4	IE.4	5	Maskable	0023H
External Interrupt 0 - 7	INT5	IE.5	6	Maskable	002BH
External Interrupt 8	INT6	IE1.0	7	Maskable	0033H
-	INT7	IE1.1	8	Maskable	003BH
USI0 I2C Interrupt	INT8	IE1.2	9	Maskable	0043H
USI0 Rx Interrupt	INT9	IE1.3	10	Maskable	004BH
USI0 Tx Interrupt	INT10	IE1.4	11	Maskable	0053H
External Interrupt 12	INT11	IE1.5	12	Maskable	005BH
T0 Overflow Interrupt	INT12	IE2.0	13	Maskable	0063H
T0 Match Interrupt	INT13	IE2.1	14	Maskable	006BH
T1 Match Interrupt	INT14	IE2.2	15	Maskable	0073H
T2 Match Interrupt	INT15	IE2.3	16	Maskable	007BH
T3 Match Interrupt	INT16	IE2.4	17	Maskable	0083H
T4 Interrupt	INT17	IE2.5	18	Maskable	008BH
ADC Interrupt	INT18	IE3.0	19	Maskable	0093H
SPI 2 Interrupt	INT19	IE3.1	20	Maskable	009BH
WT Interrupt	INT20	IE3.2	21	Maskable	00A3H
WDT Interrupt	INT21	IE3.3	22	Maskable	00ABH
BIT Interrupt	INT22	IE3.4	23	Maskable	00B3H
-	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.

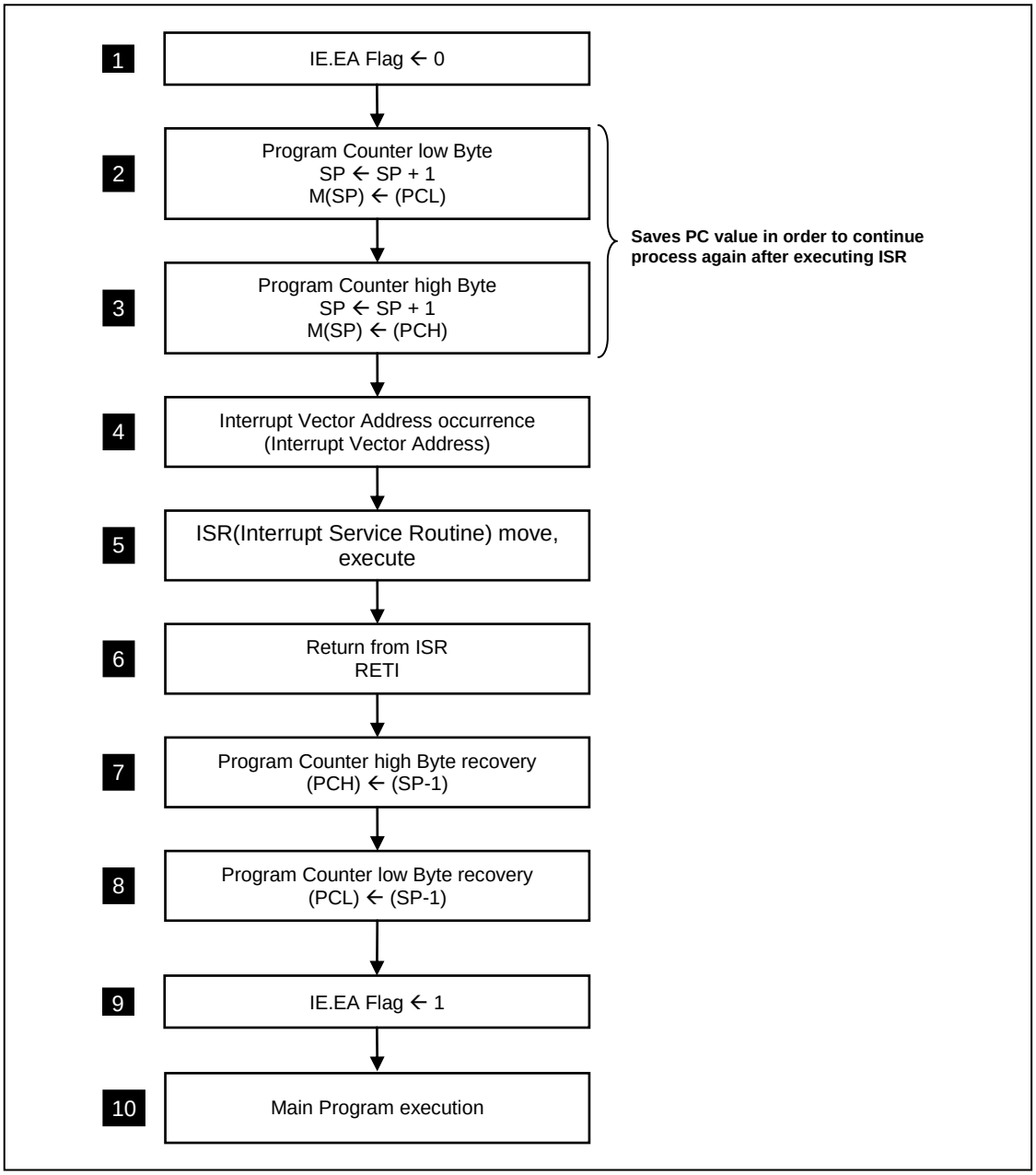


Figure 10.3 Interrupt Vector Address Table

10.6 Effective Timing after Controlling Interrupt Bit

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

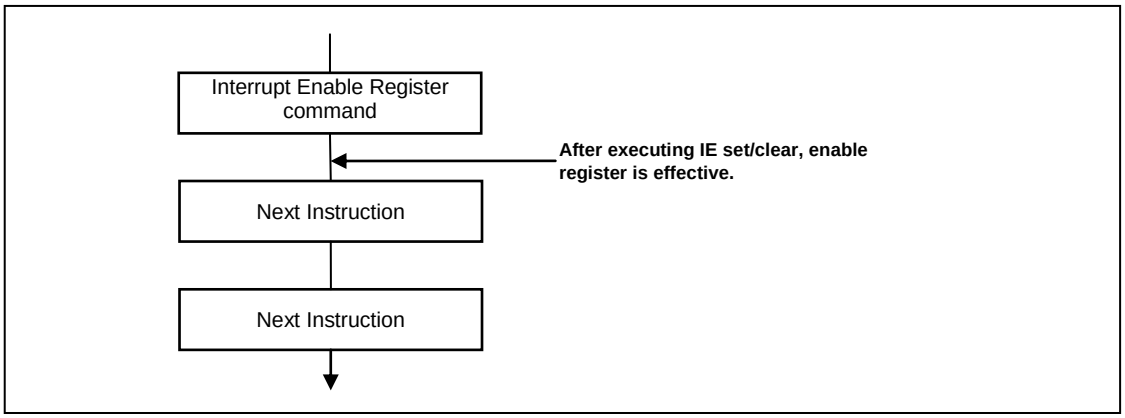


Figure 10.4 Effective Timing of Interrupt Enable Register

Case b) Interrupt flag Register

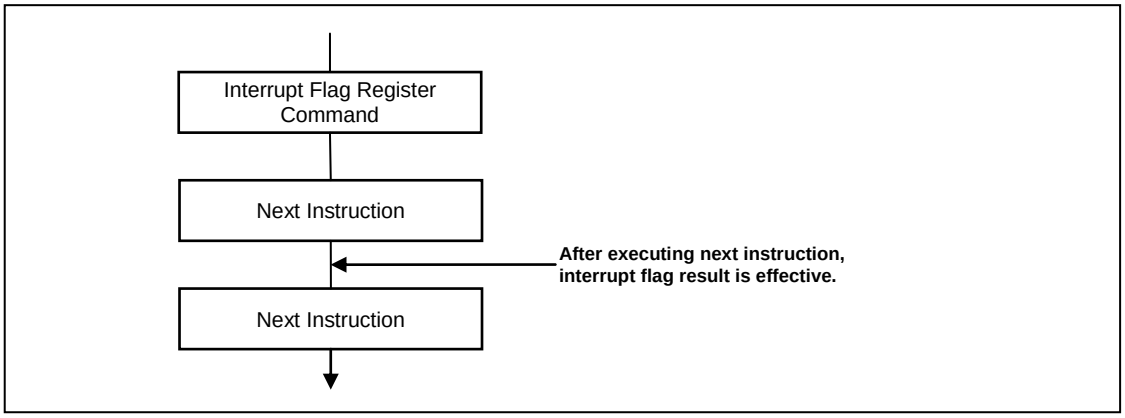


Figure 10.5 Effective Timing of Interrupt Flag Register

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.

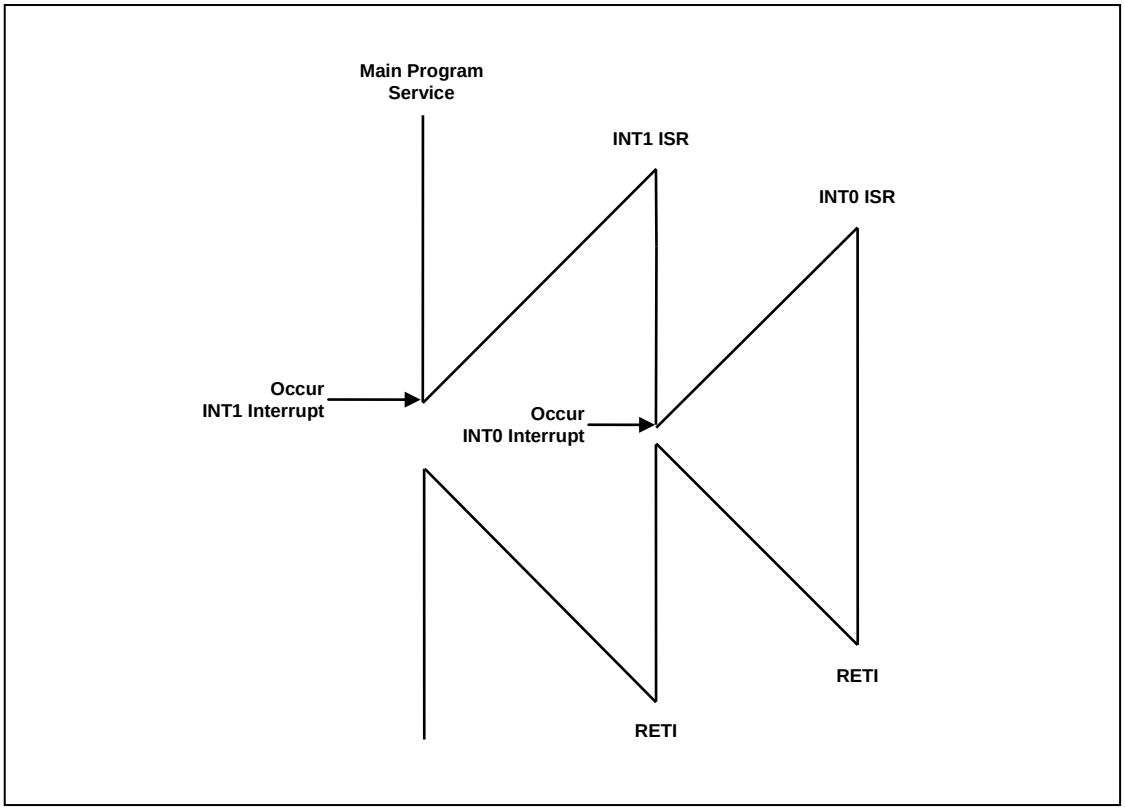


Figure 10.6 Effective Timing of Interrupt

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

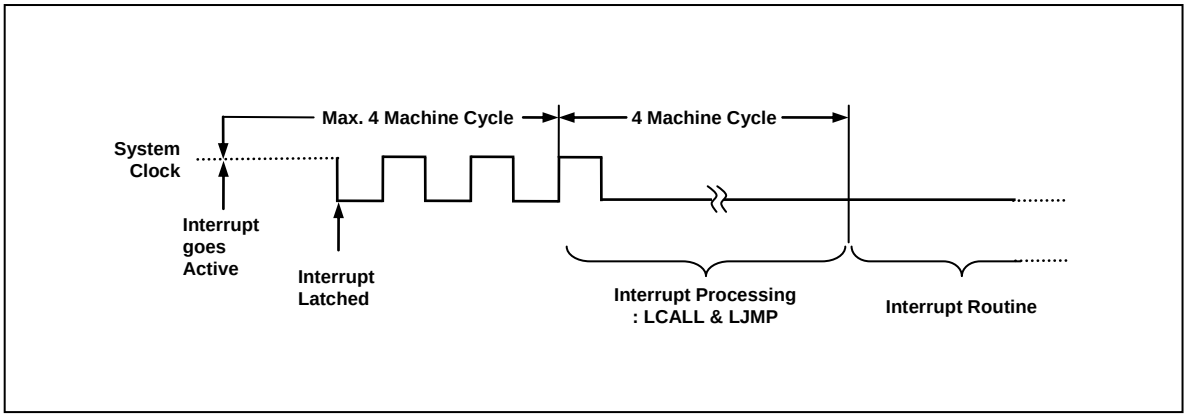


Figure 10.7 Interrupt Response Timing Diagram

10.9 Interrupt Service Routine Address

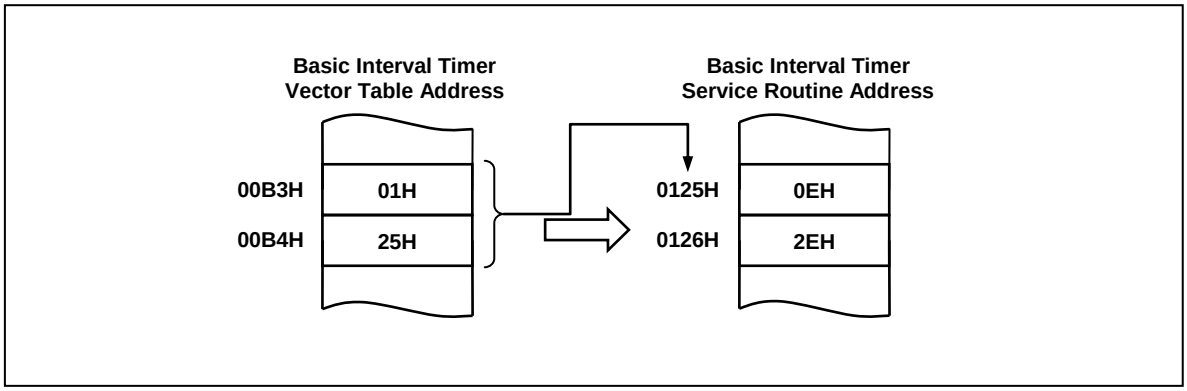


Figure 10.8 Correspondence between Vector Table Address and the Entry Address of ISP

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
```

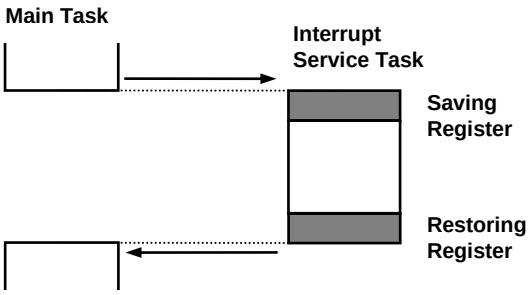


Figure 10.9 Saving/Restore Process Diagram and Sample Source

10.11 Interrupt Timing

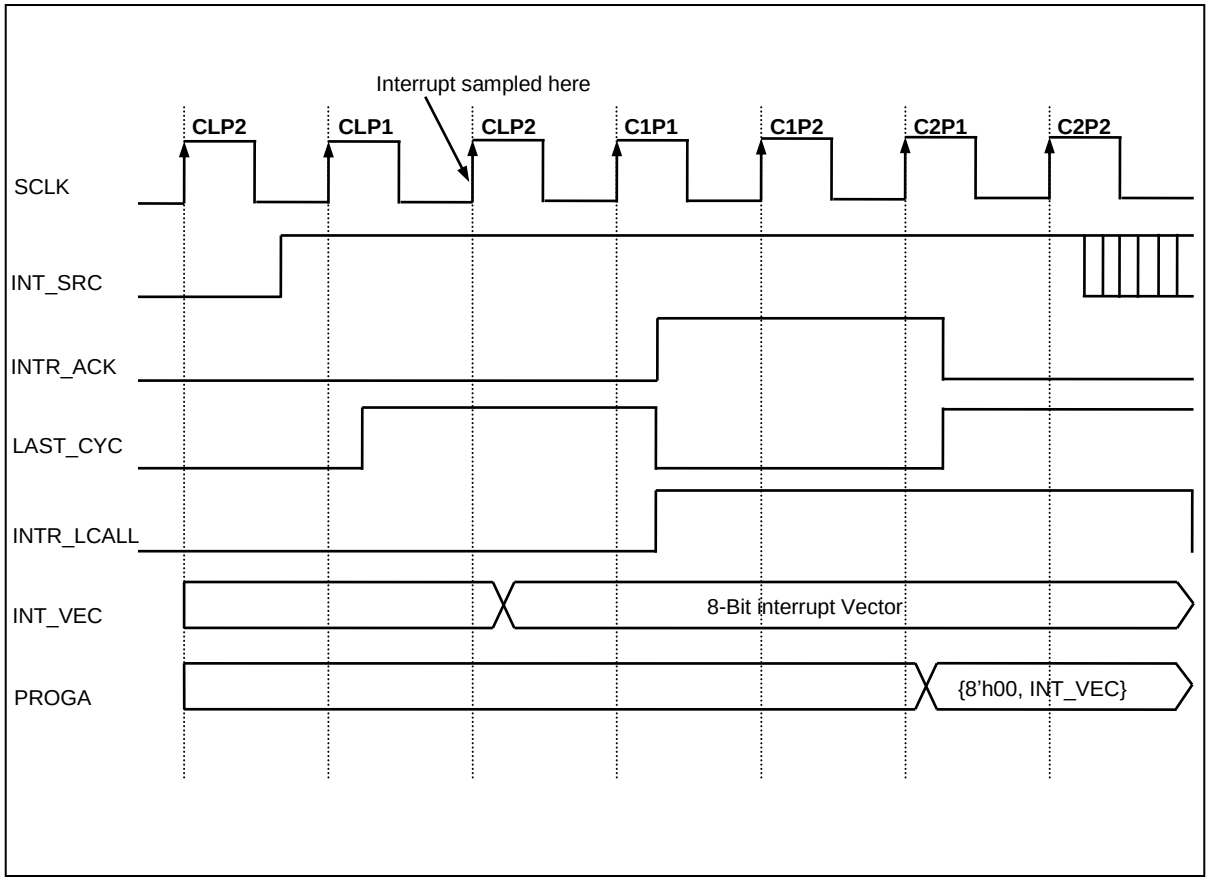


Figure 10.10 Timing Chart of Interrupt Acceptance and Interrupt Return Instruction

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 (IP, IP1)

The 24 interrupts are divided into 6 groups which have each 4 interrupt sources. A group 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 IP and IP1 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)

The external interrupt flag 0 register (EIFLAG0) and external interrupt flag 1 register (EIFLAG1) 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 (EIPOL0L, EIPOL0H, EIPOL1)

The external interrupt polarity 0 high/low register (EIPOL0H/L) and external interrupt polarity 1 register (EIPOL1) determines which type of rising/falling/both edge interrupt. Initially, default value is no interrupt at any edge.

10.12.5 Register Map

Table 10-3 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
IP	B8H	R/W	00H	Interrupt Polarity Register
IP1	F8H	R/W	00H	Interrupt Polarity Register 1
EIFLAG0	C0H	R/W	00H	External Interrupt Flag 0 Register
EIPOL0L	A4H	R/W	00H	External Interrupt Polarity 0 Low Register
EIPOL0H	A5H	R/W	00H	External Interrupt Polarity 0 High Register
EIFLAG1	A6H	R/W	00H	External Interrupt Flag 1 Register
EIPOL1	A7H	R/W	00H	External Interrupt Polarity 1 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) and external interrupt polarity 1 register (EIPOL1).

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 USI1 Tx Interrupt
- 0 Disable
- 1 Enable

INT3E Enable or Disable USI1 Rx Interrupt
- 0 Disable
- 1 Enable

INT2E Enable or Disable USI1 I2C Interrupt
- 0 Disable
- 1 Enable

INT1E Enable or Disable External Interrupt 11(EINT11)
- 0 Disable
- 1 Enable

INT0E Enable or Disable External Interrupt 10 (EINT10)
- 0 Disable
- 1 Enable

IE1 (Interrupt Enable Register 1): A9H

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

Initial value: 00H

INT11E Enable or Disable External Interrupt 12 (EINT12)
- 0 Disable
- 1 Enable

INT10E Enable or Disable USI0 Tx Interrupt
- 0 Disable
- 1 Enable

INT9E Enable or Disable USI0 Rx Interrupt
- 0 Disable
- 1 Enable

INT8E Enable or Disable USI0 I2C Interrupt
- 0 Disable
- 1 Enable

INT6E Enable or Disable External Interrupt 8 (EINT8)
- 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 4 Interrupt
0 Disable
1 Enable

INT16E Enable or Disable Timer 3 Match Interrupt
0 Disable
1 Enable

INT15E Enable or Disable Timer 2 Match Interrupt
0 Disable
1 Enable

INT14E Enable or Disable Timer 1 Match Interrupt
0 Disable
1 Enable

INT13E Enable or Disable Timer 0 I Match nterrupt
0 Disable
1 Enable

INT12E Enable or Disable Timer 0 Overflow Interrupt
0 Disable
1 Enable

IE3 (Interrupt Enable Register 3) : ABH

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

Initial value : 00H

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 Interrupt
0 Disable
1 Enable

INT19E Enable or Disable SPI 2 Interrupt
0 Disable
1 Enable

INT18E Enable or Disable ADC Interrupt
0 Disable
1 Enable

IP (Interrupt Priority Register) : B8H

7	6	5	4	3	2	1	0
–	–	IP5	IP4	IP3	IP2	IP1	IP0
–	–	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

IP1 (Interrupt Priority Register 1) : F8H

7	6	5	4	3	2	1	0
–	–	IP15	IP14	IP13	IP12	IP11	IP10
–	–	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

IP[5:0], IP1[5:0] Select Interrupt Group Priority

IP1x	IPx	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) : C0H

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[7:0] When an External Interrupt 0-7 is occurred, the flag becomes '1'. The flag is cleared only by writing '0' to the bit. 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): A5H

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): A4H

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 (EINT0, EINT1, EINT2, EINT3) 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) : A6H

7	6	5	4	3	2	1	0
T0OVIFR	T0IFR	T3IFR	–	FLAG12	FLAG11	FLAG10	FLAG8
R/W	R/W	R/W	–	R/W	R/W	R/W	R/W

Initial value : 00H

T0OVIFR When T0 overflow interrupt occurs, this bit becomes '1'. For clearing bit, write '0' to this bit or automatically clear by INT_ACK signal.

0 T0 overflow Interrupt no generation

1 T0 overflow Interrupt generation

T0IFR When T0 interrupt occurs, this bit becomes '1'. For clearing bit, write '0' to this bit or automatically clear by INT_ACK signal.

0 T0 Interrupt no generation

1 T0 Interrupt generation

T3IFR When T3 interrupt occurs, this bit becomes '1'. For clearing bit, write '0' to this bit or automatically clear by INT_ACK signal.

0 T3 Interrupt no generation

1 T3 Interrupt generation

EIFLAG1[3:0] When an External Interrupt (EINT8, EINT10-EINT12) is occurred, the flag becomes '1'. The flag is cleared by writing '0' to the bit or automatically cleared by INT_ACK signal.

0 External Interrupt not occurred

1 External Interrupt occurred

EIPOL1 (External Interrupt Polarity 1 Register): A7H

7	6	5	4	3	2	1	0
POL12		POL11		POL10		POL8	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

EIPOL1[7:0] External interrupt (EINT8,EINT10,EINT11,EINT12) 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, 10, 11 and 12

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 eight. 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 (0.4~12 MHz)
- Sub Crystal Oscillator (32.768 kHz)
- Internal WDTRC Oscillator (6 kHz)

11.1.2 Block Diagram

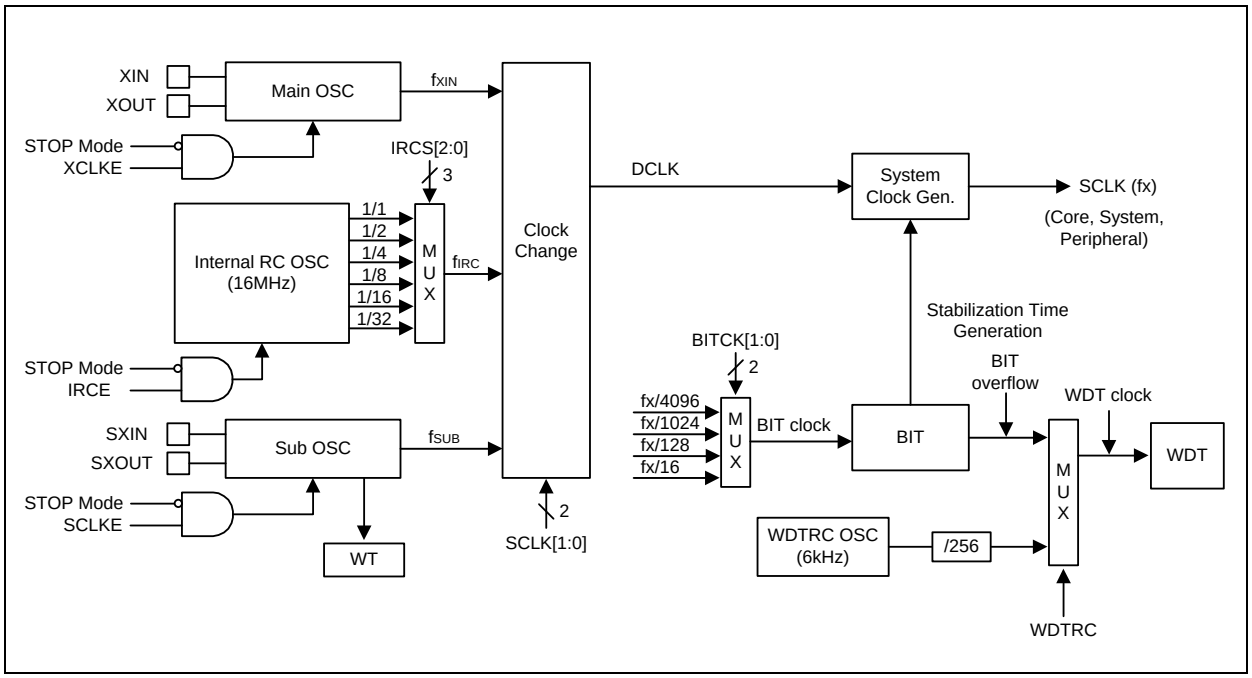


Figure 11.1 Clock Generator Block Diagram

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	20H	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 MC96F6432/F6332/F6232 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 MC96F6432/F6332/F6232 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

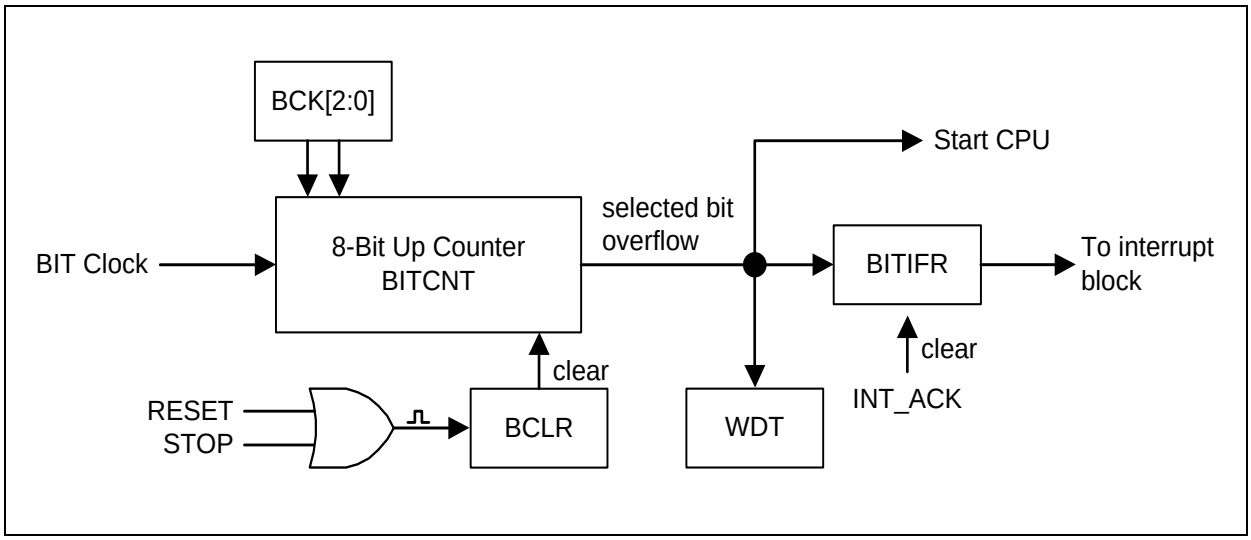


Figure 11.2 Basic Interval Timer Block Diagram

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.

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

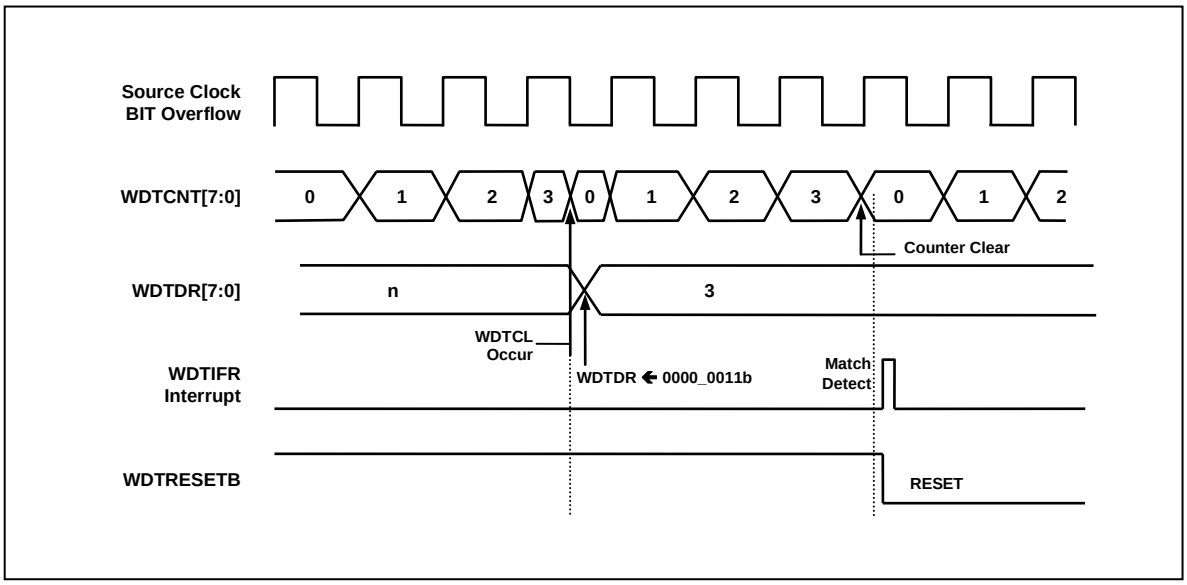


Figure 11.3 Watch Dog Timer Interrupt Timing Waveform

11.3.3 Block Diagram

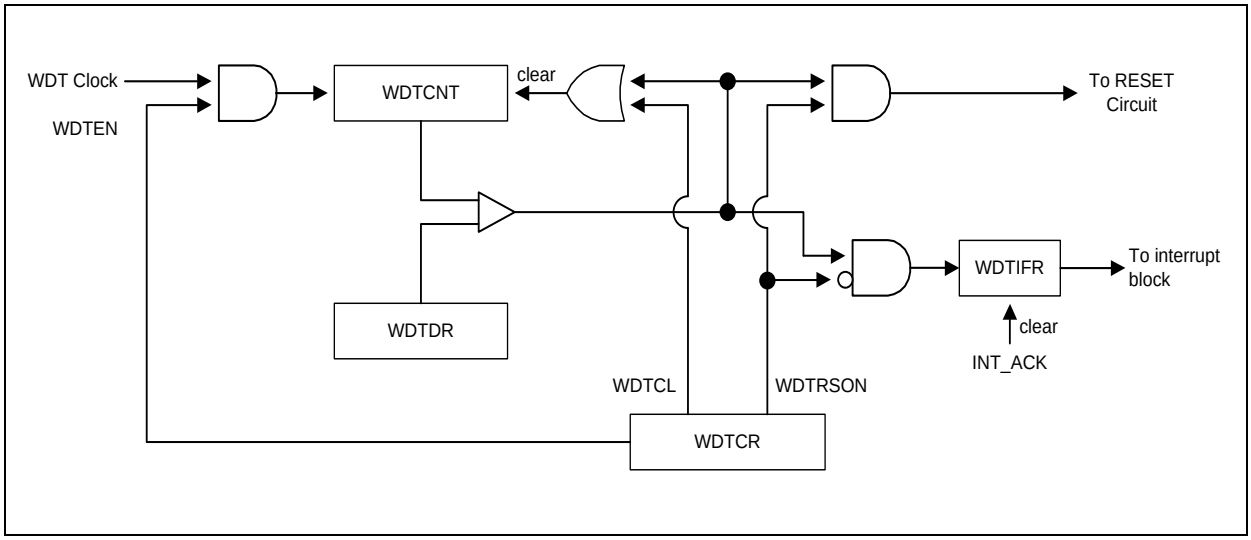


Figure 11.4 Watch Dog Timer Block Diagram

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	–	–	–	WDTRC	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)

WDTRC Control WDT Clock Selection Bit
0 BIT overflow for WDT clock (WDTRC disable)
1 WDTRC for WDT xlock (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.
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.

The watch timer supplies the clock frequency for the LCD driver (f_{LCD}). Therefore, if the watch timer is disabled, the LCD driver controller does not operate.

11.4.2 Block Diagram

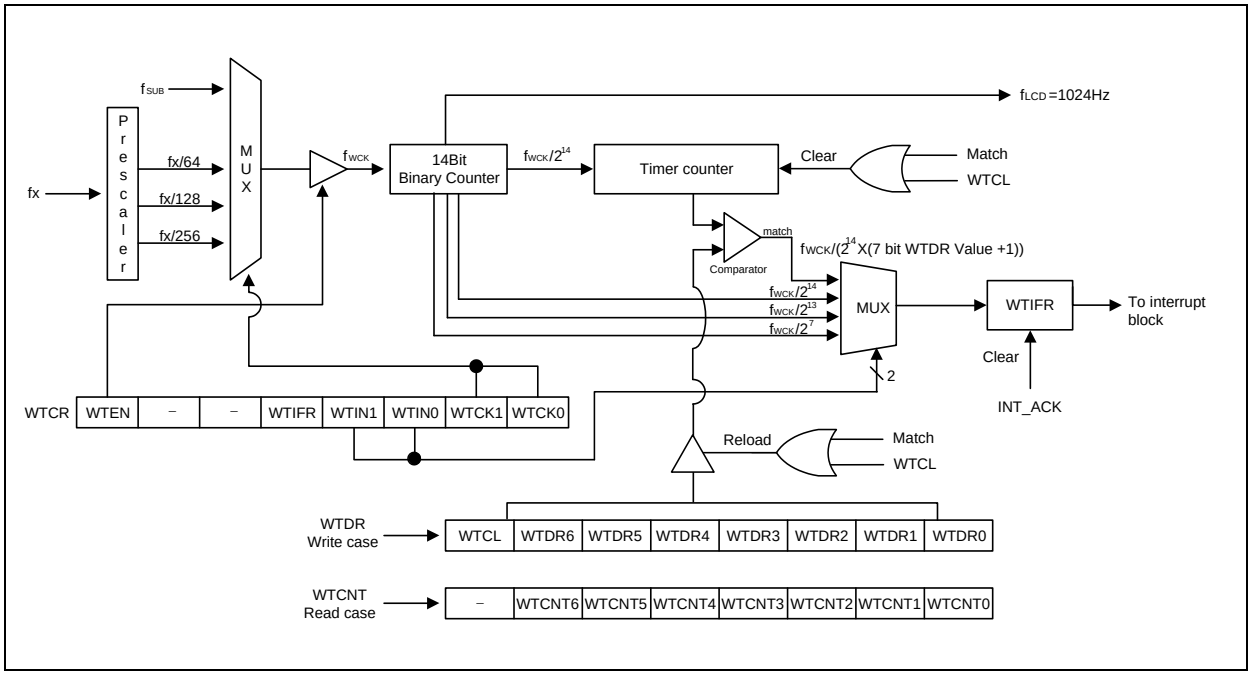


Figure 11.5 Watch Timer Block Diagram

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	96H	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
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) : 96H

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

Initial value : 00H

WTEN Control Watch Timer

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 INT_ACK signal.

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

f_{LCD} – LCD frequency (Where f_X= 4.19MHz, WTCK[1:0]='10'; f_{LCD}= 1024Hz)

11.5 Timer 0

11.5.1 Overview

The 8-bit timer 0 consists of multiplexer, timer 0 counter register, timer 0 data register, timer 0 capture data register and timer 0 control register (T0CNT, T0DR, T0CDR, T0CR).

It has three operating modes:

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

The timer/counter 0 can be clocked by an internal or an external clock source (EC0). The clock source is selected by clock selection logic which is controlled by the clock selection bits (T0CK[2:0]).

- TIMER 0 clock source: f_X/2, 4, 8, 32, 128, 512, 2048 and EC0

In the capture mode, by EINT10, the data is captured into input capture data register (T0CDR). In timer/counter mode, whenever counter value is equal to T0DR, T0O port toggles. Also the timer 0 outputs PWM waveform through PWM0O port in the PWM mode.

Table 11-5 Timer 0 Operating Modes

T0EN	T0MS[1:0]	T0CK[2:0]	Timer 0
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 can use the input clock with one of 2, 4, 8, 32, 128, 512 and 2048 prescaler division rates (T0CK[2:0]). When the value of T0CNT and T0DR is identical in timer 0, a match signal is generated and the interrupt of Timer 0 occurs. T0CNT value is automatically cleared by match signal. It can be also cleared by software (T0CC).

The external clock (EC0) counts up the timer at the rising edge. If the EC0 is selected as a clock source by T0CK[2:0], EC0 port should be set to the input port by P52IO bit.

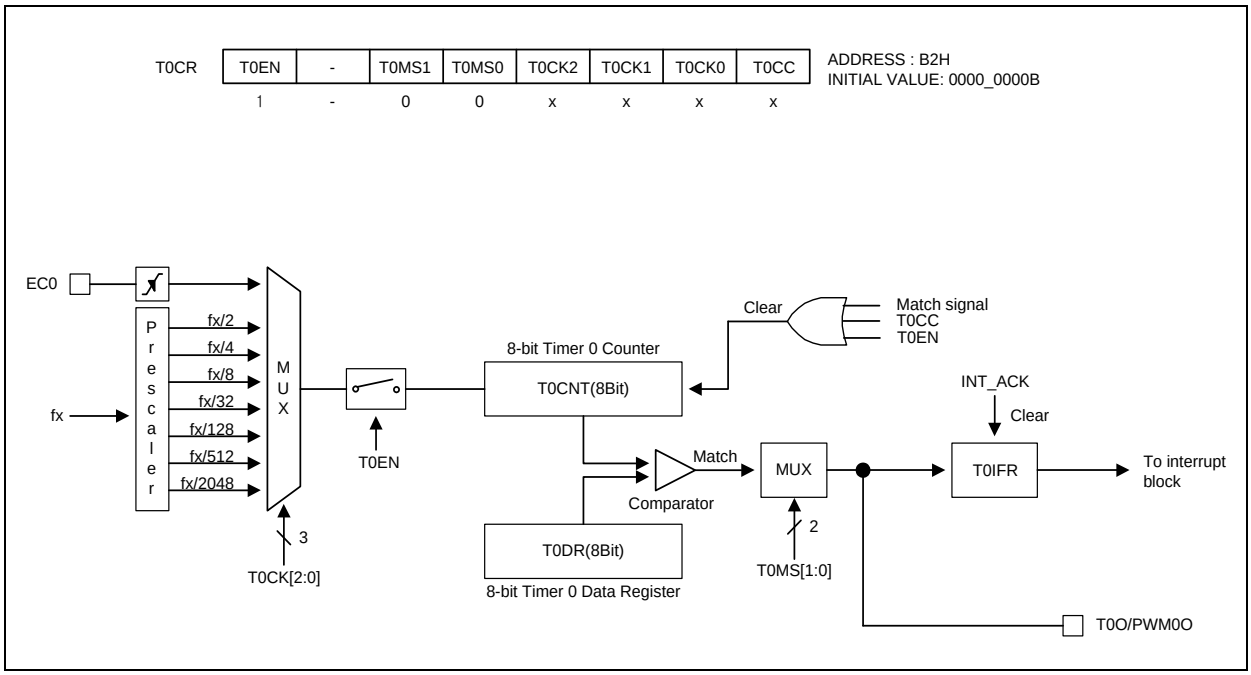


Figure 11.6 8-Bit Timer/Counter Mode for Timer 0

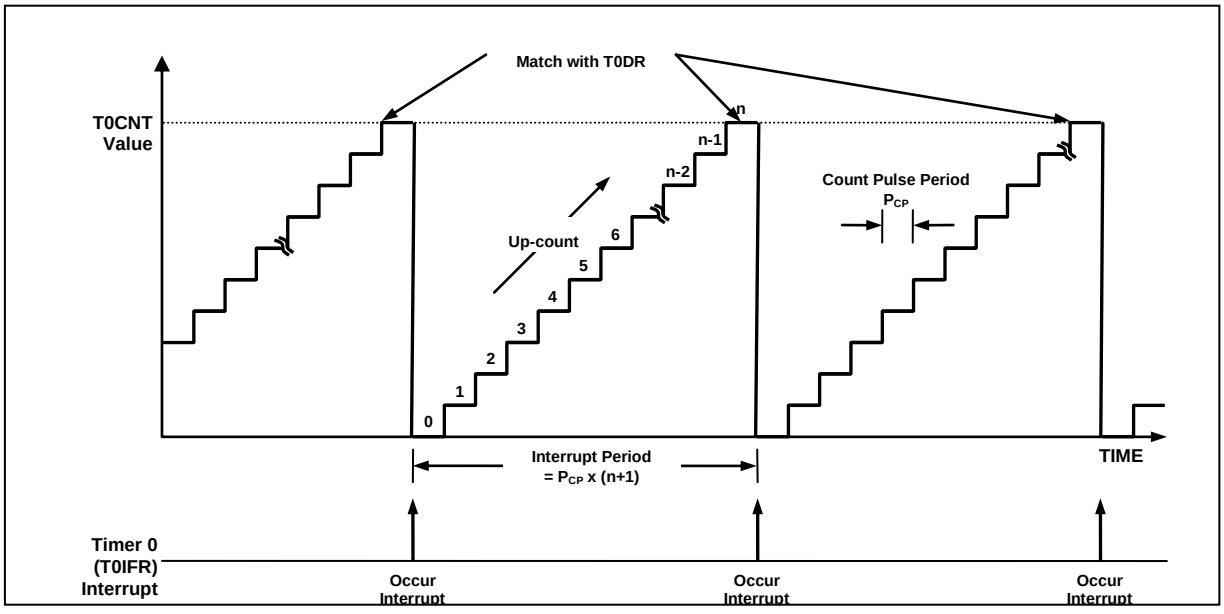


Figure 11.7 8-Bit Timer/Counter 0 Example

11.5.3 8-Bit PWM Mode

The timer 0 has a high speed PWM (Pulse Width Modulation) function. In PWM mode, T0O/PWM0O pin outputs up to 8-bit resolution PWM output. This pin should be configured as a PWM output by setting the T0O/PWM0O function by P5FSR[4:3] bits. In the 8-bit timer/counter mode, a match signal is generated when the counter value is identical to the value of T0DR. When the value of T0CNT and T0DR is identical in timer 0, a match signal is generated and the interrupt of timer 0 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 overflow interrupt is generated whenever a counter overflow occurs. T0CNT value is cleared by software (T0CC) bit.

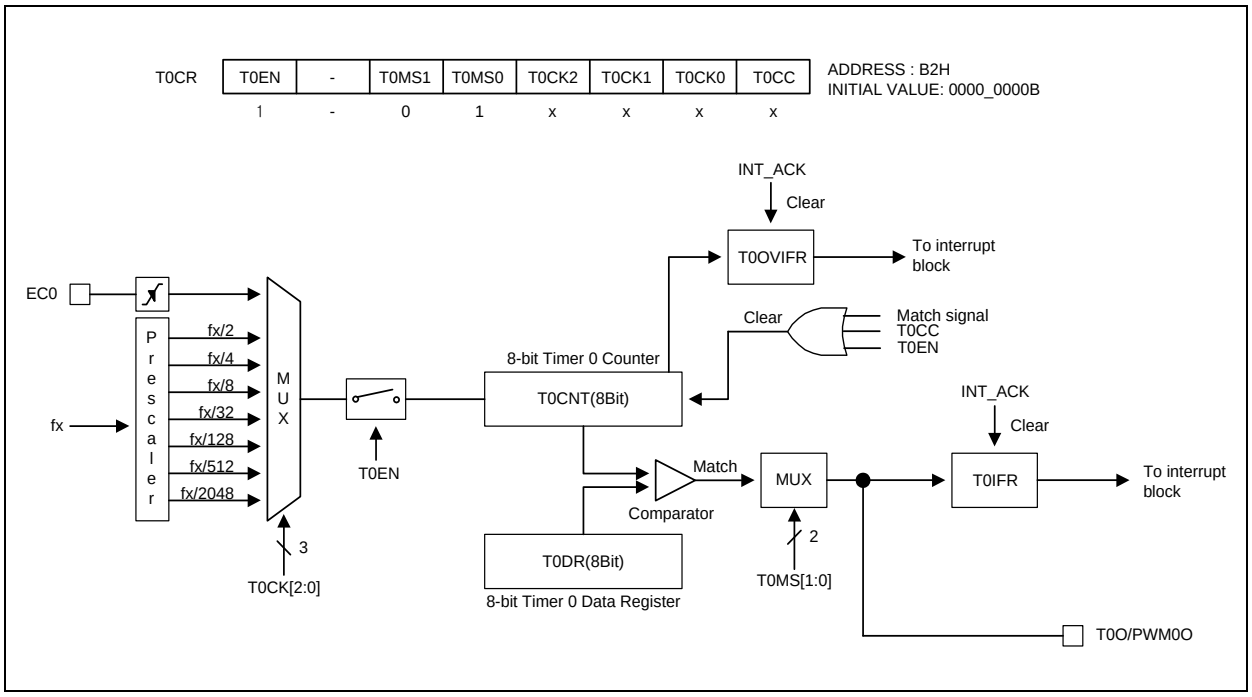


Figure 11.8 8-Bit PWM Mode for Timer 0

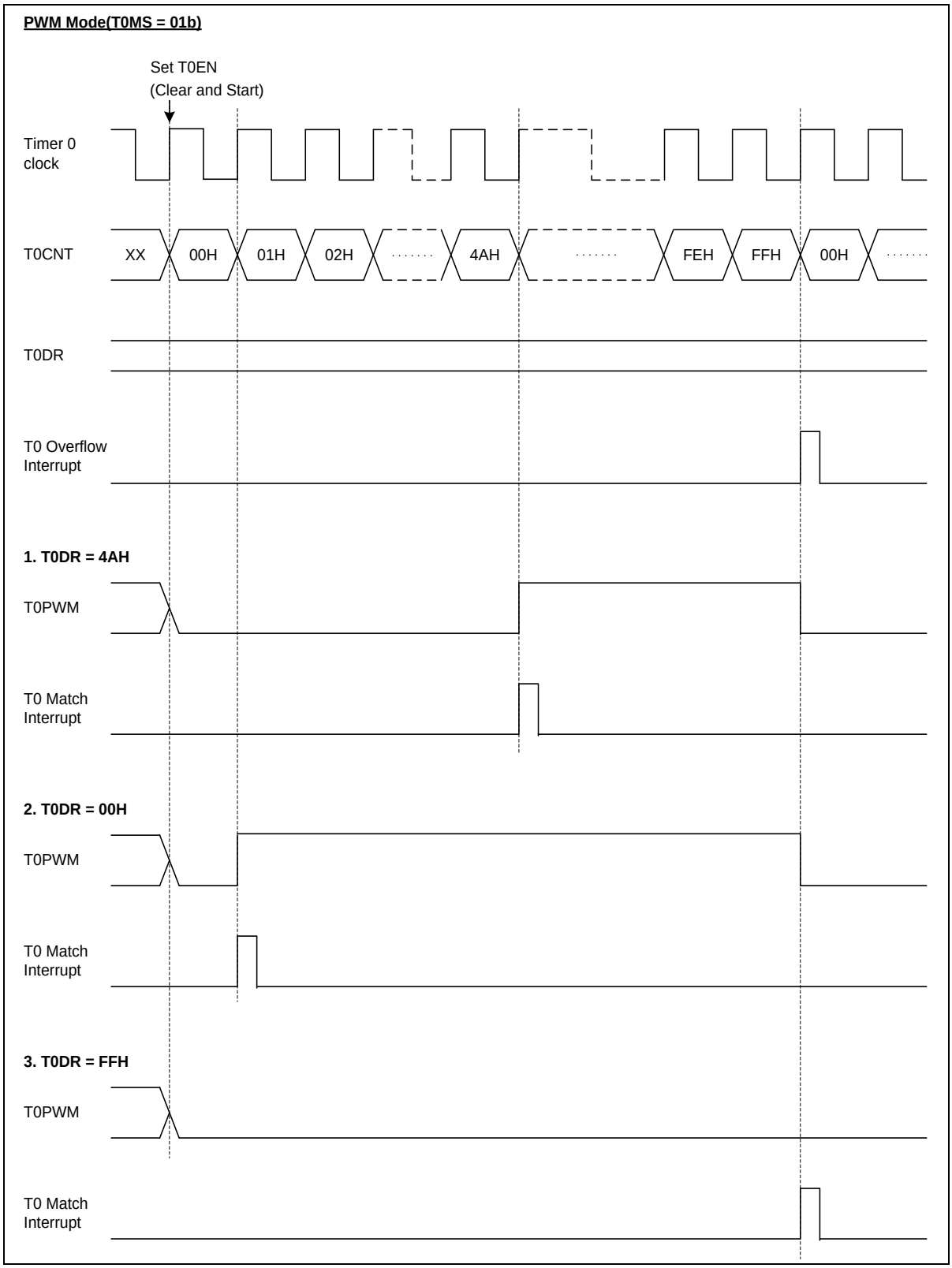


Figure 11.9 PWM Output Waveforms in PWM Mode for Timer 0

11.5.4 8-Bit Capture Mode

The timer 0 capture mode is set by T0MS[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 T0CNT is equal to T0DR. T0CNT value is automatically cleared by match signal and it can be also cleared by software (T0CC).

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 T0CDR. In the timer 0 capture mode, timer 0 output (T0O) waveform is not available.

According to EIPOL1 registers setting, the external interrupt EINT10 function is chosen. Of cource, the EINT10 pin must be set to an input port.

T0CDR and T0DR are in the same address. In the capture mode, reading operation reads T0CDR, not T0DR and writing operation will update T0DR.

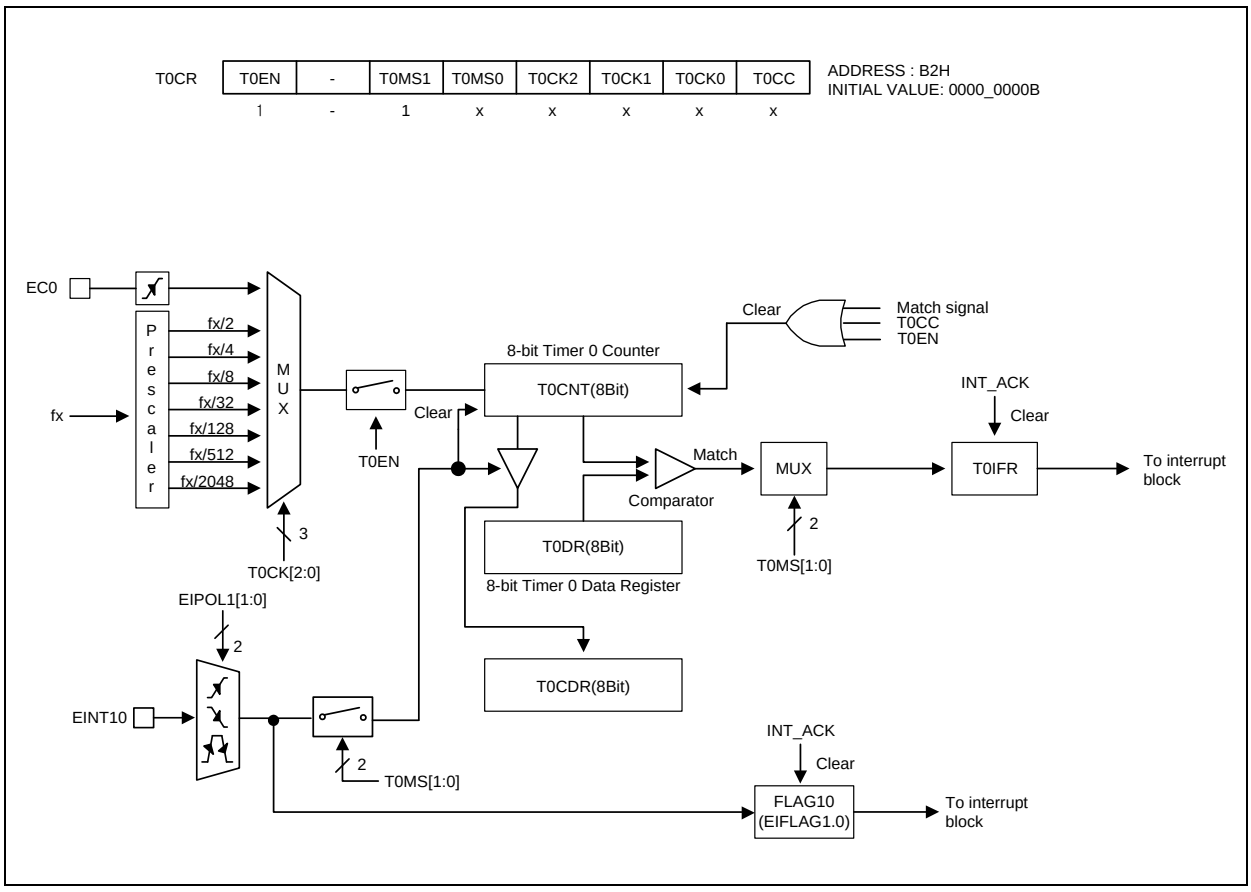


Figure 11.10 8-Bit Capture Mode for Timer 0

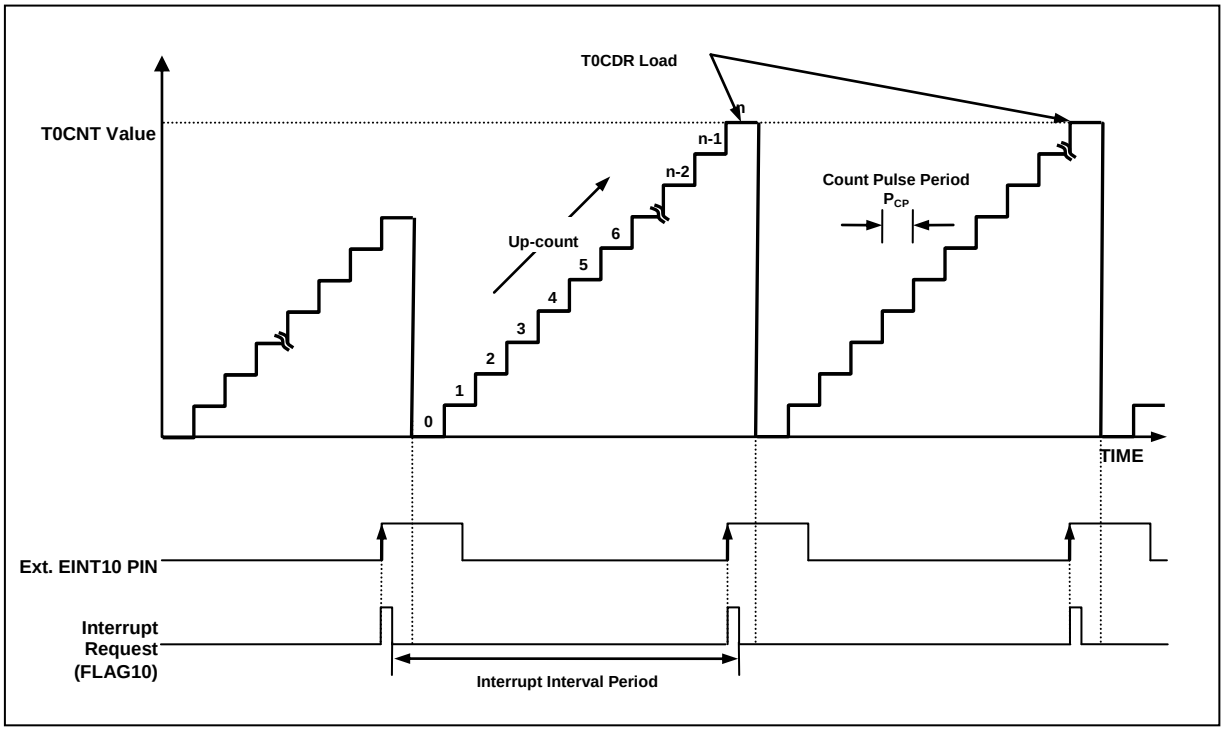


Figure 11.11 Input Capture Mode Operation for Timer 0

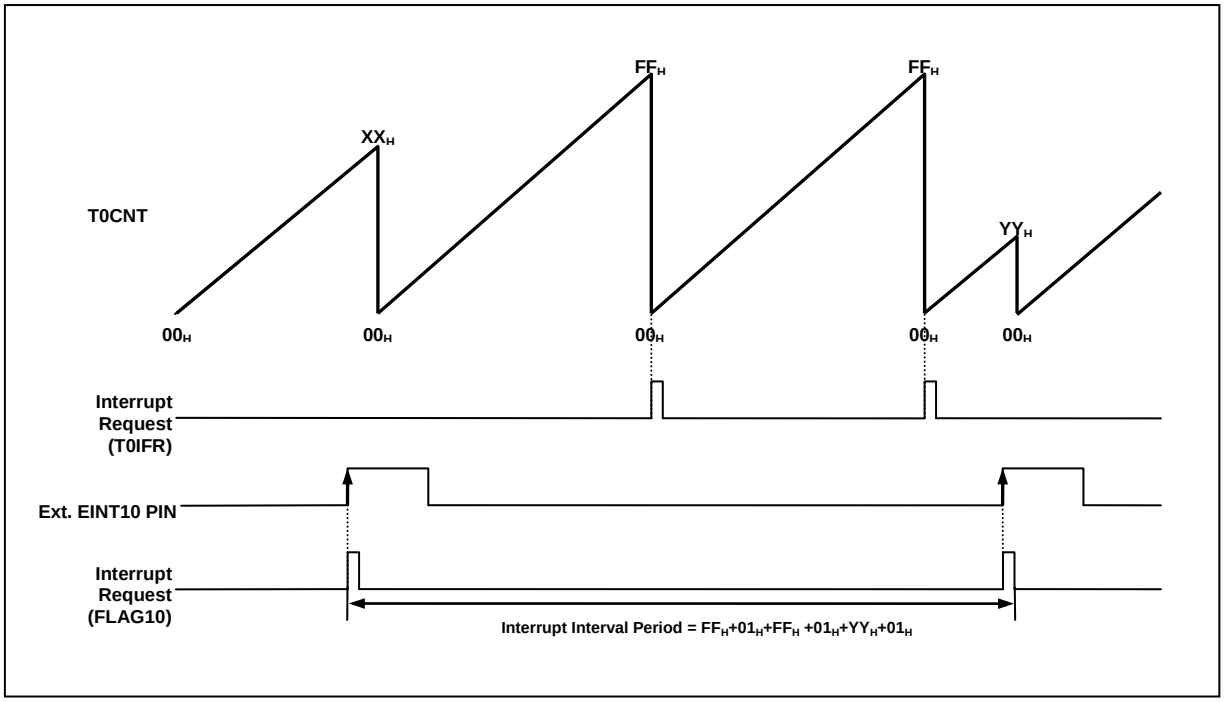


Figure 11.12 Express Timer Overflow in Capture Mode

11.5.5 Block Diagram

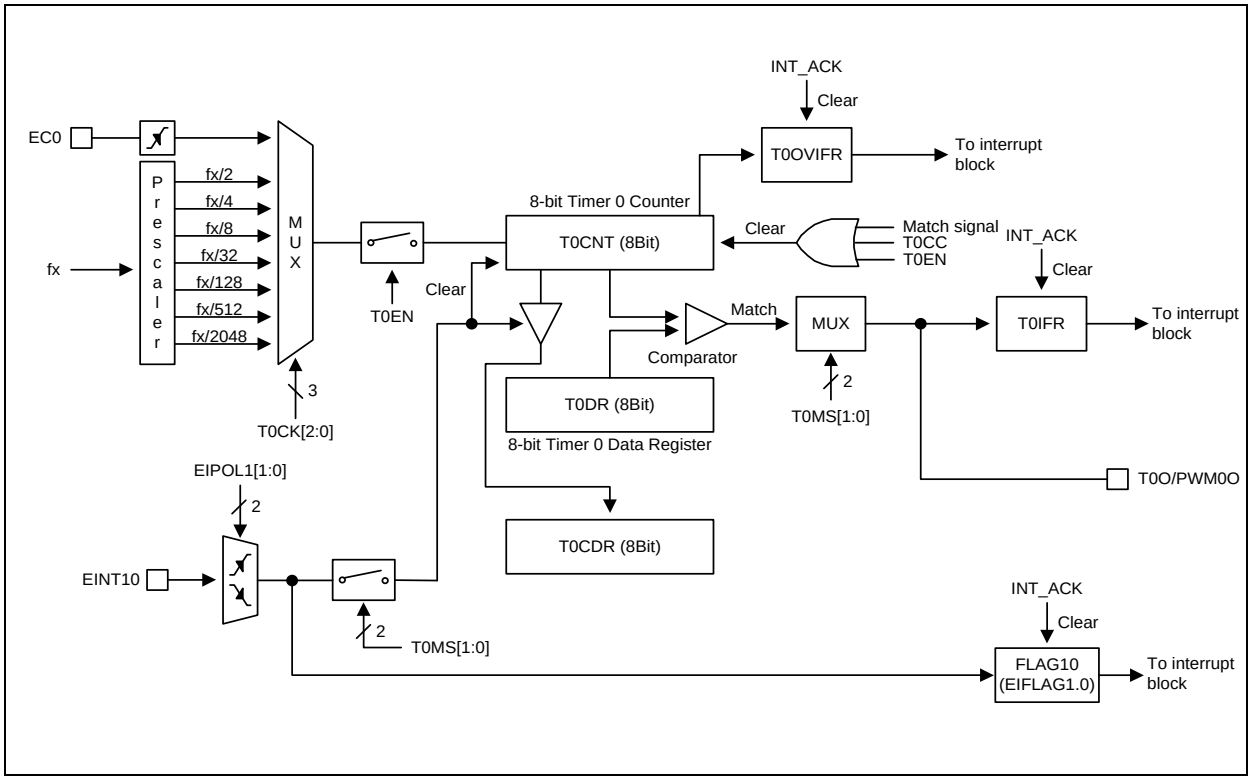


Figure 11.13 8-Bit Timer 0 Block Diagram

11.5.6 Register Map

Table 11-6 Timer 0 Register Map

Name	Address	Dir	Default	Description
T0CNT	B3H	R	00H	Timer 0 Counter Register
T0DR	B4H	R/W	FFH	Timer 0 Data Register
T0CDR	B4H	R	00H	Timer 0 Capture Data Register
T0CR	B2H	R/W	00H	Timer 0 Control Register

11.5.6.1 Timer/Counter 0 Register Description

The timer/counter 0 register consists of timer 0 counter register (T0CNT), timer 0 data register (T0DR), timer 0 capture data register (T0CDR), and timer 0 control register (T0CR). T0IFR and T0OVIFR bits are in the external interrupt flag 1 register (EIFLAG1).

11.5.6.2 Register Description for Timer/Counter 0

T0CNT (Timer 0 Counter Register) : B3H

7	6	5	4	3	2	1	0
T0CNT7	T0CNT6	T0CNT5	T0CNT4	T0CNT3	T0CNT2	T0CNT1	T0CNT0
R	R	R	R	R	R	R	R

Initial value : 00H

T0CNT[7:0] T0 Counter

T0DR (Timer 0 Data Register) : B4H

7	6	5	4	3	2	1	0
T0DR7	T0DR6	T0DR5	T0DR4	T0DR3	T0DR2	T0DR1	T0DR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FFH

T0DR[7:0] T0 Data

T0CDR (Timer 0 Capture Data Register: Read Case, Capture mode only) : B4H

7	6	5	4	3	2	1	0
T0CDR7	T0CDR6	T0CDR5	T0CDR4	T0CDR3	T0CDR2	T0CDR1	T0CDR0
R	R	R	R	R	R	R	R

Initial value : 00H

T0CDR[7:0] T0 Capture Data

T0CR (Timer 0 Control Register) : B2H

7	6	5	4	3	2	1	0
T0EN	–	T0MS1	T0MS0	T0CK2	T0CK1	T0CK0	T0CC
R/W	–	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

T0EN Control Timer 0

0	Timer 0 disable
1	Timer 0 enable (Counter clear and start)

T0MS[1:0] Control Timer 0 Operation Mode

T0MS1	T0MS0	Description
0	0	Timer/counter mode
0	1	PWM mode
1	x	Capture mode

T0CK[2:0] Select Timer 0 clock source. fx is a system clock frequency

T0CK2	T0CK1	T0CK0	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 (EC0)

T0CC Clear timer 0 Counter

0	No effect
1	Clear the Timer 0 counter (When write, automatically cleared “0” after being cleared counter)

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

2. Refer to the external interrupt flag 1 register (EIFLAG1) for the T0 interrupt flags.

11.6 Timer 1

11.6.1.1 Overview

The 16-bit timer 1 consists of multiplexer, timer 1 A data register high/low, timer 1 B data register high/low and timer 1 control register high/low (T1ADRH, T1ADRL, T1BDRH, T1BDRL, T1CRH, T1CRL).

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 1 can be clocked by an internal or an external clock source (EC1). The clock source is selected by clock selection logic which is controlled by the clock selection bits (T1CK[2:0]).

- TIMER 1 clock source: f_X/1, 2, 4, 8, 64, 512, 2048 and EC1

In the capture mode, by EINT11, the data is captured into input capture data register (T1BDRH/T1BDRL). Timer 1 outputs the comparision result between counter and data register through T1O port in timer/counter mode. Also Ttimer 1 outputs PWM wave form through PWM1O port in the PPG mode.

Table 11-7 Timer 1 Operating Modes

T1EN	P1FSRL[5:4]	T1MS[1:0]	T1CK[2:0]	Timer 1
1	11	00	XXX	16 Bit Timer/Counter Mode
1	00	01	XXX	16 Bit Capture Mode
1	11	10	XXX	16 Bit PPG Mode (one-shot mode)
1	11	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 1 can use the input clock with one of 1, 2, 4, 8, 64, 512 and 2048 prescaler division rates (T1CK[2:0]). When the value of T1CNTH, T1CNTL and the value of T1ADRH, T1ADRL are identical in Timer 1 respectively, a match signal is generated and the interrupt of Timer 1 occurs. The T1CNTH, T1CNTL value is automatically cleared by match signal. It can be also cleared by software (T1CC).

The external clock (EC1) counts up the timer at the rising edge. If the EC1 is selected as a clock source by T1CK[2:0], EC1 port should be set to the input port by P13IO bit.

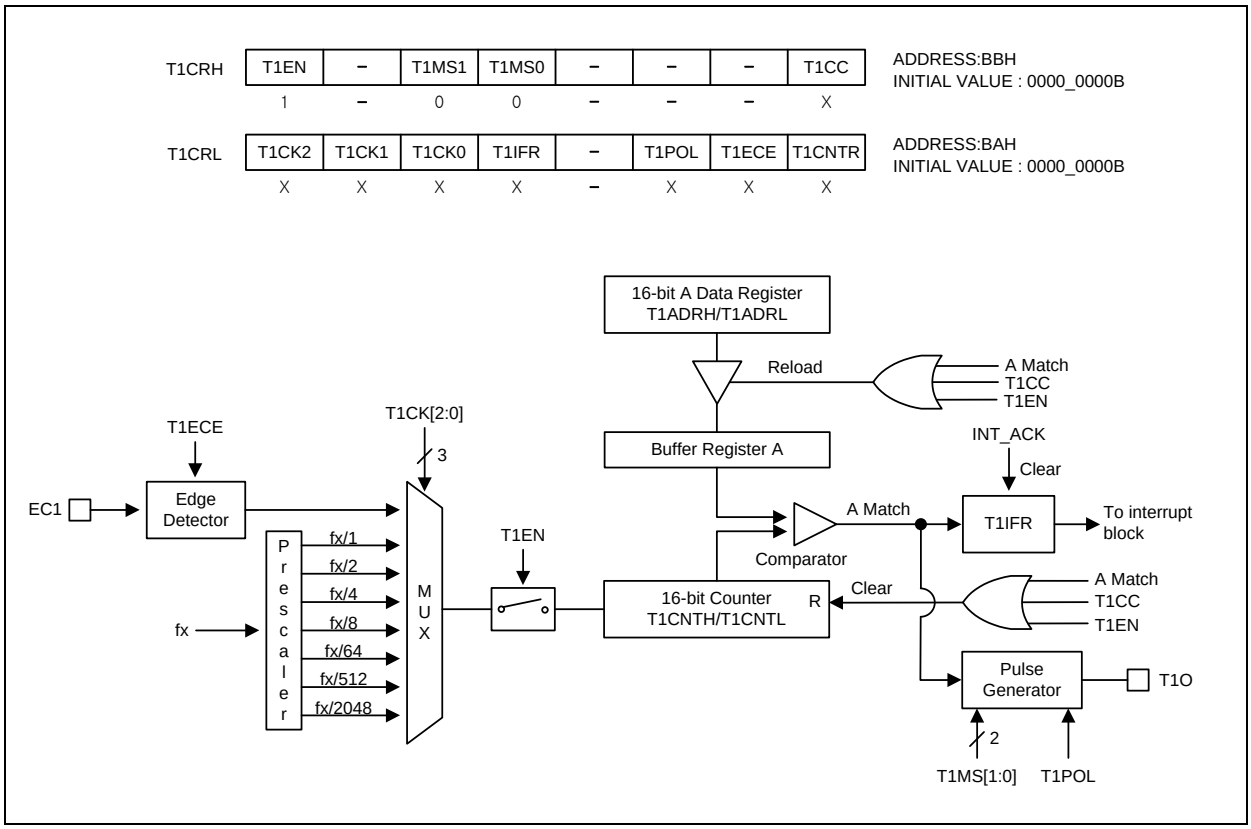


Figure 11.14 16-Bit Timer/Counter Mode for Timer 1

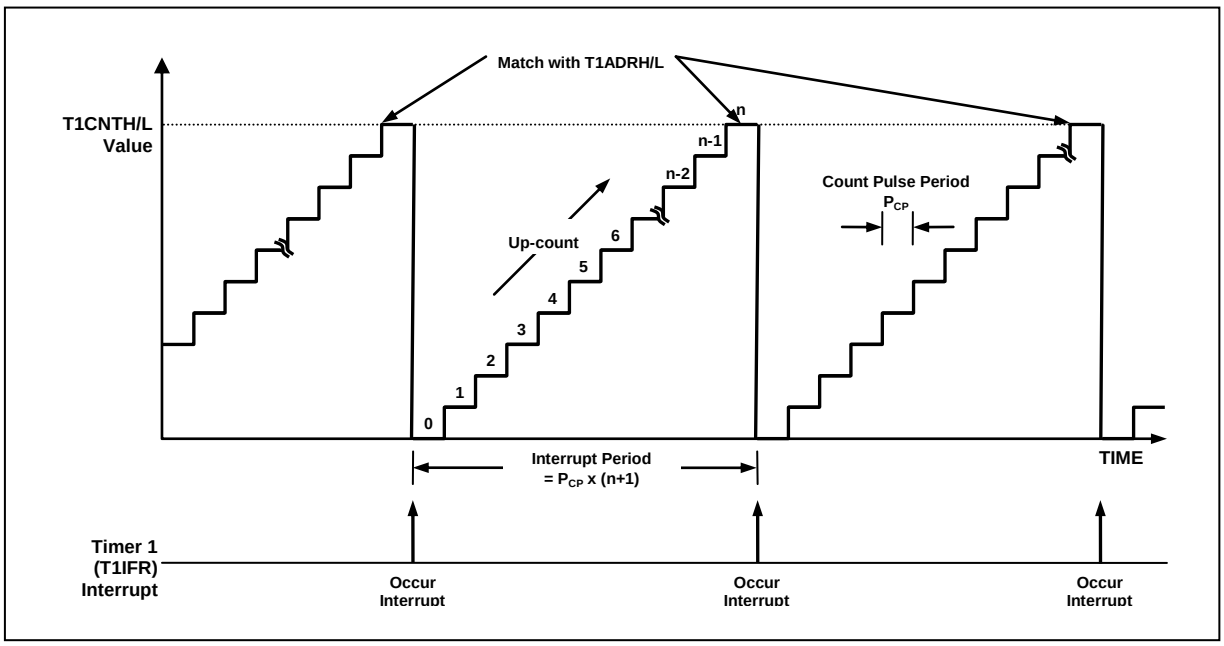


Figure 11.15 16-Bit Timer/Counter 1 Example

11.6.3 16-Bit Capture Mode

The 16-bit timer 1 capture mode is set by T1MS[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 T1CNTH/T1CNTL is equal to T1ADRH/T1ADRL. The T1CNTH, T1CNTL values are automatically cleared by match signal. It can be also cleared by software (T1CC).

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 T1BDRH/T1BDRL.

According to EIPOL1 registers setting, the external interrupt EINT11 function is chosen. Of cource, the EINT11 pin must be set as an input port.

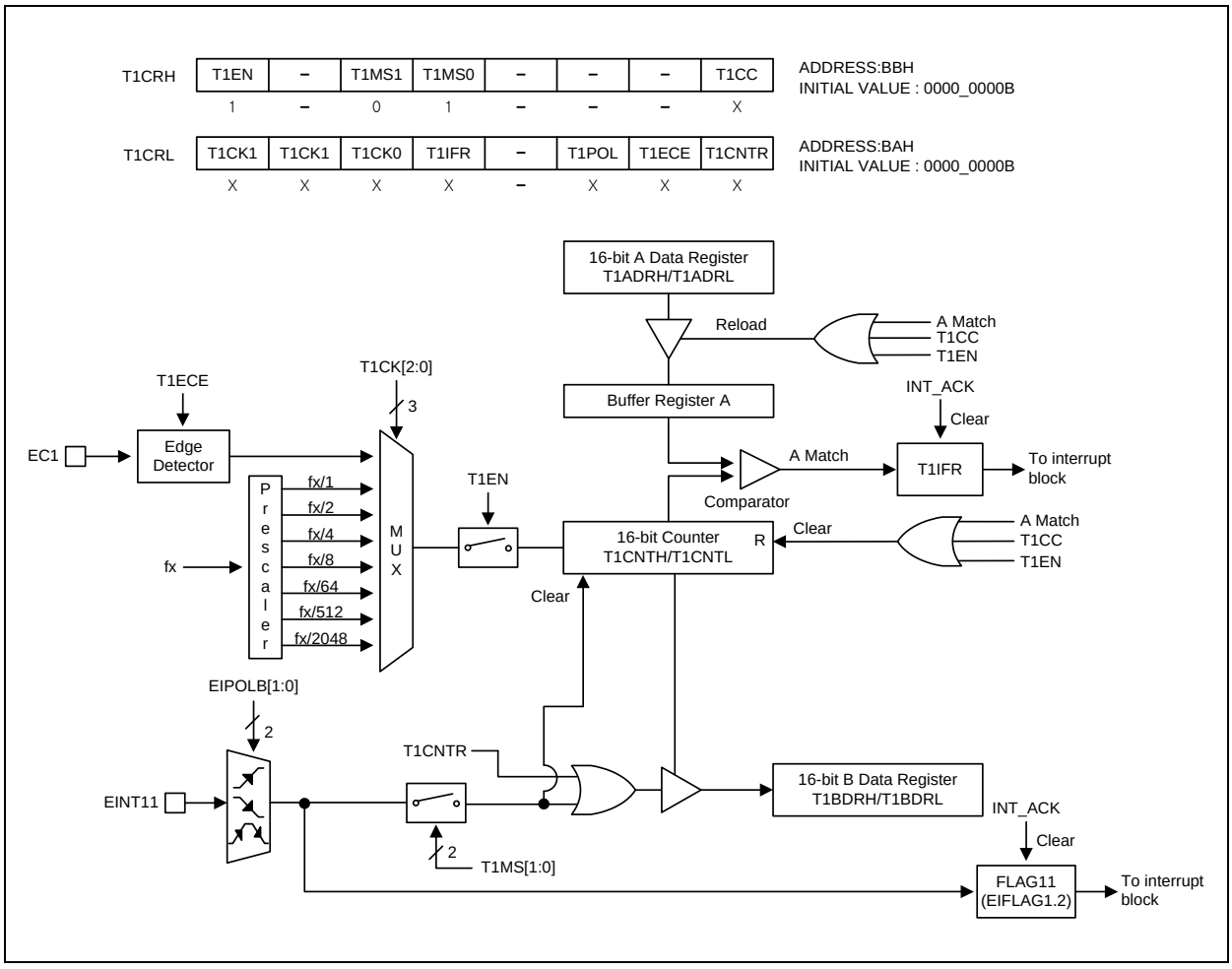


Figure 11.16 16-Bit Capture Mode for Timer 1

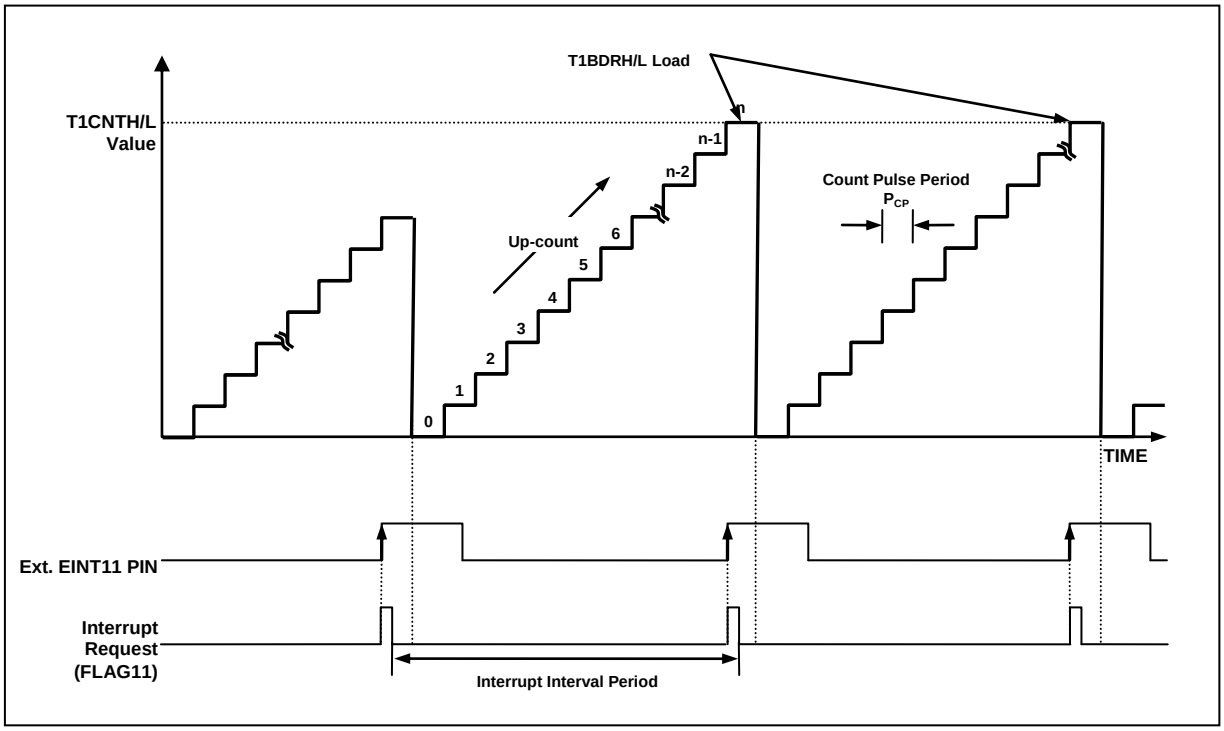


Figure 11.17 Input Capture Mode Operation for Timer 1

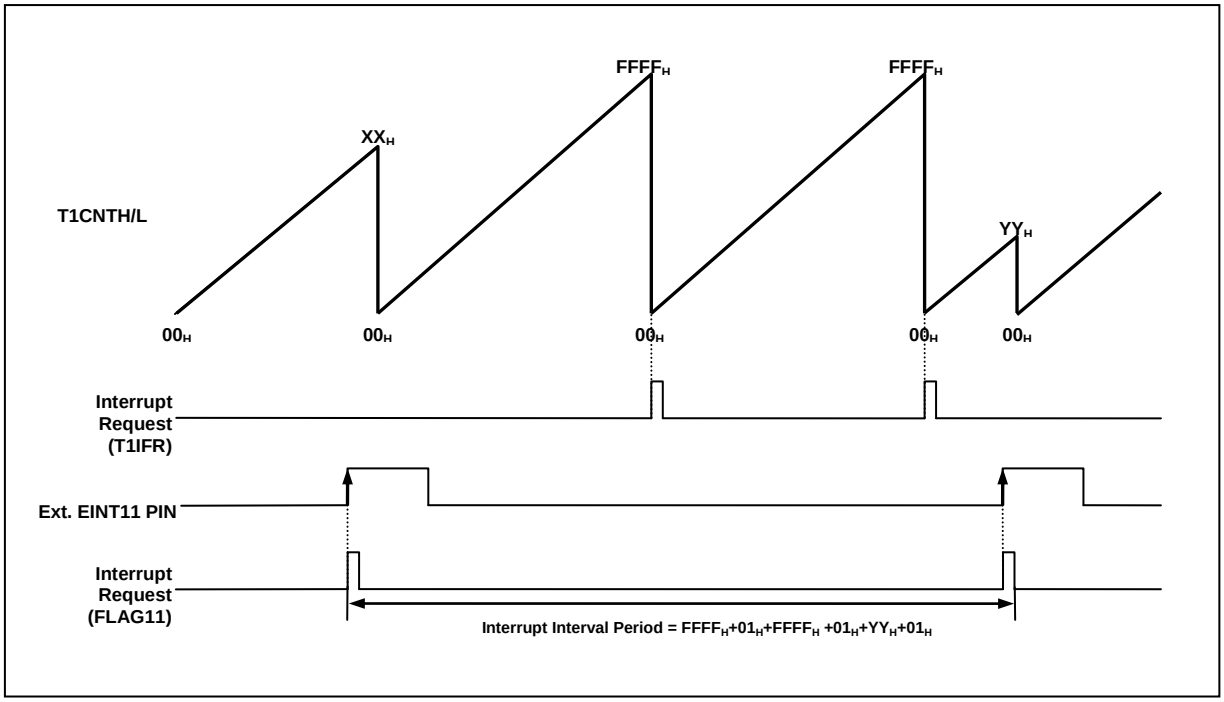
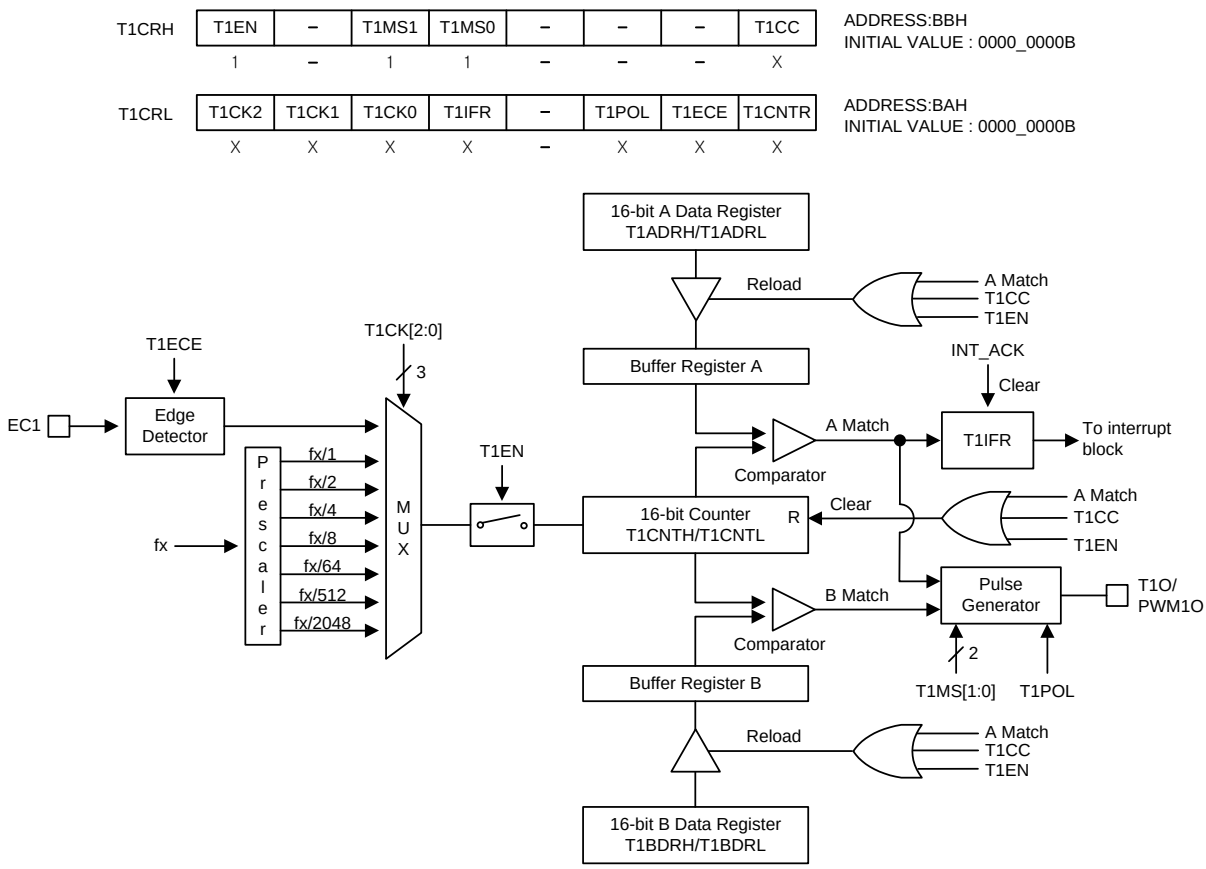


Figure 11.18 Express Timer Overflow in Capture Mode

11.6.4 16-Bit PPG Mode

The timer 1 has a PPG (Programmable Pulse Generation) function. In PPG mode, T1O/PWM1O pin outputs up to 16-bit resolution PWM output. This pin should be configured as a PWM output by setting P1FSRL[5:4] to '11' . The period of the PWM output is determined by the T1ADRH/T1ADRL. And the duty of the PWM output is determined by the T1BDRH/T1BDRL.



NOTE) The T1EN 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 1

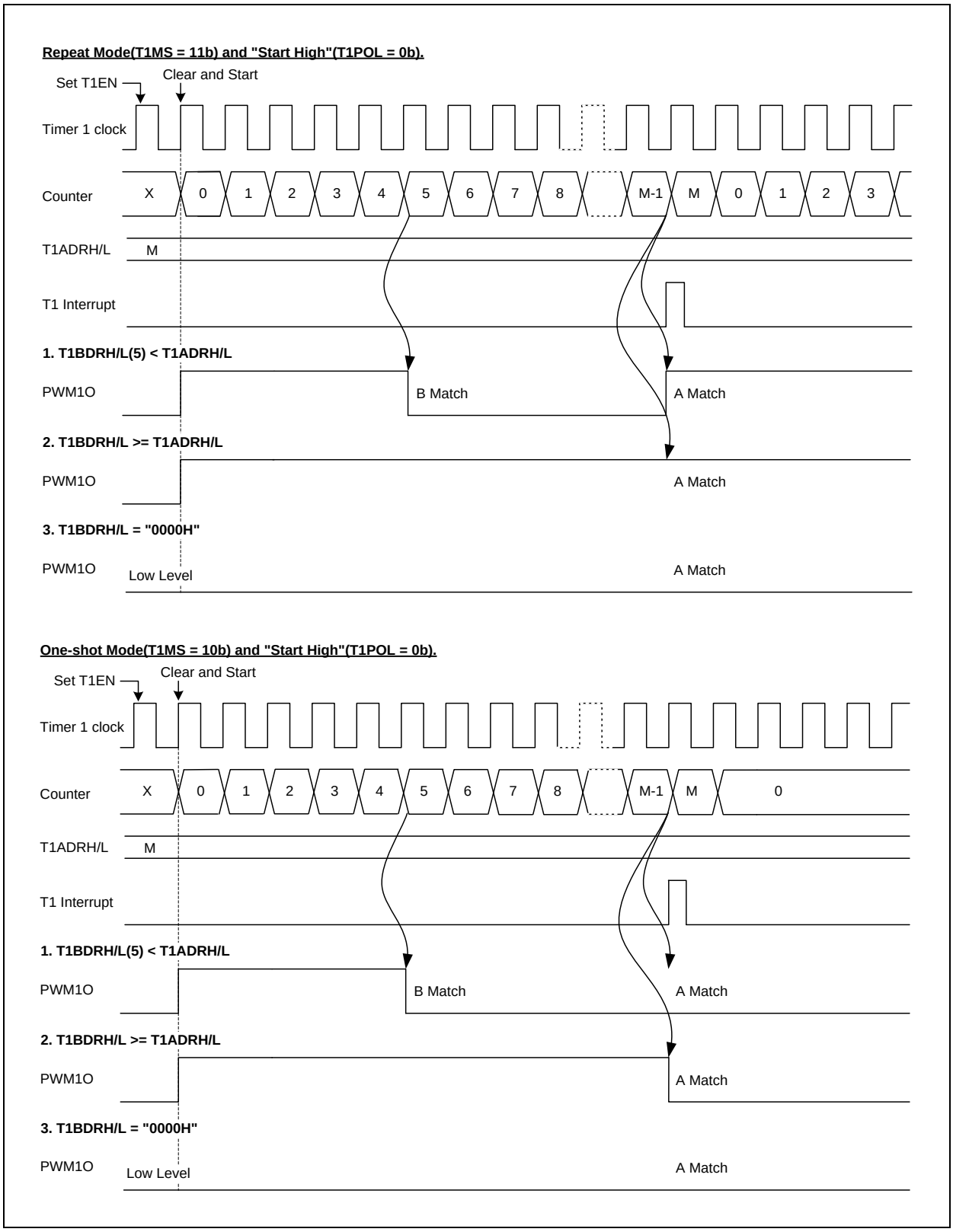


Figure 11.20 16-Bit PPG Mode Timming chart for Timer 1

11.6.5 Block Diagram

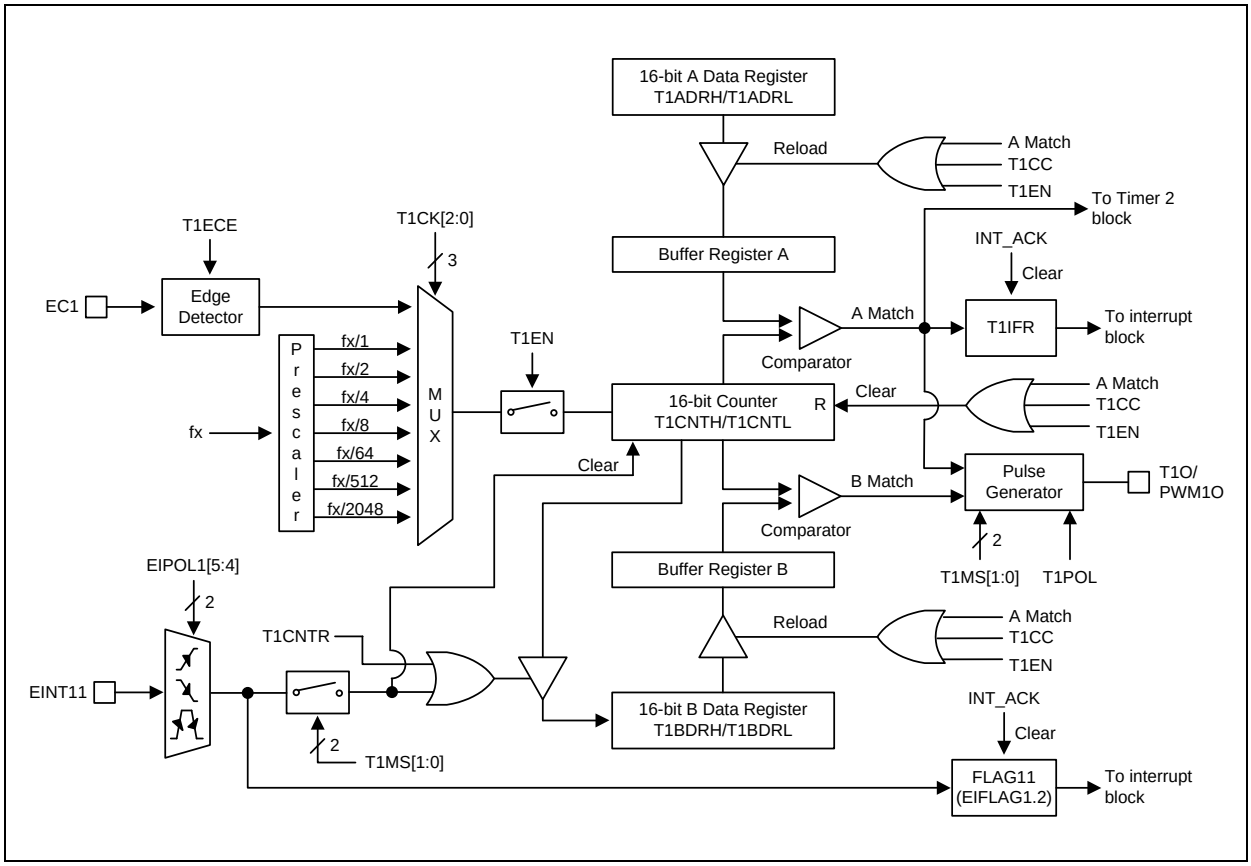


Figure 11.21 16-Bit Timer/Counter Mode for Timer 1 and Block Diagram

11.6.6 Register Map

Table 11-8 Timer 2 Register Map

Name	Address	Dir	Default	Description
T1ADRH	BDH	R/W	FFH	Timer 1 A Data High Register
T1ADRL	BCH	R/W	FFH	Timer 1 A Data Low Register
T1BDRH	BFH	R/W	FFH	Timer 1 B Data High Register
T1BDRL	BEH	R/W	FFH	Timer 1 B Data Low Register
T1CRH	BBH	R/W	00H	Timer 1 Control High Register
T1CRL	BAH	R/W	00H	Timer 1 Control Low Register

11.6.6.1 Timer/Counter 1 Register Description

The timer/counter 1 register consists of timer 1 A data high register (T1ADRH), timer 1 A data low register (T1ADRL), timer 1 B data high register (T1BDRH), timer 1 B data low register (T1BDRL), timer 1 control High register (T1CRH) and timer 1 control low register (T1CRL).

11.6.6.2 Register Description for Timer/Counter 1

T1ADRH (Timer 1 A data High Register) : BDH

7	6	5	4	3	2	1	0
T1ADRH7	T1ADRH6	T1ADRH5	T1ADRH4	T1ADRH3	T1ADRH2	T1ADRH1	T1ADRH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FFH

T1ADRH[7:0] T1 A Data High Byte

T1ADRL (Timer 1 A Data Low Register) : BCH

7	6	5	4	3	2	1	0
T1ADRL7	T1ADRL6	T1ADRL5	T1ADRL4	T1ADRL3	T1ADRL2	T1ADRL1	T1ADRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FFH

T1ADRL[7:0] T1 A Data Low Byte

NOTE) Do not write “0000H” in the T1ADRH/T1ADRL register when PPG mode

T1BDRH (Timer 1 B Data High Register) : BFH

7	6	5	4	3	2	1	0
T1BDRH7	T1BDRH6	T1BDRH5	T1BDRH4	T1BDRH3	T1BDRH2	T1BDRH1	T1BDRH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FFH

T1BDRH[7:0] T1 B Data High Byte

T1BDRL (Timer 1 B Data Low Register) : BEH

7	6	5	4	3	2	1	0
T1BDRL7	T1BDRL6	T1BDRL5	T1BDRL4	T1BDRL3	T1BDRL2	T1BDRL1	T1BDRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FFH

T1BDRL[7:0] T1 B Data Low Byte

T1CRH (Timer 1 Control High Register) : BBH

7	6	5	4	3	2	1	0
T1EN	–	T1MS1	T1MS0	–	–	–	T1CC
R/W	–	R/W	R/W	–	–	–	R/W

Initial value : 00H

T1EN Control Timer 1

0 Timer 1 disable

1 Timer 1 enable (Counter clear and start)

T1MS[1:0] Control Timer 1 Operation Mode

T1MS1	T1MS0	Description
0	0	Timer/counter mode (T1O: toggle at A match)
0	1	Capture mode (The A match interrupt can occur)
1	0	PPG one-shot mode (PWM1O)
1	1	PPG repeat mode (PWM1O)

T1CC Clear Timer 1 Counter

0 No effect

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

T1CRL (Timer 1 Control Low Register) : BAH

7	6	5	4	3	2	1	0
T1CK2	T1CK1	T1CK0	T1IFR	–	T1POL	T1ECE	T1CNTR
R/W	R/W	R/W	R/W	–	R/W	R/W	R/W

Initial value : 00H

T1CK[2:0] Select Timer 1 clock source. fx is main system clock frequency

T1CK2	T1CK1	T1CK0	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 (EC1)

T1IFR When T1 Interrupt occurs, this bit becomes ‘1’. For clearing bit, write ‘0’ to this bit or auto clear by INT_ACK signal.

0 T1 Interrupt no generation

1 T1 Interrupt generation

T1POL T1O/PWM1O Polarity Selection

0 Start High (T1O/PWM1O is low level at disable)

1 Start Low (T1O/PWM1O is high level at disable)

T1ECE Timer 1 External Clock Edge Selection

0 External clock falling edge

1 External clock rising edge

T1CNTR Timer 1 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 2

11.7.1.1 Overview

The 16-bit timer 2 consists of multiplexer, timer 2 A data high/low register, timer 2 B data high/low register and timer 2 control high/low register (T2ADRH, T2ADRL, T2BDRH, T2BDRL, T2CRH, T2CRL).

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 2 can be divided clock of the system clock selectd from prescaler output and T1 A Match (timer 1 A match signal). The clock source is selected by clock selection logic which is controlled by the clock selection bits (T2CK[2:0]).

- TIMER 2 clock source: f_X/1, 2, 4, 8, 32, 128, 512 and T1 A Match

In the capture mode, by EINT12, the data is captured into input capture data register (T2BDRH/T2BDRL). In timer/counter mode, whenever counter value is equal to T2ADRH/L, T2O port toggles. Also the timer 2 outputs PWM wave form to PWM2O port in the PPG mode.

Table 11-9 Timer 2 Operating Modes

T2EN	P1FSRL[3:2]	T2MS[1:0]	T2CK[2:0]	Timer 2
1	11	00	XXX	16 Bit Timer/Counter Mode
1	00	01	XXX	16 Bit Capture Mode
1	11	10	XXX	16 Bit PPG Mode (one-shot mode)
1	11	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 timer 1 A match clock input. Timer 2 can use the input clock with one of 1, 2, 4, 8, 32, 128, 512 and T1 A Match prescaler division rates (T2CK[2:0]). When the values of T2CNTH/T2CNTL and T2ADRH/T2ADRL are identical in timer 2, a match signal is generated and the interrupt of Timer 2 occurs. The T2CNTH/T2CNTL values are automatically cleared by match signal. It can be also cleared by software (T2CC).

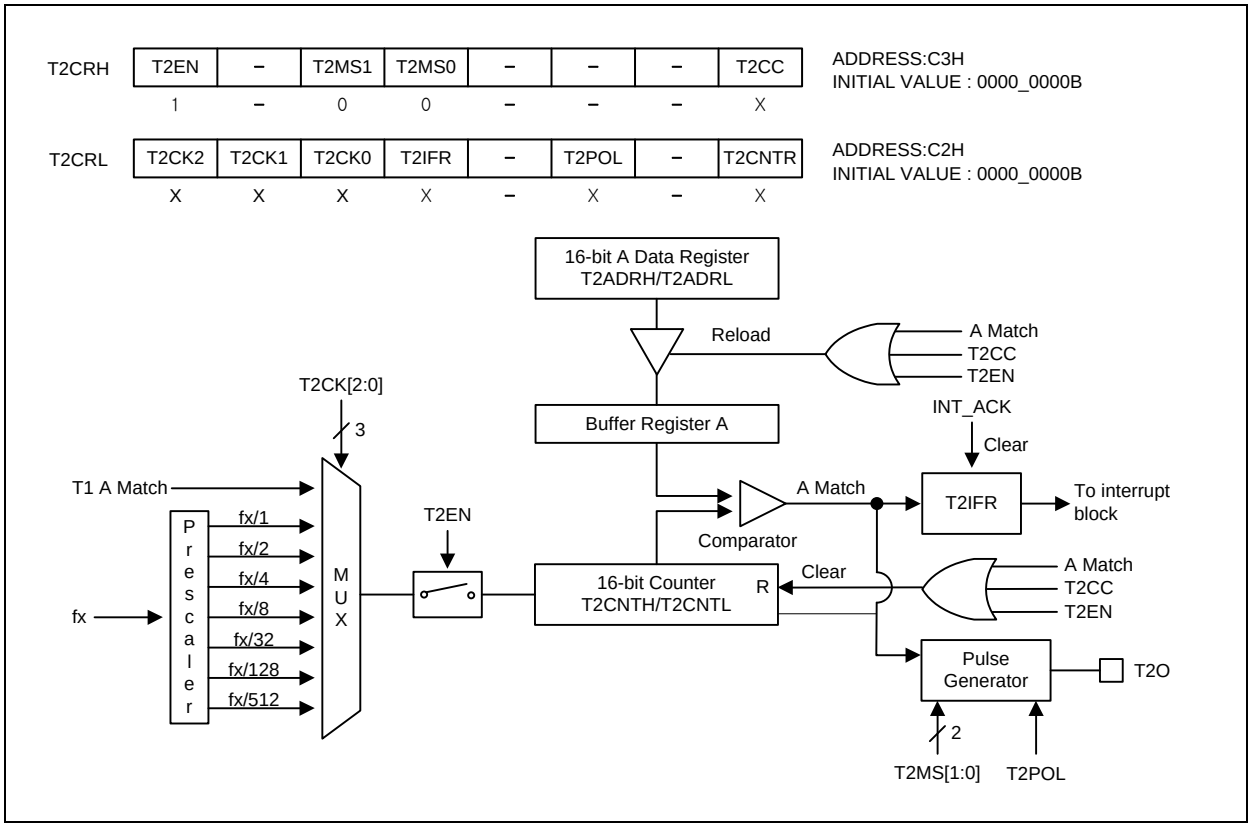


Figure 11.22 16-Bit Timer/Counter Mode for Timer 2

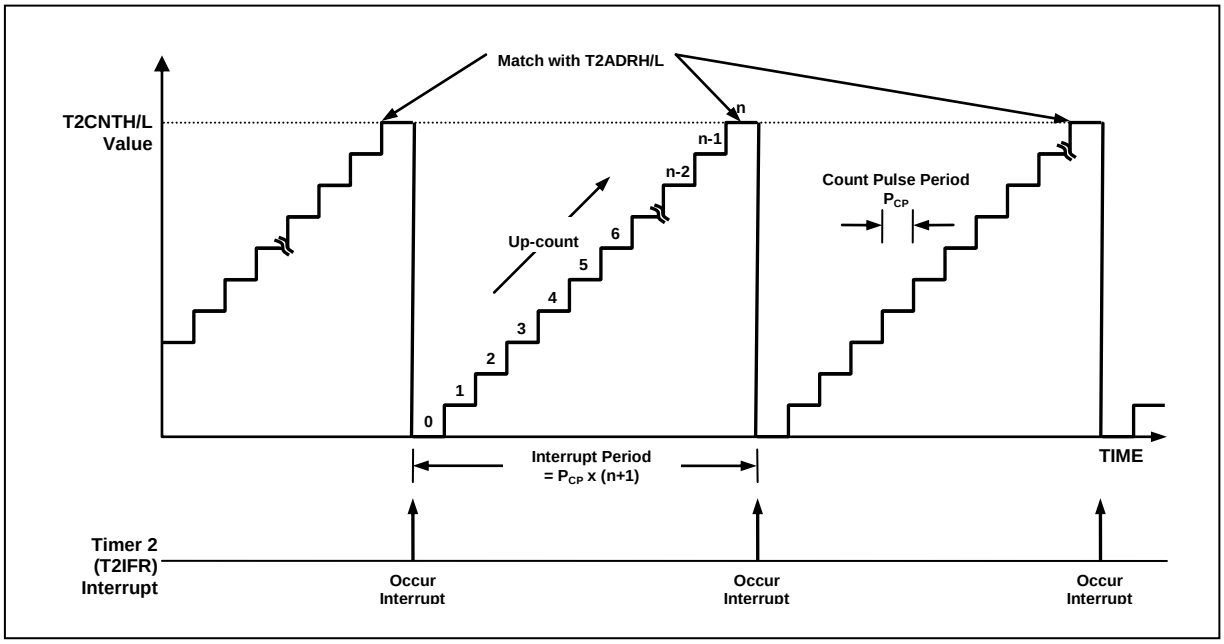


Figure 11.23 16-Bit Timer/Counter 2 Example

11.7.3 16-Bit Capture Mode

The timer 2 capture mode is set by T2MS[1:0] as '01'. The clock source can use the internal clock. Basically, it has the same function as the 16-bit timer/counter mode and the interrupt occurs when T2CNTH/T2CNTL is equal to T2ADRH/T2ADRL. T2CNTH/T2CNTL values are automatically cleared by match signal and it can be also cleared by software (T2CC).

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 T2BDRH/T2BDRL. In the timer 2 capture mode, timer 2 output(T2O) waveform is not available.

According to EIPOL1 registers setting, the external interrupt EINT12 function is chosen. Of cource, the EINT12 pin must be set to an input port.

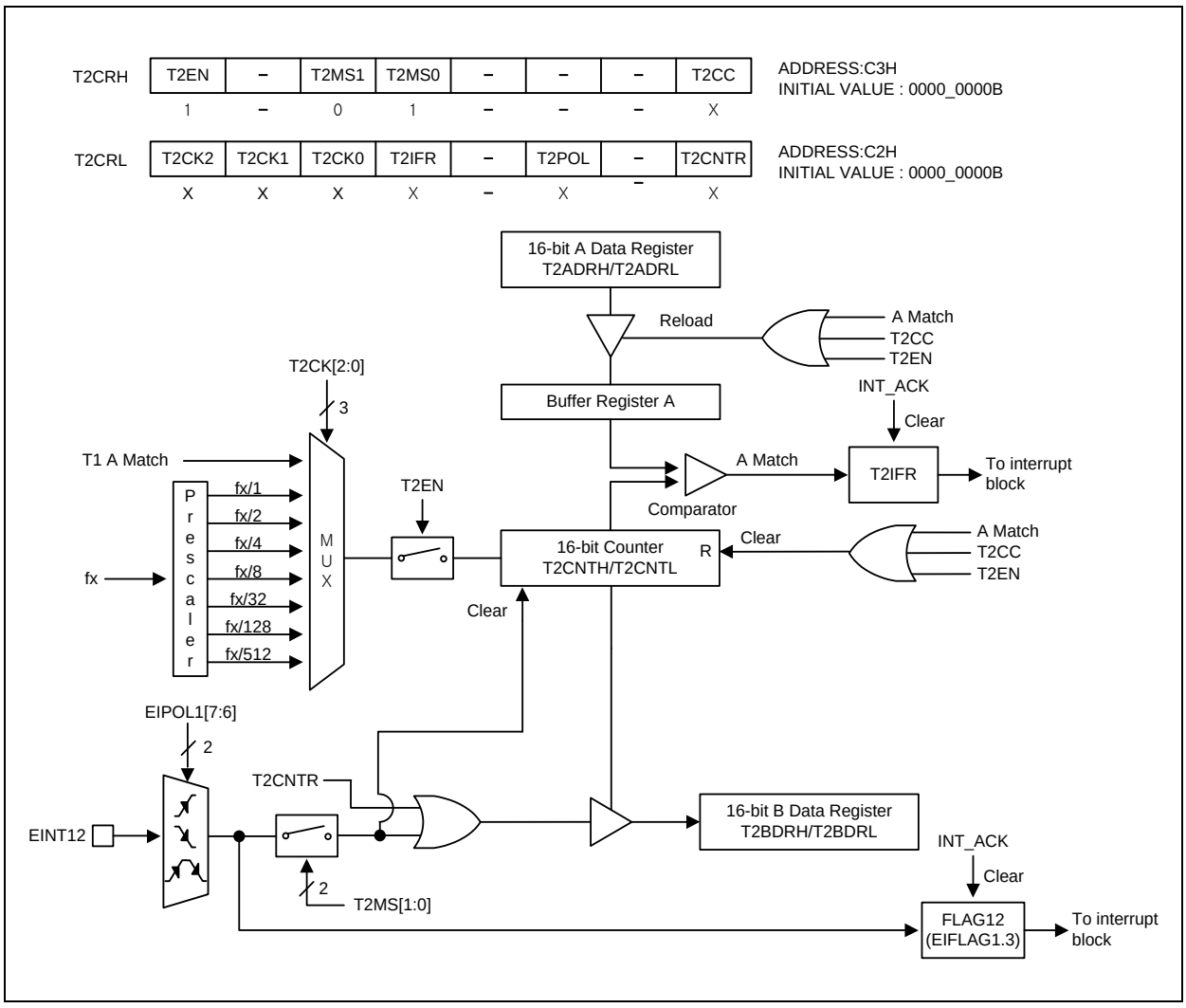


Figure 11.24 16-Bit Capture Mode for Timer 2

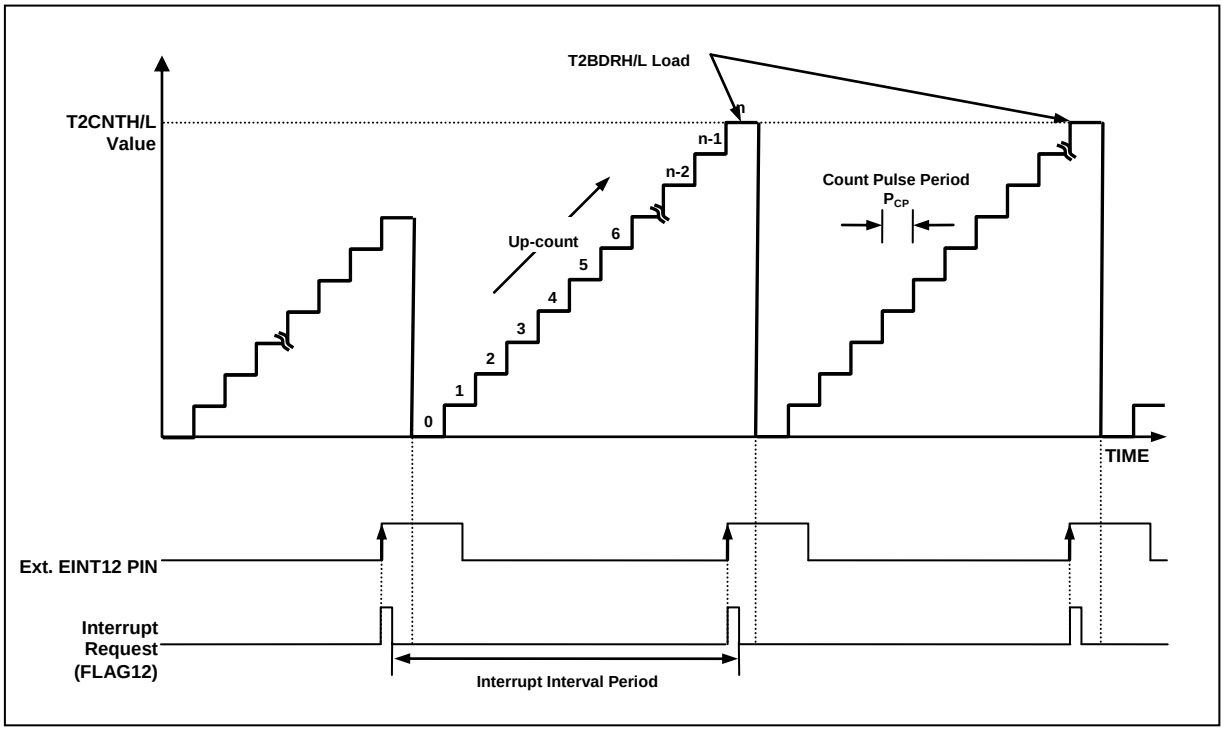


Figure 11.25 Input Capture Mode Operation for Timer 2

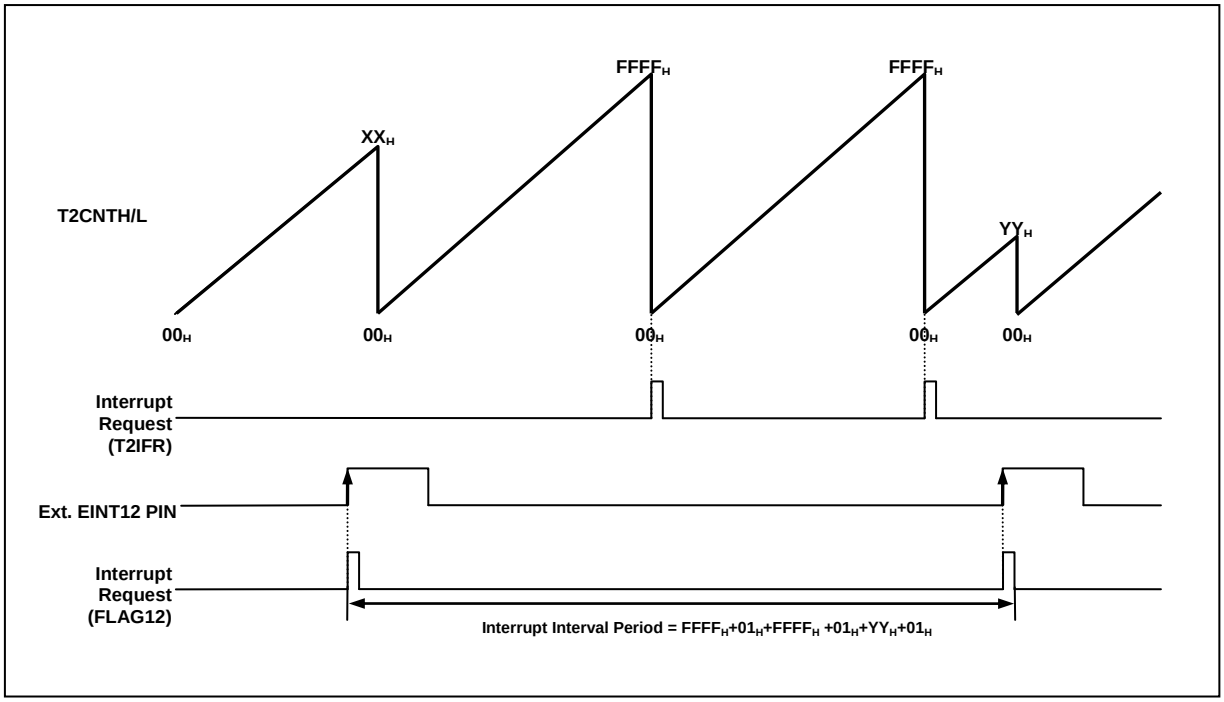
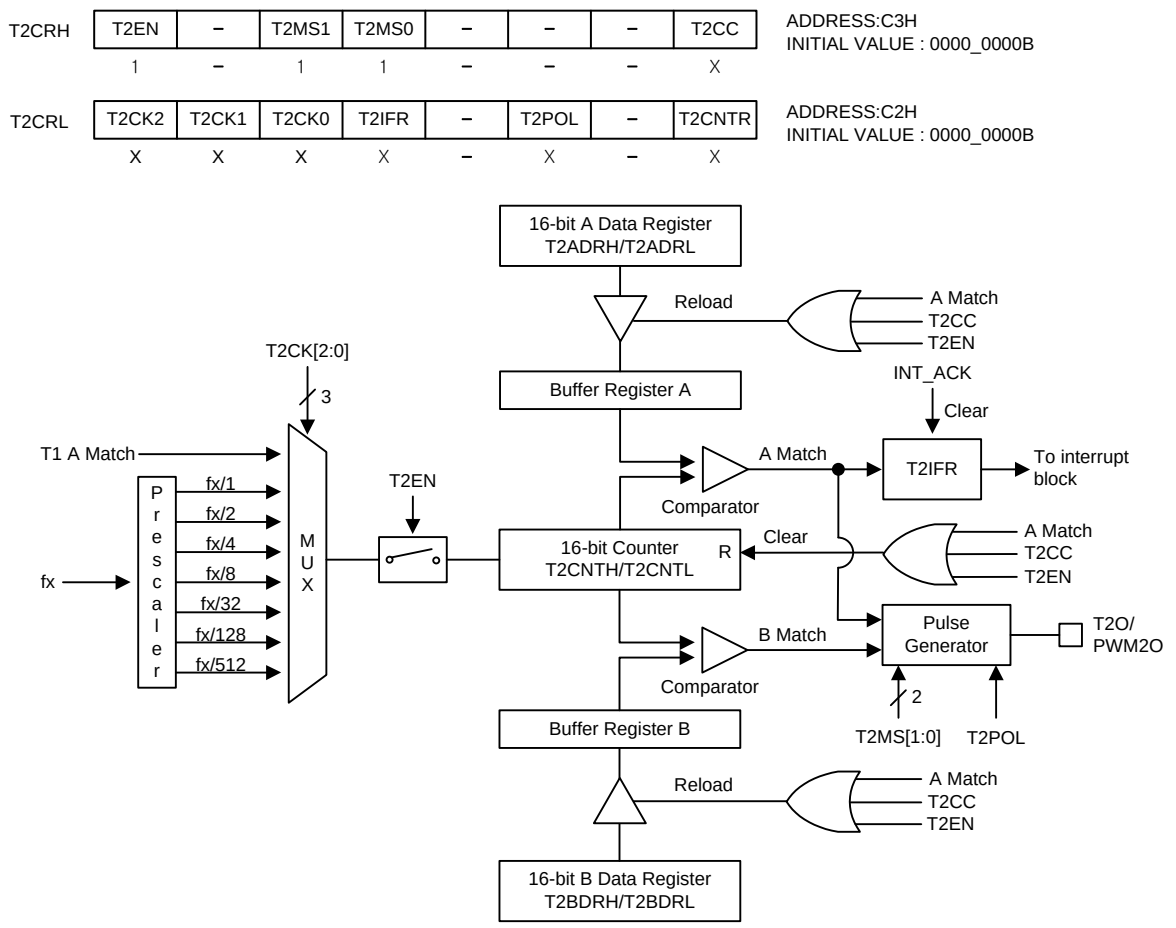


Figure 11.26 Express Timer Overflow in Capture Mode

11.7.4 16-Bit PPG Mode

The timer 2 has a PPG (Programmable Pulse Generation) function. In PPG mode, the T2O/PWM2O pin outputs up to 16-bit resolution PWM output. This pin should be configured as a PWM output by set P1FSRL[3:2] to '11' . The period of the PWM output is determined by the T2ADRH/T2ADRL. And the duty of the PWM output is determined by the T2BDRH/T2BDRL.



NOTE) The T2EN 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 2

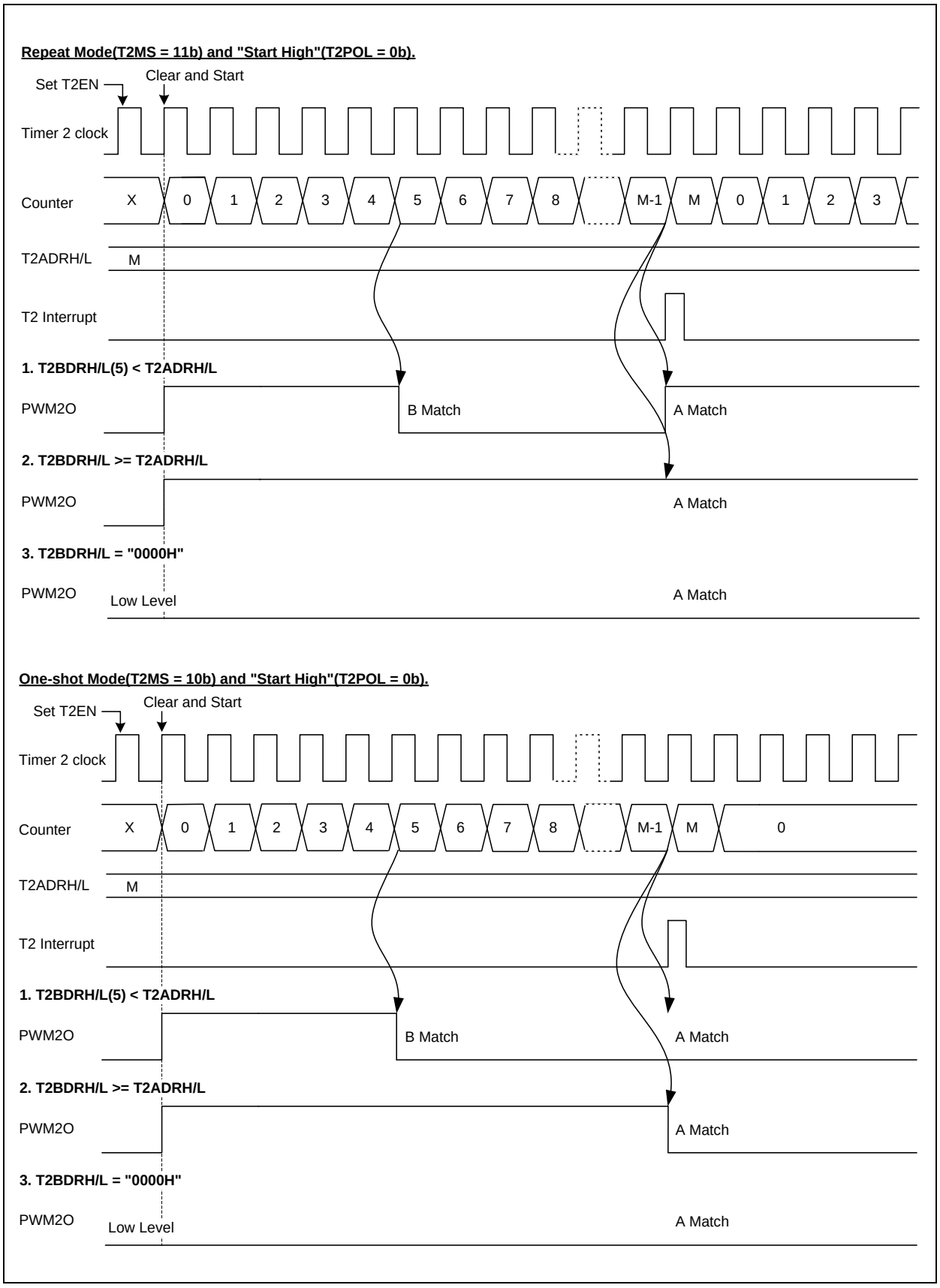


Figure 11.28 16-Bit PPG Mode Timming chart for Timer 2

11.7.5 Block Diagram

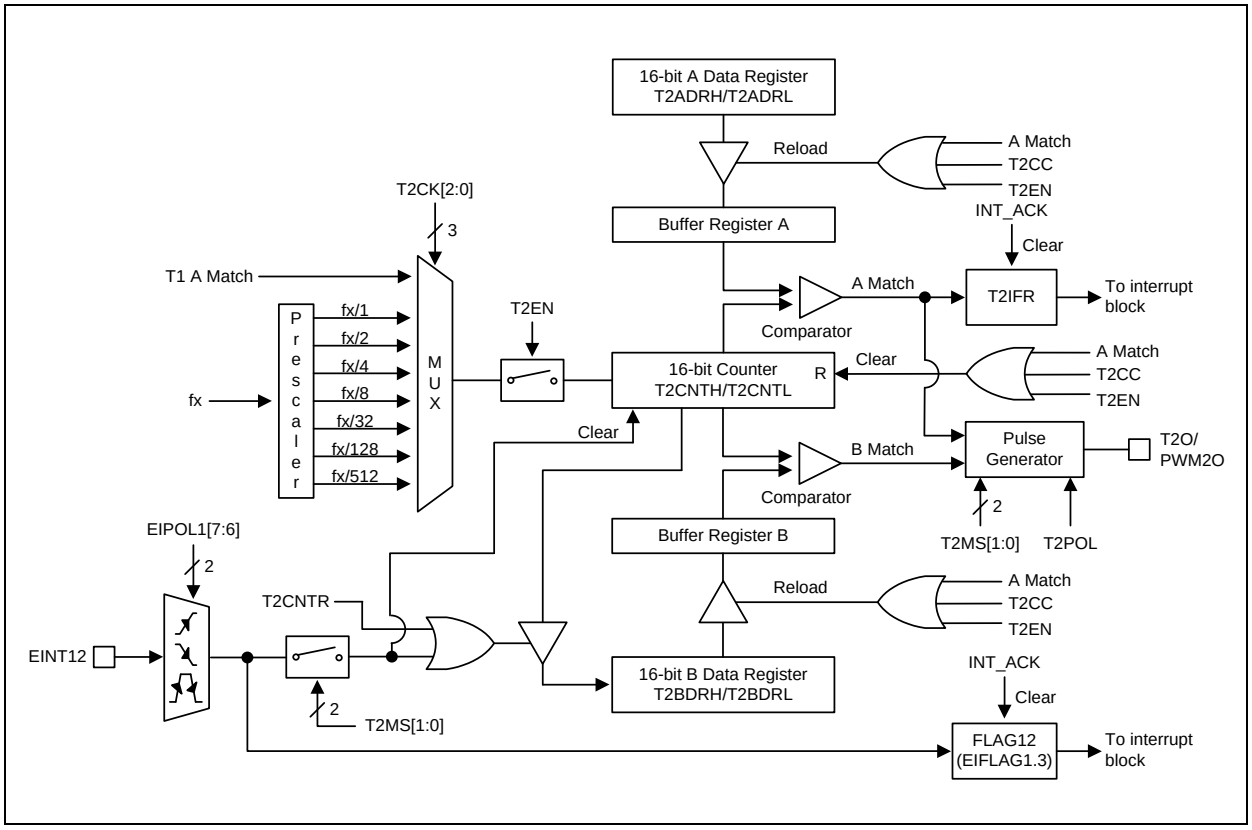


Figure 11.29 16-Bit Timer/Counter Mode for Timer 2 and Block Diagram

11.7.6 Register Map

Table 11-10 Timer 3 Register Map

Name	Address	Dir	Default	Description
T2ADRH	C5H	R/W	FFH	Timer 2 A Data High Register
T2ADRL	C4H	R/W	FFH	Timer 2 A Data Low Register
T2BDRH	C7H	R/W	FFH	Timer 2 B Data High Register
T2BDRL	C6H	R/W	FFH	Timer 2 B Data Low Register
T2CRH	C3H	R/W	00H	Timer 2 Control High Register
T2CRL	C2H	R/W	00H	Timer 2 Control Low Register

11.7.6.1 Timer/Counter 2 Register Description

The timer/counter 2 register consists of timer 2 A data high register (T2ADRH), timer 2 A data low register (T2ADRL), timer 2 B data high register (T2BDRH), timer 2 B data low register (T2BDRL), timer 2 control High register (T2CRH) and timer 2 control low register (T2CRL).

11.7.6.2 Register Description for Timer/Counter 2

T2ADRH (Timer 2 A data High Register) : C5H

7	6	5	4	3	2	1	0
T2ADRH7	T2ADRH6	T2ADRH5	T2ADRH4	T2ADRH3	T2ADRH2	T2ADRH1	T2ADRH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FFH

T2ADRH[7:0] T2 A Data High Byte

T2ADRL (Timer 2 A Data Low Register) : C4H

7	6	5	4	3	2	1	0
T2ADRL7	T2ADRL6	T2ADRL5	T2ADRL4	T2ADRL3	T2ADRL2	T2ADRL1	T2ADRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FFH

T2ADRL[7:0] T2 A Data Low Byte

NOTE) Do not write “0000H” in the T2ADRH/T2ADRL register when PPG mode.

T2BDRH (Timer 2 B Data High Register) : C7H

7	6	5	4	3	2	1	0
T2BDRH7	T2BDRH6	T2BDRH5	T2BDRH4	T2BDRH3	T2BDRH2	T2BDRH1	T2BDRH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FFH

T2BDRH[7:0] T2 B Data High Byte

T2BDRL (Timer 2 B Data Low Register) : C6H

7	6	5	4	3	2	1	0
T2BDRL7	T2BDRL6	T2BDRL5	T2BDRL4	T2BDRL3	T2BDRL2	T2BDRL1	T2BDRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FFH

T2BDRL[7:0] T2 B Data Low

T2CRH (Timer 2 Control High Register) : C3H

7	6	5	4	3	2	1	0
T2EN	–	T2MS1	T2MS0	–	–	–	T2CC
R/W	–	R/W	R/W	–	–	–	R/W

Initial value : 00H

T2EN Control Timer 2

0 Timer 2 disable

1 Timer 2 enable (Counter clear and start)

T2MS[1:0] Control Timer 2 Operation Mode

T2MS1	T2MS0	Description
0	0	Timer/counter mode (T2O: toggle at A match)
0	1	Capture mode (The A match interrupt can occur)
1	0	PPG one-shot mode (PWM2O)
1	1	PPG repeat mode (PWM2O)

T2CC Clear Timer 2 Counter

0 No effect

1 Clear the Timer 2 counter (When write, automatically cleared “0” after being cleared counter)

T2CRL (Timer 2 Control Low Register) : CAH

7	6	5	4	3	2	1	0
T2CK2	T2CK1	T2CK0	T2IFR	–	T2POL	–	T2CNTR
R/W	R/W	R/W	R/W	–	R/W	–	R/W

Initial value : 00H

T2CK[2:0] Select Timer 2 clock source. fx is main system clock frequency

T2CK2	T2CK1	T2CK0	Description
0	0	0	fx/512
0	0	1	fx/128
0	1	0	fx/32
0	1	1	fx/8
1	0	0	fx/4
1	0	1	fx/2
1	1	0	fx/1
1	1	1	T1 A Match

T2IFR When T2 Match Interrupt occurs, this bit becomes ‘1’. For clearing bit, write ‘0’ to this bit or auto clear by INT_ACK signal.

0 T2 interrupt no generation

1 T2 interrupt generation

T2POL T2O/PWM2O Polarity Selection

0 Start High (T2O/PWM2O is low level at disable)

1 Start Low (T2O/PWM2O is high level at disable)

T2CNTR Timer 2 Counter Read Control

0 No effect

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

11.8 Timer 3, 4

11.8.1 Overview

Timer 3 and timer 4 can be used either two 8-bit timer/counter or one 16-bit timer/counter with combine them. Each 8-bit timer/event counter module has multiplexer, comparator, 8-bit timer data register, 8-bit counter register, control register and capture data register (T3CNT, T3DR, T3CAPR, T3CR, T4CNT, T4DR, T4CAPR, T4CR). For PWM, it has PWM register (T4PPRL. T4PPRH, T4ADRL, T4ADRH, T4BDRL, T4BDRH, T4CDRL, T4CDRH, T4DLYA, T4DLYB, T4DLYC).

It has five operating modes:

- 8-bit timer/counter mode
- 8-bit capture mode
- 16-bit timer/counter mode
- 16-bit capture mode
- 10-bit PWM mode

The timer/counter 3 and 4 can be clocked by an internal or an external clock source (EC3). The clock source is selected by clock selection logic which is controlled by the clock selection bits (T3CK[2:0], T4CK[3:0]). Also the timer/counter 4 can use more clock sources than timer/counter 3.

- TIMER 3 clock source: f_X/2, 4, 8, 32, 128, 512, 2048 and EC3
- TIMER 4 clock source: f_X/1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384 and T3 clock

In the capture mode, by EINT0/EINT1, the data is captured into input capture data register (T3CAPR, T4CAPR).
In 8-bit timer/counter 3/4 mode, whenever counter value is equal to T3DR/T4DR, T3O/T4O port toggles. Also In 16-bit timer/counter 3 mode,
The timer 3 outputs the comparison result between counter and data register through T3O port. The PWM wave form to PWMAA, PWMAB, PWMBA, PWMBB, PWMCA, PWMCB Port (6-channel) in the PWM mode.

Table 11-11 Timer 3, 4 Operating Modes

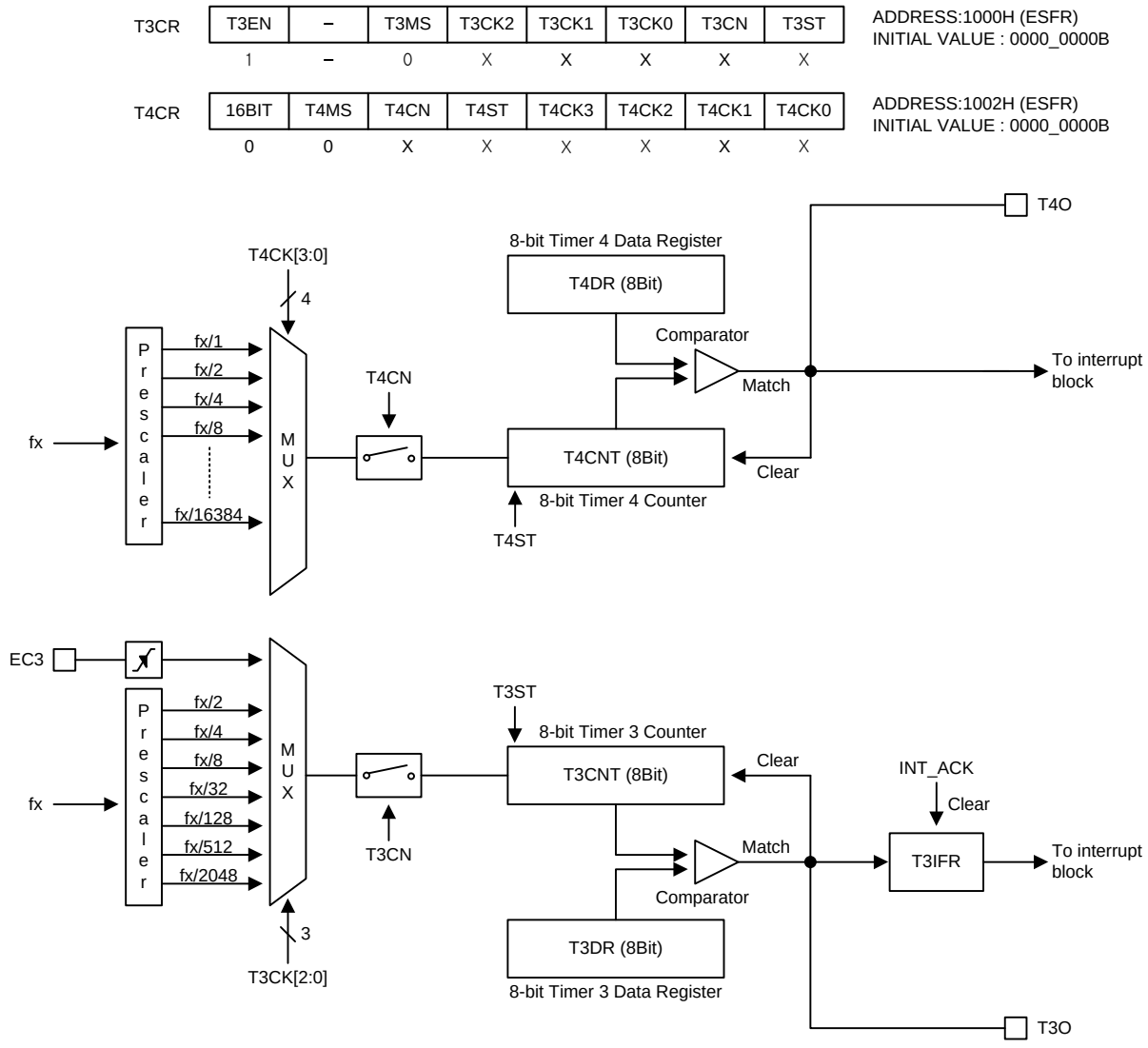
16BIT	T3MS	T4MS	PWM4E	T3CK[2:0]	T4CK[3:0]	Timer 3	Timer 4
0	0	0	0	XXX	XXXX	8 Bit Timer/Counter Mode	8 Bit Timer/Counter Mode
0	1	1	0	XXX	XXXX	8 Bit Capture Mode	8 Bit Capture Mode
1	0	0	0	XXX	XXXX	16 Bit Tmer/Counter Mode	
1	1	1	0	XXX	XXXX	16 Bit Capture Mode	
0	X	X	1	XXX	XXXX	10 Bit PWM Mode	

11.8.2 8-Bit Timer/Counter 3, 4 Mode

The 8-bit timer/counter mode is selected by control register as shown in Figure 11.30.

The two 8-bit timers have each counter and data register. The counter register is increased by internal or external clock input. Timer 3 can use the input clock with one of 2, 4, 8, 32, 128, 512, 2048 and EC3 prescaler division rates (T3CK[2:0]). Timer 4 can use the input clock with one of 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384 and timer 3 clock prescaler division rates (T4CK[3:0]). When the value of T3CNT, T4CNT and T3DR, T4DR are respectively identical in Timer 3, 4, the interrupt Timer 3, 4 occurs.

The external clock (EC3) counts up the timer at the rising edge. If the EC3 is selected as a clock source by T3CK[2:0], EC3 port should be set to the input port by P00IO bit. Timer 4 can't use the external EC3 clock.



NOTE: Do not set to "1111b" in the T4CK[3:0], when two 8-bit timer 3/4 modes.

Figure 11.30 8-Bit Timer/Counter Mode for Timer 3, 4

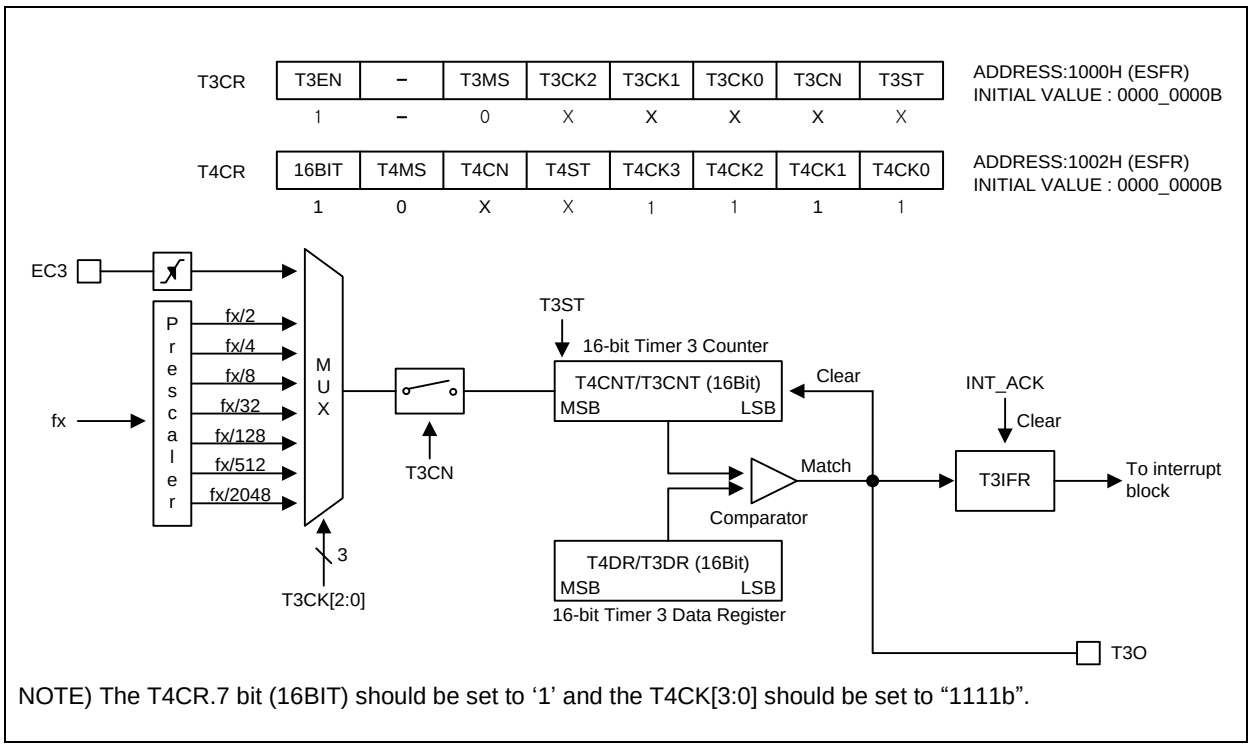
11.8.3 16-Bit Timer/Counter 3 Mode

The 16-bit timer/counter mode is selected by control register as shown in Figure 11.31.

The 16-bit timer have counter and data register. The counter register is increased by internal or external clock input. Timer 3 can use the input clock with one of 2, 4, 8, 32, 128, 512 and 2048 prescaler division rates (T3CK[2:0]).

A 16-bit timer/counter register T3CNT, T4CNT are incremented from 0000H to FFFFH until it matches T3DR, T4DR and then cleared to 0000H. The match signal output generates the Timer 3 Interrupt (No timer 4 interrupt). The clock source is selected from T3CK[2:0] and 16BIT bit must be set to '1'. Timer 3 is LSB 8-bit, the timer 4 is MSB 8-bit.

The external clock (EC3) counts up the timer at the rising edge. f the EC3 is selected as a clock source by T3CK[2:0], EC3 port should be set to the input port by P00IO bit.



NOTE) The T4CR.7 bit (16BIT) should be set to '1' and the T4CK[3:0] should be set to "1111b".

Figure 11.31 16-Bit Timer/Counter Mode for Timer 3

11.8.4 8-Bit Timer 3, 4 Capture Mode

The 8-bit Capture 3 and 4 mode is selected by control register as shown in Figure 11.32.

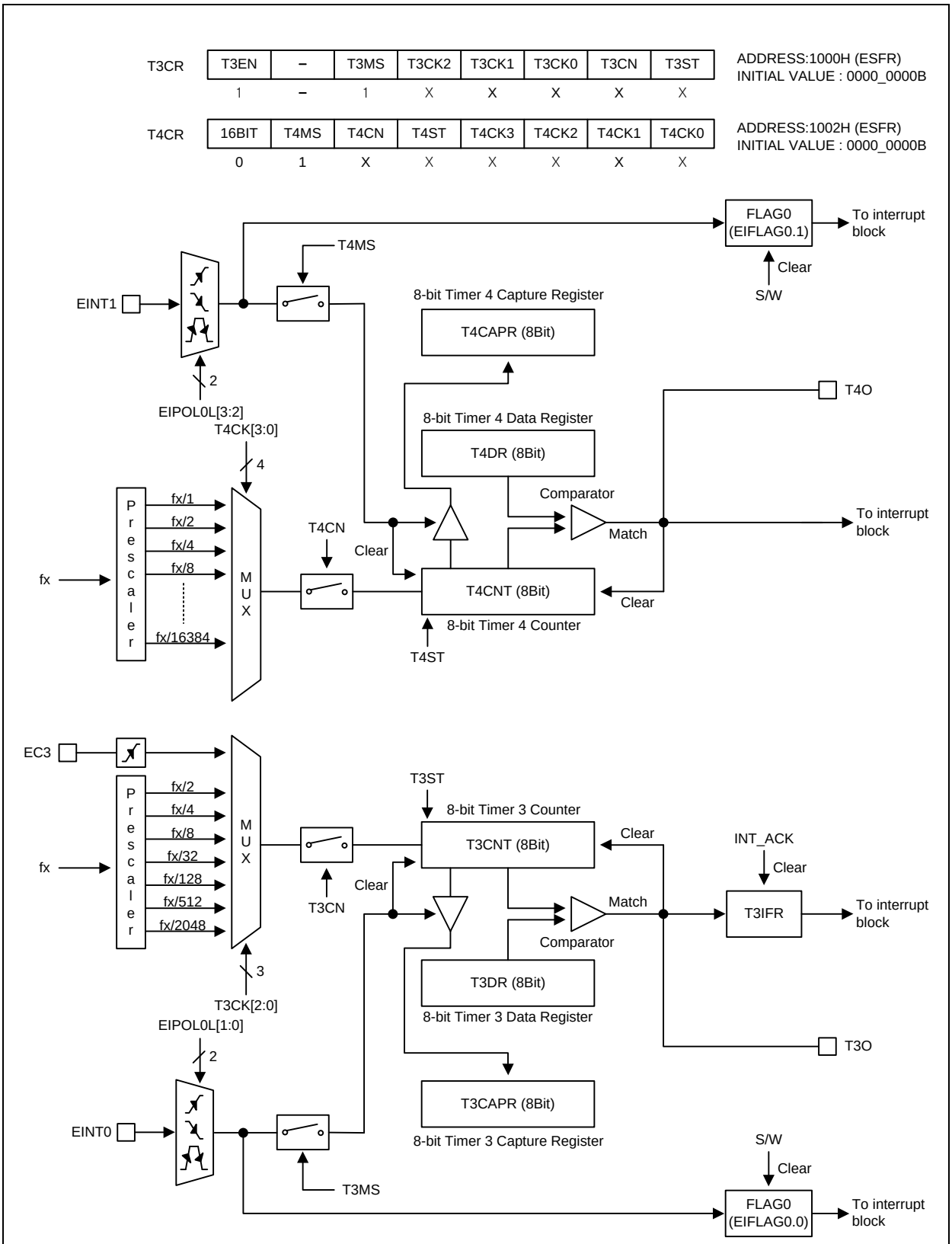
The timer 3, 4 capture mode is set by T3MS, T4MS as '1'. The clock source can use the internal/external clock. Basically, it has the same function as the 8-bit timer/counter mode and the interrupt occurs when T3CNT, T4CNT is equal to T3DR, T4DR. The T3CNT, T4CNT value is automatically cleared by match signal.

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 T3CAPR, T4CAPR. In the timer 3, 4 capture mode, timer 3, 4 output (T3O, T4O) waveform is not available.

According to the EIPOL0L register setting, the external interrupt EINT0 and EINT1 function is chose. Of cource, the EINT0 and EINT1 pins must be set to an input port.

The T3CAPR and T3DR are in the same address. In the capture mode, reading operation reads T3CAPR, not T3DR and writing operation will update T3DR. The T4CAPR has the same function.



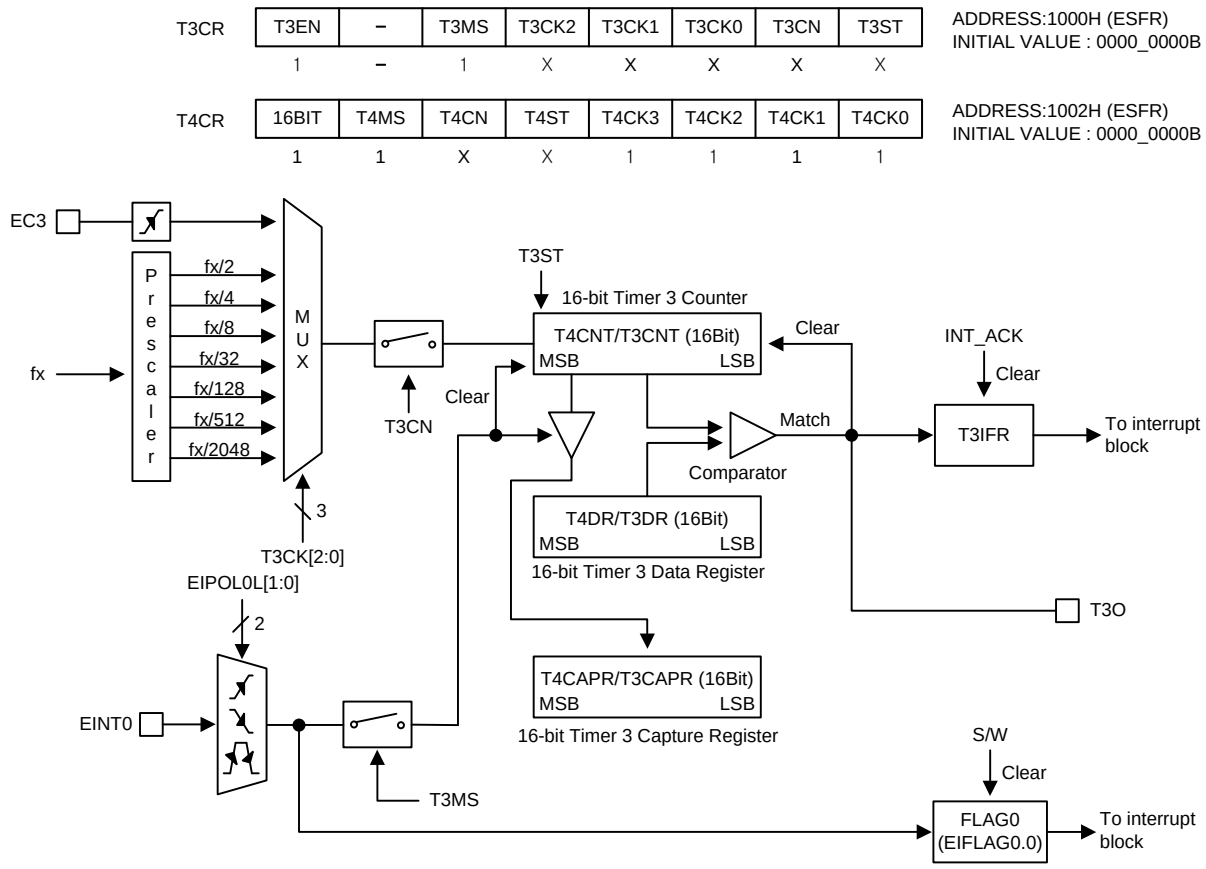
NOTE: Do not set to “1111b” in the T4CK[3:0], when two 8-bit timer 3/4 modes.

Figure 11.32 8-Bit Capture Mode for Timer 3, 4

11.8.5 16-Bit Timer 3 Capture Mode

The 16-bit Capture mode is selected by control register as shown in Figure 11.33.

The 16-bit capture mode is the same operation as 8-bit capture mode, except that the timer register uses 16 bits. The 16-bit timer 3 capture mode is set by T3MS, T4MS as '1'. The clock source is selected from T3CK[2:0] and 16BIT bit must be set to '1'. Timer 3 is LSB 8-bit, the timer 4 is MSB 8-bit.



NOTE) The T4CR.7 bit (16BIT) should be set to '1' and the T4CK[3:0] should be set to "1111b".

Figure 11.33 16-Bit Capture Mode for Timer 3

11.8.6 10-Bit Timer 4 PWM Mode

The timer 4 has a high speed PWM (Pulse Width Modulation) function. In PWM mode, the 6-channel pins output up to 10-bit resolution PWM output. This pin should be configured as a PWM output by set PWM4E to '1'. When the value of 2bit +T4CNT and T4PPRH/L are identical in timer 4, a period match signal is generated and the interrupt of timer 4 occurs. In 10-bit PWM mode, A, B, C, bottom(underflow) match signal are generated when the 10-bit counter value are identical to the value of T4xADRH/L. The period of the PWM output is determined by the T4PPRH/L (PWM period register), T4xDRH/L (each channel PWM duty register).

PWM Period = [T4PPRH/T4PPRL] X Source Clock

PWM Duty(A-ch) = [T4ADRH/T4ADRL] X Source Clock

Table 11-12 PWM Frequency vs. Resolution at 8 MHz

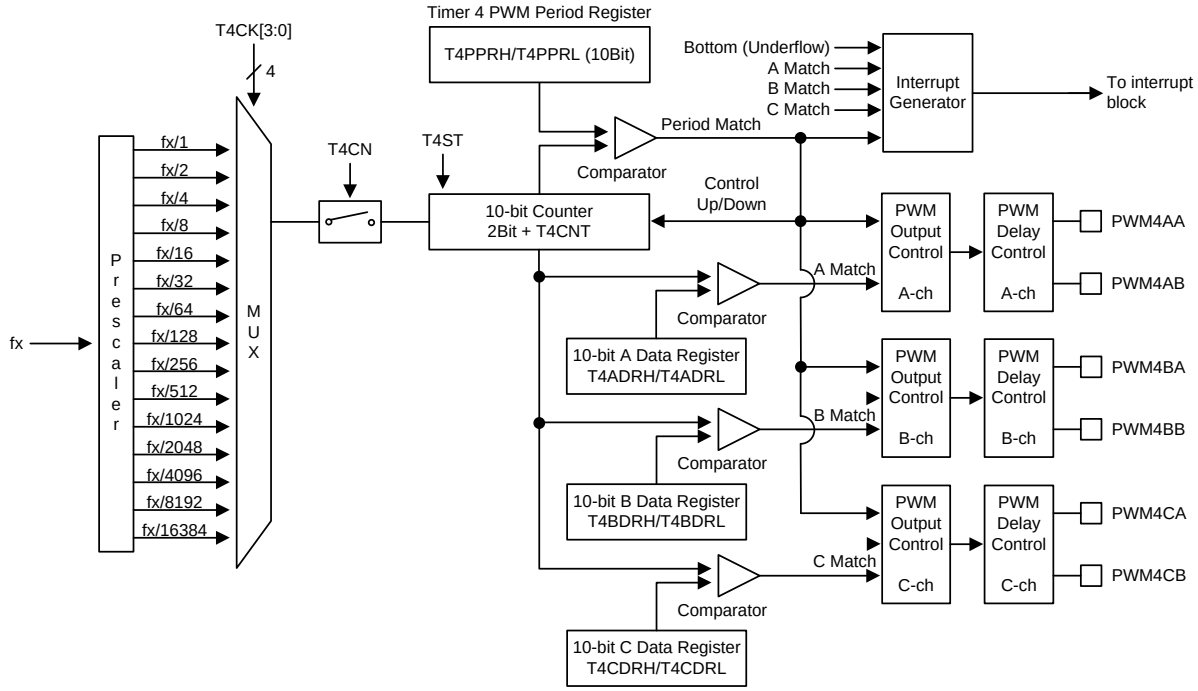
Resolution	Frequency		
	T4CK[3:0]=0001 (250ns)	T4CK[3:0]=0010 (500ns)	T4CK[3:0]=0100 (2us)
10 Bit	3.9KHz	1.95KHz	0.49KHz
9 Bit	7.8KHz	3.9KHz	0.98KHz
8 Bit	15.6KHz	7.8KHz	1.95KHz
7 Bit	31.2KHz	15.6KHz	3.91KHz

The POLxA bit of T4PCR3 register decides the polarity of duty cycle. If the duty value is set same to the period value, the PWM output is determined by the bit POLxA (1: High, 0: Low). And if the duty value is set to "00H", the PWM output is determined by the bit POLxA (1: Low, 0: High).

Table 11-13 PWM Channel Polarity

PHLT:PxxOE	POLxA	POLBO	POLxB	PWM4xA Pin Output	PWM4xB Pin Output
0x, x0, 00	0	0	0	Low-level	Low-level
			1	Low-level	High-level
		1	x	Low-level	Low-level
	1	0	0	High-level	High-level
			1	High-level	Low-level
		1	x	High-level	High-level
11	0	x	0	Positive-phase	Positive-Phase
			1	Positive-phase	Negative-Phase
	1	x	0	Negative-Phase	Negative-Phase
			1	Negative-Phase	Positive-phase

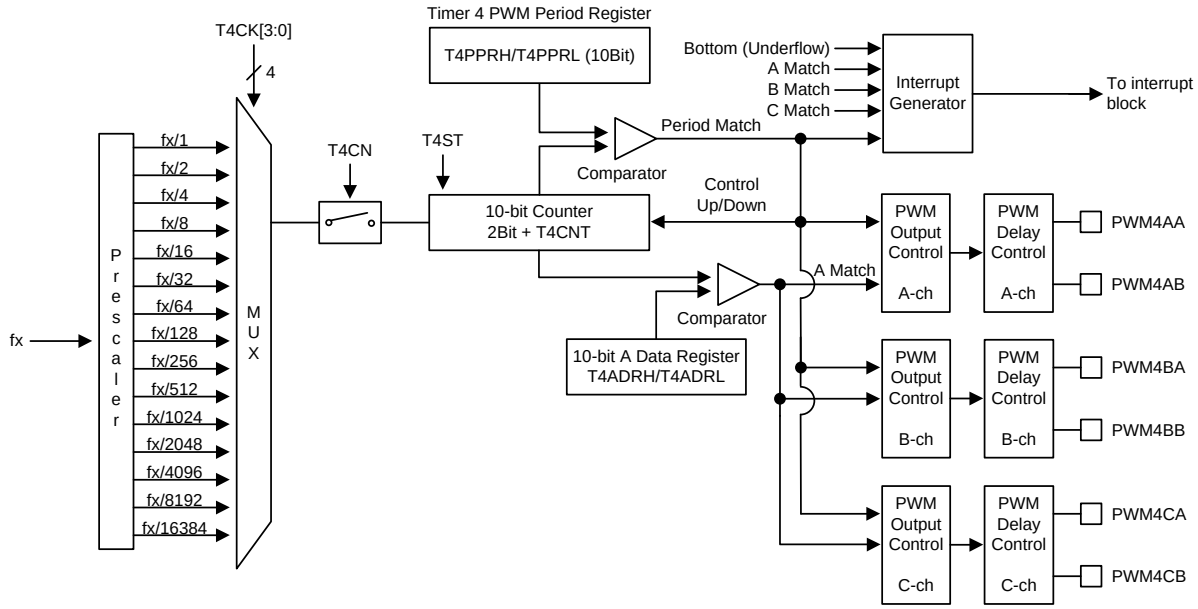
T4CR	16BIT	T4MS	T4CN	T4ST	T4CK3	T4CK2	T4CK1	T4CK0	ADDRESS:1002H (ESFR) INITIAL VALUE : 0000_0000B
	0	X	X	X	X	X	X	X	
T4PCR1	PWM4E	ESYNC	BMOD	PHLT	UPDT	UALL	NOPS1	NOPS0	ADDRESS:1003H (ESFR) INITIAL VALUE : 0000_0000B
	1	X	X	X	X	X	X	X	
T4PCR2	FORCA	–	PAAOE	PABOE	PBAOE	PBBOE	PCAOE	PCBOE	ADDRESS:1004H (ESFR) INITIAL VALUE : 0000_0000B
	0	–	X	X	X	X	X	X	
T4PCR3	HZCLR	POLBO	POLAA	POLAB	POLBA	POLBB	POLCA	POLCB	ADDRESS:1005H (ESFR) INITIAL VALUE : 0000_0000B
	X	X	X	X	X	X	X	X	



NOTE: Do not set to “1111b” in the T4CK[3:0], when two 8-bit timer 3/4 modes.

Figure 11.34 10-Bit PWM Mode (Force 6-ch)

T4CR	16BIT	T4MS	T4CN	T4ST	T4CK3	T4CK2	T4CK1	T4CK0	ADDRESS:1002H (ESFR) INITIAL VALUE : 0000_0000B
	0	X	X	X	X	X	X	X	
T4PCR1	PWM4E	ESYNC	BMOD	PHLT	UPDT	UALL	NOPS1	NOPS0	ADDRESS:1003H (ESFR) INITIAL VALUE : 0000_0000B
	1	X	X	X	X	X	X	X	
T4PCR2	FORCA	–	PAAOE	PABOE	PBAOE	PBBOE	PCAOE	PCBOE	ADDRESS:1004H (ESFR) INITIAL VALUE : 0000_0000B
	1	–	X	X	X	X	X	X	
T4PCR3	HZCLR	POLBO	POLAA	POLAB	POLBA	POLBB	POLCA	POLCB	ADDRESS:1005H (ESFR) INITIAL VALUE : 0000_0000B
	X	X	X	X	X	X	X	X	



NOTE: Do not set to “1111b” in the T4CK[3:0], when two 8-bit timer 3/4 modes.

Figure 11.35 10-Bit PWM Mode (Force All-ch)

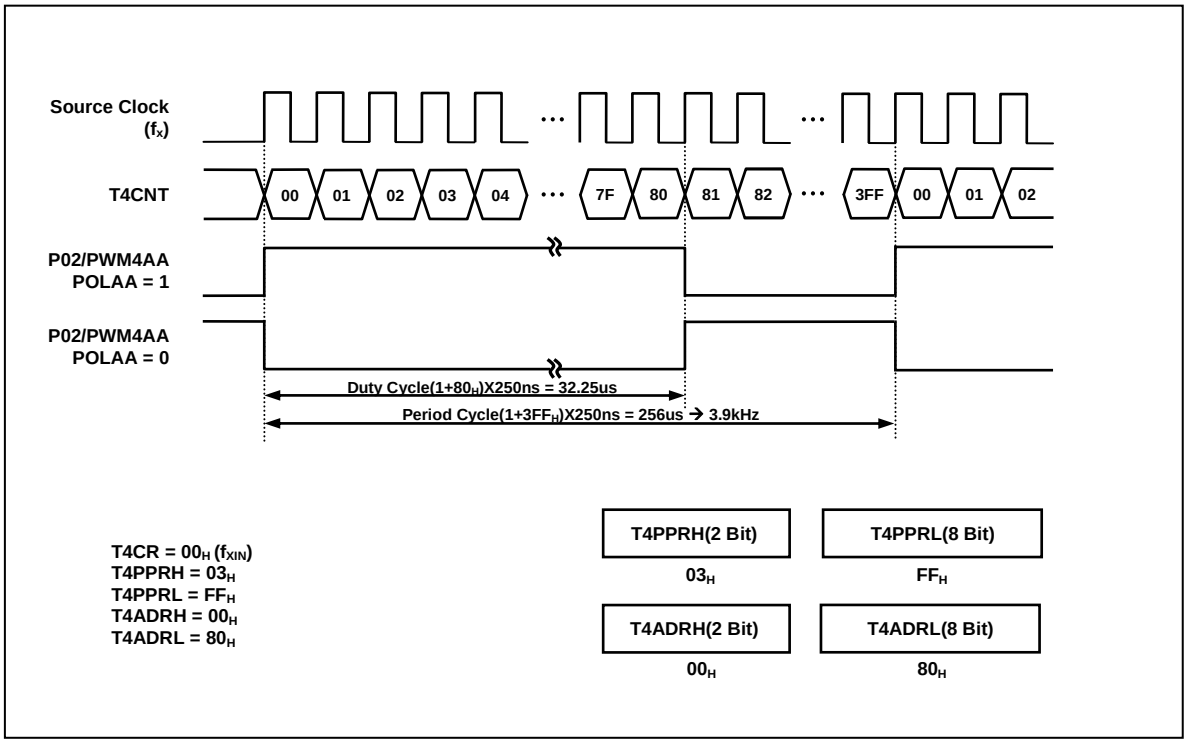


Figure 11.36 Example of PWM at 4 MHz

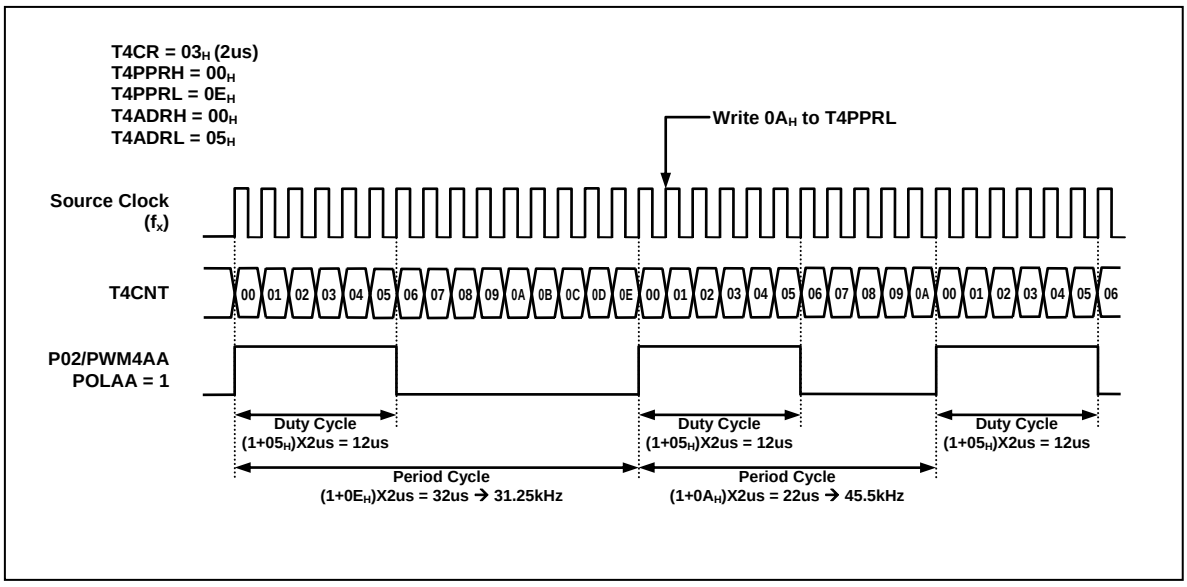


Figure 11.37 Example of Changing the Period in Absolute Duty Cycle at 4 MHz

Update period & duty register value at once

The period and duty of PWM comes to move from temporary registers to T4PPRH/L (PWM Period Register) and T4ADRH/L/T4BDRH/L/T4CDRH/L (PWM Duty Register) when always period match occurs. If you want that the period and duty is immediately changed, the UPDT bit in the T4PCR1 register must set to '1'. It should be noted that it needs the 3 cycle of timer clock for data transfer in the internal clock synchronization circuit. So the update data is written before 3 cycle of timer clock to get the right output waveform.

Phase correction & Frequency correction

On operating PWM, it is possible that it is changed the phase and the frequency by using BMOD bit (back-to-back mode) in T4PCR1 register. (Figure 1.38, Figure 11.39, Figure 11.40 referred)

In the back-to-back mode, the counter of PWM repeats up/down count. In fact, the effective duty and period becomes twofold of the register set values. (Figure 1.38, Figure 11.39 referred)

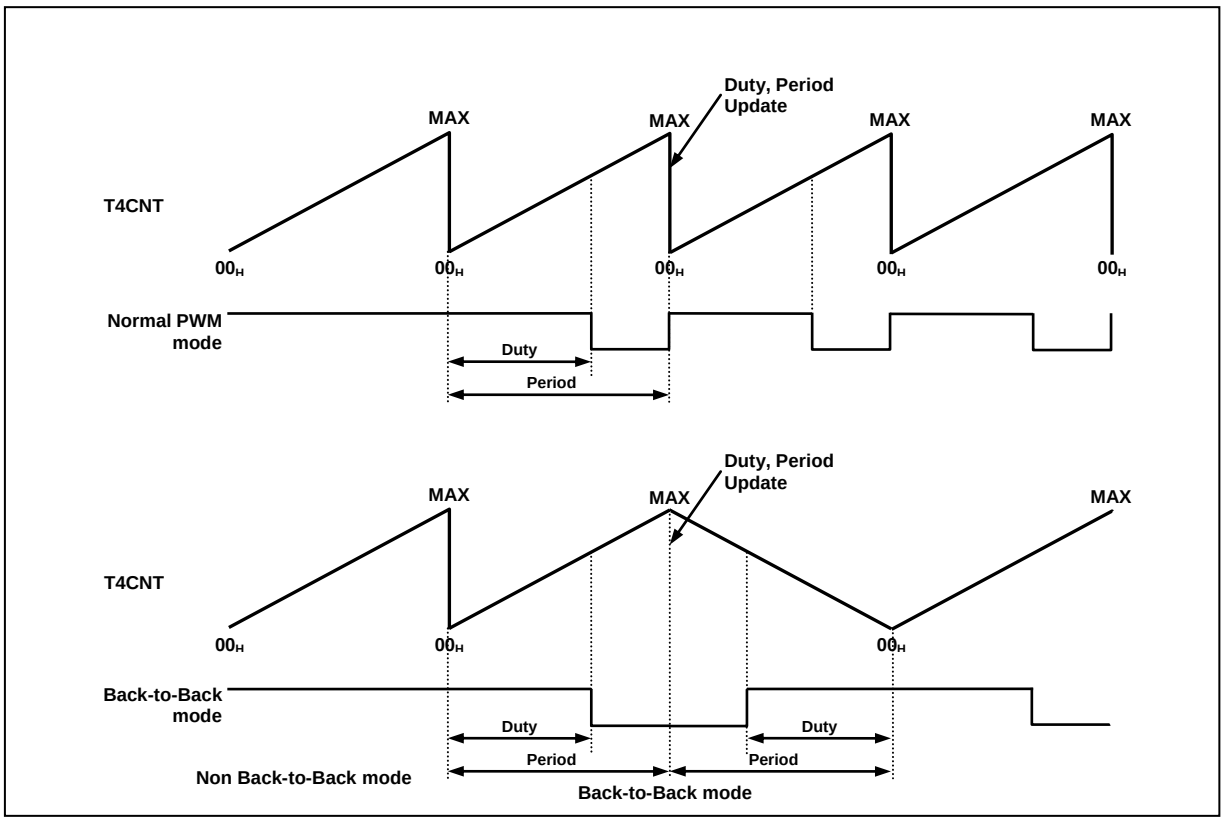


Figure 11.38 Example of PWM Output Waveform

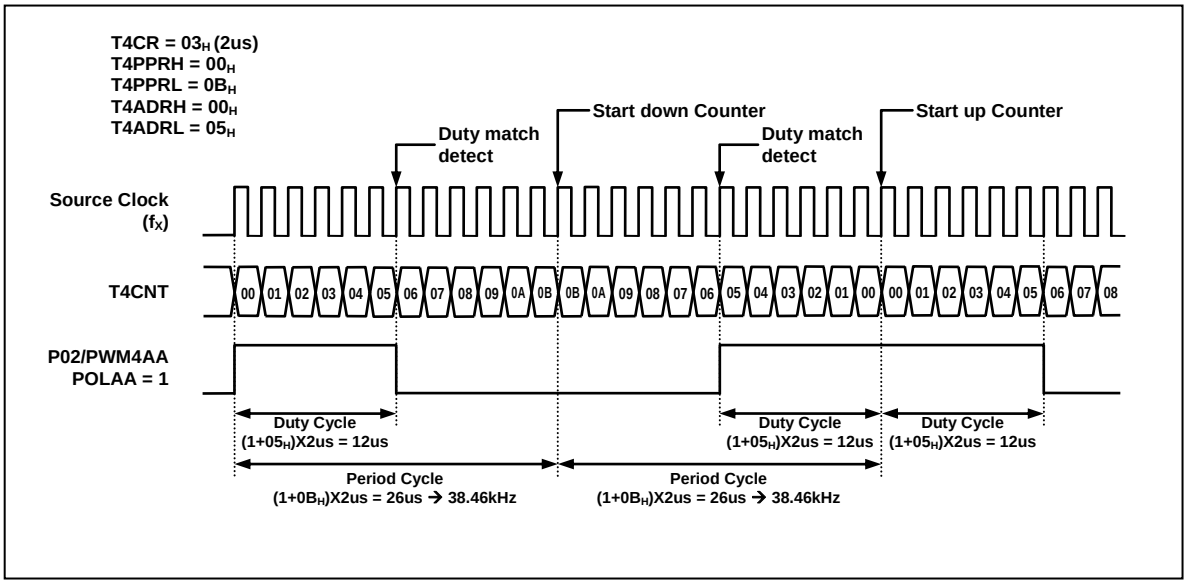


Figure 11.39 Example of PWM waveform in Back-to-Back mode at 4 MHz

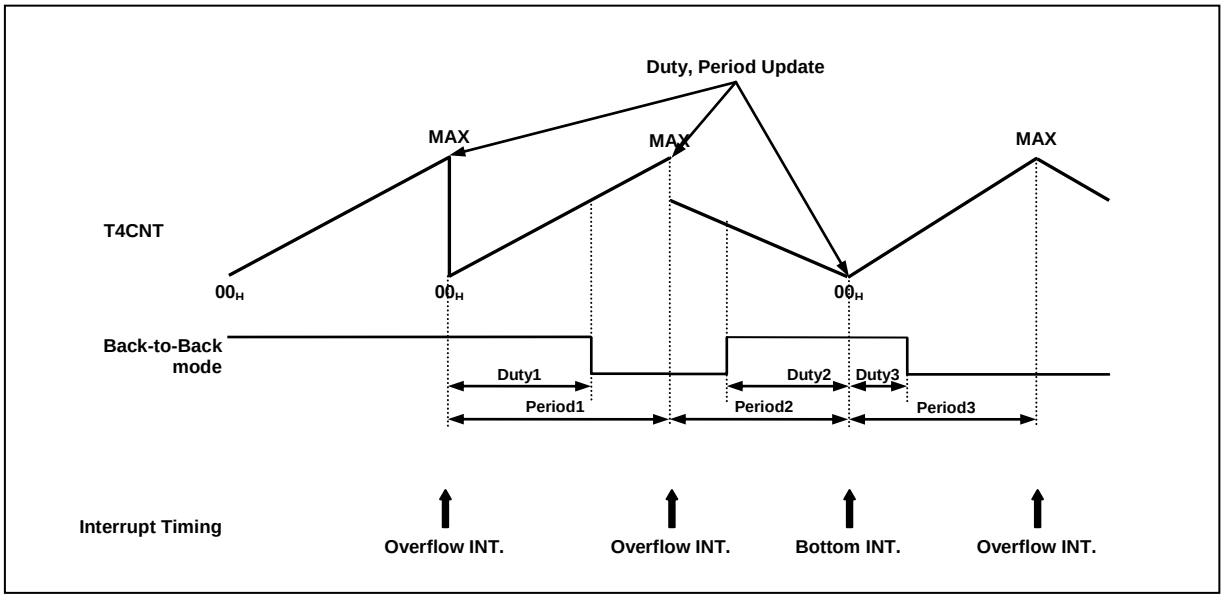


Figure 11.40 Example of Phase Correction and Frequency correction of PWM

External Sync

If using ESYNC bit of T4PCR1 register, it is possible to synchronize the output of PWM from external signal.

If ESYNC bit sets to '1', the external signal moves to PWM module through the BLNK pin. If BLNK signal is low, immediately PWM output becomes a reset value, and internal counter becomes reset. If BLNK signal returns to '1', the counter is started again and PWM output is normally generated. (Figure 11.41 referred)

PWM Halt

If using PHLT bit of T4PCR1 register, it is possible to stop PWM operation by the software. During PHLT bit being '1', PWM output becomes a reset value, and internal counter becomes reset as 0. Without changing PWM setting, temporarily it is able to stop PWM. In case of T4CNT, when stopping counter, PWM output pin remains before states. But if PHLT bit sets to '1', PWM output pin has reset value.

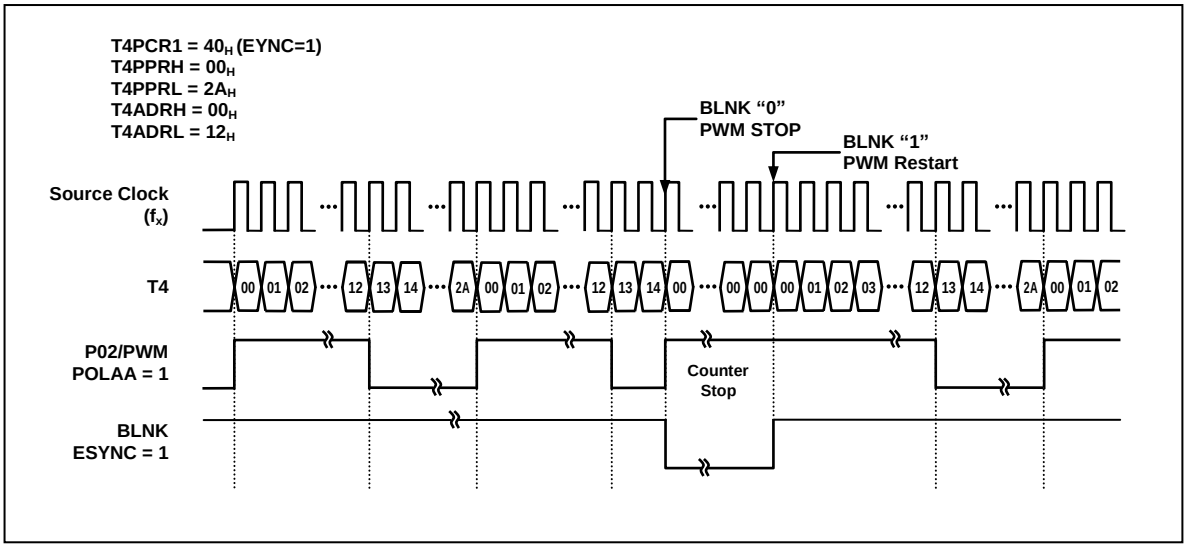


Figure 11.41 Example of PWM External Synchronization with BLNK Input

FORCE Drive ALL Channel with A-ch mode

If FORCA bit sets to '1', it is possible to enable or disable all PWM output pins through PWM outputs which occur from A-ch duty counter. It is noted that the inversion outputs of A, B, C channel have the same A-ch output waveform. According to POLAA/BB/CC, it is able to control the inversion of outputs.

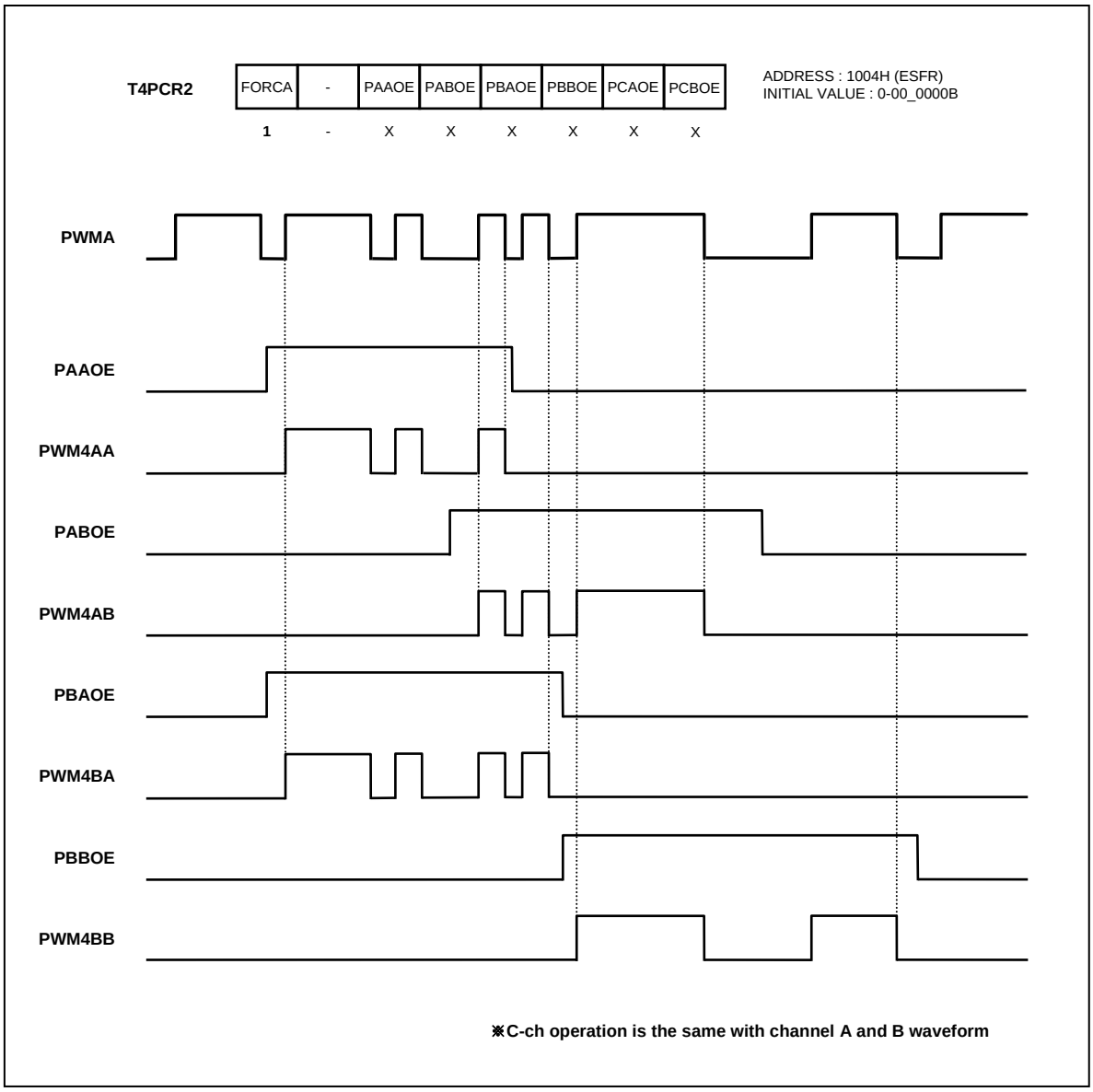


Figure 11.42 Example of Force Drive All Channel with A-ch

FORCE 6-Ch Drive

If FORCA bit sets to ‘0’, it is possible to enable or disable PWM output pin and inversion output pin generated through the duty counter of each channel. The inversion output is the reverse phase of the PWM output. A AA/AB output of the A-channel duty register, a BA/BB output of the B-channel duty register, a CA/CB output of the C-channel duty register are controlled respectively. If the UALL bit is set to ‘1’, it is updated B/C channel duty at the same time, when it is written by a A-channel duty register.

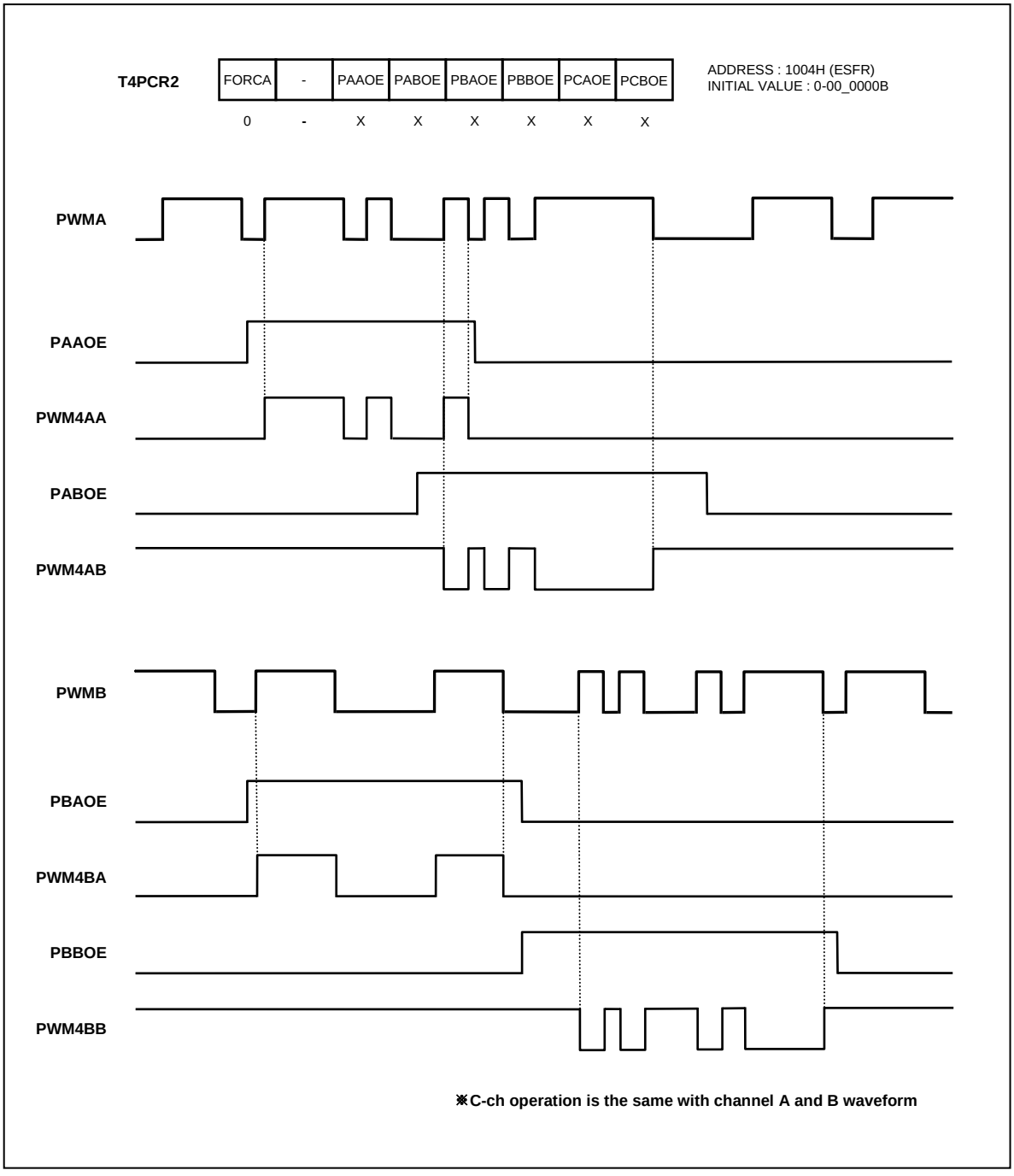


Figure 11.43 Example of Force Drive 6-ch Mode

PWM output Delay

If using the T4DLYA, T4DLYB, T4DLYC register, it can delay PWM output based on the rising edge. At that time, it does not change the falling edge, so the duty is reduced as the time delay. In POLAA/BA/CA setting to '0', the delay is applied to the falling edge. In POLAA/BA/CA setting to '1', the delay is applied to the rising edge. It can produce a pair of Non-overlapping clock. The each channel is able to have 4-bit delay. As it can select the clock up to 1/8 divided clock using NOPS[1:0] the delay of its maximum 128 timer clock cycle is produced.

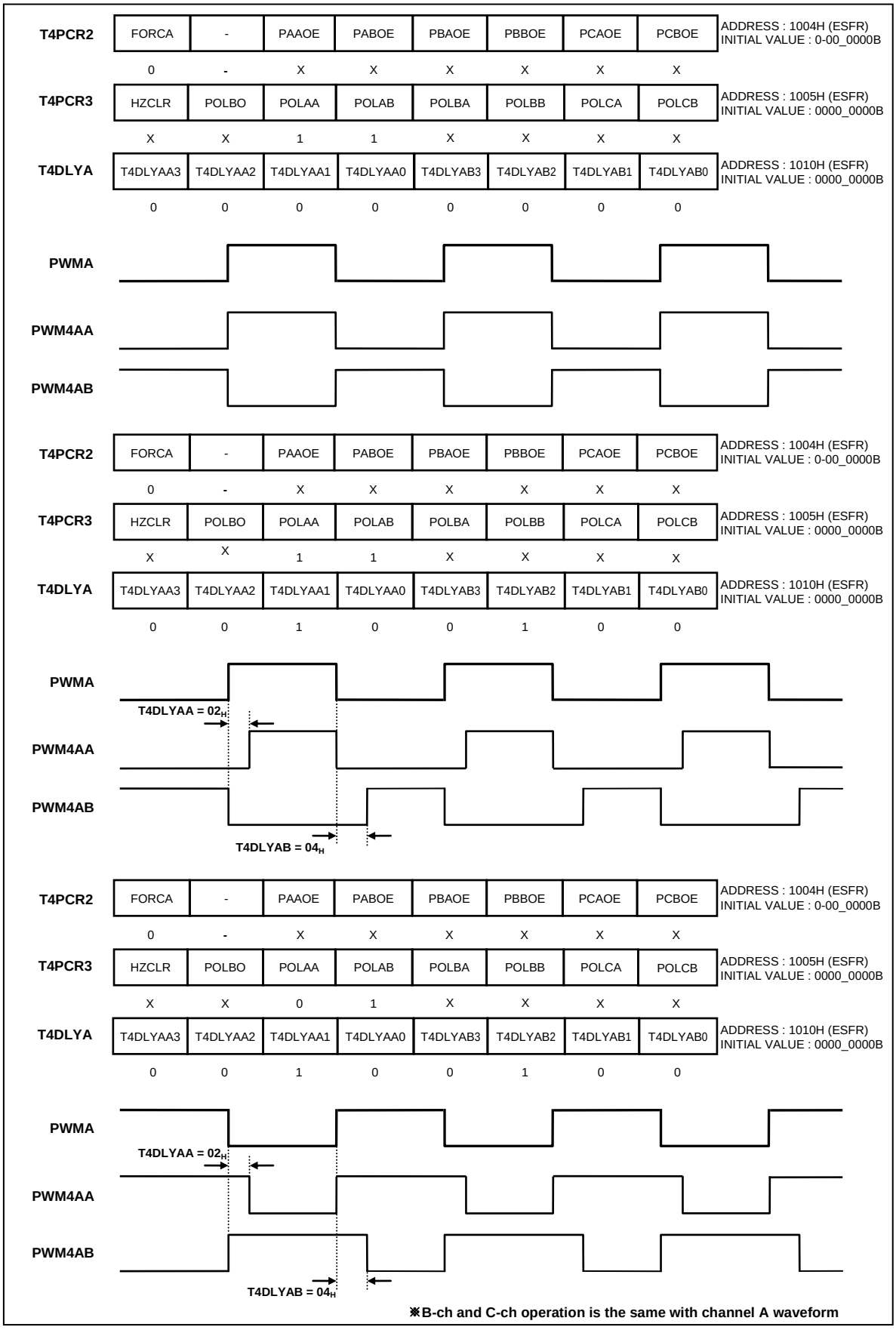
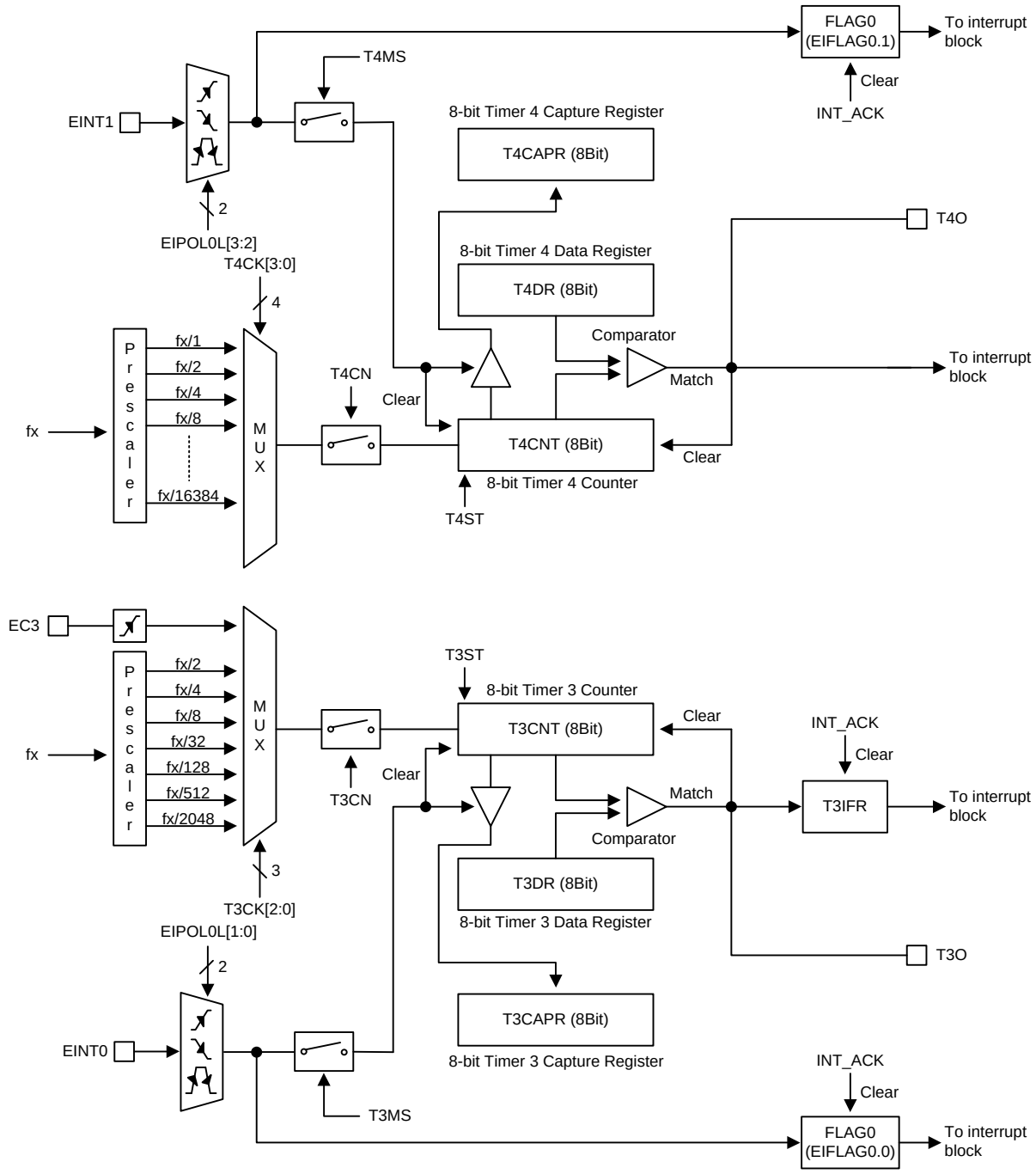


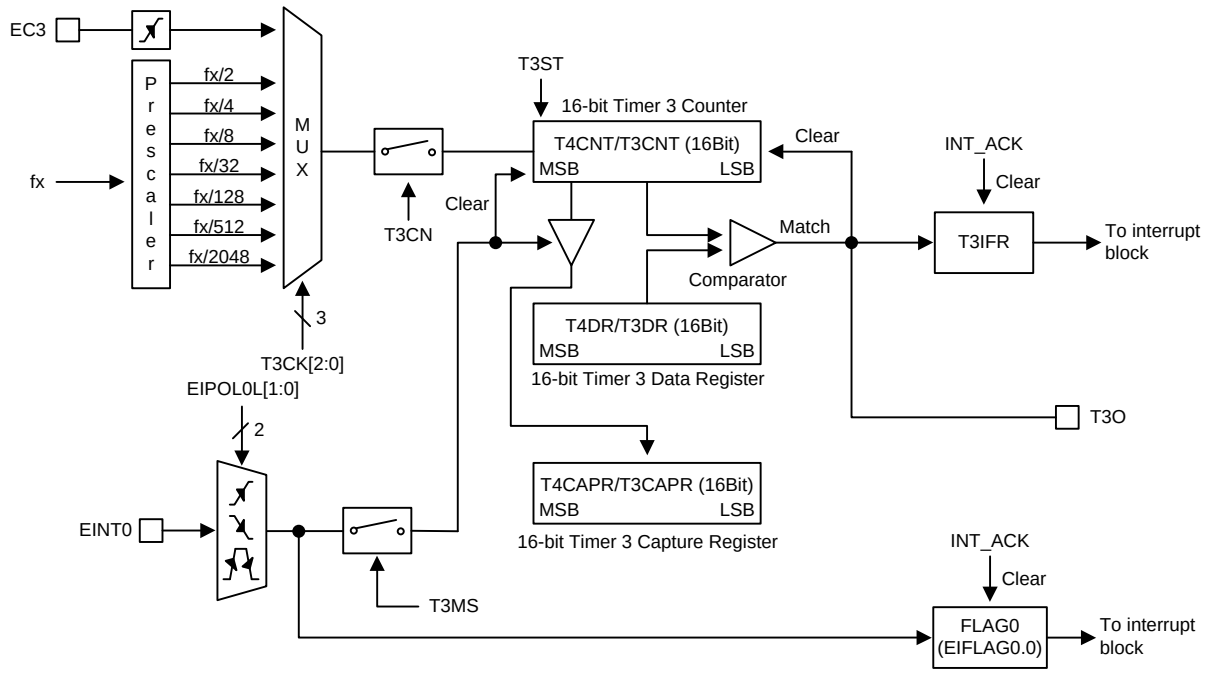
Figure 11.44 Example of PWM Delay

11.8.7 Block Diagram



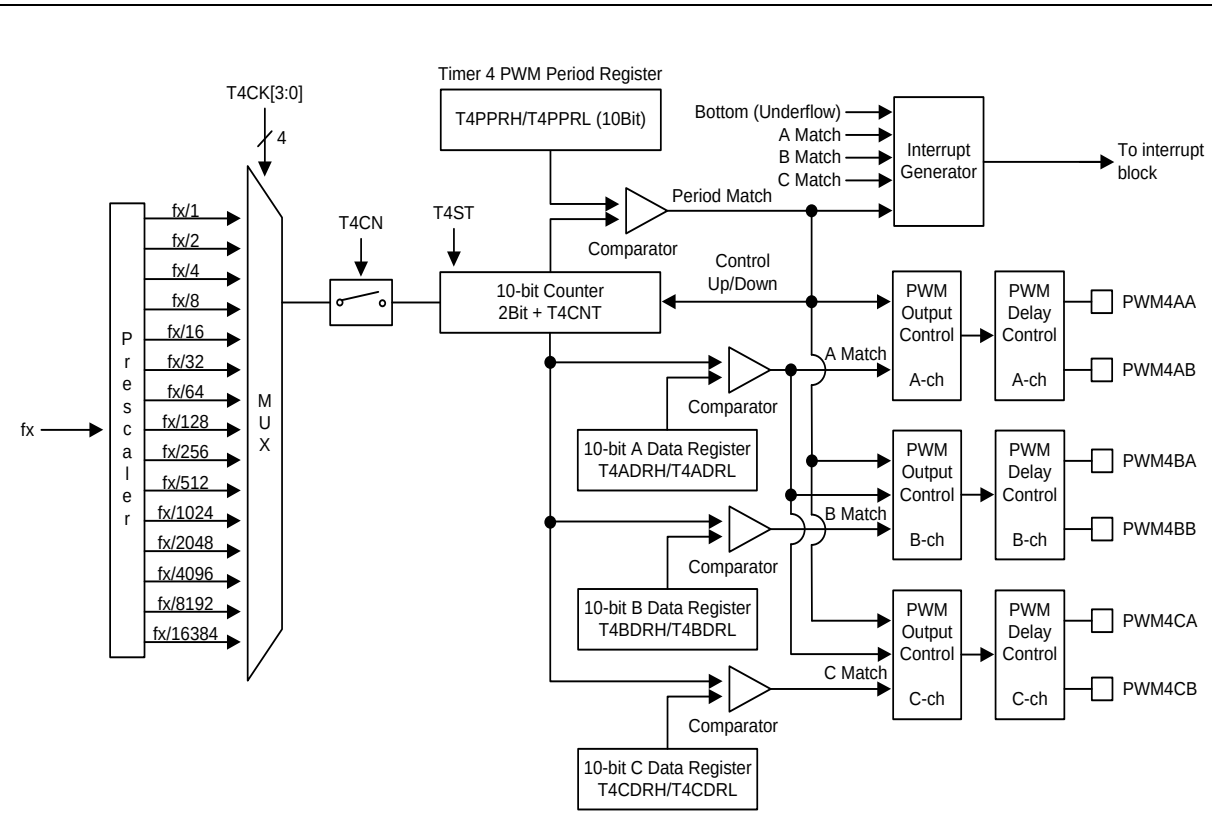
NOTE: Do not set to “1111b” in the T4CK[3:0], when two 8-bit timer 3/4 modes.

Figure 11.45 Two 8-Bit Timer 3, 4 Block Diagram



NOTE) The T4CR.7 bit (16BIT) should be set to '1' and the T4CK[3:0] should be set to "1111b".

Figure 11.46 16-Bit Timer 3 Block Diagram



NOTE: Do not set to "1111b" in the T4CK[3:0], when two 8-bit timer 3/4 modes.

Figure 11.47 10-Bit PWM Timer 4 Block Diagram

11.8.8 Register Map

Table 11-14 Timer 3, 4 Register Map

Name	Address	Dir	Default	Description
T3CNT	1001H (ESFR)	R	00H	Timer 3 Counter Register
T3DR	1001H (ESFR)	W	FFH	Timer 3 Data Register
T3CAPR	1001H (ESFR)	R	00H	Timer 3 Capture Data Register
T3CR	1000H (ESFR)	R/W	00H	Timer 3 Control Register
T4PPRH	1009H (ESFR)	R/W	00H	Timer 4 PWM Period High Register
T4PPRL	1008H (ESFR)	R/W	FFH	Timer 4 PWM Period Low Register
T4ADRH	100BH (ESFR)	R/W	00H	Timer 4 PWM A Duty High Register
T4ADRL	100AH (ESFR)	R/W	7FH	Timer 4 PWM A Duty Low Register
T4BDRH	100DH (ESFR)	R/W	00H	Timer 4 PWM B Duty High Register
T4BDRL	100CH (ESFR)	R/W	7FH	Timer 4 PWM B Duty Low Register
T4CDRH	100FH (ESFR)	R/W	00H	Timer 4 PWM C Duty High Register
T4CDRL	100EH (ESFR)	R/W	7FH	Timer 4 PWM C Duty Low Register
T4DLYA	1010H (ESFR)	R/W	00H	Timer 4 PWM A Delay Register
T4DLYB	1011H (ESFR)	R/W	00H	Timer 4 PWM B Delay Register
T4DLYC	1012H (ESFR)	R/W	00H	Timer 4 PWM C Delay Register
T4DR	1013H (ESFR)	R/W	FFH	Timer 4 Data Register
T4CAPR	1014H (ESFR)	R	00H	Timer 4 Capture Data Register
T4CNT	1015H (ESFR)	R	00H	Timer 4 Counter Register
T4CR	1002H (ESFR)	R/W	00H	Timer 4 Control Register
T4PCR1	1003H (ESFR)	R/W	00H	Timer 4 PWM Control Register 1
T4PCR2	1004H (ESFR)	R/W	00H	Timer 4 PWM Control Register 2
T4PCR3	1005H (ESFR)	R/W	00H	Timer 4 PWM Control Register 3
T4ISR	1006H (ESFR)	R/W	00H	Timer 4 Interrupt Status Register
T4MSK	1007H (ESFR)	R/W	00H	Timer 4 Interrupt Mask Register

11.8.8.1 Timer/Counter 3 Register Description

The timer/counter 3 register consists of timer 3 counter register (T3CNT), timer 3 data register (T3DR), timer 3 capture data register (T3CAPR) and timer 3 control register (T3CR).

11.8.8.2 Register Description for Timer/Counter 3

T3CNT (Timer 3 Counter Register: Read Case, Timer mode only) : 1001H (ESFR)

7	6	5	4	3	2	1	0
T3CNT7	T3CNT6	T3CNT5	T3CNT4	T3CNT3	T3CNT2	T3CNT1	T3CNT0
R	R	R	R	R	R	R	R

Initial value : 00H

T3CNT[7:0] T3 Counter

T3DR (Timer 3 Data Register: Write Case) : 1001H (ESFR)

7	6	5	4	3	2	1	0
T3DR7	T3DR6	T3DR5	T3DR4	T3DR3	T3DR2	T3DR1	T3DR0
W	W	W	W	W	W	W	W

Initial value : FFH

T3DR[7:0] T3 Data

T3CAPR (Timer 3 Capture Data Register: Read Case, Capture mode only) : 1001H (ESFR)

7	6	5	4	3	2	1	0
T3CAPR7	T3CAPR6	T3CAPR5	T3CAPR4	T3CAPR3	T3CAPR2	T3CAPR1	T3CAPR0
R	R	R	R	R	R	R	R

Initial value : 00H

T3CAPR[7:0] T3 Capture Data

T3CR (Timer 3 Control Register) : 1000H (ESFR)

7	6	5	4	3	2	1	0
T3EN	–	T3MS	T3CK2	T3CK1	T3CK0	T3CN	T3ST
R/W	–	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

T3EN Control Timer 3
- 0 Timer 3 disable
- 1 Timer 3 enable

T3MS Control Timer 3 Operation Mode
- 0 Timer/counter mode (T3O: toggle at match)
- 1 Capture mode (the match interrupt can occur)

T3CK[2:0] Select Timer 3 clock source. fx is main system clock frequency

T3CK2	T3CK1	T3CK0	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 (EC3)

T3CN Control Timer 3 Count Pause/Continue
- 0 Temporary count stop
- 1 Continue count

T3ST Control Timer 3 Start/Stop
- 0 Counter stop
- 1 Clear counter and start

NOTE) Refer to the external interrupt flag 1 register (EIFLAG1) tor the T3 interrupt flag.

11.8.8.3 Timer/Counter 4 Register Description

The timer/counter 4 register consists of timer 4 PWM period high/low register (T4PPRH/L), timer 4 PWM A duty high/low register (T4ADRH/L), timer 4 PWM B duty high/low register (T4BDRH/L),), timer 4 PWM C duty high/low register (T4CDRH/L), timer 4 PWM A delay register (T4DLYA), timer 4 PWM B delay register (T4DLYB), timer 4 PWM C delay register (T4DLYC), timer 4 data register (T4DR), timer 4 capture data register (T4CAPR), timer 4 counter register (T4CNT), timer 4 control register (T4CR), timer 4 PWM control register 1 (T4PCR1), timer 4 PWM control register 2 (T4PCR2), timer 4 PWM control register 3 (T4PCR3), timer 4 interrupt status register (T4ISR) and timer 4 interrupt mask register (T4MSK).

11.8.8.4 Register Description for Timer/Counter 4

T4PPRH (Timer 4 PWM Period High Register : 6-ch PWM mode only) : 1009H (ESFR)

7	6	5	4	3	2	1	0
–	–	–	–	–	–	T4PPRH1	T4PPRH0
–	–	–	–	–	–	R/W	R/W

Initial value : 00H

T4PPRL[1:0] T4 PWM Period Data High Byte

T4PPRL (Timer 4 PWM Period Low Register : 6-ch PWM mode only) : 1008H (ESFR)

7	6	5	4	3	2	1	0
T4PPRL7	T4PPRL6	T4PPRL5	T4PPRL4	T4PPRL3	T4PPRL2	T4PPRL1	T4PPRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FFH

T4PPRL[7:0] T4 PWM Period Data Low Byte

T4ADRH (Timer 4 PWM A Duty High Register : 6-ch PWM mode only) : 100BH (ESFR)

7	6	5	4	3	2	1	0
–	–	–	–	–	–	T4ADRH1	T4ADRH0
–	–	–	–	–	–	R/W	R/W

Initial value : 00H

T4ADRL[1:0] T4 PWM A Duty Data High Byte

T4ADRL (Timer 4 PWM A Duty Low Register : 6-ch PWM mode only) : 100AH (ESFR)

7	6	5	4	3	2	1	0
T4ADRL7	T4ADRL6	T4ADRL5	T4ADRL4	T4ADRL3	T4ADRL2	T4ADRL1	T4ADRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 7FH

T4ADRL[7:0] T4 PWM A Duty Data Low Byte

T4BDRH (Timer 4 PWM B Duty High Register : 6-ch PWM mode only) : 100DH (ESFR)

7	6	5	4	3	2	1	0
–	–	–	–	–	–	T4BDRH1	T4BDRH0
–	–	–	–	–	–	R/W	R/W

Initial value : 00H

T4BDRL[1:0] T4 PWM B Duty Data High Byte

T4BDRL (Timer 4 PWM B Duty Low Register : 6-ch PWM mode only) : 100CH (ESFR)

7	6	5	4	3	2	1	0
T4BDRL7	T4BDRL6	T4BDRL5	T4BDRL4	T4BDRL3	T4BDRL2	T4BDRL1	T4BDRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 7FH

T4BDRL[7:0] T4 PWM B Duty Data Low Byte

T4CDRH (Timer 4 PWM C Duty High Register : 6-ch PWM mode only) : 100FH (ESFR)

7	6	5	4	3	2	1	0
–	–	–	–	–	–	T4CDRH1	T4CDRH0
–	–	–	–	–	–	R/W	R/W

Initial value : 00H

T4CDRL[1:0] T4 PWM C Duty Data High Byte

T4CDRL (Timer 4 PWM C Duty Low Register : 6-ch PWM mode only) : 100EH (ESFR)

7	6	5	4	3	2	1	0
T4CDRL7	T4CDRL6	T4CDRL5	T4CDRL4	T4CDRL3	T4CDRL2	T4CDRL1	T4CDRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 7FH

T4CDRL[7:0] T4 PWM C Duty Data Low Byte

T4DLYA (Timer 4 PWM A Delay Register : 6-ch PWM mode only) : 1010H (ESFR)

7	6	5	4	3	2	1	0
T4DLYAA3	T4DLYAA2	T4DLYAA1	T4DLYAA0	T4DLYAB3	T4DLYAB2	T4DLYAB1	T4DLYAB0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

T4DLYAA[3:0] PWM4AA Delay Data (Rising edge only)

T4DLYAB[3:0] PWM4AB Delay Data (Rising edge only)

T4DLYB (Timer 4 PWM B Delay Register : 6-ch PWM mode only) : 1011H (ESFR)

7	6	5	4	3	2	1	0
T4DLYBA3	T4DLYBA2	T4DLYBA1	T4DLYBA0	T4DLYBB3	T4DLYBB2	T4DLYBB1	T4DLYBB0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

T4DLYBA[3:0] PWM4BA Delay Data (Rising edge only)

T4DLYBB[3:0] PWM4BB Delay Data (Rising edge only)

T4DLYC (Timer 4 PWM C Delay Register : 6-ch PWM mode only) : 1012H (ESFR)

7	6	5	4	3	2	1	0
T4DLYCA3	T4DLYCA2	T4DLYCA1	T4DLYCA0	T4DLYCB3	T4DLYCB2	T4DLYCB1	T4DLYCB0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

T4DLYCA[3:0] PWM4CA Delay Data (Rising edge only)

T4DLYCB[3:0] PWM4CB Delay Data (Rising edge only)

T4DR (Timer 4 Data Register: Timer and Capture mode only) : 1013H (ESFR)

7	6	5	4	3	2	1	0
T4DR7	T4DR6	T4DR5	T4DR4	T4DR3	T4DR2	T4DR1	T4DR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FFH

T4DR[7:0] T4 Data

T4CAPR (Timer 4 Capture Data Register: Read Case, Capture mode only) : 1014H (ESFR)

7	6	5	4	3	2	1	0
T4CAPR7	T4CAPR6	T4CAPR5	T4CAPR4	T4CAPR3	T4CAPR2	T4CAPR1	T4CAPR0
R	R	R	R	R	R	R	R

Initial value : 00H

T4CAPR[7:0] T4 Capture Data

T4CNT (Timer 4 Counter Register: Read Case, Timer mode only) : 1015H (ESFR)

7	6	5	4	3	2	1	0
T4CNT7	T4CNT6	T4CNT5	T4CNT4	T4CNT3	T4CNT2	T4CNT1	T4CNT0
R	R	R	R	R	R	R	R

Initial value : 00H

T4CNT[7:0] T4 Counter

T4CR (Timer 4 Control Register) : 1002H (ESFR)

7	6	5	4	3	2	1	0
16BIT	T4MS	T4CN	T4ST	T4CK3	T4CK2	T4CK1	T4CK0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

16BIT Select Two 8-bit or 16-bit Mode for Timer 3/4

0 Two 8-bit Timer 3/4

1 16-bit Timer 3

T4MS Control Timer 4 Operation Mode

0 Timer/counter mode (T4O: toggle at match)

1 Capture mode (the match interrupt can occur)

T4CN Control Timer 4 Count Pause/Continue

0 Temporary count stop

1 Continue count

T4ST Control Timer 4 Start/Stop

0 Counter stop

1 Clear counter and start

T4CK[3:0] Select Timer 4 clock source. fx is main system clock frequency

T4CK3	T4CK2	T4CK1	T4CK0	Description
0	0	0	0	fx/1
0	0	0	1	fx/2
0	0	1	0	fx/3
0	0	1	1	fx/8
0	1	0	0	fx/16
0	1	0	1	fx/32
0	1	1	0	fx/64
0	1	1	1	fx/128
1	0	0	0	fx/256
1	0	0	1	fx/512
1	0	1	0	fx/1024
1	0	1	1	fx/2048
1	1	0	0	fx/406
1	1	0	1	fx/8192
1	1	1	0	fx/16384
1	1	1	1	Timer 3 clock (only 16-Bit Timer 3)

T4PCR1 (Timer 4 PWM Control Register 1) : 1003H (ESFR)

7	6	5	4	3	2	1	0
PWM4E	ESYNC	BMOD	PHLT	UPDT	UALL	NOPS1	NOPS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

PWM4E Control Timer 4 Mode

0 Select timer/counter or capture mode of Timer 4

1 Select 10-bit PWM mode of Timer 4

ESYNC Select the Operation of External Sync with the BLNK pin

0 Disable external sync operation

1 Enable external sync operation
(The all PWM4xA/PWM4xB pins are high-impedance outputs on rising edge of the BLNK input pin. Where x= A, B and C)

BMOD Control Back-to-Back Mode Operation

0 Disable back-to-back mode (up count only)

1 Enable back-to-back mode (up/down count only)

PHLT Control Timer 4 PWM Operation

0 Run 10-bit PWM

1 Stop 10-bit PWM (counter hold and output disable)

UPDT Select the Update Timer of T4PPR/T4ADR/T4BDR/T4CDR

0 Update at period match of T4CNT and T4PPR

1 Update at any time when written

UALL Control Update All Duty Registers (T4ADR/T4BDR/T4CDR)

0 Write a duty register separately

1 Wrtie all duty registers via Timer 4 PWM A dury register (T4ADR)

NOPS[1:0] Select on-Overlap Prescaler

NOPS1	NOPS0	Description
0	0	f_{PWM}/1
0	1	f_{PWM} /2
1	0	f_{PWM} /4
1	1	f_{PWM} /8

NOTE) Where the f_{PWM} is the clock frequency of the Timer 4 PWM.

T4PCR2 (Timer 4 PWM Control Register 2) : 1004H (ESFR)

7	6	5	4	3	2	1	0
FORCA	–	PAAOE	PABOE	PBAOE	PBBOE	PCAOE	PCBOE
R/W	–	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

FORCA Control The PWM outputs Mode

0 6-channel mode
(The PWM4xA/PWM4xB pins are output according to the T4xDR registers, respectively. Where x = A, B and C)

1 Force A-channel mode
(The all PWM4xA/PWM4xB pins are output according to the only T4ADR registers. Where x = A, B and C)

PAAOE Select Channel PWM4AA Operation

0 Disable PWM4AA output

1 Enable PWM4AA output

PABOE Select Channel PWM4AB Operation

0 Disable PWM4AB output

1 Enable PWM4AB output

PBAOE Select Channel PWM4BA Operation

0 Disable PWM4BA output

1 Enable PWM4BA output

PBBOE Select Channel PWM4BB Operation

0 Disable PWM4BB output

1 Enable PWM4BB output

PCAOE Select Channel PWM4CA Operation

0 Disable PWM4CA output

1 Enable PWM4CA output

PCBOE Select Channel PWM4CB Operation

0 Disable PWM4CB output

1 Enable PWM4CB output

T4PCR3 (Timer 4 PWM Control Register 3) : 1005H (ESFR)

7	6	5	4	3	2	1	0
HZCLR	POLBO	POLAA	POLAB	POLBA	POLBB	POLCA	POLCB
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

HZCLR High-Impedance Output Clear Bit

0 No effect

1 Clear high-impedance output
(The PWM4xA/PWM4xB pins are back to output and this bit is automatically cleared to logic '0'. where x = A, B and C)

POLBO Configure PWM4AB/PWM4BB/PWMCB Channel Polarity When these pins are disabled

0 These pins are output according to the polarity setting when disable (POLAB/POLBB/POLCB bits)

1 These pins are same level as the PWM4xA pins regardless of the polarity setting when disable (POLAB/POLBB/POLCB bits, where x = A, B and C)

POLAA Configure PWM4AA Channel Polarity

0 Start at high level (This pin is low level when disable)

1 Start at low level (This pin is high level when disable)

POLAB Configure PWM4AB Channel Polarity

0 Non-inversion signal of PWM4AA pin

1 Inversion signal of PWM4AA pin

POLBA Configure PWM4AA Channel Polarity

0 Start at high level (This pin is low level when disable)

1 Start at low level (This pin is high level when disable)

POLBB Configure PWM4AB Channel Polarity

0 Non-inversion signal of PWM4BA pin

1 Inversion signal of PWM4BA pin

POLCA Configure PWM4CA Channel Polarity

0 Start at high level (This pin is low level when disable)

1 Start at low level (This pin is high level when disable)

POLCB Configure PWM4CB Channel Polarity

0 Non-inversion signal of PWM4CA pin

1 Inversion signal of PWM4CA pin

T4ISR (Timer 4 Interrupt Status Register) : 1006H (ESFR)

7	6	5	4	3	2	1	0
IOVR	IBTM	ICMA	ICMB	ICMC	–	–	–
R/W	R/W	R/W	R/W	R/W	–	–	–

Initial value : 00H

IOVR Timer 4 Overflow Interrupt Status, Write '1' to this bit for clear
- 0 Overflow occurrence
- 1 Overflow no occurrence

IBTM Timer 4 Bottom Interrupt Status, Write '1' to this bit for clear
(In the Back-to-Back mode)
- 0 Bottom occurrence
- 1 Bottom no occurrence

ICMA Timer 4 Compare Match or PWM A-ch Match Interrupt Staus, Write '1' to this bit for clear
- 0 Compare match or PWM A-ch match occurrence
- 1 Compare match or PWM A-ch match no occurrence

ICMB Timer 4 PWM B-ch Match Interrupt Status, Write '1' to this bit for clear
- 0 PWm B-ch match occurrence
- 1 PWm B-ch match no occurrence

ICMC Timer 4 PWM C-ch Match Interrupt Status, Write '1' to this bit for clear
- 0 PWm C-ch match occurrence
- 1 PWm C-ch match no occurrence

T4MSK (Timer 4 Interrupt Mask Register) : 1007H (ESFR)

7	6	5	4	3	2	1	0
OVRMSK	BTMMSK	CMAMSK	CMBMSK	CMCMSK	–	–	–
R/W	R/W	R/W	R/W	R/W	–	–	–

Initial value : 00H

OVRMSK Control Timer 4 Overflow Interrupt
- 0 Disble overflow interrupt
- 1 Enable overflow interrupt

BTMMSK Control Timer 4 Bottom Interrupt
- 0 Disble bottom interrupt
- 1 Enable bottom interrupt

CMAMSK Control Timer 4 Compare Match or PWM A-ch Match Interrupt
- 0 Disble compare match or PWM A-ch match interrupt
- 1 Enable compare match or PWM A-ch match interrupt

CMBMSK Control Timer 4 PWM B-ch Match Interrupt
- 0 Disble PWM B-ch match interrupt
- 1 Enable PWM B-ch match interrupt

CMCMSK Control Timer 4 PWM C-ch Match Interrupt
- 0 Disble PWM C-ch match interrupt
- 1 Enable PWM C-ch match interrupt

11.9 Buzzer Driver

11.9.1 Overview

The Buzzer consists of 8 bit counter, buzzer data register (BUZDR), and buzzer control register (BUZCR). The Square Wave (61.035Hz~125.0 kHz @8MHz) is outputted through P13/SEG17/AN10/EC1/BUZO pin. The buzzer data register (BUZDR) controls the bsuzzer frequency (look at the following expression). In buzzer control register (BUZCR), BUCK[1:0] selects source clock divided by prescaler.

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

Table 11-15 Buzzer Frequency at 8 MHz

BUZDR[7:0]	Buzzer Frequency (kHz)			
	BUZCR[2:1]=00	BUZCR[2:1]=01	BUZCR[2:1]=10	BUZCR[2:1]=11
0000_0000	125kHz	62.5kHz	31.25kHz	15.625kHz
0000_0001	62.5kHz	31.25kHz	15.625kHz	7.812kHz
…	…	…	…	…
1111_1101	492.126Hz	246.063Hz	123.031Hz	61.515Hz
1111_1110	490.196Hz	245.098Hz	122.549Hz	61.274Hz
1111_1111	488.281Hz	244.141Hz	122.07Hz	61.035Hz

11.9.2 Block Diagram

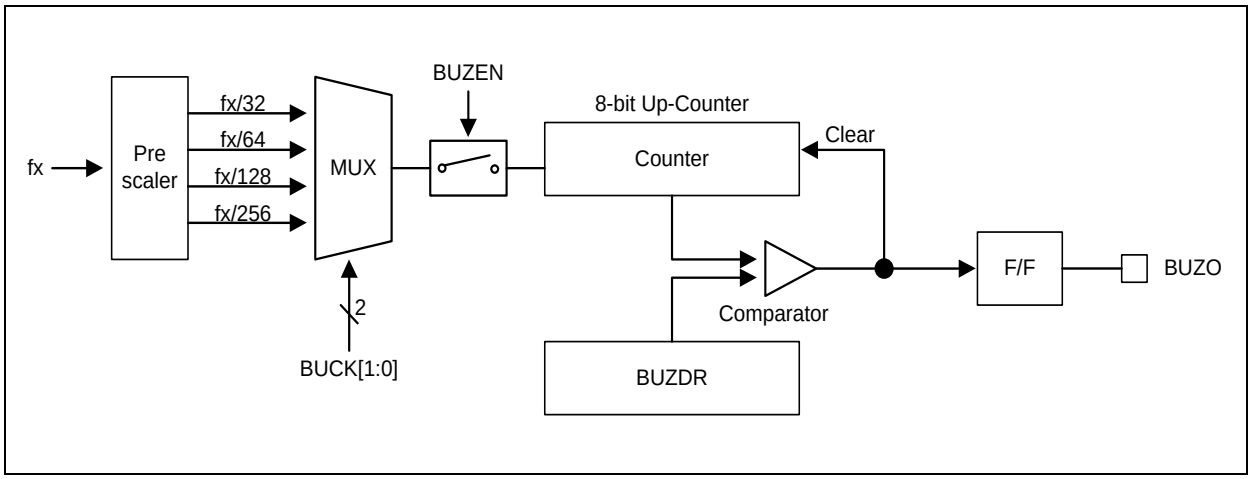


Figure 11.48 Buzzer Driver Block Diagram

11.9.3 Register Map

Table 11-16 Buzzer Driver Register Map

Name	Address	Dir	Default	Description
BUZDR	8FH	R/W	FFH	Buzzer Data Register
BUZCR	97H	R/W	00H	Buzzer Control Register

11.9.4 Buzzer Driver Register Description

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

11.9.5 Register Description for Buzzer Driver

BUZDR (Buzzer Data Register) : 8FH

7	6	5	4	3	2	1	0
BUZDR7	BUZDR6	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

BUZDR[7:0] This bits control the Buzzer frequency
Its resolution is 00H ~ FFH

BUZCR (Buzzer Control Register) : 97H

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

Initial value : 00H

BUCK[1:0] Buzzer Driver Source Clock Selection

BUCK1	BUCK0	Description
0	0	fx/32
0	1	fx/64
1	0	fx/128
1	1	fx/256

BUZEN Buzzer Driver Operation Control

0	Buzzer Driver disable
1	Buzzer Driver enable

NOTE) fx: System clock oscillation frequency.

11.10 SPI 2

11.10.1 Overview

There is serial peripheral interface (SPI 2) one channel in MC96F6432. The SPI 2 allows synchronous serial data transfer between the external serial devices. It can do Full-duplex communication by 4-wire (MOSI2, MISO2, SCK2, SS2), support master/slave mode, can select serial clock (SCK2) polarity, phase and whether LSB first data transfer or MSB first data transfer.

11.10.2 Block Diagram

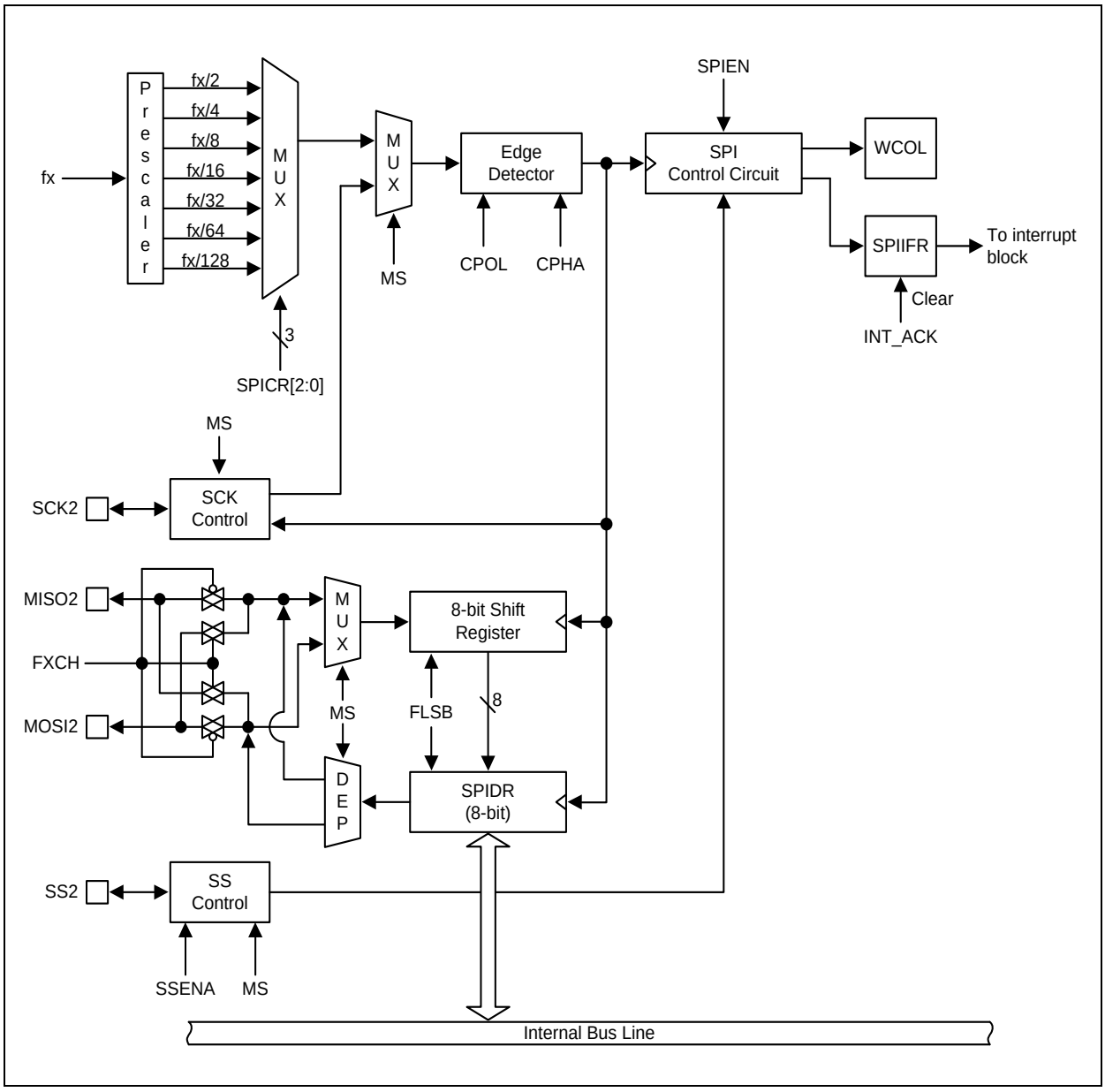


Figure 11.49 SPI 2 Block Diagram

11.10.3 Data Transmit / Receive Operation

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

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

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

11.10.4 SS2 pin function

1. When the SPI 2 is configured as a Slave, the SS2 pin is always input. If LOW signal come into SS2 pin, the SPI 2 logic is active. And if ‘HIGH’ signal come into SS2 pin, the SPI 2 logic is stop. In this time, SPI 2 logic will be reset, and invalidated any received data.
2. When the SPI 2 is configured as a Master, the user can select the direction of the SS2 pin by port direction register (P17IO). If the SS2 pin is configured as an output, user can use general P17IO output mode. If the SS2 pin is configured as an input, ‘HIGH’ signal must come into SS2 pin to guarantee Master operation. If ‘LOW’ signal come into SS2 pin, the SPI 2 logic interprets this as another master selecting the SPI 2 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 becomes a Slave and then, SPIIFR bit of SPISR will be set, and if the SPI 2 interrupt is enabled, an interrupt is requested.

NOTES)

- When the SS2 pin is configured as an output at Master mode, SS2 pin’s output value is defined by user’s software (P17IO). Before SPICR setting, the direction of SS2 pin must be defined
- If you don’t need to use SS2 pin, clear the SSENA bit of SPISR. So, you can use disabled pin by P17IO freely. In this case, SS2 signal is driven by ‘HIGH’ or ‘LOW’ internally. In other words, master is ‘HIGH’, salve is ‘LOW’
- When SS2 pin is configured as input, if ‘HIGH’ signal come into SS2 pin, SS_HIGH flag bit will be set. And you can clear it by writing ‘0’.

11.10.5 SPI 2 Timing Diagram

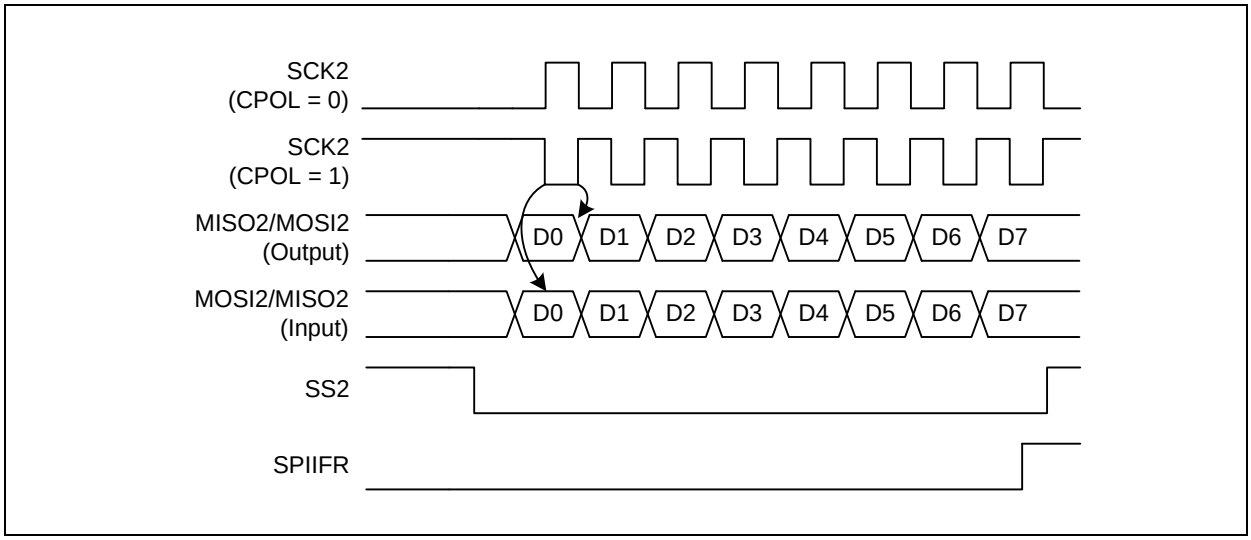


Figure 11.50 SPI 2 Transmit/Receive Timing Diagram at CPHA = 0

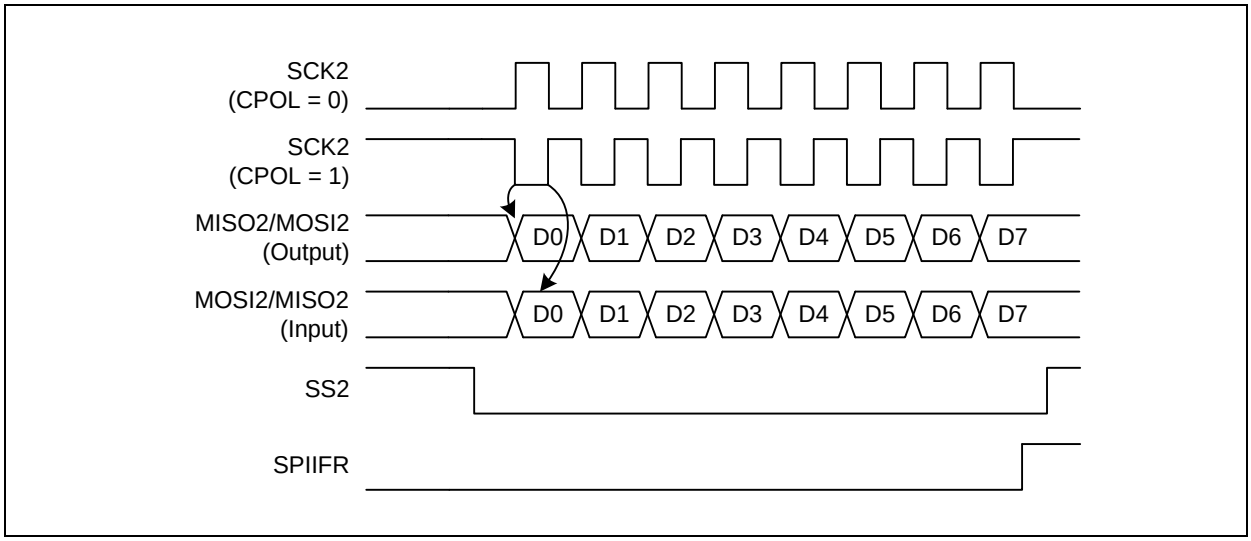


Figure 11.51 SPI 2 Transmit/Receive Timing Diagram at CPHA = 1

11.10.6 Register Map

Table 11-17 SPI 2 Register Map

Name	Address	Dir	Default	Description
SPISR	B7H	R/W	00H	SPI 2 Status Register
SPIDR	B6H	R/W	00H	SPI 2 Data Register
SPICR	B5H	R/W	00H	SPI 2 Control Register

11.10.7 SPI 2 Register Description

The SPI 2 register consists of SPI 2 control register (SPICR), SPI 2 status register (SPISR) and SPI 2 data register (SPIDR)

11.10.8 Register Description for SPI 2

SPIDR (SPI 2 Data Register) : B6H

7	6	5	4	3	2	1	0
SPIDR7	SPIDR6	SPIDR5	SPIDR4	SPIDR3	SPIDR2	SPIDR1	SPIDR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

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

SPISR (SPI 2 Status Register) : B7H

7	6	5	4	3	2	1	0
SPIIFR	WCOL	SS_HIGH	–	FXCH	SSENA	–	–
R/W	R	R/W	–	R/W	R/W	–	–

Initial value : 00H

SPIIFR When SPI 2 Interrupt occurs, this bit becomes '1'. IF SPI 2 interrupt is enable, this bit is auto cleared by INT_ACK signal. And if SPI 2 Interrupt is disable, this bit is cleared when the status register SPISR is read, and then access (read/write) the data register SPIDR

0 SPI 2 Interrupt no generation

1 SPI 2 Interrupt generation

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

0 No collision

1 Collision

SS_HIGH When the SS2 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

FXCH SPI 2 port function exchange control bit.

0 No effect

1 Exchange MOSI2 and MISO2 function

SSENA This bit controls the SS2 pin operation

0 Disable

1 Enable (The P17 should be a normal input)

SPICR (SPI 2 Control Register) : B5H

7	6	5	4	3	2	1	0
SPIEN	FLSB	MS	CPOL	CPHA	DSCR	SCR1	SCR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

SPIEN This bit controls the SPI 2 operation

0 Disable SPI 2 operation

1 Enable SPI 2 operation

FLSB This bit selects the data transmission sequence

0 MSB first

1 LSB first

MS This bit selects whether Master or Slave mode

0 Slave mode

1 Master mode

CPOL This two bits control the serial clock (SCK2) mode.

CPHA Clock polarity(CPOL) bit determine SCK2's value at idle mode.

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

CPOL	CPHA	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)

DSCR
SCR[2:0] These three bits select the SCK2 rate of the device configured as a master. When DSCR bit is written one, SCK2 will be doubled in master mode.

DSCR	SCR1	SCR0	SCK2 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

11.11 12-Bit A/D Converter

11.11.1 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 eight 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[2:0] bits should be set to 'xxx'. 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'.

11.11.2 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 μs. 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 μs = 38.28 μs at 1.5 MHz (12 MHz/8)

NOTE) The A/D converter needs at least 20 μs for conversion time. So you must set the conversion time more than 20 μs.

11.11.3 Block Diagram

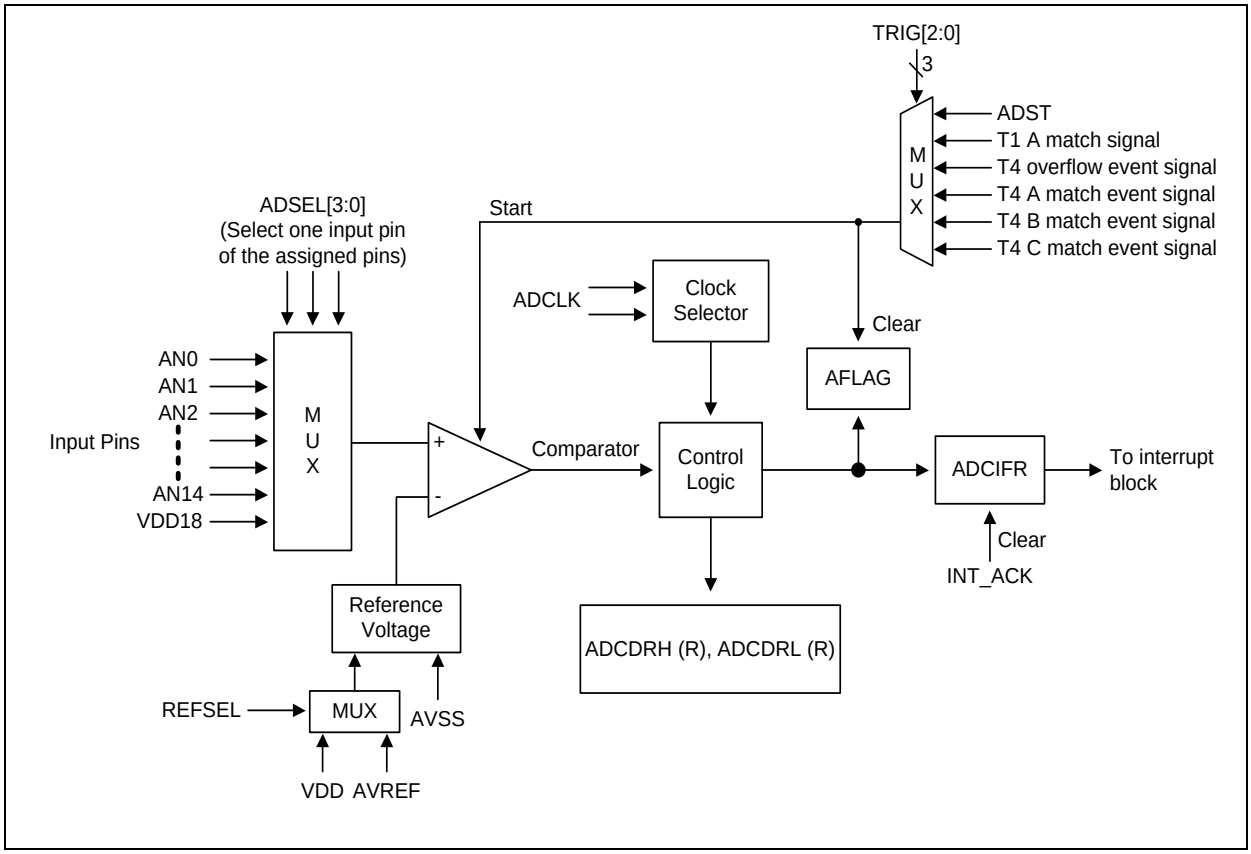


Figure 11.52 12-bit ADC Block Diagram

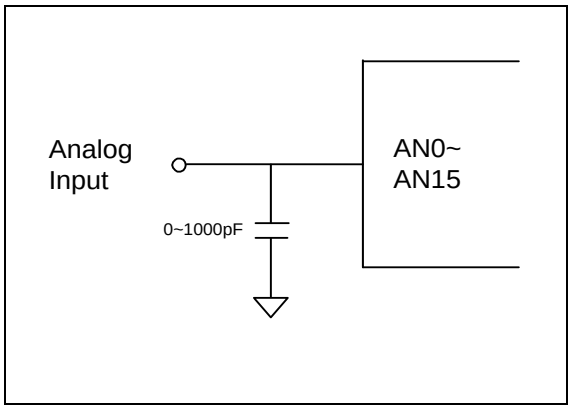


Figure 11.53 A/D Analog Input Pin with Capacitor

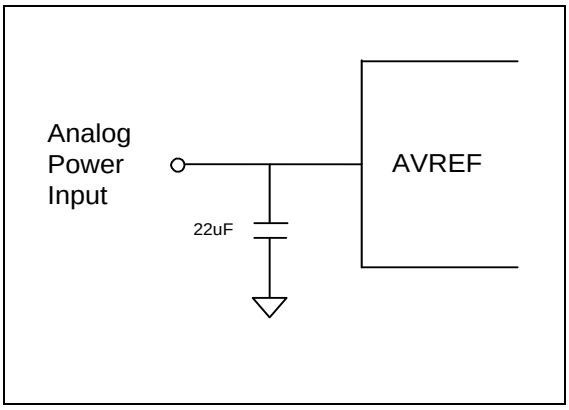


Figure 11.54 A/D Power (AVREF) Pin with Capacitor

11.11.4 ADC Operation

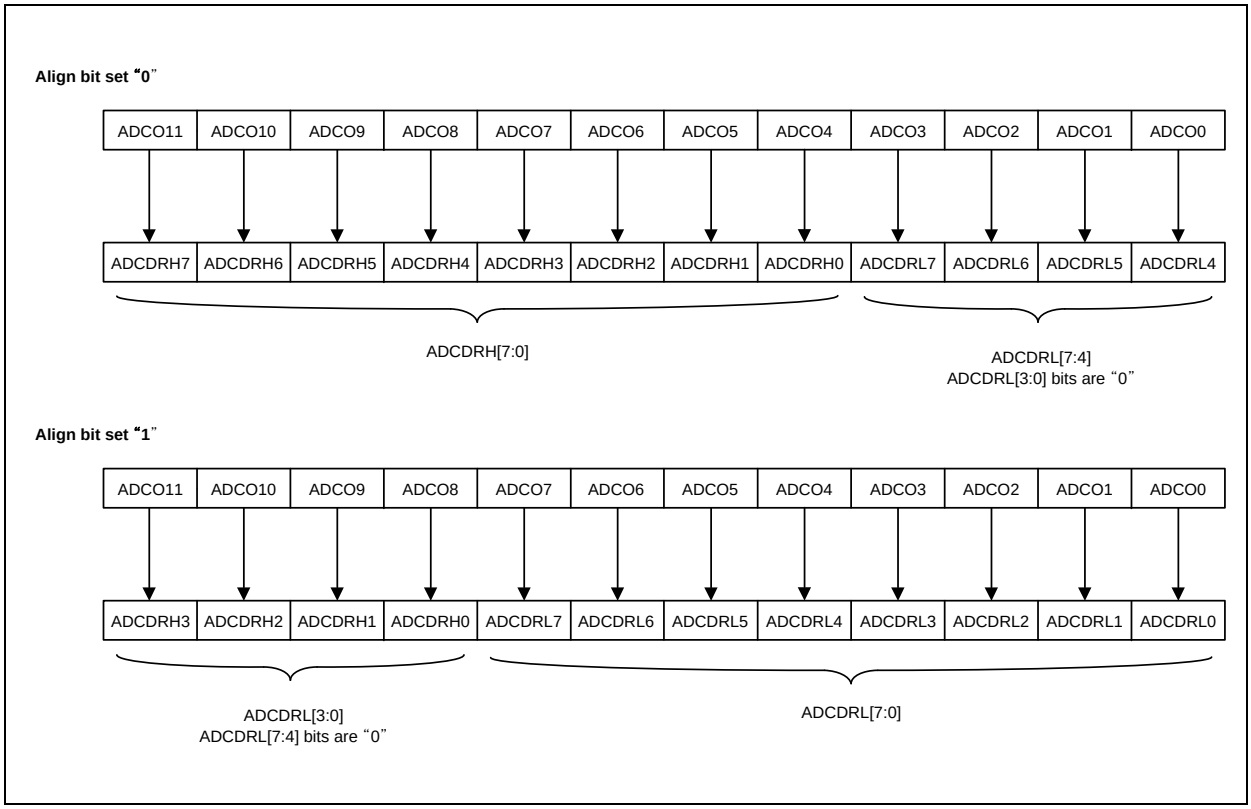


Figure 11.55 ADC Operation for Align Bit

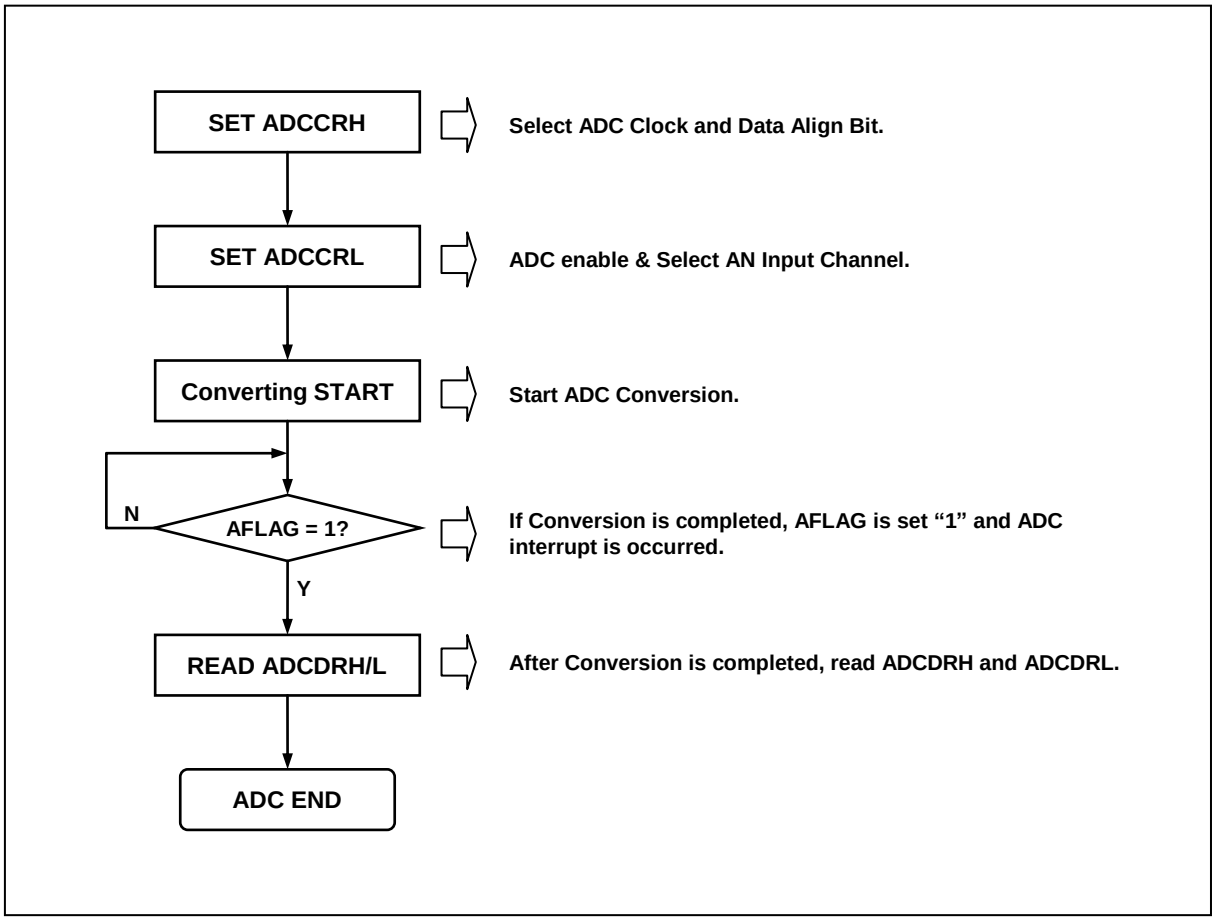


Figure 11.56 A/D Converter Operation Flow

11.11.5 Register Map

Table 11-18 ADC Register Map

Name	Address	Dir	Default	Description
ADCDRH	9FH	R	xxH	A/D Converter Data High Register
ADCDRL	9EH	R	xxH	A/D Converter Data Low Register
ADCCRH	9DH	R/W	00H	A/D Converter Control High Register
ADCCRL	9CH	R/W	00H	A/D Converter Control Low Register

11.11.6 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).

11.11.7 Register Description for ADC

ADCDRH (A/D Converter Data High Register) : 9FH

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) : 9EH

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) : 9DH

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.

0 ADC Interrupt no generation

1 ADC Interrupt generation

TRIG[2:0] A/D Trigger Signal Selection

TRIG2	TRIG1	TRIG0	Description
0	0	0	ADST
0	0	1	Timer 1 A match signal
0	1	0	Timer 4 overflow event signal
0	1	1	Timer 4 A match event signal
1	0	0	Timer 4 B match event signal
1	0	1	Timer 4 C match event signal
Other Values			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) : 9CH

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 stop/start.

0 No effect

1 ADC Conversion Start and auto clear

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

11.12 USI0 (UART + SPI + I2C)

11.12.1 Overview

The USI0 consists of USI0 control register1/2/3/4, USI0 status register 1/2, USI0 baud-rate generation register, USI0 data register, USI0 SDA hold time register, USI0 SCL high period register, USI0 SCL low period register, and USI0 slave address register (USI0CR1, USI0CR2, USI0CR3, USI0CR4, USI0ST1, USI0ST2, USI0BD, USI0DR, USI0SDHR, USI0SCHR, USI0SCLR, USI0SAR).

The operation mode is selected by the operation mode of USI0 selection bits (USI0MS[1:0]).

It has four operating modes:

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

11.12.2 USI0 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 has three main parts of clock generator, Transmitter and receiver. The clock generation logic consists of synchronization logic for external clock inut 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 (USI0DR) and control logic. The receiver supports the same frame formats as the transmitter and can detect frame error, data overrun and parity errors.

11.12.3 USI0 UART Block Diagram

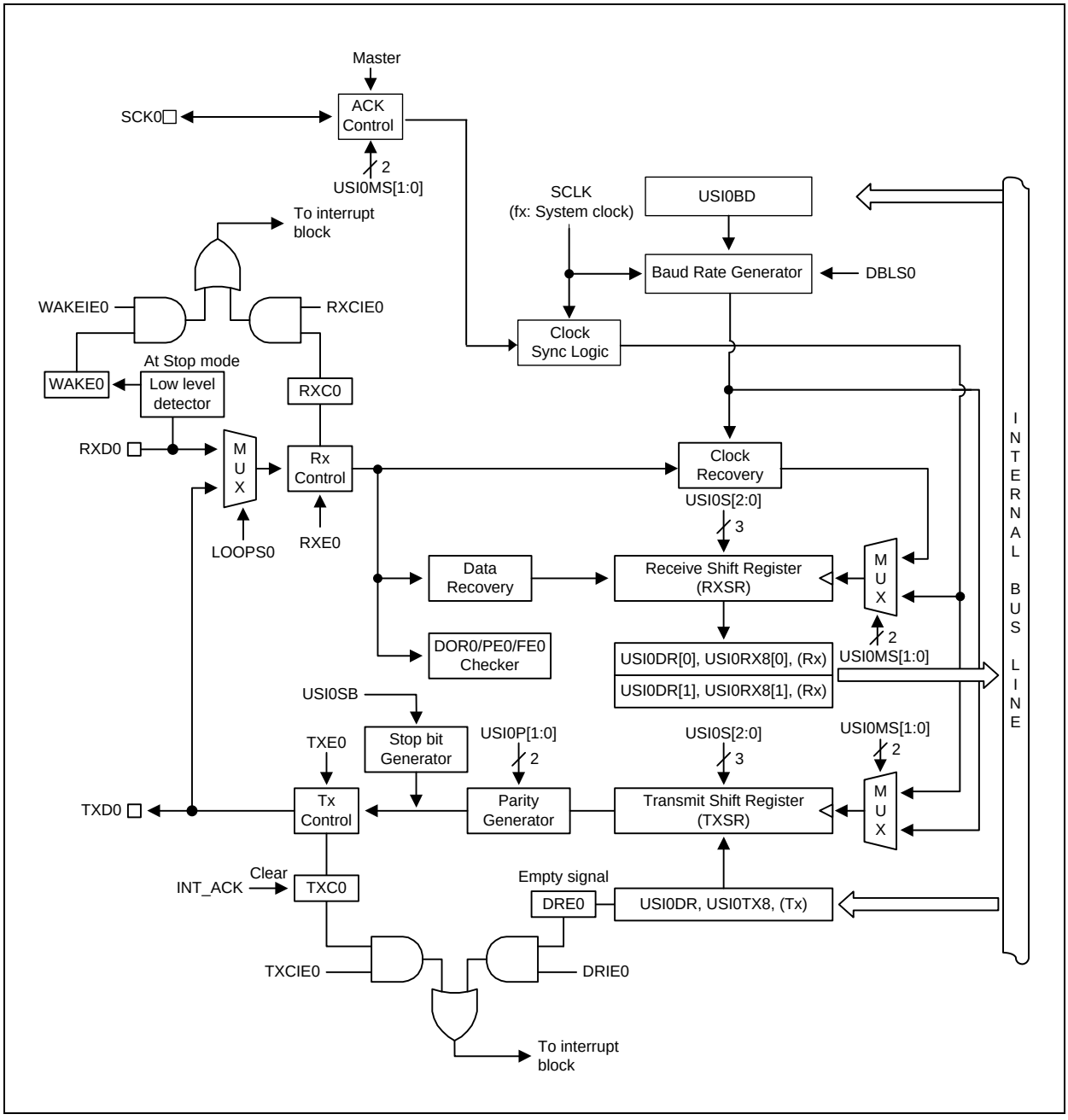


Figure 11.57 USI0 UART Block Diagram

11.12.4 USI0 Clock Generation

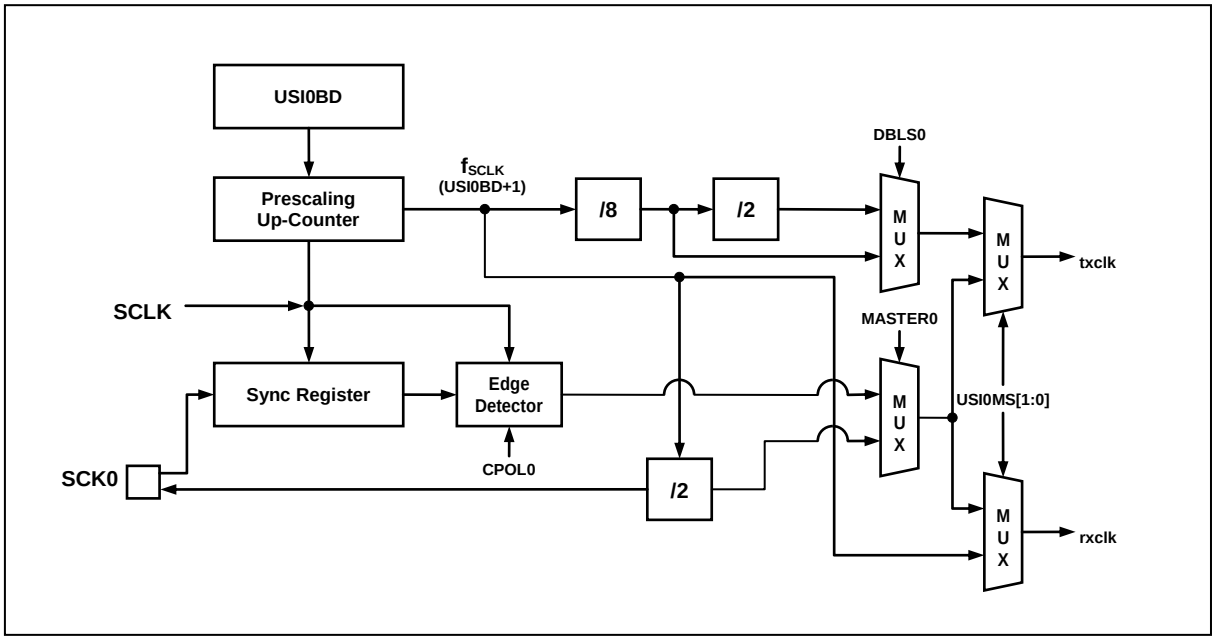


Figure 11.58 Clock Generation Block Diagram (USI0)

The clock generation logic generates the base clock for the transmitter and receiver. The USI0 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 USI0MS[1:0] bits in USI0CR1 register selects asynchronous or synchronous operation. Asynchronous double speed mode is controlled by the DBLS0 bit in the USI0CR2 register. The MASTER0 bit in USI0CR3 register controls whether the clock source is internal (master mode, output pin) or external (slave mode, input pin). The SCK0 pin is active only when the USI0 operates in synchronous or SPI mode.

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

Table 11-19 Equations for Calculating USI0 Baud Rate Register Setting

Operating Mode	Equation for Calculating Baud Rate
Asynchronous Normal Mode (DBLS0=0)	$\text{Baud Rate} = \frac{\text{fx}}{16(\text{USI0BD}+1)}$
Asynchronous Double Speed Mode (DBLS0=1)	$\text{Baud Rate} = \frac{\text{fx}}{8(\text{USI0BD}+1)}$
Synchronous or SPI Master Mode	$\text{Baud Rate} = \frac{\text{fx}}{2(\text{USI0BD}+1)}$

11.12.5 USI0 External Clock (SCK0)

External clocking is used in the synchronous mode of operation.

External clock input from the SCK0 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 SCK0 pin is limited up-to 1MHz.

11.12.6 USI0 Synchronous mode operation

When synchronous or SPI mode is used, the SCK0 pin will be used as either clock input (slave) or clock output (master). Data sampling and transmitter is issued on the different edge of SCK0 clock each other. For example, if data input on RXD0 (MISO0 in SPI mode) pin is sampled on the rising edge of SCK0 clock, data output on TXD0 (MOSI0 in SPI mode) pin is altered on the falling edge.

The CPOL0 bit in USI0CR1 register selects which SCK0 clock edge is used for data sampling and which is used for data change. As shown in the figure below, when CPOL0 is zero, the data will be changed at rising SCK0 edge and sampled at falling SCK0 edge.

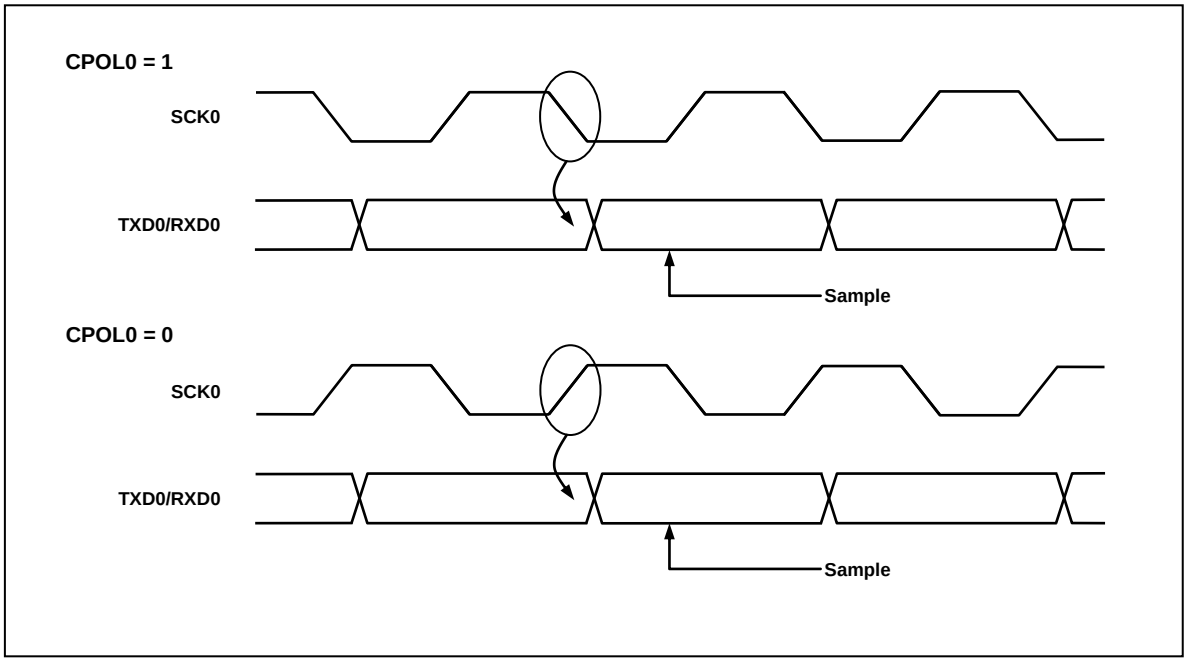


Figure 11.59 Synchronous Mode SCK0 Timing (USI0)

11.12.7 USI0 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.

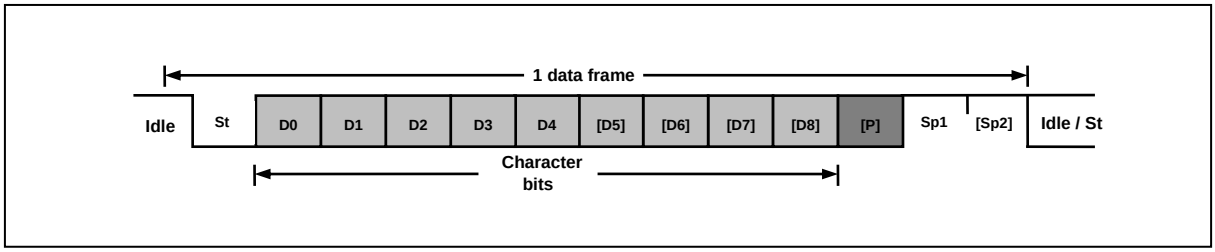


Figure 11.60 Frame Format (USI0)

1 data frame consists of the following bits

- Idle No communication on communication line (TXD0/RXD0)
- 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 USI0S[2:0], USI0PM[1:0] bits in USI0CR1 register and USI0SB bit in USI0CR3 register. The Transmitter and Receiver use the same setting.

11.12.8 USI0 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

11.12.9 USI0 UART Transmitter

The UART transmitter is enabled by setting the TXE0 bit in USI0CR2 register. When the Transmitter is enabled, the TXD0 pin should be set to TXD0 function for the serial output pin of UART by the P4FSR[3:2]. The baud-rate, operation mode and frame format must be setup once before doing any transmission. In synchronous operation mode, the SCK0 pin is used as transmission clock, so it should be selected to do SCK0 function by P4FSR[5:4] .

11.12.9.1 USI0 UART Sending Tx data

A data transmission is initiated by loading the transmit buffer (USI0DR 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 USI0TX8 bit in USI0CR3 register before it is loaded to the transmit buffer (USI0DR register).

11.12.9.2 USI0 UART Transmitter flag and interrupt

The UART transmitter has 2 flags which indicate its state. One is UART data register empty flag (DRE0) and the other is transmit complete flag (TXC0). Both flags can be interrupt sources.

DRE0 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 (DRIE0) bit in USI0CR2 register is set and the global interrupt is enabled, USI0ST1 status register empty interrupt is generated while DRE0 flag is set.

The transmit complete (TXC0) 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 TXC0 flag is automatically cleared when the transmit complete interrupt service routine is executed, or it can be cleared by writing ‘0’ to TXC0 bit in USI0ST1 register.

When the transmit complete interrupt enable (TXCIE0) bit in USI0CR2 register is set and the global interrupt is enabled, UART transmit complete interrupt is generated while TXC0 flag is set.

11.12.9.3 USI0 UART Parity Generator

The parity generator calculates the parity bit for the serial frame data to be sent. When parity bit is enabled (USI0PM1=1), the transmitter control logic inserts the parity bit between the MSB and the first stop bit of the frame to be sent.

11.12.9.4 USI0 UART Disabling Transmitter

Disabling the transmitter by clearing the TXE0 bit will not become effective until ongoing transmission is completed. When the Transmitter is disabled, the TXD0 pin can be used as a normal general purpose I/O (GPIO).

11.12.10 USI0 UART Receiver

The UART receiver is enabled by setting the RXE0 bit in the USI0CR2 register. When the receiver is enabled, the RXD0 pin should be set to RXD0 function for the serial input pin of UART by P4FSR[1:0]. The baud-rate, mode of operation and frame format must be set before serial reception. In synchronous or SPI operation mode the SCK0 pin is used as transfer clock, so it should be selected to do SCK0 function by P4FSR[5:4]. In SPI operation mode the SS0 input pin in slave mode or can be configured as SS0 output pin in master mode. This can be done by setting USI0SSEN bit in USI0CR3 register.

11.12.10.1 USI0 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 RXD0 pin. Each bit after start bit is sampled at pre-defined baud-rate (asynchronous) or sampling edge of SCK0 (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 USI0DR register.

If 9-bit characters are used (USI0S[2:0] = "111"), the ninth bit is stored in the USI0RX8 bit position in the USI0CR3 register. The 9^{th} bit must be read from the USI0RX8 bit before reading the low 8 bits from the USI0DR register. Likewise, the error flags FE0, DOR0, PE0 must be read before reading the data from USI0DR register. It's because the error flags are stored in the same FIFO position of the receive buffer.

11.12.10.2 USI0 UART Receiver Flag and Interrupt

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

The receive complete (RXC0) 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 (RXE0=0), the receiver buffer is flushed and the RXC0 flag is cleared.

When the receive complete interrupt enable (RXCIE0) bit in the USI0CR2 register is set and global interrupt is enabled, the UART receiver complete interrupt is generated while RXC0 flag is set.

The UART receiver has three error flags which are frame error (FE0), data overrun (DOR0) and parity error (PE0). These error flags can be read from the USI0ST1 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 USI0DR register, read the USI0ST1 register first which contains error flags.

The frame error (FE0) flag indicates the state of the first stop bit. The FE0 flag is '0' when the stop bit was correctly detected as "1", and the FE0 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 (DOR0) flag indicates data loss due to a receive buffer full condition. DOR0 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 DOR0 flag is set, all the incoming data are lost. To prevent data loss or clear this flag, read the receive buffer.

The parity error (PE0) flag indicates that the frame in the receive buffer had a parity error when received. If parity check function is not enabled (USI0PM1=0), the PE bit is always read "0".

11.12.10.3 USI0 UART Parity Checker

If parity bit is enabled (USI0PM1=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.

11.12.10.4 USI0 UART Disabling Receiver

In contrast to transmitter, disabling the Receiver by clearing RXE0 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 RXD0 pin can be used as a normal general purpose I/O (GPIO).

11.12.10.5 USI0 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 RXD0 pin.

The data recovery logic samples and low pass filters the incoming bits, and this removes the noise of RXD0 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 aud-rate for double speed mode (DBLS0=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.

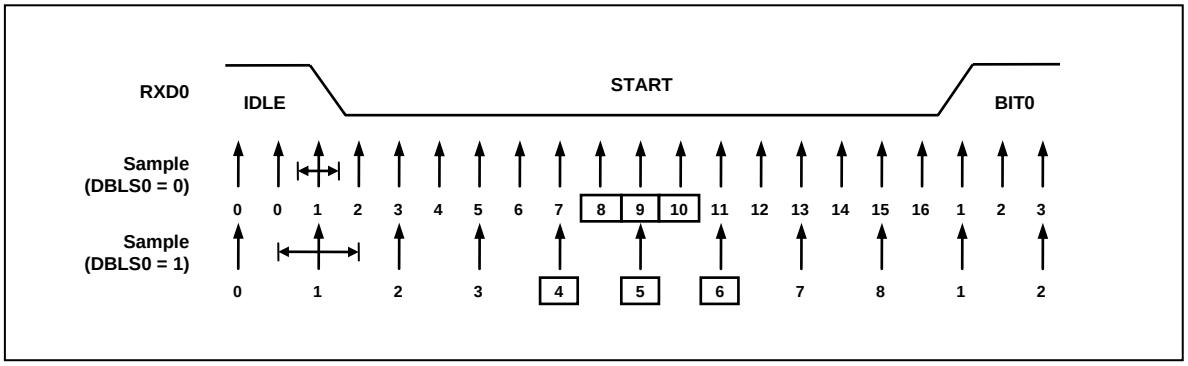


Figure 11.61 Asynchronous Start Bit Sampling (USI0)

When the receiver is enabled (RXE0=1), the clock recovery logic tries to find a high-to-low transition on the RXD0 line, the start bit condition. After detecting high to low transition on RXD0 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.

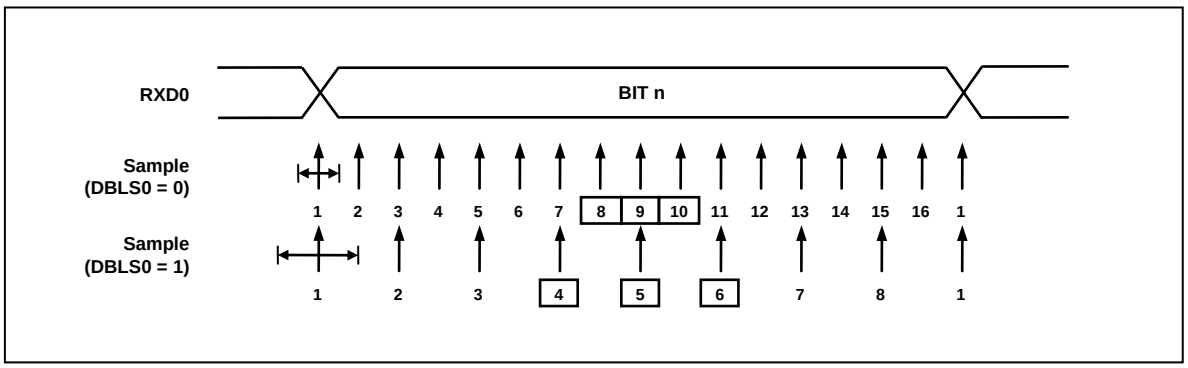


Figure 11.62 Asynchronous Sampling of Data and Parity Bit (USI0)

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 (FE0) flag is set. After deciding whether the first stop bit is valid or not, the Receiver goes to idle state and monitors the RXD0 line to check a valid high to low transition is detected (start bit detection).

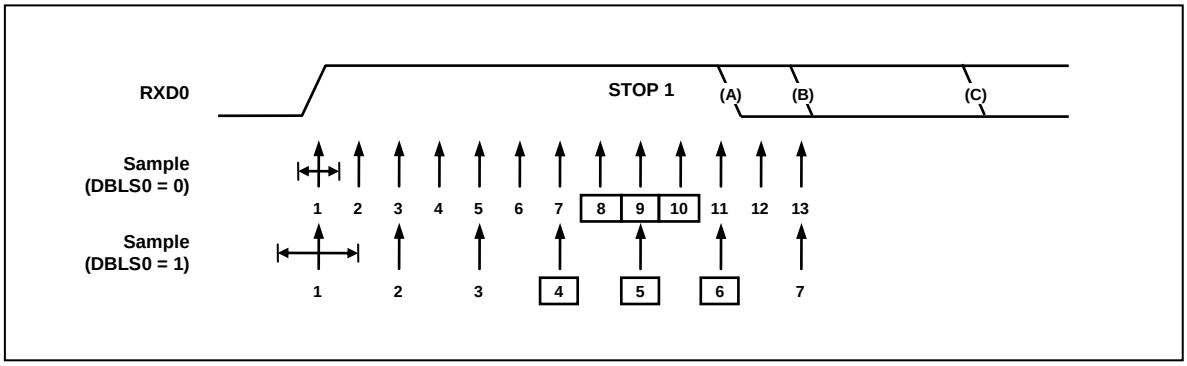


Figure 11.63 Stop Bit Sampling and Next Start Bit Sampling (USI0)

11.12.11 USI0 SPI Mode

The USI0 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 (USI0MS[1:0]=”11”), the slave select (SS0) pin becomes active LOW input in slave mode operation, or can be output in master mode operation if USI0SSEN bit is set to ‘0’.

Note that during SPI mode of operation, the pin RXD0 is renamed as MISO0 and TXD0 is renamed as MOSI0 for compatibility to other SPI devices.

11.12.12 USI0 SPI Clock Formats and Timing

To accommodate a wide variety if synchronus serial peripherals from different manufacturers, the USI0 has a clock polarity bit (CPOL0) and a clock phase control bit (CPHA0) to select one of four clock formats for data transfers. CPOL0 selectively insert an inverter in series with the clock. CPHA0 chooses between two different clock phase relationships between the clock and data. Note that CPHA0 and CPOL0 bits in USI0CR1 register have different meanings according to the USI0MS[1:0] bits which decides the operating mode of USI0.

Table below shows four combinations of CPOL0 and CPHA0 for SPI mode 0, 1, 2, and 3.

Table 11-20 CPOL0 Functionality

SPI Mode	CPOL0	CPHA0	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)

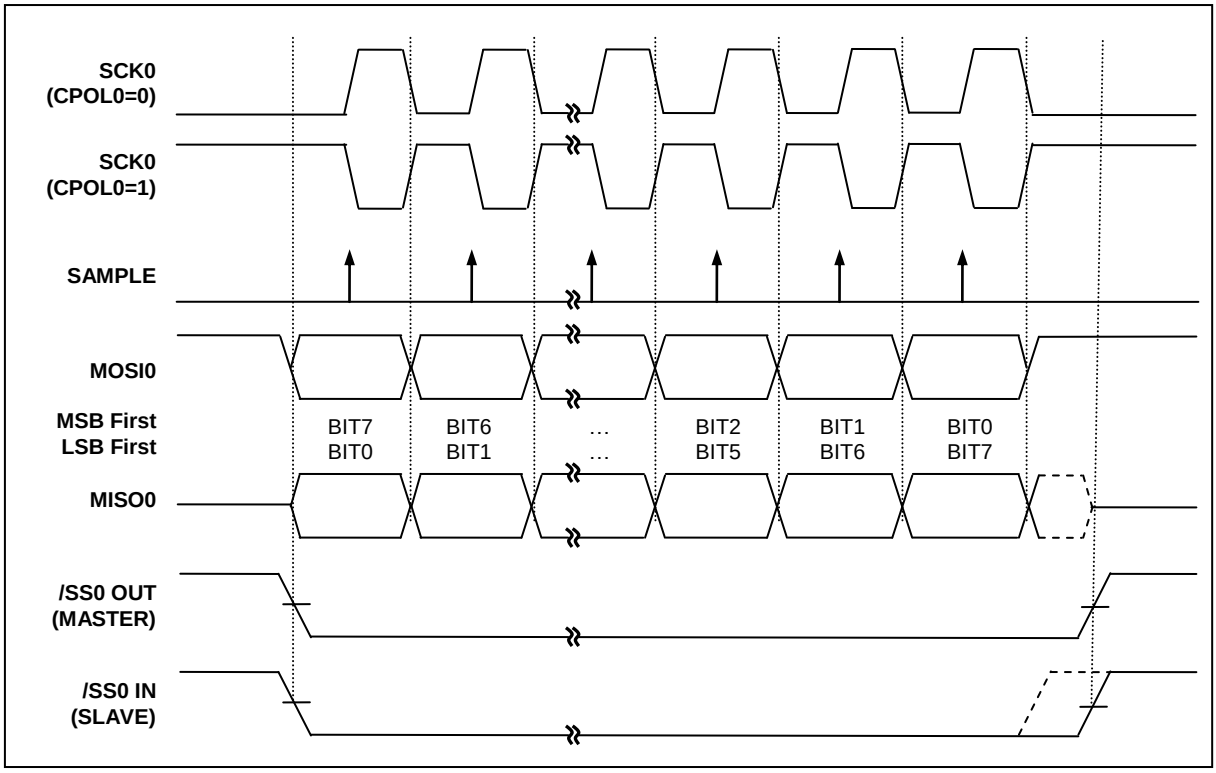


Figure 11.64 USI0 SPI Clock Formats when CPHA0=0

When CPHA0=0, the slave begins to drive its MISO0 output with the first data bit value when SS0 goes to active low. The first SCK0 edge causes both the master and the slave to sample the data bit value on their MISO0 and MOSI0 inputs, respectively. At the second SCK0 edge, the USI0 shifts the second data bit value out to the MOSI0 and MISO0 outputs of the master and slave, respectively. Unlike the case of CPHA0=1, when CPHA0=0, the slave's SS0 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 SS0 input.

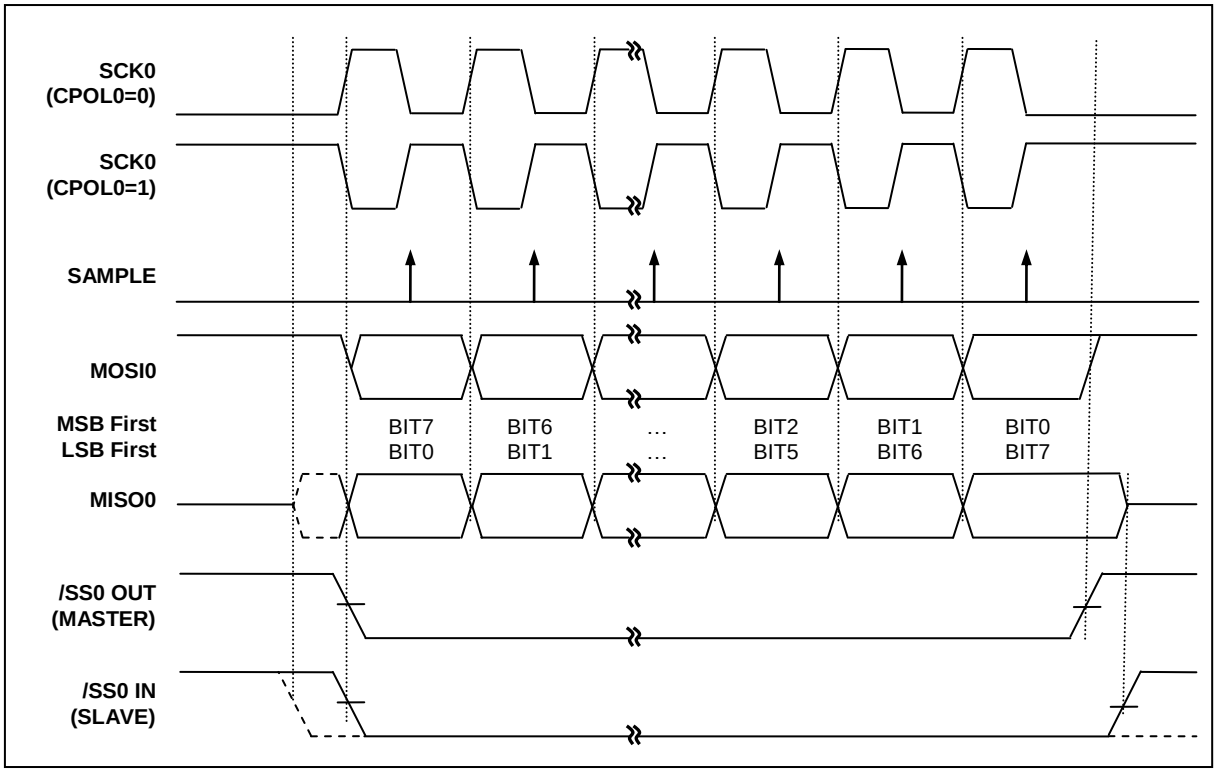


Figure 11.65 USI0 SPI Clock Formats when CPHA0=1

When CPHA0=1, the slave begins to drive its MISO0 output when SS0 goes active low, but the data is not defined until the first SCK0 edge. The first SCK0 edge shifts the first bit of data from the shifter onto the MOSI0 output of the master and the MISO0 output of the slave. The next SCK0 edge causes both the master and slave to sample the data bit value on their MISO0 and MOSI0 inputs, respectively. At the third SCK0 edge, the USI0 shifts the second data bit value out to the MOSI0 and MISO0 output of the master and slave respectively. When CPHA0=1, the slave's SS0 input is not required to go to its inactive high level between transfers.

Because the SPI logic reuses the USI0 resources, SPI mode of operation is similar to that of synchronous or asynchronous operation. An SPI transfer is initiated by checking for the USI0 Data Register Empty flag (DRE0=1) and then writing a byte of data to the USI0DR Register. In master mode of operation, even if transmission is not enabled (TXE0=0), writing data to the USI0DR register is necessary because the clock SCK0 is generated from transmitter block.

11.12.13 USI0 SPI Block Diagram

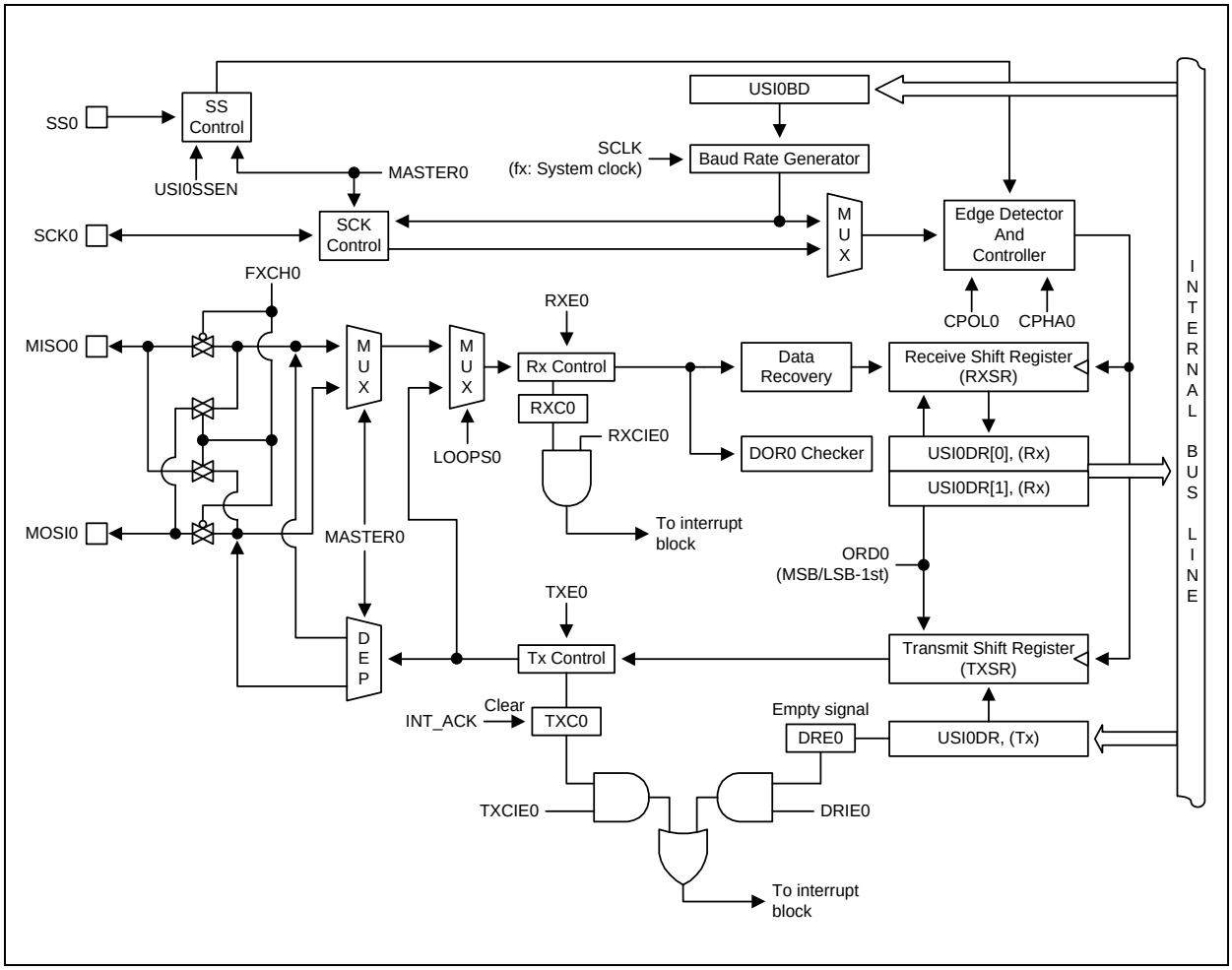


Figure 11.66 USI0 SPI Block Diagram

11.12.14 USI0 I2C Mode

The USI0 can be set to operate in industrial standard serial communicatin protocols mode. The I2C mode uses 2 bus lines serial data line (SDA0) and serial clock line (SCL0) to exchange data. Because both SDA0 and SCL0 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

11.12.15 USI0 I2C Bit Transfer

The data on the SDA0 line must be stable during HIGH period of the clock, SCL0. The HIGH or LOW state of the data line can only change when the clock signal on the SCL0 line is LOW. The exceptions are START(S), repeated START(Sr) and STOP(P) condition where data line changes when clock line is high.

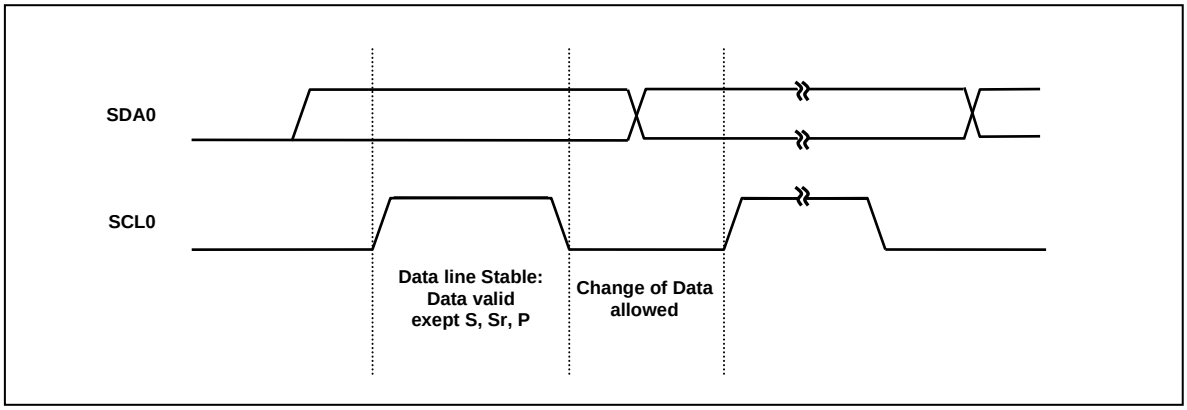


Figure 11.67 Bit Transfer on the I2C-Bus (USI0)

11.12.16 USI0 I2C Start / Repeated Start / Stop

One master can issue a START (S) condition to notice other devices connected to the SCL0, SDA0 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 SDA0 line while SCL0 is high defines a START (S) condition.
A low to high transition on the SDA0 line while SCL0 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.

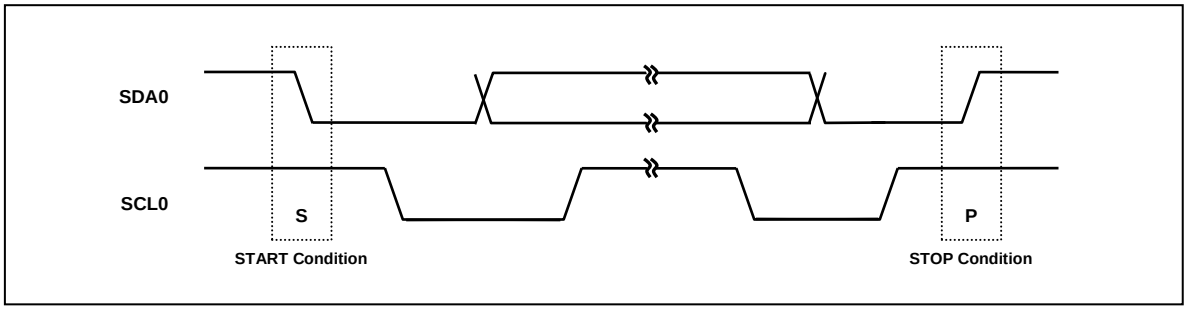


Figure 11.68 START and STOP Condition (USI0)

11.12.17 USI0 I2C Data Transfer

Every byte put on the SDA0 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 SCL0 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 SCL0.

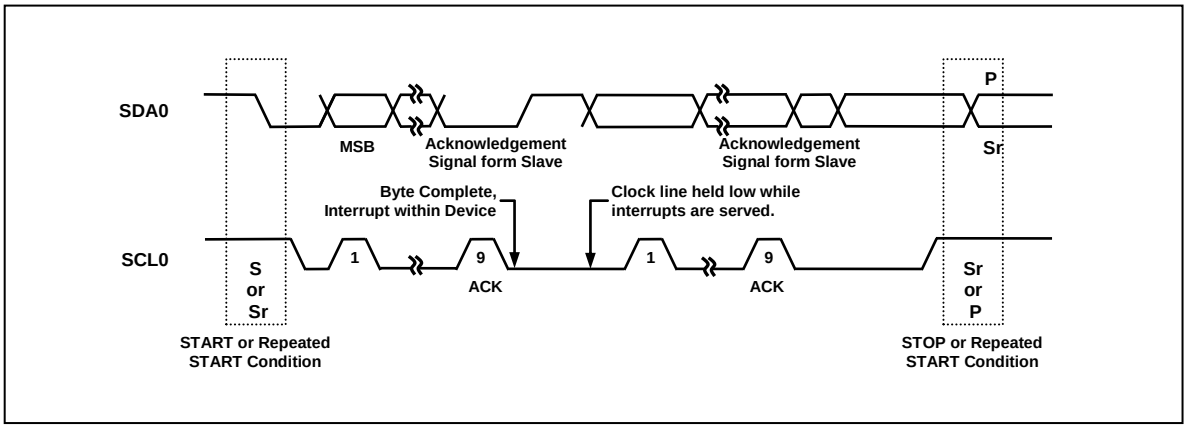


Figure 11.69 Data Transfer on the I2C-Bus (USI0)

11.12.18 USI0 I2C Acknowledge

The acknowledge related clock pulse is generated by the master. The transmitter releases the SDA0 line (HIGH) during the acknowledge clock pulse. The receiver must pull down the SDA0 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 SDA0 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.

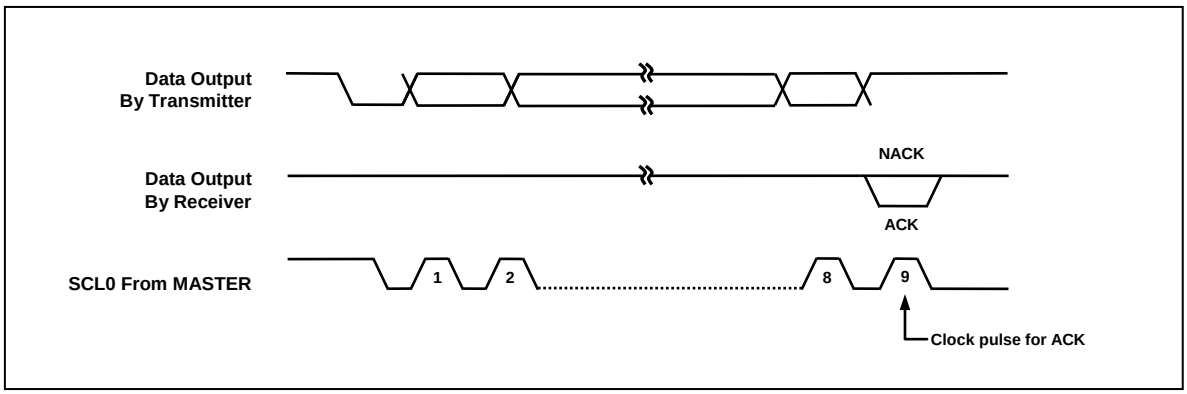


Figure 11.70 Acknowledge on the I2C-Bus (USI0)

11.12.19 USI0 I2C Synchronization / Arbitration

Clock synchronization is performed using the wired-AND connection of I2C interfaces to the SCL0 line. This means that a HIGH to LOW transition on the SCL0 line will cause the devices concerned to start counting off their LOW period and it will hold the SCL0 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 SCL0 line if another clock is still within its LOW period. In this way, a synchronized SCL0 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 SDA0 line, while the SCL0 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.

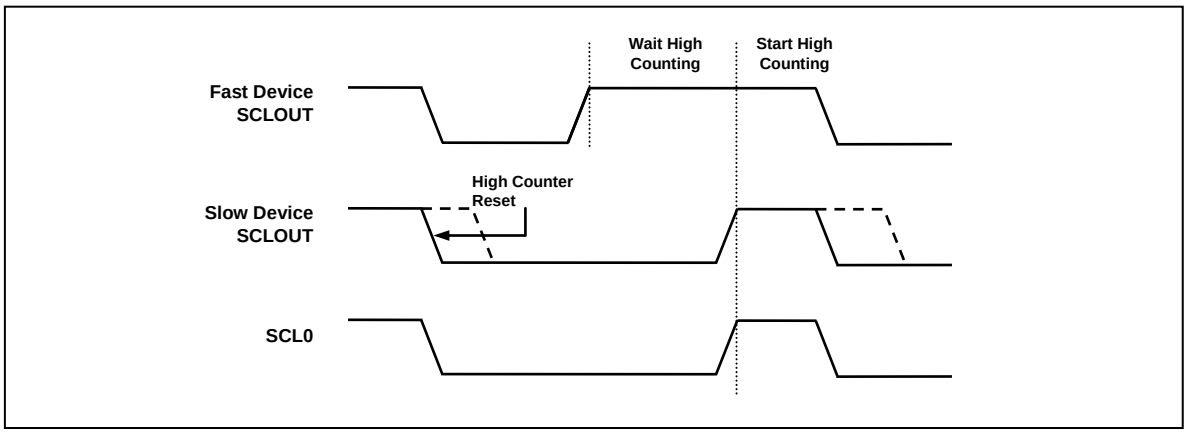


Figure 11.71 Clock Synchronization during Arbitration Procedure (USI0)

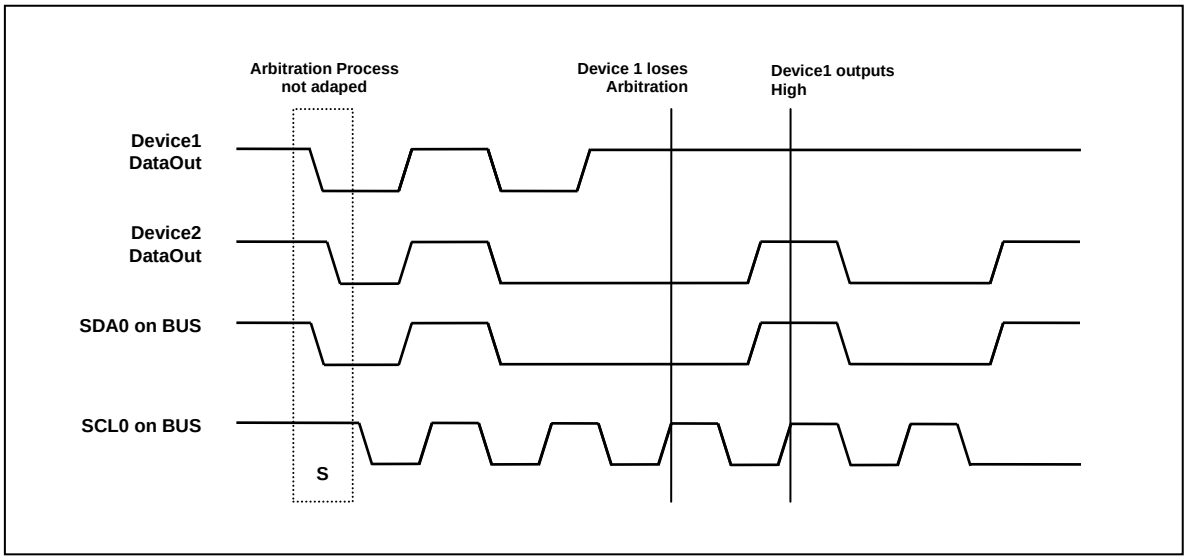


Figure 11.72 Arbitration Procedure of Two Masters (USI0)

11.12.20 USI0 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, IIC0IFR flag in USI0CR4 register is set, it is cleared by writing an any value to USI0ST2. When I2C interrupt occurs, the SCL0 line is hold LOW until writing any value to USI0ST2. When the IIC0IFR flag is set, the USI0ST2 contains a value indicating the current state of the I2C bus. According to the value in USI0ST2, 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.

11.12.20.1 USI0 I2C Master Transmitter

To operate I2C in master transmitter, follow the recommended steps below.

1. Enable I2C by setting USI0MS[1:0] bits in USI0CR1 and USI0EN bit in USI0CR2. This provides main clock to the peripheral.
2. Load SLA0+W into the USI0DR where SLA0 is address of slave device and W is transfer direction from the viewpoint of the master. For master transmitter, W is '0'. Note that USI0DR is used for both address and data.
3. Configure baud rate by writing desired value to both USI0SCLR and USI0SCHR for the Low and High period of SCL0 line.
4. Configure the USI0SDHR to decide when SDA0 changes value from falling edge of SCL0. If SDA0 should change in the middle of SCL0 LOW period, load half the value of USI0SCLR to the USI0SDHR.
5. Set the STARTC0 bit in USI0CR4. This transmits a START condition. And also configure how to handle interrupt and ACK signal. When the STARTC0 bit is set, 8-bit data in USI0DR 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 SCL0. 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 MLOST0 bit in USI0ST2 is set, and I2C waits in idle state or can be operate as an addressed slave. To operate as a slave when the MLOST0 bit in USI0ST2 is set, the ACK0EN bit in USI0CR4 must be set and the received 7-bit address must equal to the USI0SLA[6:0] bits in USI0SAR. In this case I2C operates as a slave transmitter or a slave receiver (go to appropriate section). In this stage, I2C holds the SCL0 LOW. This is because to decide whether I2C continues serial transfer or stops 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 USI0DR.
   2) Master stops data transfer even if it receives ACK signal from slave. In this case, set the STOPC0 bit in USI0CR4.
   3) Master transmits repeated START condition with not checking ACK signal. In this case, load SLA0+R/W into the USI0DR and set STARTC0 bit in USI0CR4.

   After doing one of the actions above, write any arbitrary to USI0ST2 to release SCL0 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 USI0DR 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 SCL0 LOW. When I2C loses bus mastership while transmitting data arbitrating other masters, the MLOST0 bit in USI0ST2 is set. If then, I2C waits in idle state. When the data in USI0DR is transmitted completely, I2C generates TEND0 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 USI0DR.
   2) Master stops data transfer even if it receives ACK signal from slave. In this case, set the STOPC0 bit in USI0CR4.
   3) Master transmits repeated START condition with not checking ACK signal. In this case, load SLA0+R/W into the USI0DR and set the STARTC0 bit in USI0CR4.

   After doing one of the actions above, write any arbitrary to USI0ST2 to release SCL0 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 USI0DR, 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 USI0ST2, write any value to USI0ST2. After this, I2C enters idle state.

The next figure depicts above process for master transmitter operation of I2C.

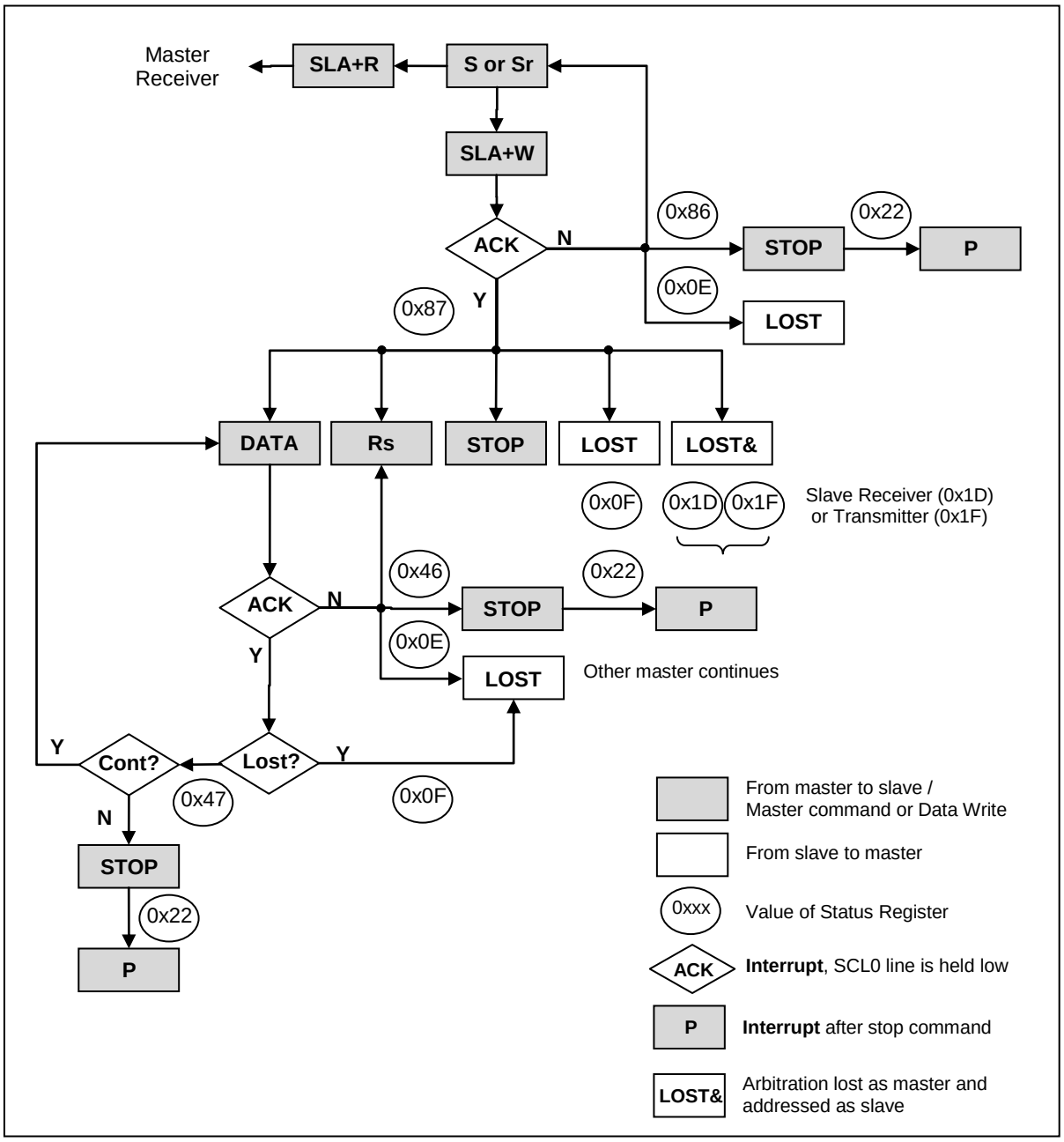


Figure 11.73 Formats and States in the Master Transmitter Mode (USI0)

11.12.20.2 USI0 I2C Master Receiver

To operate I2C in master receiver, follow the recommended steps below.

1. Enable I2C by setting USI0MS[1:0] bits in USI0CR1 and USI0EN bit in USI0CR2. This provides main clock to the peripheral.
2. Load SLA0+R into the USI0DR 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 USI0DR is used for both address and data.
3. Configure baud rate by writing desired value to both USI0SCLR and USI0SCHR for the Low and High period of SCL0 line.
4. Configure the USI0SDHR to decide when SDA0 changes value from falling edge of SCL0. If SDA0 should change in the middle of SCL0 LOW period, load half the value of USI0SCLR to the USI0SDHR.
5. Set the STARTC0 bit in USI0CR4. This transmits a START condition. And also configure how to handle interrupt and ACK signal. When the STARTC0 bit is set, 8-bit data in USI0DR 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 SCL0. 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 MLOST0 bit in USI0ST2 is set, and I2C waits in idle state or can be operate as an addressed slave. To operate as a slave when the MLOST0 bit in USI0ST2 is set, the ACK0EN bit in USI0CR4 must be set and the received 7-bit address must equal to the USI0SLA[6:0] bits in USI0SAR. In this case I2C operates as a slave transmitter or a slave receiver (go to appropriate section). In this stage, I2C holds the SCL0 LOW. This is because to decide whether I2C continues serial transfer or stops 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 ACK0EN bit in USI0CR4 to decide whether I2C ACKnowledges the next data to be received or not.
   2) Master stops data transfer because it receives no ACK signal from slave. In this case, set the STOPC0 bit in USI0CR4.
   3) Master transmits repeated START condition due to no ACK signal from slave. In this case, load SLA0+R/W into the USI0DR and set STARTC0 bit in USI0CR4.

   After doing one of the actions above, write arbitrary value to USI0ST2 to release SCL0 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 USI0DR 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 SCL0 LOW. When 1-Byte of data is received completely, I2C generates TEND0 interrupt.

   I2C0 can choose one of the following cases according to the RXACK0 flag in USI0ST2.

   1) Master continues receiving data from slave. To do this, set ACK0EN bit in USI0CR4 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 ACK0EN bit in USI0CR4.
   3) Because no ACK signal is detected, master terminates data transfer. In this case, set the STOPC0 bit in USI0CR4.
   4) No ACK signal is detected, and master transmits repeated START condition. In this case, load SLA0+R/W into the USI0DR and set the STARTC0 bit in USI0CR4.

   After doing one of the actions above, write arbitrary value to USI0ST2 to release SCL0 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 USI0DR, 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 USI0ST2, write any value to USI0ST2. After this, I2C enters idle state.

The processes described above for master receiver operation of I2C can be depicted as the following figure.

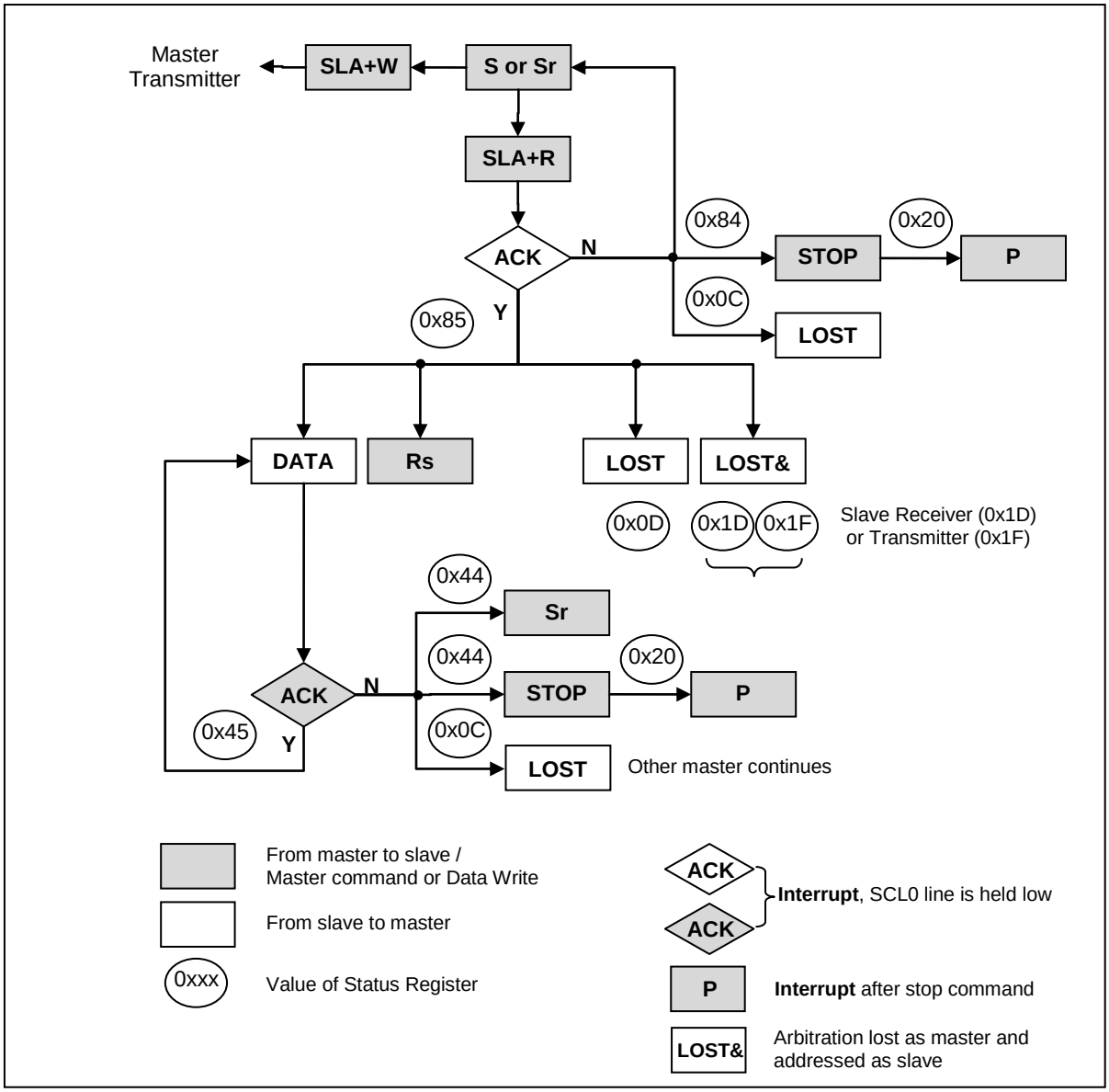


Figure 11.74 Formats and States in the Master Receiver Mode (USI0)

11.12.20.3 USI0 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 SCL0, load value 0x00 into USI0SDHR to make SDA0 change within one system clock period from the falling edge of SCL0. Note that the hold time of SDA0 is calculated by SDAH x period of SCLK where SDAH is multiple of number of SCLK coming from USI0SDHR. When the hold time of SDA0 is longer than the period of SCLK, I2C (slave) cannot transmit serial data properly.
2. Enable I2C by setting USI0MS[1:0] bits in USI0CR1 , IIC0IE bit in USI0CR4 and USI0EN bit in USI0CR2. This provides main clock to the peripheral.
3. When a START condition is detected, I2C receives one byte of data and compares it with USI0SLA[6:0] bits in USI0SAR. If the GCALL0 bit in USI0SAR is enabled, I2C compares the received data with value 0x00, the general call address.
4. If the received address does not equal to USI0SLA[6:0] bits in USI0SAR, I2C enters idle state ie, waits for another START condition. Else if the address equals to USI0SLA[6:0] bits and the ACK0EN bit is enabled, I2C generates SSEL0 interrupt and the SCL0 line is held LOW. Note that even if the address equals to USI0SLA[6:0] bits, when the ACK0EN bit is disabled, I2C enters idle state. When SSEL0 interrupt occurs, load transmit data to USI0DR and write arbitrary value to USI0ST2 to release SCL0 line.
5. 1-Byte of data is being transmitted.
6. In this step, I2C generates TEND0 interrupt and holds the SCL0 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 USI0DR.

   After doing one of the actions above, write arbitrary value to USI0ST2 to release SCL0 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 STOPC0 bit indicates that data transfer between master and slave is over. To clear USI0ST2, write any value to USI0ST2. After this, I2C enters idle state.

The next figure shows flow chart for handling slave transmitter function of I2C.

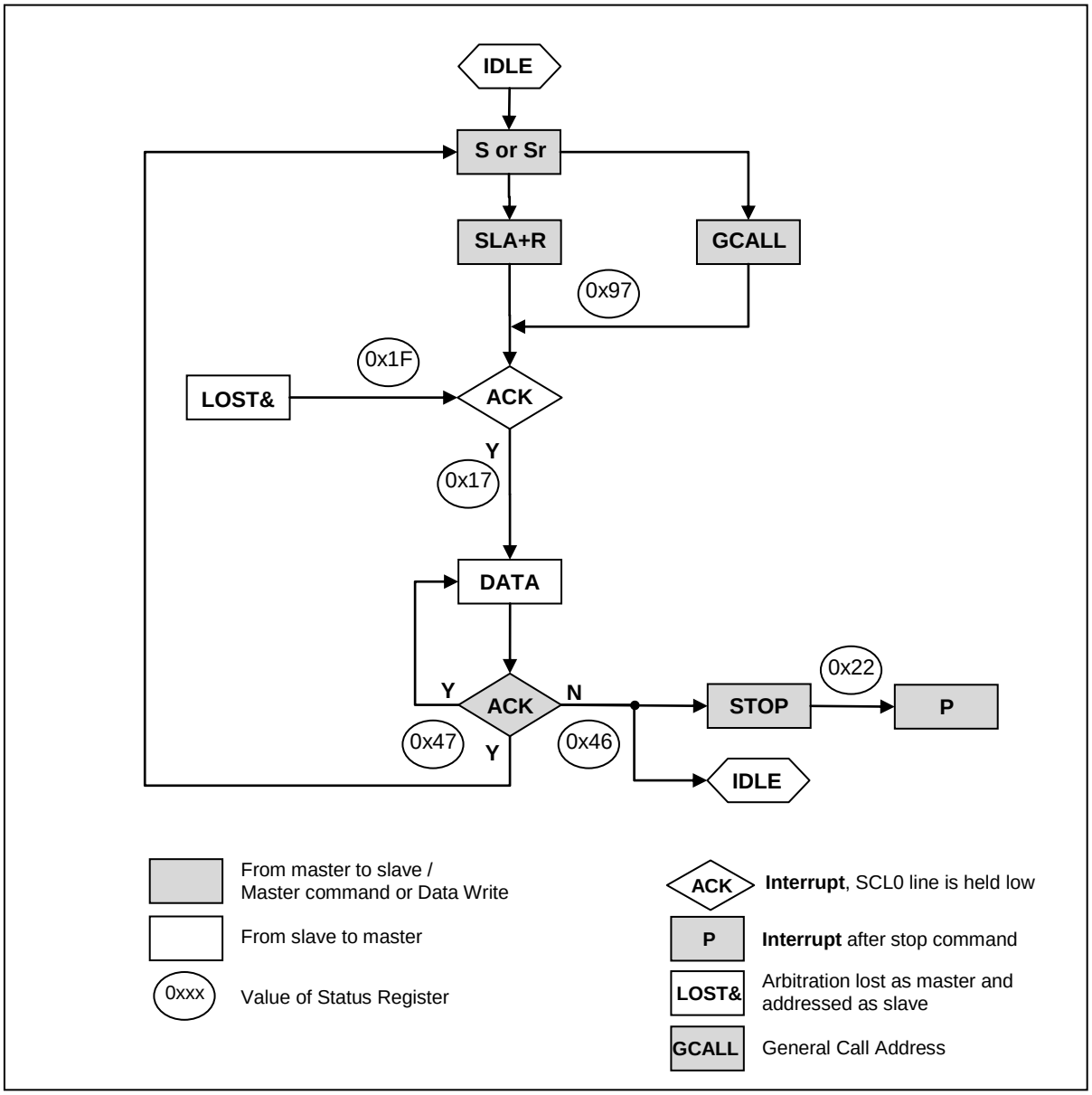


Figure 11.75 Formats and States in the Slave Transmitter Mode (USI0)

11.12.20.4 USI0 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 SCL0, load value 0x00 into USI0SDHR to make SDA0 change within one system clock period from the falling edge of SCL0. Note that the hold time of SDA0 is calculated by SDAH x period of SCLK where SDAH is multiple of number of SCLK coming from USI0SDHR. When the hold time of SDA0 is longer than the period of SCLK, I2C (slave) cannot transmit serial data properly.
2. Enable I2C by setting USI0MS[1:0] bits in USI0CR1, IIC0IE bit in USI0CR4 and USI0EN bit in USI0CR2. This provides main clock to the peripheral.
3. When a START condition is detected, I2C receives one byte of data and compares it with USI0SLA[6:0] bits in USI0SAR. If the GCALL0 bit in USI0SAR is enabled, I2C0 compares the received data with value 0x00, the general call address.
4. If the received address does not equal to SLA0bits in USI0SAR, I2C enters idle state ie, waits for another START condition. Else if the address equals to SLA0 bits and the ACK0EN bit is enabled, I2C generates SSEL0 interrupt and the SCL0 line is held LOW. Note that even if the address equals to SLA0 bits, when the ACK0EN bit is disabled, I2C enters idle state. When SSEL0 interrupt occurs and I2C is ready to receive data, write arbitrary value to USI0ST2 to release SCL0 line.
5. 1-Byte of data is being received.
6. In this step, I2C generates TEND0 interrupt and holds the SCL0 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 (ACK0EN=0) and I2C waits STOP or repeated START condition.
   2) ACK signal is detected (ACK0EN=1) and I2C can continue to receive data from master.

   After doing one of the actions above, write arbitrary value to USI0ST2 to release SCL0 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 STOPC0 bit indicates that data transfer between master and slave is over. To clear USI0ST2, write any value to USI0ST2. After this, I2C enters idle state.

The process can be depicted as following figure when I2C operates in slave receiver mode.

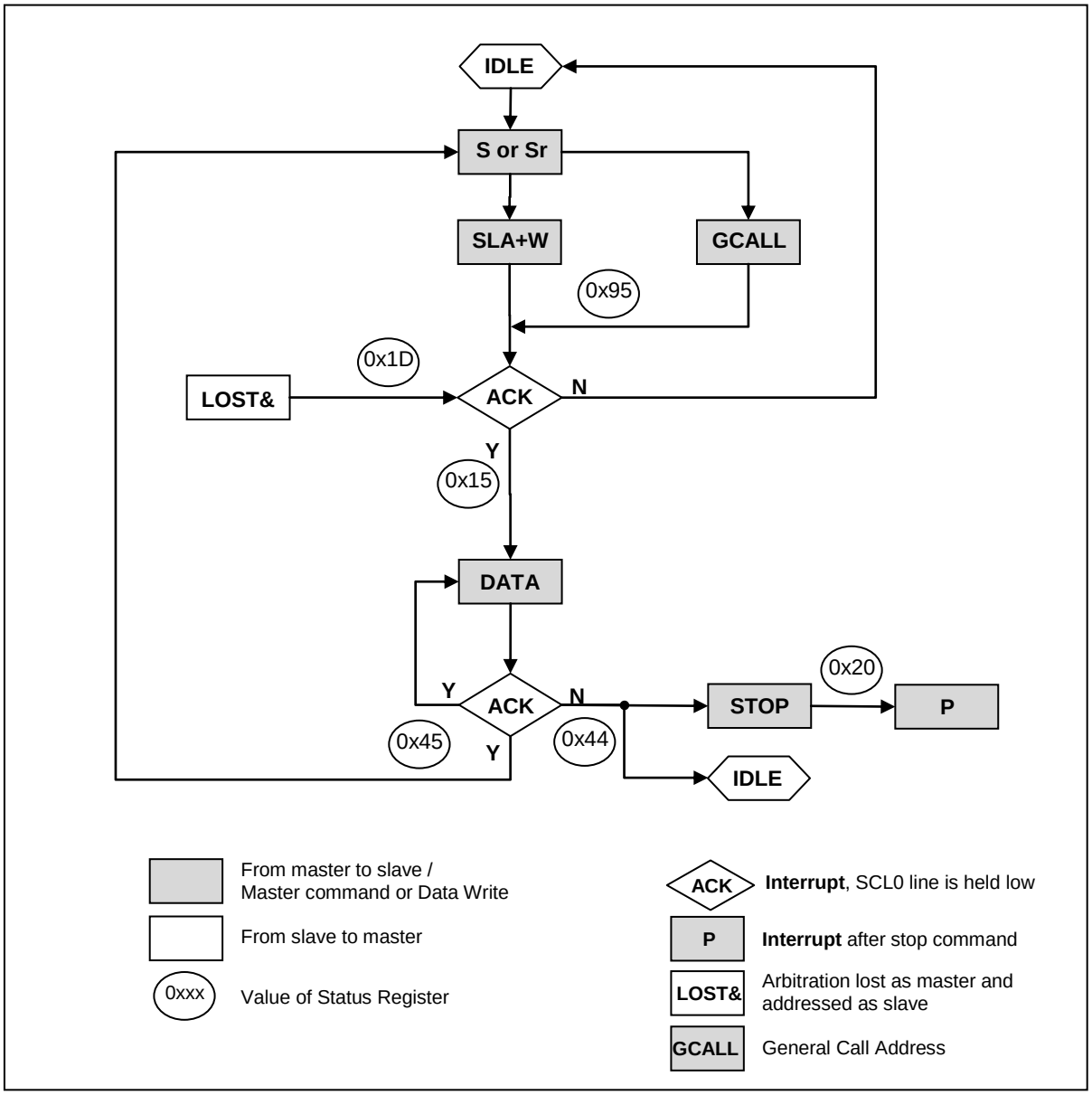
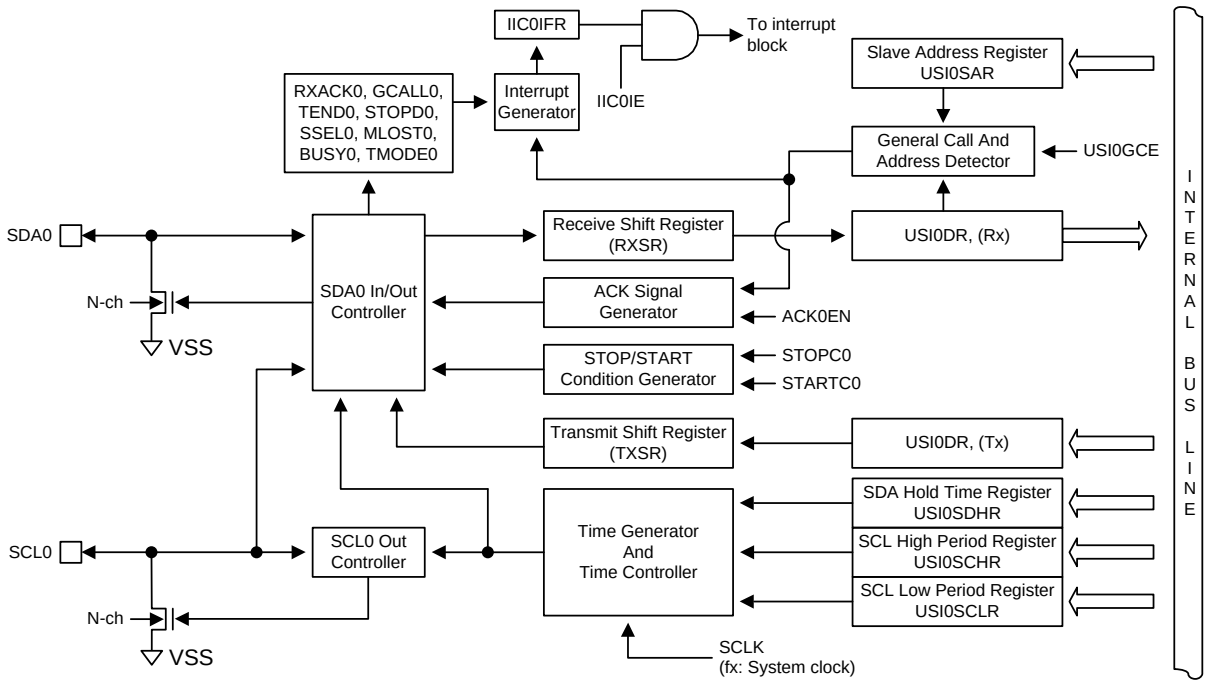


Figure 11.76 Formats and States in the Slave Receiver Mode (USI0)

11.12.21 USI0 I2C Block Diagram



NOTE) When the USI0 block is an I2C mode and the corresponding port is an sub-function for SCL0/SDA0 pin, The SCL0/SDA0 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 11.77 USI0 I2C Block Diagram

11.12.22 Register Map

Table 11-21 USI0 Register Map

Name	Address	Dir	Default	Description
USI0BD	E3H	R/W	FFH	USI0 Baud Rate Generation Register
USI0DR	E5H	R/W	00H	USI0 Data Register
USI0SDHR	E4H	R/W	01H	USI0 SDA Hold Time Register
USI0SCHR	E7H	R/W	3FH	USI0 SCL High Period Register
USI0SCLR	E6H	R/W	3FH	USI0 SCL Low Period Register
USI0SAR	DDH	R/W	00H	USI0 Slave Address Register
USI0CR1	D9H	R/W	00H	USI0 Control Register 1
USI0CR2	DAH	R/W	00H	USI0 Control Register 2
USI0CR3	DBH	R/W	00H	USI0 Control Register 3
USI0CR4	DCH	R/W	00H	USI0 Control Register 4
USI0ST1	E1H	R/W	80H	USI0 Status Register 1
USI0ST2	E2H	R	00H	USI0 Status Register 2

11.12.23 USI0 Register Description

USI0 module consists of USI0 baud rate generation register (USI0BD), USI0 data register (USI0DR), USI0 SDA hold time register (USI0SDHR), USI0 SCL high period register (USI0SCHR), USI0 SCL low period Register (USI0SCLR), USI0 slave address register (USI0SAR), USI0 control register 1/2/3/4 (USI0CR1/2/3/4), USI0 status register 1/2 (USI0ST1/2).

11.12.24 Register Description for USI0

USI0BD (USI0 Baud- Rate Generation Register: For UART and SPI mode) : E3H

7	6	5	4	3	2	1	0
USI0BD7	USI0BD6	USI0BD5	USI0BD4	USI0BD3	USI0BD2	USI0BD1	USI0BD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FFH

USI0BD[7:0] The value in this register is used to generate internal baud rate in asynchronous mode or to generate SCK0 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 USI0SAR register, USI0BD register is used for slave address register when the USI0 I2C mode.

USI0DR (USI0 Data Register: For UART, SPI, and I2C mode) : E5H

7	6	5	4	3	2	1	0
USI0DR7	USI0DR6	USI0DR5	USI0DR4	USI0DR3	USI0DR2	USI0DR1	USI0DR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

USI0DR[7:0] The USI0 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 USI0DR register. Reading the USI0DR register returns the contents of the receive buffer.

Write to this register only when the DRE0 flag is set. In SPI master mode, the SCK clock is generated when data are written to this register.

USI0SDHR (USI0 SDA Hold Time Register: For I2C mode) : E4H

7	6	5	4	3	2	1	0
USI0SDHR7	USI0SDHR6	USI0SDHR5	USI0SDHR4	USI0SDHR3	USI0SDHR2	USI0SDHR1	USI0SDHR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

USI0SDHR[7:0] The register is used to control SDA0 output timing from the falling edge of SCI in I2C mode.

NOTE) That SDA0 is changed after t_{SCLK} X USI0SDHR, in master SDA 0 change in the middle of SCL0.

In slave mode, configure this register regarding the frequency of SCL0 from master.

The SDA0 is changed after tsclk X (USI0SDHR+2) in master mode. So, to insure operation in slave mode, the value

t_{SCLK} X (USI0SDHR +1) must be smaller than the period of SCL.

USI0SCHR (USI0 SCL High Period Register: For I2C mode) : E7H

7	6	5	4	3	2	1	0
USI0SCHR7	USI0SCHR6	USI0SCHR5	USI0SCHR4	USI0SCHR3	USI0SCHR2	USI0SCHR1	USI0SCHR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

USI0SCHR[7:0] This register defines the high period of SCL0 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 USI0SCHR) +2 where

t_{SCLK} is the period of SCLK.

So, the operating frequency of I2C master mode is calculated by the following equation.

$$f_{I2C} = \frac{1}{t_{SCLK} \text{ X } (4 \text{ X } (USI0SCLR + USI0SCHR + 4))}$$

USI0SCLR (USI0 SCL Low Period Register: For I2C mode) : E6H

7	6	5	4	3	2	1	0
USI0SCLR7	USI0SCLR6	USI0SCLR5	USI0SCLR4	USI0SCLR3	USI0SCLR2	USI0SCLR1	USI0SCLR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

USI0SCLR[7:0] This register defines the high period of SCL0 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 USI0SCLR) +2 where t_{SCLK} is the period of SCLK.

USI0SAR (USI0 Slave Address Register: For I2C mode) : DDH

7	6	5	4	3	2	1	0
USI0SLA6	USI0SLA5	USI0SLA4	USI0SLA3	USI0SLA2	USI0SLA1	USI0SLA0	USI0GCE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

USI0SLA[6:0] These bits configure the slave address of I2C when it operaties in I2C slave mode.

UPM[1:0] 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

USI0CR1 (USI0 Control Register 1: For UART, SPI, and I2C mode) : D9H

7	6	5	4	3	2	1	0
USI0MS1	USI0MS0	USI0PM1	USI0PM0	USI0S2	USI0S1 ORD0	USI0S0 CPHA0	CPOL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

USI0MS[1:0] Selects operation mode of USI0

USI0MS1	USI0MS0	Operation mode
0	0	Asynchronous Mode (UART)
0	1	Synchronous Mode
1	0	I2C mode
1	1	SPI mode

USI0PM[1:0] Selects parity generation and check methods (only UART mode)

USI0PM1	USI0PM0	Parity
0	0	No Parity
0	1	Reserved
1	0	Even Parity
1	1	Odd Parity

USI0S[2:0] When in asynchronous or synchronous mode of operation, selects the length of data bits in frame

USI0S2	USI0S1	USI0S0	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

ORD0 This bit in the same bit position with USI0S1. 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

CPOL0 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

CPHA0 This bit is in the same bit position with USI0S0. This bit determines if data are sampled on the leading or trailing edge of SCK0 (only SPI mode).

CPOL0	CPHA0	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)

USI0CR2 (USI0 Control Register 2: For UART, SPI, and I2C mode) : DAH

7	6	5	4	3	2	1	0
DRIE0	TXCIE0	RXCIE0	WAKEIE0	TXE0	RXE0	USI0EN	DBLS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

DRIE0 Interrupt enable bit for data register empty (only UART and SPI mode).

0 Interrupt from DRE0 is inhibited (use polling)

1 When DRE0 is set, request an interrupt

TXCIE0 Interrupt enable bit for transmit complete (only UART and SPI mode).

0 Interrupt from TXC0 is inhibited (use polling)

1 When TXC0 is set, request an interrupt

RXCIE0 Interrupt enable bit for receive complete (only UART and SPI mode).

0 Interrupt from RXC0 is inhibited (use polling)

1 When RXC0 is set, request an interrupt

WAKEIE0 Interrupt enable bit for asynchronous wake in STOP mode. When device is in stop mode, if RXD0 goes to low level an interrupt can be requested to wake-up system. (only UART mode). At that time the DRIE0 bit and USI0ST1 register value should be set to '0b' and "00H", respectively.

0 Interrupt from Wake is inhibited

1 When WAKE0 is set, request an interrupt

TXE0 Enables the transmitter unit (only UART and SPI mode).

0 Transmitter is disabled

1 Transmitter is enabled

RXE0 Enables the receiver unit (only UART and SPI mode).

0 Receiver is disabled

1 Receiver is enabled

USI0EN Activate USI0 function block by supplying.

0 USI0 is disabled

1 USI0 is enabled

DBLS0 This bit selects receiver sampling rate (only UART).

0 Normal asynchronous operation

1 Double Speed asynchronous operation

USI0CR3 (USI0 Control Register 3: For UART, SPI, and I2C mode) : DBH

7	6	5	4	3	2	1	0
MASTER0	LOOPS0	DISSCK0	USI0SSEN	FXCH0	USI0SB	USI0TX8	USI0RX8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

MASTER0 Selects master or slave in SPI and synchronous mode operation and controls the direction of SCK0 pin

0 Slave mode operation (External clock for SCK0).

1 Master mode operation(Internal clock for SCK0).

LOOPS0 Controls the loop back mode of USI0 for test mode (only UART and SPI mode)

0 Normal operation

1 Loop Back mode

DISSCK0 In synchronous mode of operation, selects the waveform of SCK0 output

0 ACK is free-running while UART is enabled in synchronous master mode

1 ACK is active while any frame is on transferring

USI0SSEN This bit controls the SS0 pin operation (only SPI mode)

0 Disable

1 Enable (The SS0 pin should be a normal input)

FXCH0 SPI port function exchange control bit (only SPI mode)

0 No effect

1 Exchange MOSI0 and MISO0 function

USI0SB Selects the length of stop bit in asynchronous or synchronous mode of operation.

0 1 Stop Bit

1 2 Stop Bit

USI0TX8 The ninth bit of data frame in asynchronous or synchronous mode of operation. Write this bit first before loading the USI0DR register

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

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

USI0RX8 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’

USI0CR4 (USI0 Control Register 4: For I2C mode) : DCH

7	6	5	4	3	2	1	0
IIC0IFR	–	TXDLYENB0	IIC0IE	ACK0EN	IMASTER0	STOPC0	STARTC0
R	–	R/W	R/W	R/W	R	R/W	R/W

Initial value : 00H

IIC0IFR 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 USI0ST2.

0 I2C interrupt no generation

1 I2C interrupt generation

TXDLYENB0 USI0SDHR register control bit

0 Enable USI0SDHR register

1 Disable USI0SDHR register

IIC0IE Interrupt Enable bit for I2C mode

0 Interrupt from I2C is inhibited (use polling)

1 Enable interrupt for I2C

ACK0EN Controls ACK signal Generation at ninth SCL0 period.

0 No ACK signal is generated (SDA0 =1)

1 ACK signal is generated (SDA0 =0)

NOTES) ACK signal is output (SDA =0) for the following 3 cases.

1. When received address packet equals to USI0SLA bits in USI0SAR.

2. When received address packet equals to value 0x00 with GCALL0 enabled.

3. When I2C operates as a receiver (master or slave)

IMASTER0 Represent operating mode of I2C

0 I2C is in slave mode

1 I2C is in master mode

STOPC0 When I2C is master, STOP condition generation

0 No effect

1 STOP condition is to be generated

STARTC0 When I2C is master, START condition generation

0 No effect

1 START or repeated START condition is to be generated

USI0ST1 (USI0 Status Register 1: For UART and SPI mode) : E1H

7	6	5	4	3	2	1	0
DRE0	TXC0	RXC0	WAKE0	USI0RST	DOR0	FE0	PE0
R/W	R/W	R	R/W	R/W	R	R/W	R/W

Initial value : 80H

DRE0 The DRE0 flag indicates if the transmit buffer (USI0DR) is ready to receive new data. If DRE0 is '1', the buffer is empty and ready to be written. This flag can generate a DRE0 interrupt.

0 Transmit buffer is not empty.

1 Transmit buffer is empty.

TXC0 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 TXC0 interrupt is executed. This flag can generate a TXC0 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.

RXC0 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 RXC0 flag can be used to generate a RXC0 interrupt. This bit is automatically cleared.

0 There is no data unread in the receive buffer

1 There are more than 1 data in the receive buffer

WAKE0 This flag is set when the RXD0 pin is detected low while the CPU is in STOP mode. This flag can be used to generate a WAKE0 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

USI0RST This is an internal reset and only has effect on USI0. Writing '1' to this bit initializes the internal logic of USI0 and this bit is automatically cleared to '0'.

0 No operation

1 Reset USI0

DOR0 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

FE0 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

PE0 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

USI0ST2 (USI0 Status Register 2: For I2C mode) : E2H

7	6	5	4	3	2	1	0
GCALL0	TEND0	STOPD0	SSEL0	MLOST0	BUSY0	TMODE0	RXACK0
R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

GCALL0(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)

TEND0(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

STOPD0(NOTE) This bit is set when a STOP condition is detected.

0 No STOP condition is detected

1 STOP condition is detected

SSEL0(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

MLOST0(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

BUSY0 This bit reflects bus status.

0 I2C bus is idle, so a master can issue a START condition

1 I2C bus is busy

TMODE0 This bit is used to indicate whether I2C is transmitter or receiver.

0 I2C is a receiver

1 I2C is a transmitter

RXACK0 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 SCL0 line is hold LOW. To release SCL0, write rbitrary value to USI0ST2. When USI0ST2 is written, the TEND0, STOPD0, SSEL0, MLOST0, and RXACK0 bits are cleared.

11.13 USI1 (UART + SPI + I2C)

11.13.1 Overview

The USI1 consists of USI1 control register1/2/3/4, USI1 status register 1/2, USI1 baud-rate generation register, USI1 data register, USI1 SDA hold time register, USI1 SCL high period register, USI1 SCL low period register, and USI1 slave address register (USI1CR1, USI1CR2, USI1CR3, USI1CR4, USI1ST1, USI1ST2, USI1BD, USI1DR, USI1SDHR, USI1SCHR, USI1SCLR, USI1SAR).

The operation mode is selected by the operation mode of USI1 selection bits (USI1MS[1:0]).

It has four operating modes:

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

11.13.2 USI1 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

USI1 has three main parts of clock generator, Transmitter and receiver. The clock generation logic consists of synchronization logic for external clock inut 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 (USI1DR) and control logic. The receiver supports the same frame formats as the transmitter and can detect frame error, data overrun and parity errors.

11.13.3 USI1 UART Block Diagram

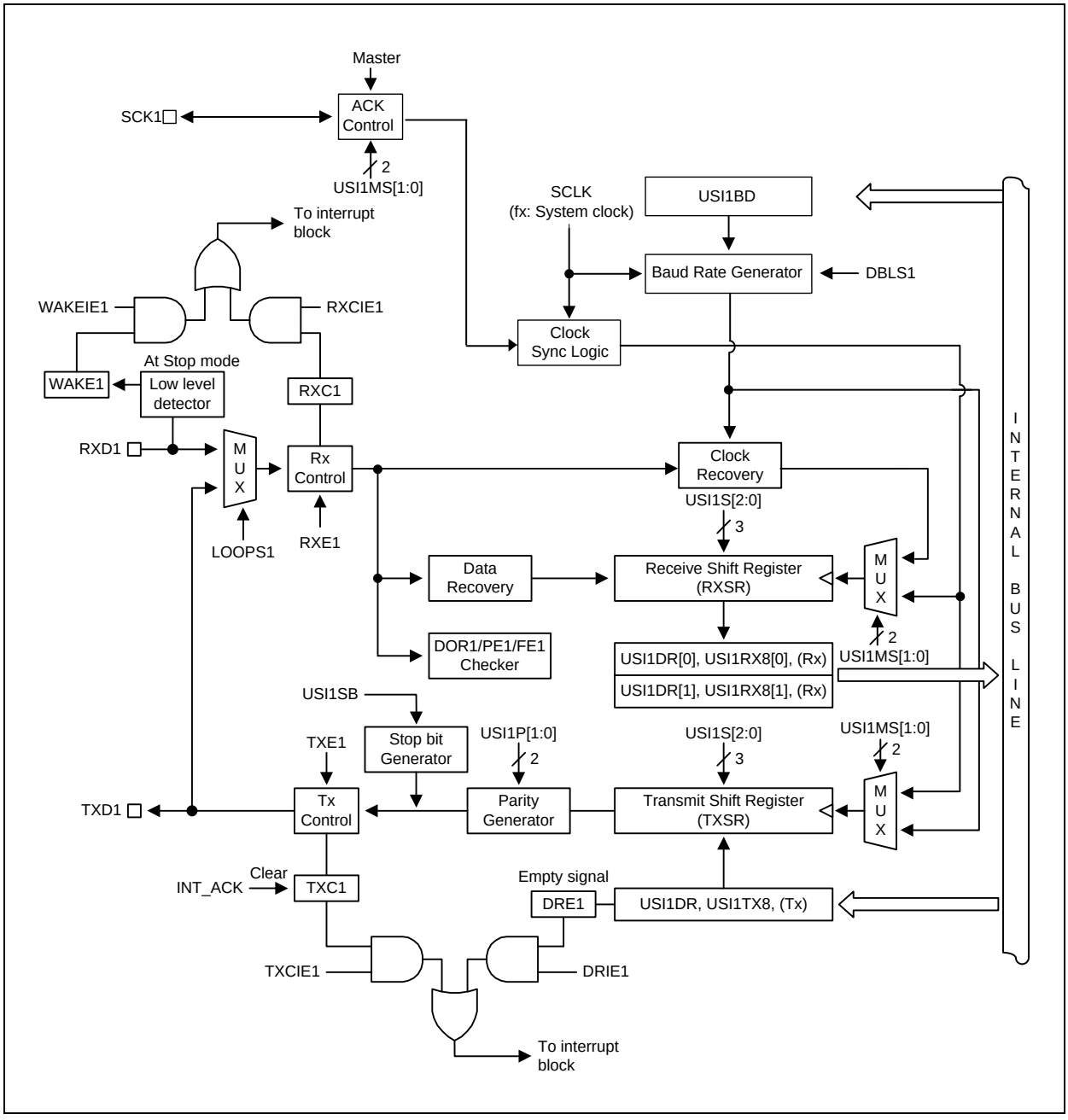


Figure 11.78 USI1 UART Block Diagram

11.13.4 USI1 Clock Generation

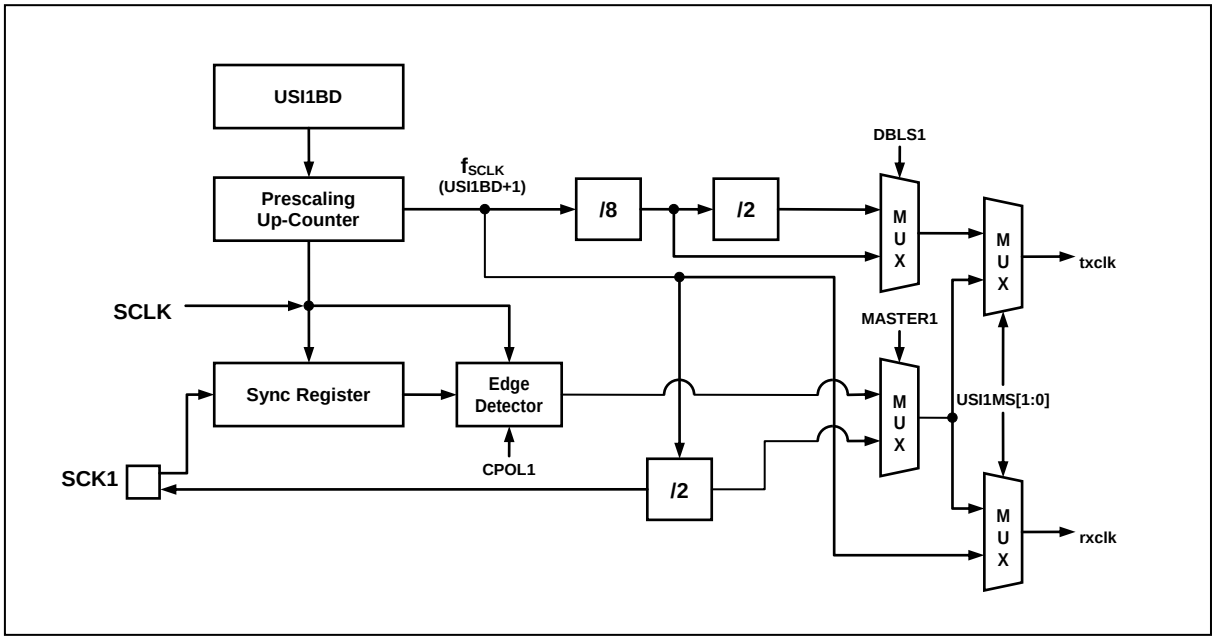


Figure 11.79 Clock Generation Block Diagram (USI1)

The clock generation logic generates the base clock for the transmitter and receiver. The USI1 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 USI1MS[1:0] bits in USI1CR1 register selects asynchronous or synchronous operation. Asynchronous double speed mode is controlled by the DBLS1 bit in the USI1CR2 register. The MASTER1 bit in USI1CR3 register controls whether the clock source is internal (master mode, output pin) or external (slave mode, input pin). The SCK1 pin is active only when the USI1 operates in synchronous or SPI mode.

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

Table 11-22 Equations for Calculating USI1 Baud Rate Register Setting

Operating Mode	Equation for Calculating Baud Rate
Asynchronous Normal Mode (DBLS1=0)	$\text{Baud Rate} = \frac{\text{fx}}{16(\text{USI1BD}+1)}$
Asynchronous Double Speed Mode (DBLS1=1)	$\text{Baud Rate} = \frac{\text{fx}}{8(\text{USI1BD}+1)}$
Synchronous or SPI Master Mode	$\text{Baud Rate} = \frac{\text{fx}}{2(\text{USI1BD}+1)}$

11.13.5 USI1 External Clock (SCK1)

External clocking is used in the synchronous mode of operation.

External clock input from the SCK1 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 SCK1 pin is limited up-to 1MHz.

11.13.6 USI1 Synchronous mode operation

When synchronous or SPI mode is used, the SCK1 pin will be used as either clock input (slave) or clock output (master). Data sampling and transmitter is issued on the different edge of SCK1 clock each other. For example, if data input on RXD1 (MISO1 in SPI mode) pin is sampled on the rising edge of SCK1 clock, data output on TXD1 (MOSI1 in SPI mode) pin is altered on the falling edge.

The CPOL1 bit in USI1CR1 register selects which SCK1 clock edge is used for data sampling and which is used for data change. As shown in the figure below, when CPOL1 is zero, the data will be changed at rising SCK1 edge and sampled at falling SCK1 edge.

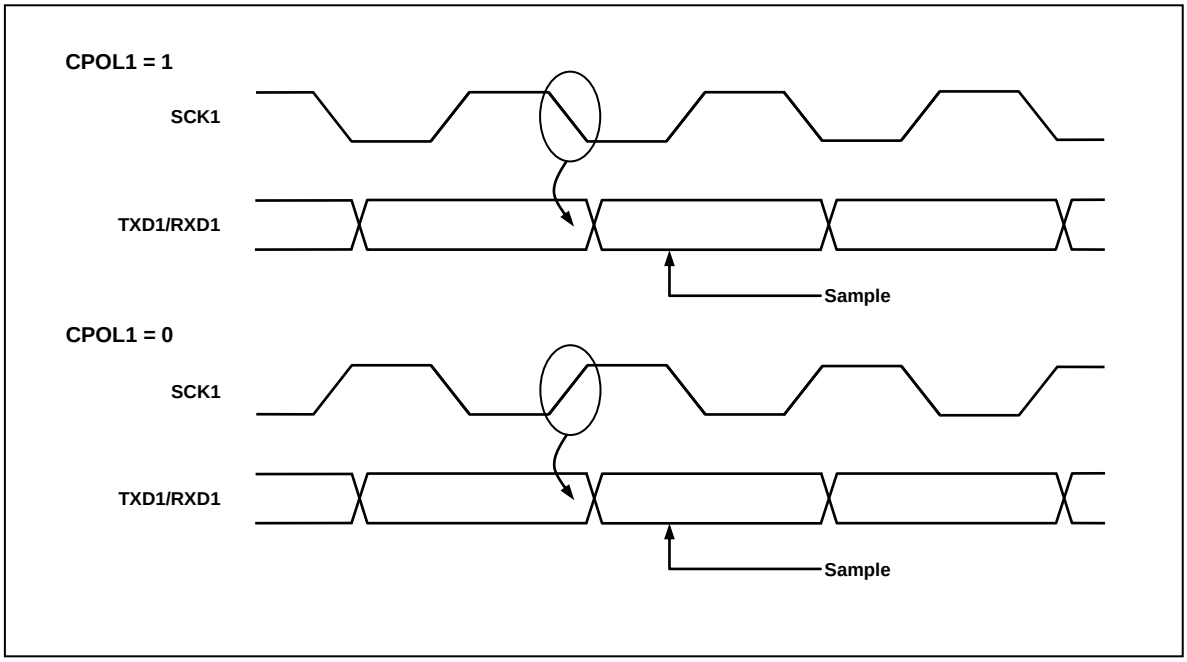


Figure 11.80 Synchronous Mode SCK1 Timing (USI1)

11.13.7 USI1 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.

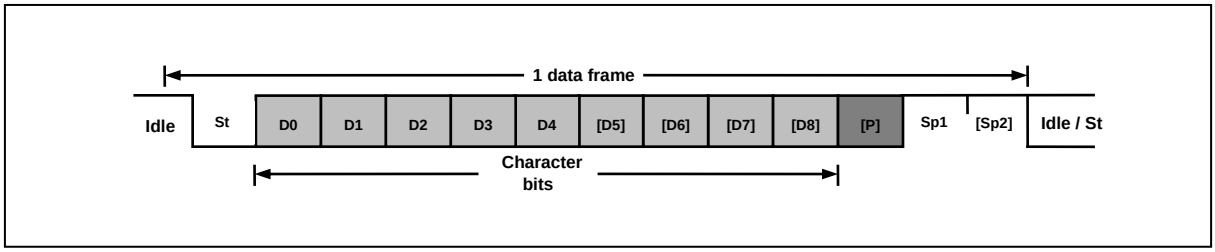


Figure 11.81 Frame Format (USI1)

1 data frame consists of the following bits

- Idle No communication on communication line (TXD0/RXD0)
- 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 USI1S[2:0], USI1PM[1:0] bits in USI1CR1 register and USI1SB bit in USI1CR3 register. The Transmitter and Receiver use the same setting.

11.13.8 USI1 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

11.13.9 USI1 UART Transmitter

The UART transmitter is enabled by setting the TXE1 bit in USI1CR2 register. When the Transmitter is enabled, the TXD1 pin should be set to TXD1 function for the serial output pin of UART by the P2FSR[1:0]. The baud-rate, operation mode and frame format must be setup once before doing any transmission. In synchronous operation mode, the SCK1 pin is used as transmission clock, so it should be selected to do SCK1 function by P2FSR[3:2] .

11.13.9.1 USI1 UART Sending Tx data

A data transmission is initiated by loading the transmit buffer (USI1DR 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 USI1TX8 bit in USI1CR3 register before it is loaded to the transmit buffer (USI1DR register).

11.13.9.2 USI1 UART Transmitter flag and interrupt

The UART transmitter has 2 flags which indicate its state. One is UART data register empty flag (DRE1) and the other is transmit complete flag (TXC1). Both flags can be interrupt sources.

DRE1 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 (DRIE1) bit in USI1CR2 register is set and the global interrupt is enabled, USI1ST1 status register empty interrupt is generated while DRE1 flag is set.

The transmit complete (TXC1) 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 TXC1 flag is automatically cleared when the transmit complete interrupt service routine is executed, or it can be cleared by writing ‘0’ to TXC1 bit in USI1ST1 register.

When the transmit complete interrupt enable (TXCIE1) bit in USI1CR2 register is set and the global interrupt is enabled, UART transmit complete interrupt is generated while TXC1 flag is set.

11.13.9.3 USI1 UART Parity Generator

The parity generator calculates the parity bit for the serial frame data to be sent. When parity bit is enabled (USI1PM1=1), the transmitter control logic inserts the parity bit between the MSB and the first stop bit of the frame to be sent.

11.13.9.4 USI1 UART Disabling Transmitter

Disabling the transmitter by clearing the TXE1 bit will not become effective until ongoing transmission is completed. When the Transmitter is disabled, the TXD1 pin can be used as a normal general purpose I/O (GPIO).

11.13.10 USI1 UART Receiver

The UART receiver is enabled by setting the RXE1 bit in the USI1CR2 register. When the receiver is enabled, the RXD1 pin should be set to RXD1 function for the serial input pin of UART by P1FSR[1:0]. The baud-rate, mode of operation and frame format must be set before serial reception. In synchronous or SPI operation mode the SCK1 pin is used as transfer clock, so it should be selected to do SCK1 function by P2FSR[3:2]. In SPI operation mode the SS1 input pin in slave mode or can be configured as SS1 output pin in master mode. This can be done by setting USI1SSEN bit in USI1CR3 register.

11.13.10.1 USI1 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 RXD1 pin. Each bit after start bit is sampled at pre-defined baud-rate (asynchronous) or sampling edge of SCK1 (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 USI1DR register.

If 9-bit characters are used (USI1S[2:0] = "111"), the ninth bit is stored in the USI1RX8 bit position in the USI1CR3 register. The 9^{th} bit must be read from the USI1RX8 bit before reading the low 8 bits from the USI1DR register. Likewise, the error flags FE1, DOR1, PE1 must be read before reading the data from USI1DR register. It's because the error flags are stored in the same FIFO position of the receive buffer.

11.13.10.2 USI1 UART Receiver Flag and Interrupt

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

The receive complete (RXC1) 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 (RXE1=1), the receiver buffer is flushed and the RXC1 flag is cleared.

When the receive complete interrupt enable (RXCIE1) bit in the USI1CR2 register is set and global interrupt is enabled, the UART receiver complete interrupt is generated while RXC1 flag is set.

The UART receiver has three error flags which are frame error (FE1), data overrun (DOR1) and parity error (PE1). These error flags can be read from the USI1ST1 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 USI1DR register, read the USI1ST1 register first which contains error flags.

The frame error (FE1) flag indicates the state of the first stop bit. The FE1 flag is '0' when the stop bit was correctly detected as "1", and the FE1 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 (DOR1) flag indicates data loss due to a receive buffer full condition. DOR1 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 DOR1 flag is set, all the incoming data are lost. To prevent data loss or clear this flag, read the receive buffer.

The parity error (PE1) flag indicates that the frame in the receive buffer had a parity error when received. If parity check function is not enabled (USI1PM1=0), the PE bit is always read "0".

11.13.10.3 USI1 UART Parity Checker

If parity bit is enabled (USI1PM1=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.

11.13.10.4 USI1 UART Disabling Receiver

In contrast to transmitter, disabling the Receiver by clearing RXE1 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 RXD1 pin can be used as a normal general purpose I/O (GPIO).

11.13.10.5 USI1 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 RXD1 pin.

The data recovery logic samples and low pass filters the incoming bits, and this removes the noise of RXD1 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 aud-rate for double speed mode (DBLS1=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.

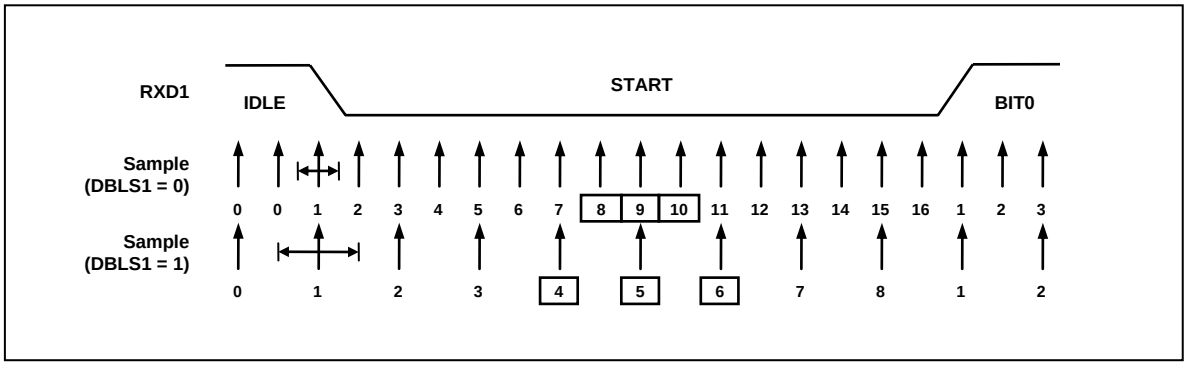


Figure 11.82 Asynchronous Start Bit Sampling (USI1)

When the receiver is enabled (RXE1=1), the clock recovery logic tries to find a high-to-low transition on the RXD1 line, the start bit condition. After detecting high to low transition on RXD1 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.

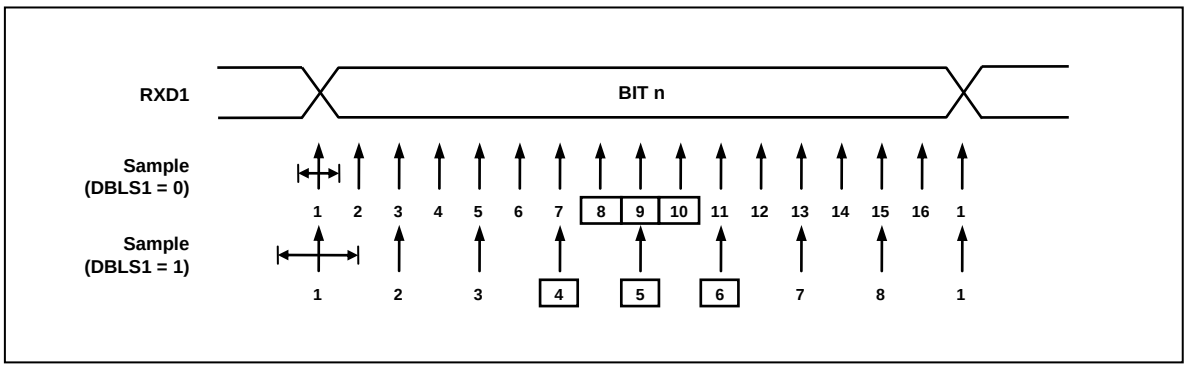


Figure 11.83 Asynchronous Sampling of Data and Parity Bit (USI1)

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 (FE1) flag is set. After deciding whether the first stop bit is valid or not, the Receiver goes to idle state and monitors the RXD1 line to check a valid high to low transition is detected (start bit detection).

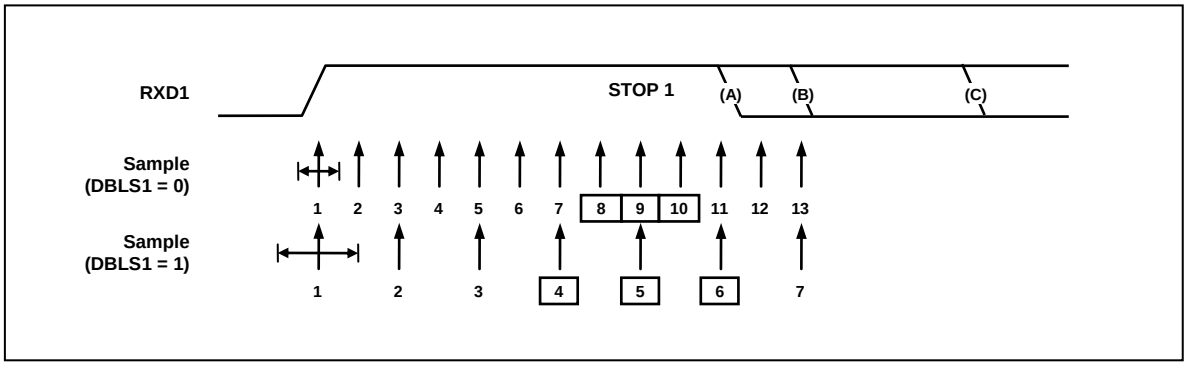


Figure 11.84 Stop Bit Sampling and Next Start Bit Sampling (USI1)

11.13.11 USI1 SPI Mode

The USI1 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 (USI1MS[1:0]=”11”), the slave select (SS1) pin becomes active LOW input in slave mode operation, or can be output in master mode operation if USI1SSEN bit is set to ‘0’.

Note that during SPI mode of operation, the pin RXD1 is renamed as MISO1 and TXD1 is renamed as MOSI1 for compatibility to other SPI devices.

11.13.12 USI1 SPI Clock Formats and Timing

To accommodate a wide variety if synchronus serial peripherals from different manufacturers, the USI1 has a clock polarity bit (CPOL1) and a clock phase control bit (CPHA1) to select one of four clock formats for data transfers. CPOL1 selectively insert an inverter in series with the clock. CPHA1 chooses between two different clock phase relationships between the clock and data. Note that CPHA1 and CPOL1 bits in USI1CR1 register have different meanings according to the USI1MS[1:0] bits which decides the operating mode of USI1.

Table below shows four combinations of CPOL1 and CPHA1 for SPI mode 0, 1, 2, and 3.

Table 11-23 CPOL1 Functionality

SPI Mode	CPOL1	CPHA1	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)

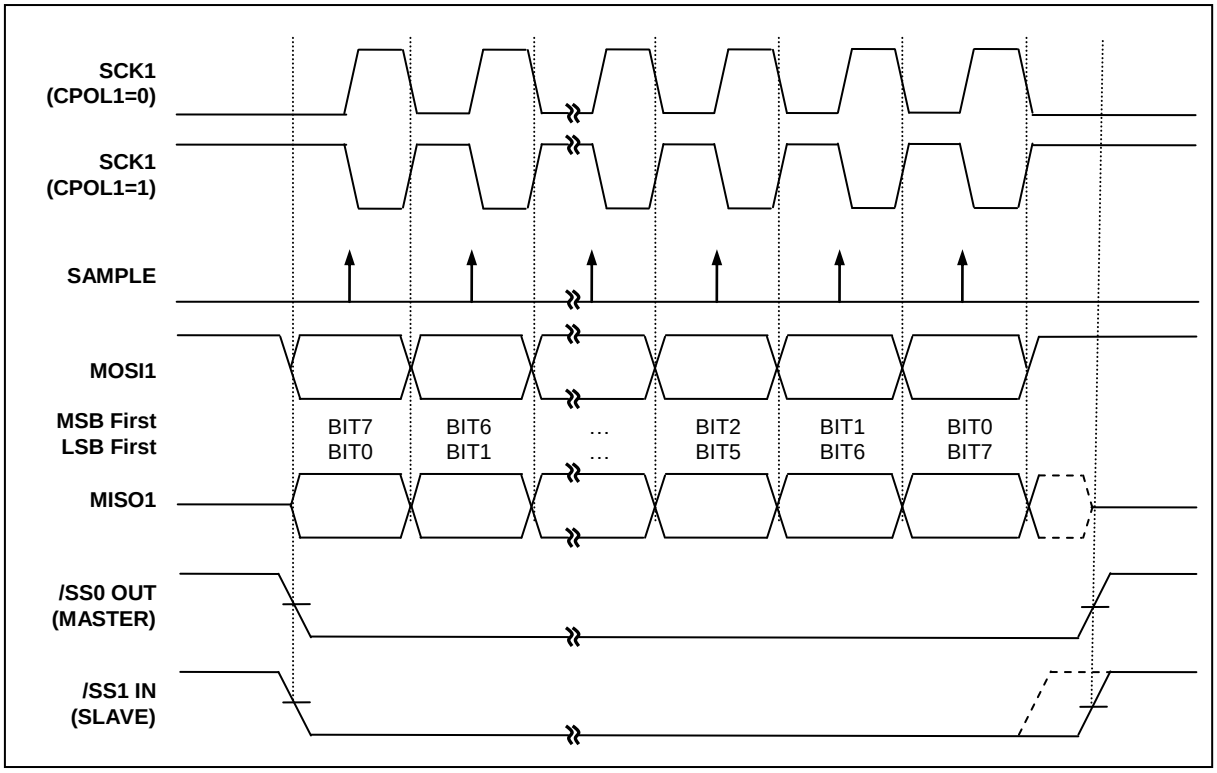


Figure 11.85 USI1 SPI Clock Formats when CPHA1=0

When CPHA1=0, the slave begins to drive its MISO1 output with the first data bit value when SS1 goes to active low. The first SCK1 edge causes both the master and the slave to sample the data bit value on their MISO1 and MOSI1 inputs, respectively. At the second SCK1 edge, the USI1 shifts the second data bit value out to the MOSI1 and MISO1 outputs of the master and slave, respectively. Unlike the case of CPHA1=1, when CPHA1=0, the slave's SS1 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 SS1 input.

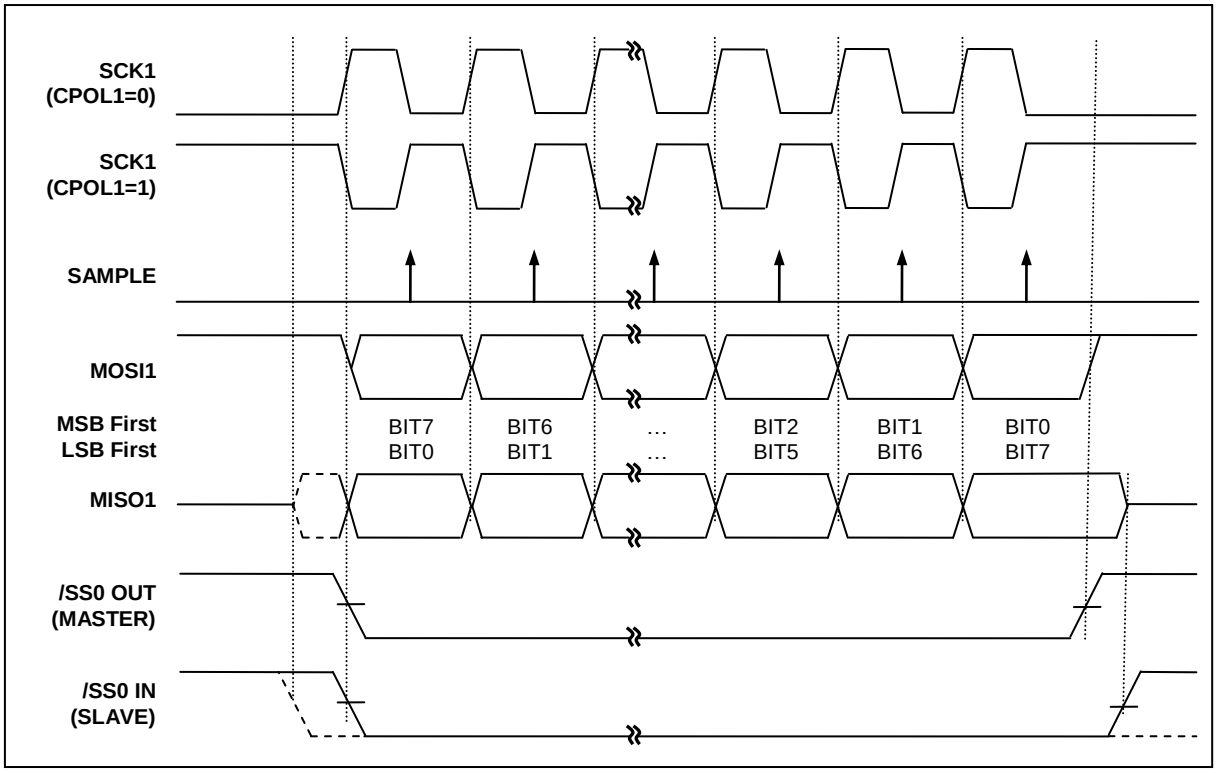


Figure 11.86 USI1 SPI Clock Formats when CPHA1=1

When CPHA1=1, the slave begins to drive its MISO1 output when SS1 goes active low, but the data is not defined until the first SCK1 edge. The first SCK1 edge shifts the first bit of data from the shifter onto the MOSI1 output of the master and the MISO1 output of the slave. The next SCK1 edge causes both the master and slave to sample the data bit value on their MISO1 and MOSI1 inputs, respectively. At the third SCK1 edge, the USI1 shifts the second data bit value out to the MOSI1 and MISO1 output of the master and slave respectively. When CPHA1=1, the slave's SS1 input is not required to go to its inactive high level between transfers.

Because the SPI logic reuses the USI1 resources, SPI mode of operation is similar to that of synchronous or asynchronous operation. An SPI transfer is initiated by checking for the USI1 Data Register Empty flag (DRE1=1) and then writing a byte of data to the USI1DR Register. In master mode of operation, even if transmission is not enabled (TXE1=0), writing data to the USI1DR register is necessary because the clock SCK1 is generated from transmitter block.

11.13.13 USI1 SPI Block Diagram

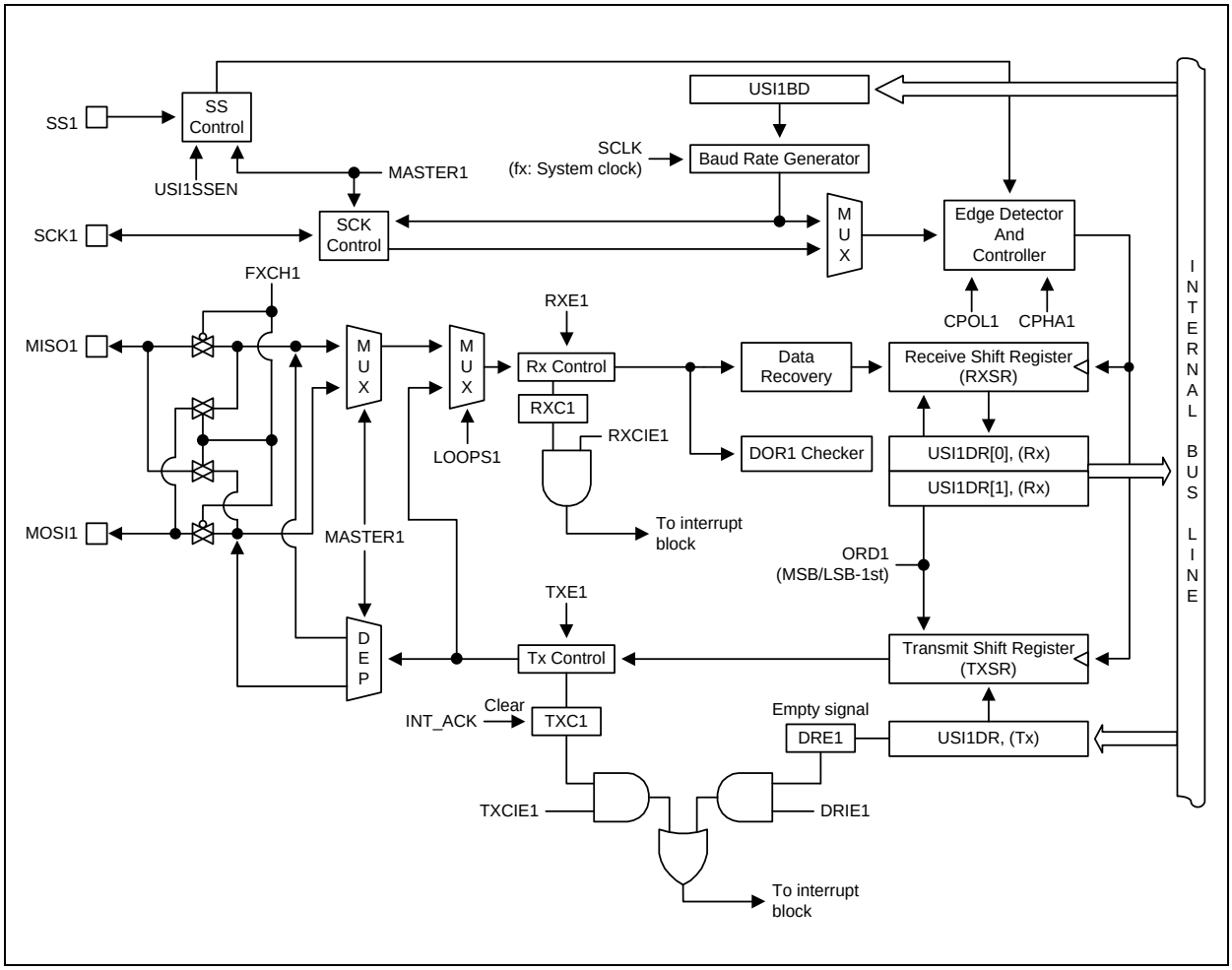


Figure 11.87 USI1 SPI Block Diagram

11.13.14 USI1 I2C Mode

The USI1 can be set to operate in industrial standard serial communicatin protocols mode. The I2C mode uses 2 bus lines serial data line (SDA1) and serial clock line (SCL1) to exchange data. Because both SDA1 and SCL1 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

11.13.15 USI1 I2C Bit Transfer

The data on the SDA1 line must be stable during HIGH period of the clock, SCL1. The HIGH or LOW state of the data line can only change when the clock signal on the SCL1 line is LOW. The exceptions are START(S), repeated START(Sr) and STOP(P) condition where data line changes when clock line is high.

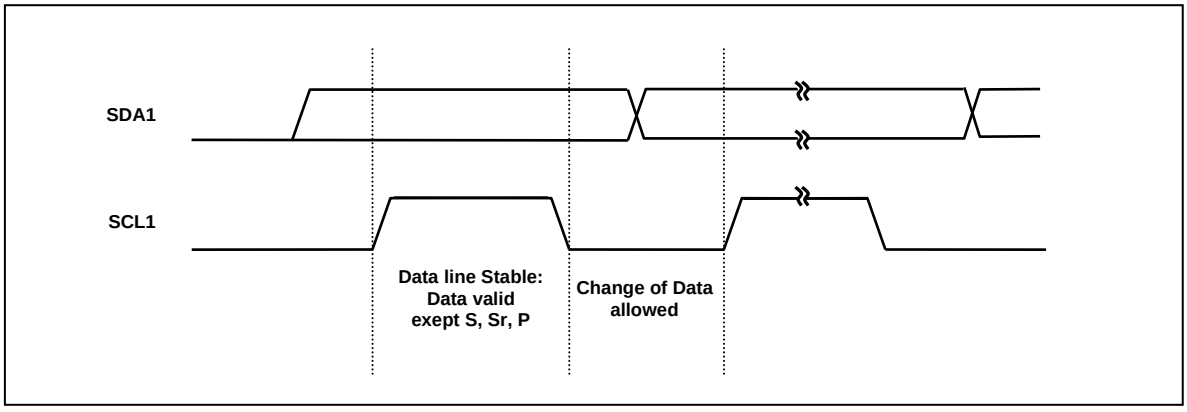


Figure 11.88 Bit Transfer on the I2C-Bus (USI1)

11.13.16 USI1 I2C Start / Repeated Start / Stop

One master can issue a START (S) condition to notice other devices connected to the SCL1, SDA1 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 SDA1 line while SCL1 is high defines a START (S) condition.
A low to high transition on the SDA1 line while SCL1 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.

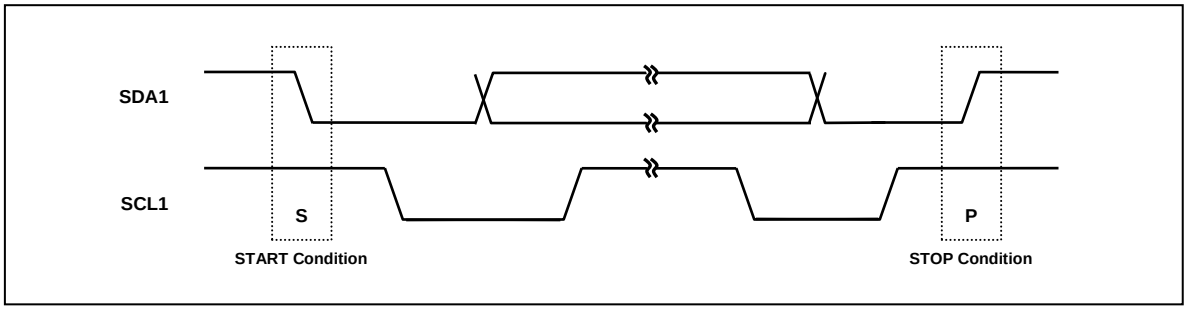


Figure 11.89 START and STOP Condition (USI1)

11.13.17 USI1 I2C Data Transfer

Every byte put on the SDA1 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 SCL1 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 SCL1.

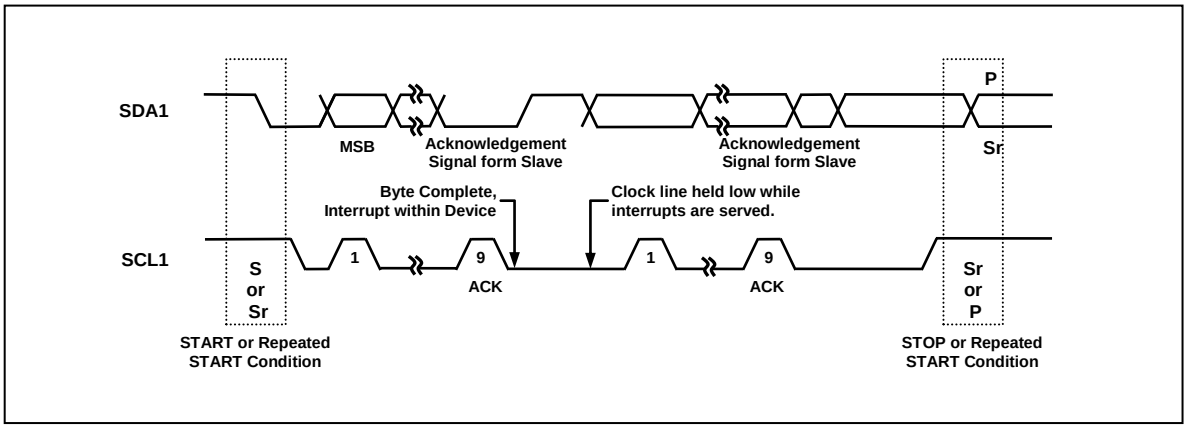


Figure 11.90 Data Transfer on the I2C-Bus (USI1)

11.13.18 USI1 I2C Acknowledge

The acknowledge related clock pulse is generated by the master. The transmitter releases the SDA1 line (HIGH) during the acknowledge clock pulse. The receiver must pull down the SDA1 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 SDA1 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.

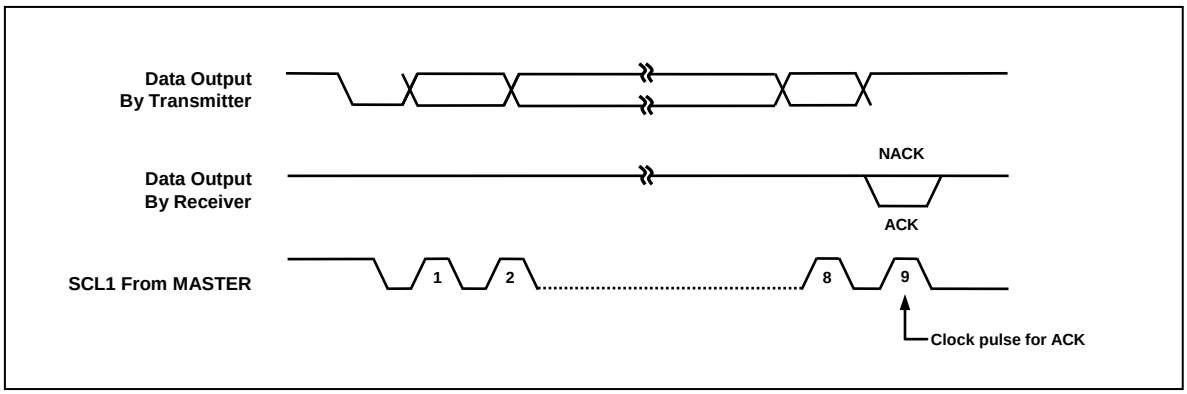


Figure 11.91 Acknowledge on the I2C-Bus (USI1)

11.13.19 USI1 I2C Synchronization / Arbitration

Clock synchronization is performed using the wired-AND connection of I2C interfaces to the SCL1 line. This means that a HIGH to LOW transition on the SCL1 line will cause the devices concerned to start counting off their LOW period and it will hold the SCL1 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 SCL1 line if another clock is still within its LOW period. In this way, a synchronized SCL1 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 SDA1 line, while the SCL1 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.

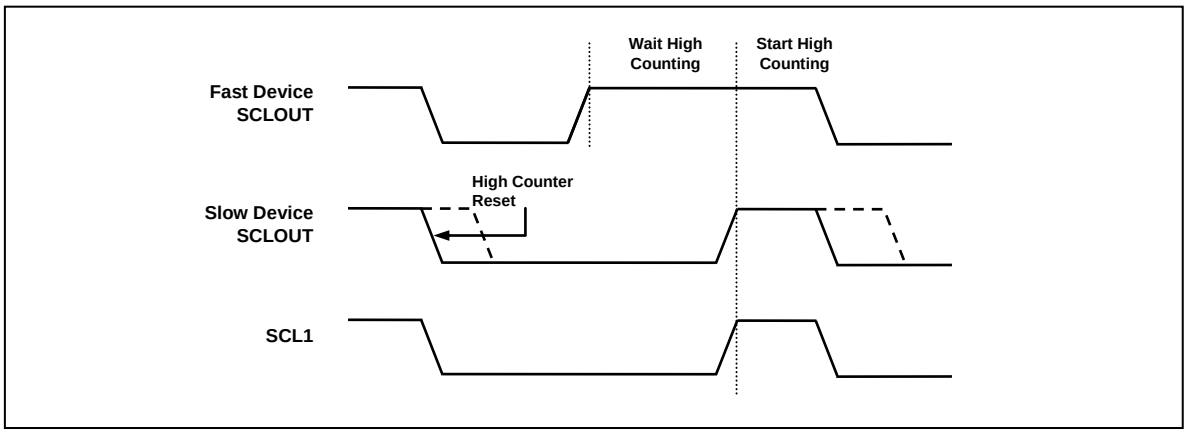


Figure 11.92 Clock Synchronization during Arbitration Procedure (USI1)

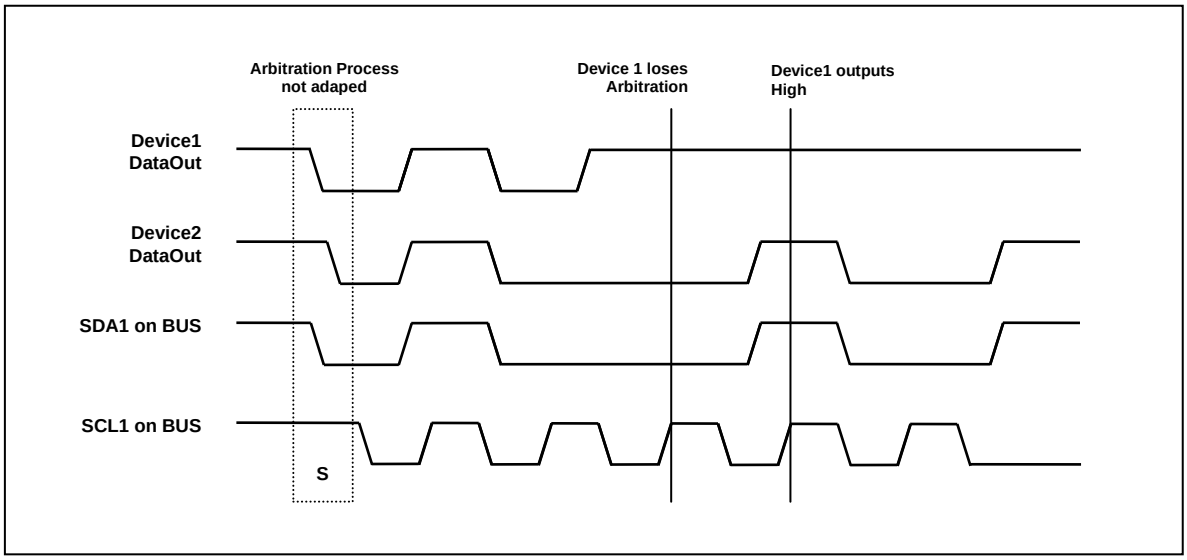


Figure 11.93 Arbitration Procedure of Two Masters (USI1)

11.13.20 USI1 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, IIC1IFR flag in USI1CR4 register is set, it is cleared by writing an any value to USI1ST2. When I2C interrupt occurs, the SCL1 line is hold LOW until writing any value to USI1ST2. When the IIC1IFR flag is set, the USI1ST2 contains a value indicating the current state of the I2C bus. According to the value in USI1ST2, 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.

11.13.20.1 USI1 I2C Master Transmitter

To operate I2C in master transmitter, follow the recommended steps below.

10. Enable I2C by setting USI1MS[1:0] bits in USI1CR1 and USI1EN bit in USI1CR2. This provides main clock to the peripheral.
11. Load SLA1+W into the USI1DR where SLA1 is address of slave device and W is transfer direction from the viewpoint of the master. For master transmitter, W is '0'. Note that USI1DR is used for both address and data.
12. Configure baud rate by writing desired value to both USI1SCLR and USI1SCHR for the Low and High period of SCL1 line.
13. Configure the USI0SDHR to decide when SDA1 changes value from falling edge of SCL1. If SDA1 should change in the middle of SCL1 LOW period, load half the value of USI1SCLR to the USI1SDHR.
14. Set the STARTC1 bit in USI1CR4. This transmits a START condition. And also configure how to handle interrupt and ACK signal. When the STARTC1 bit is set, 8-bit data in USI1DR is transmitted out according to the baud-rate.
15. 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 SCL1. 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 MLOST1 bit in USI1ST2 is set, and I2C waits in idle state or can be operate as an addressed slave. To operate as a slave when the MLOST1 bit in USI1ST2 is set, the ACK1EN bit in USI1CR4 must be set and the received 7-bit address must equal to the USI1SLA[6:0] bits in USI1SAR. In this case I2C operates as a slave transmitter or a slave receiver (go to appropriate section). In this stage, I2C holds the SCL1 LOW. This is because to decide whether I2C continues serial transfer or stops 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 USI1DR.
    2) Master stops data transfer even if it receives ACK signal from slave. In this case, set the STOPC1 bit in USI1CR4.
    3) Master transmits repeated START condition with not checking ACK signal. In this case, load SLA1+R/W into the USI1DR and set STARTC1 bit in USI1CR4.

    After doing one of the actions above, write any arbitrary to USI1ST2 to release SCL1 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 USI1DR and if transfer direction bit is '1' go to master receiver section.
16. 1-Byte of data is being transmitted. During data transfer, bus arbitration continues.
17. This is ACK signal processing stage for data packet transmitted by master. I2C holds the SCL1 LOW. When I2C loses bus mastership while transmitting data arbitrating other masters, the MLOST1 bit in USI1ST2 is set. If then, I2C waits in idle state. When the data in USI1DR is transmitted completely, I2C generates TEND1 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 USI1DR.
    2) Master stops data transfer even if it receives ACK signal from slave. In this case, set the STOPC1 bit in USI1CR4.
    3) Master transmits repeated START condition with not checking ACK signal. In this case, load SLA1+R/W into the USI1DR and set the STARTC1 bit in USI1CR4.

    After doing one of the actions above, write any arbitrary to USI1ST2 to release SCL1 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 USI1DR, and if transfer direction bit is '1' go to master receiver section.
18. 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 USI1ST2, write any value to USI1ST2. After this, I2C enters idle state.

The next figure depicts above process for master transmitter operation of I2C.

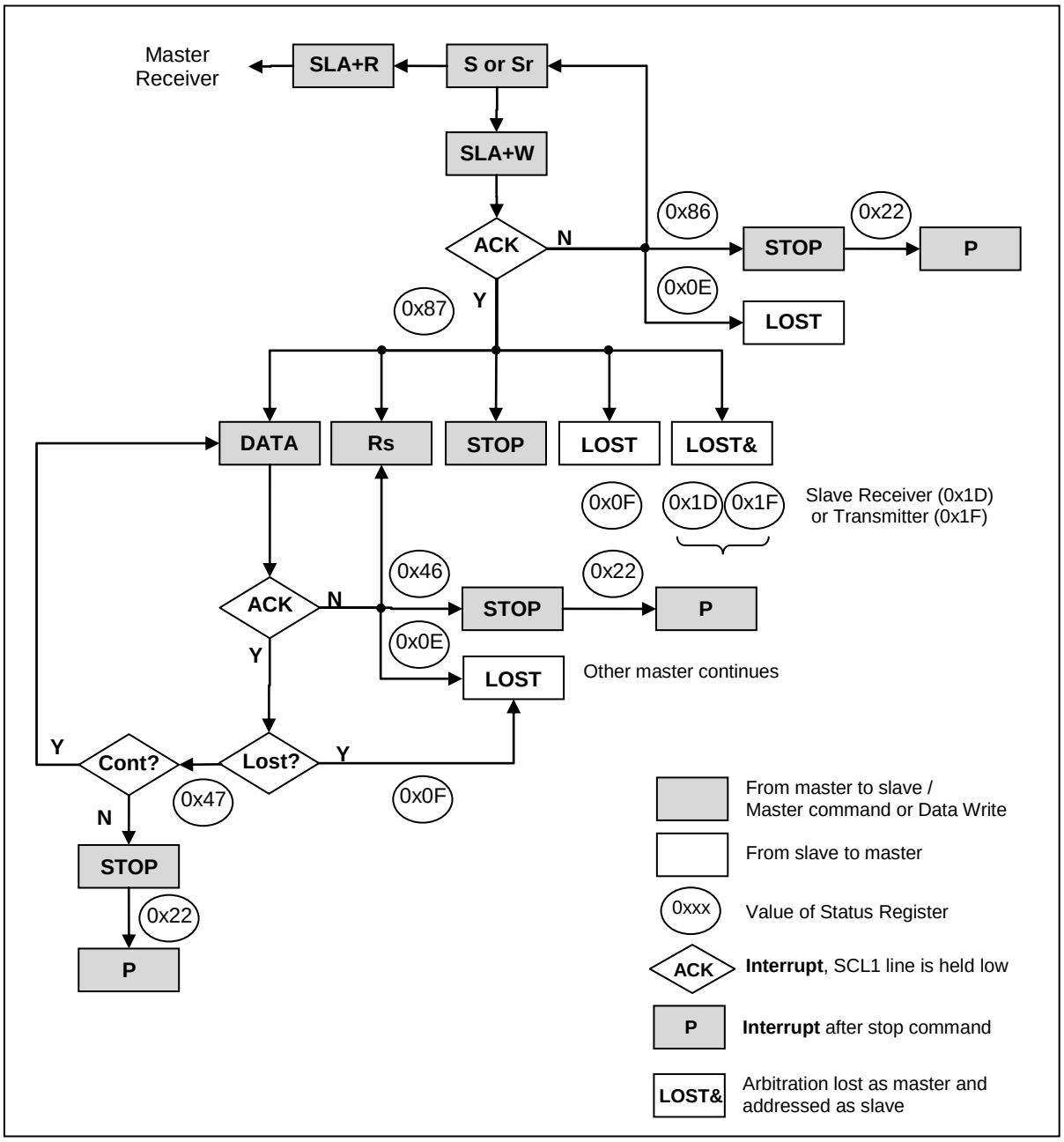


Figure 11.94 Formats and States in the Master Transmitter Mode (USI1)

11.13.20.2 USI1 I2C Master Receiver

To operate I2C in master receiver, follow the recommended steps below.

10. Enable I2C by setting USI1MS[1:0] bits in USI1CR1 and USI1EN bit in USI1CR2. This provides main clock to the peripheral.
11. Load SLA1+R into the USI1DR 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 USI1DR is used for both address and data.
12. Configure baud rate by writing desired value to both USI1SCLR and USI1SCHR for the Low and High period of SCL1 line.
13. Configure the USI1SDHR to decide when SDA1 changes value from falling edge of SCL1. If SDA1 should change in the middle of SCL1 LOW period, load half the value of USI1SCLR to the USI1SDHR.
14. Set the STARTC1 bit in USI1CR4. This transmits a START condition. And also configure how to handle interrupt and ACK signal. When the STARTC1 bit is set, 8-bit data in USI1DR is transmitted out according to the baud-rate.
15. 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 SCL1. 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 MLOST1 bit in USI1ST2 is set, and I2C waits in idle state or can be operate as an addressed slave. To operate as a slave when the MLOST1 bit in USI1ST2 is set, the ACK1EN bit in USI1CR4 must be set and the received 7-bit address must equal to the USI1SLA[6:0] bits in USI1SAR. In this case I2C operates as a slave transmitter or a slave receiver (go to appropriate section). In this stage, I2C holds the SCL1 LOW. This is because to decide whether I2C continues serial transfer or stops 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 ACK0EN bit in USI0CR4 to decide whether I2C ACKnowledges the next data to be received or not.
    2) Master stops data transfer because it receives no ACK signal from slave. In this case, set the STOPC1 bit in USI1CR4.
    3) Master transmits repeated START condition due to no ACK signal from slave. In this case, load SLA1+R/W into the USI1DR and set STARTC1 bit in USI1CR4.

    After doing one of the actions above, write arbitrary value to USI1ST2 to release SCL1 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 USI1DR and if transfer direction bit is '0' go to master transmitter section.

16. 1-Byte of data is being received.
17. This is ACK signal processing stage for data packet transmitted by slave. I2C holds the SCL1 LOW. When 1-Byte of data is received completely, I2C generates TEND1 interrupt.

    I2C can choose one of the following cases according to the RXACK1 flag in USI1ST2.

    1) Master continues receiving data from slave. To do this, set ACK1EN bit in USI0CR4 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 ACK1EN bit in USI1CR4.
    3) Because no ACK signal is detected, master terminates data transfer. In this case, set the STOPC1 bit in USI1CR4.
    4) No ACK signal is detected, and master transmits repeated START condition. In this case, load SLA1+R/W into the USI1DR and set the STARTC1 bit in USI1CR4.

    After doing one of the actions above, write arbitrary value to USI1ST2 to release SCL1 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 USI1DR, and if transfer direction bit is '0' go to master transmitter section.

18. 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 USI1ST2, write any value to USI1ST2. After this, I2C enters idle state.

The processes described above for master receiver operation of I2C can be depicted as the following figure.

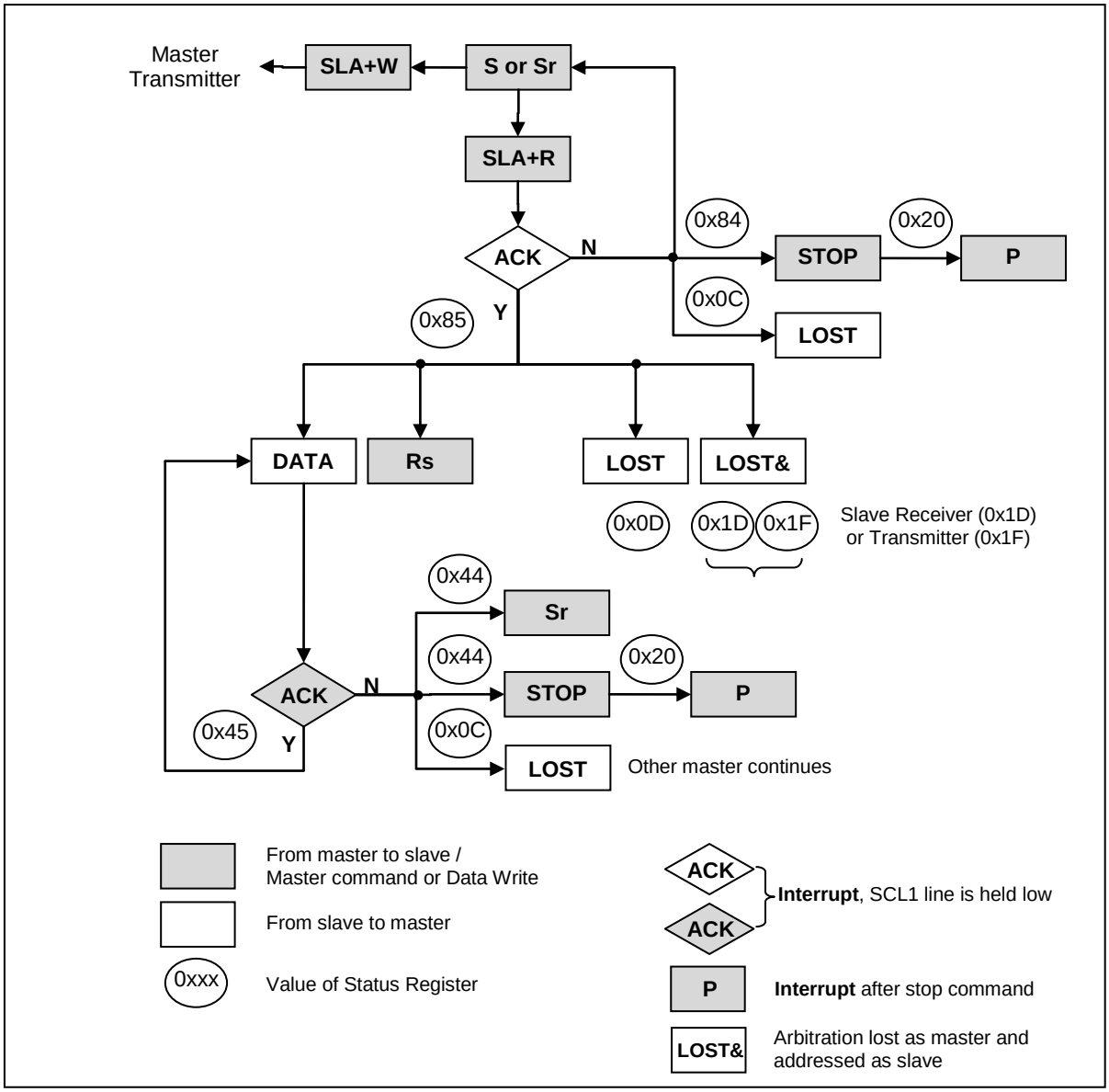


Figure 11.95 Formats and States in the Master Receiver Mode (USI1)

11.13.20.3 USI1 I2C Slave Transmitter

To operate I2C in slave transmitter, follow the recommended steps below.

8. If the main operating clock (SCLK) of the system is slower than that of SCL1, load value 0x00 into USI1SDHR to make SDA1 change within one system clock period from the falling edge of SCL1. Note that the hold time of SDA1 is calculated by SDAH x period of SCLK where SDAH is multiple of number of SCLK coming from USI1SDHR. When the hold time of SDA1 is longer than the period of SCLK, I2C (slave) cannot transmit serial data properly.
9. Enable I2C by setting USI1MS[1:0] bits in USI1CR1 , IIC1IE bit in USI1CR4 and USI1EN bit in USI1CR2. This provides main clock to the peripheral.
10. When a START condition is detected, I2C receives one byte of data and compares it with USI1SLA[6:0] bits in USI1SAR. If the GCALL1 bit in USI1SAR is enabled, I2C compares the received data with value 0x00, the general call address.
11. If the received address does not equal to USI1SLA[6:0] bits in USI1SAR, I2C enters idle state ie, waits for another START condition. Else if the address equals to USI1SLA[6:0] bits and the ACK1EN bit is enabled, I2C generates SSEL1 interrupt and the SCL1 line is held LOW. Note that even if the address equals to USI1SLA[6:0] bits, when the ACK1EN bit is disabled, I2C enters idle state. When SSEL1 interrupt occurs, load transmit data to USI1DR and write arbitrary value to USI1ST2 to release SCL1 line.
12. 1-Byte of data is being transmitted.
13. In this step, I2C generates TEND1 interrupt and holds the SCL1 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 USI1DR.

    After doing one of the actions above, write arbitrary value to USI1ST2 to release SCL1 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.
14. This is the final step for slave transmitter function of I2C, handling STOP interrupt. The STOPC1 bit indicates that data transfer between master and slave is over. To clear USI1ST2, write any value to USI1ST2. After this, I2C enters idle state.

The next figure shows flow chart for handling slave transmitter function of I2C.

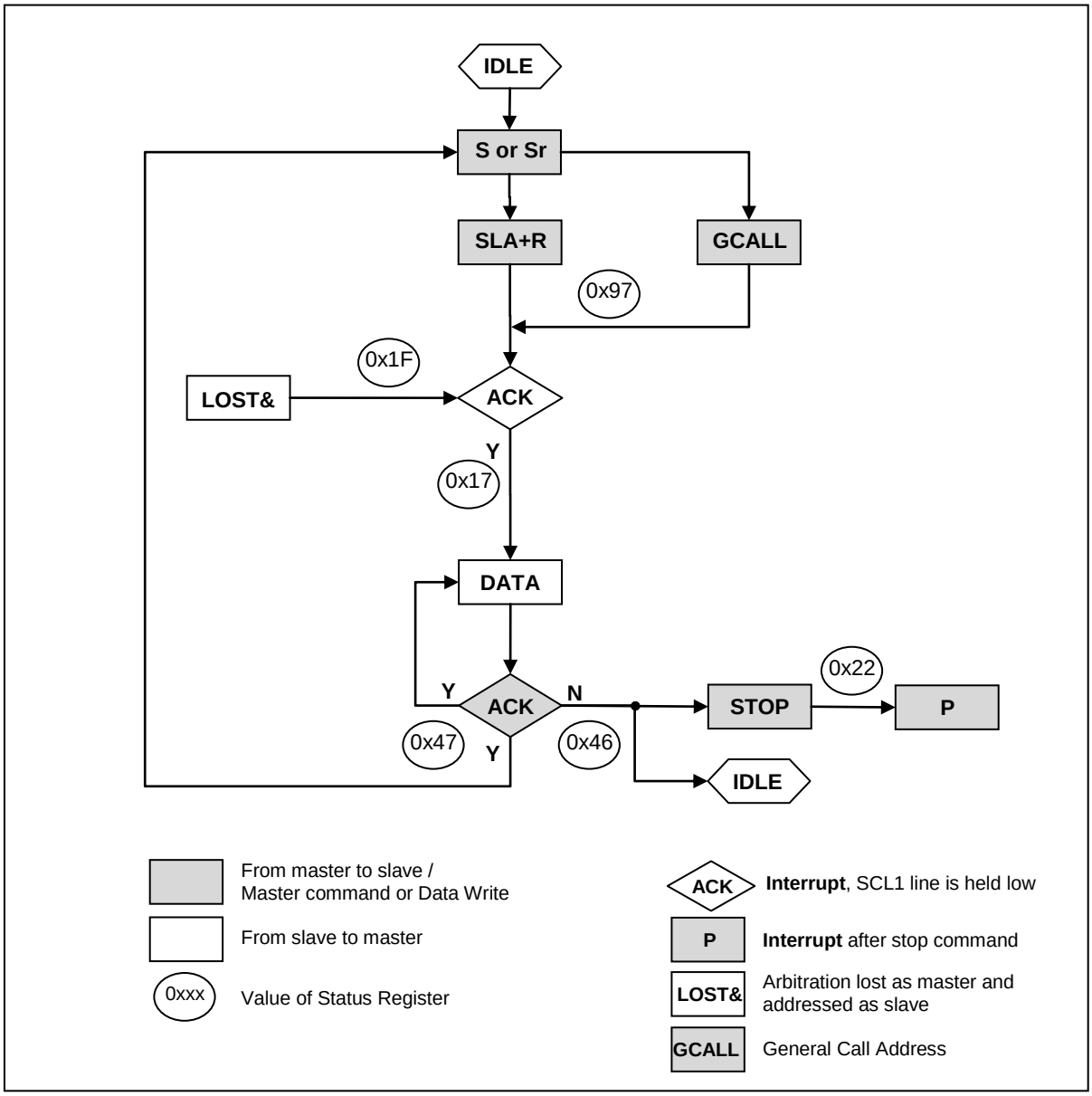


Figure 11.96 Formats and States in the Slave Transmitter Mode (USI1)

11.13.20.4 USI1 I2C Slave Receiver

To operate I2C in slave receiver, follow the recommended steps below.

8. If the main operating clock (SCLK) of the system is slower than that of SCL1, load value 0x00 into USI1SDHR to make SDA1 change within one system clock period from the falling edge of SCL1. Note that the hold time of SDA1 is calculated by SDAH x period of SCLK where SDAH is multiple of number of SCLK coming from USI1SDHR. When the hold time of SDA1 is longer than the period of SCLK, I2C (slave) cannot transmit serial data properly.
9. Enable I2C by setting USI1MS[1:0] bits in USI1CR1, IIC1IE bit in USI1CR4 and USI1EN bit in USI1CR2. This provides main clock to the peripheral.
10. When a START condition is detected, I2C receives one byte of data and compares it with USI1SLA[6:0] bits in USI1SAR. If the GCALL1 bit in USI1SAR is enabled, I2C1 compares the received data with value 0x00, the general call address.
11. If the received address does not equal to SLA1 bits in USI1SAR, I2C enters idle state ie, waits for another START condition. Else if the address equals to SLA1 bits and the ACK1EN bit is enabled, I2C generates SSEL1 interrupt and the SCL1 line is held LOW. Note that even if the address equals to SLA1 bits, when the ACK1EN bit is disabled, I2C enters idle state. When SSEL1 interrupt occurs and I2C is ready to receive data, write arbitrary value to USI1ST2 to release SCL1 line.
12. 1-Byte of data is being received.
13. In this step, I2C generates TEND1 interrupt and holds the SCL1 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 (ACK1EN=0) and I2C waits STOP or repeated START condition.
    2) ACK signal is detected (ACK1EN=1) and I2C can continue to receive data from master.

    After doing one of the actions above, write arbitrary value to USI1ST2 to release SCL1 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.
14. This is the final step for slave receiver function of I2C, handling STOP interrupt. The STOPC1 bit indicates that data transfer between master and slave is over. To clear USI1ST2, write any value to USI1ST2. After this, I2C enters idle state.

The process can be depicted as following figure when I2C operates in slave receiver mode.

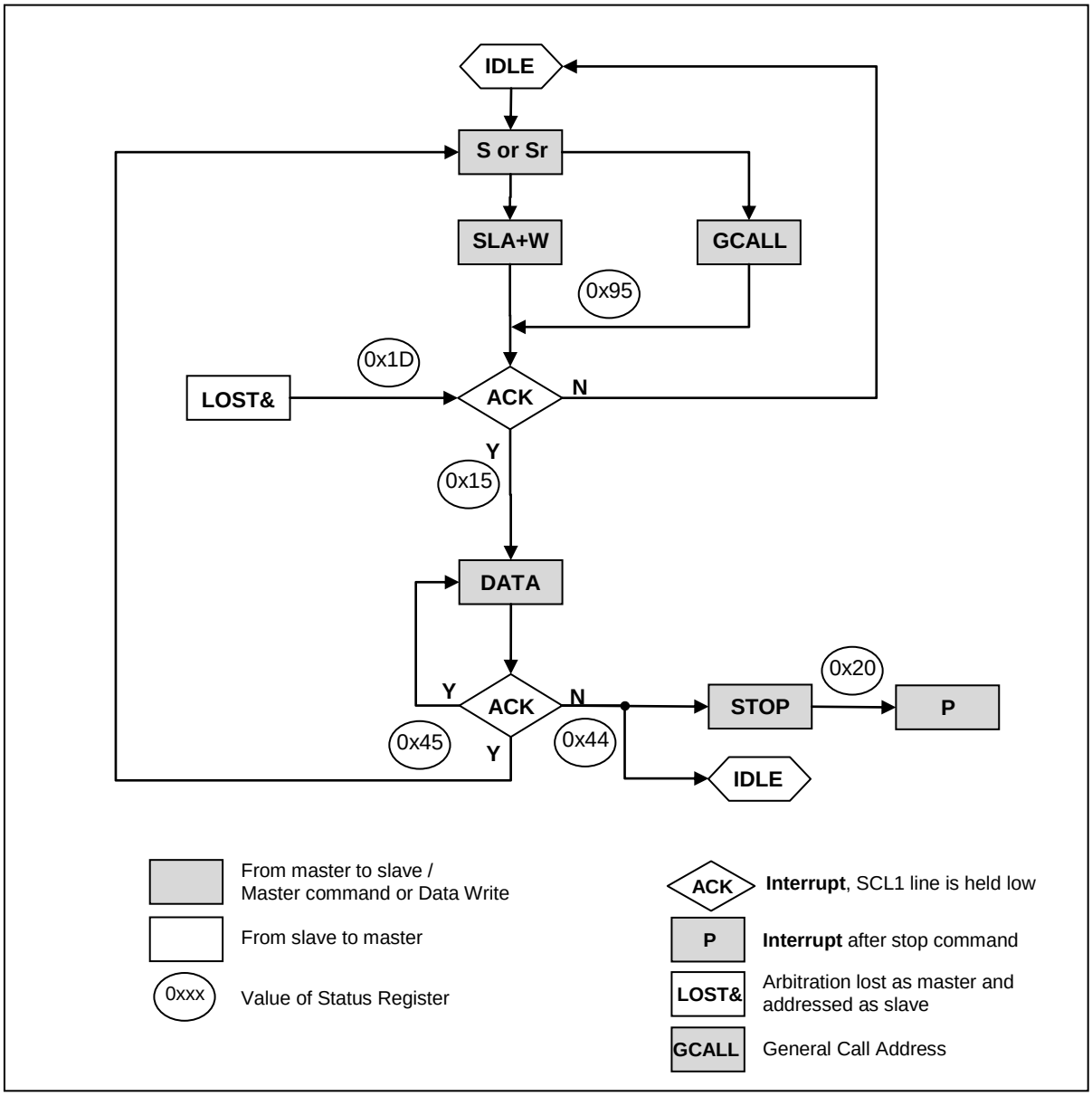
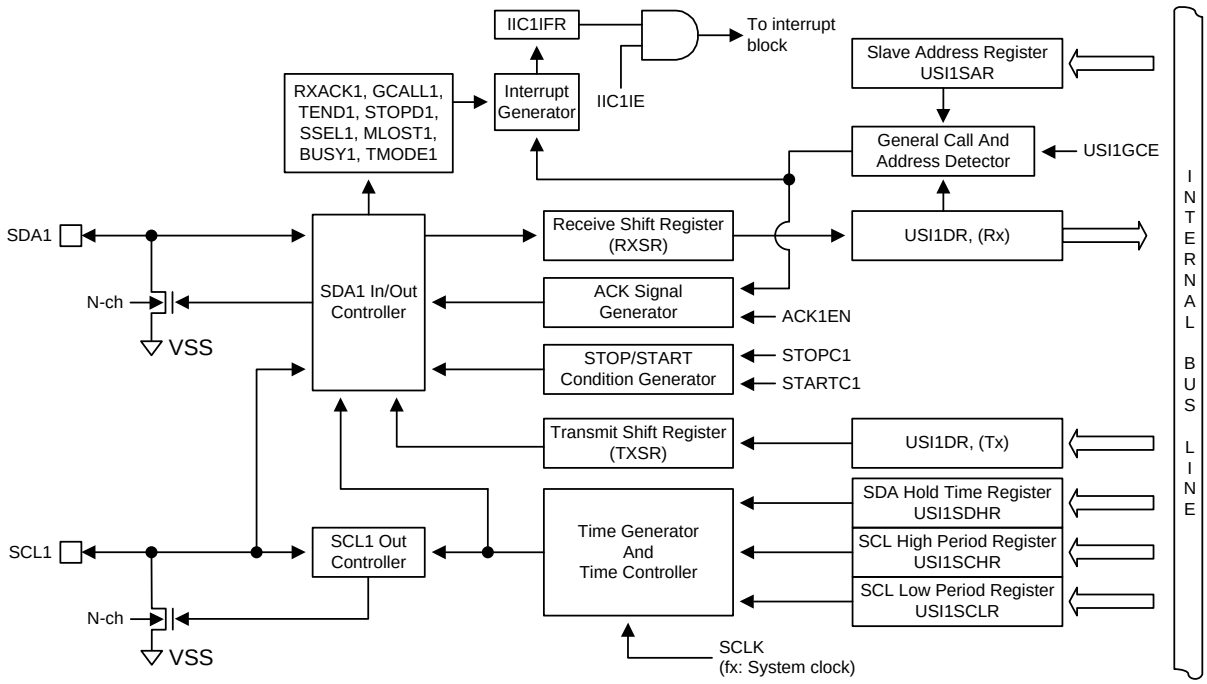


Figure 11.97 Formats and States in the Slave Receiver Mode (USI1)

11.13.21 USI1 I2C Block Diagram



NOTE) When the USI1 block is an I2C mode and the corresponding port is an sub-function for SCL1/SDA1 pin, The SCL1/SDA1 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 11.98 USI1 I2C Block Diagram

11.13.22 Register Map

Table 11-24 USI1 Register Map

Name	Address	Dir	Default	Description
USI1BD	F3H	R/W	FFH	USI1 Baud Rate Generation Register
USI1DR	F5H	R/W	00H	USI1 Data Register
USI1SDHR	F4H	R/W	01H	USI1 SDA Hold Time Register
USI1SCHR	F7H	R/W	3FH	USI1 SCL High Period Register
USI1SCLR	F6H	R/W	3FH	USI1 SCL Low Period Register
USI1SAR	EDH	R/W	00H	USI1 Slave Address Register
USI1CR1	E9H	R/W	00H	USI1 Control Register 1
USI1CR2	EAH	R/W	00H	USI1 Control Register 2
USI1CR3	EBH	R/W	00H	USI1 Control Register 3
USI1CR4	ECH	R/W	00H	USI1 Control Register 4
USI1ST1	F1H	R/W	80H	USI1 Status Register 1
USI1ST2	F2H	R	00H	USI1 Status Register 2

11.13.23 USI1 Register Description

USI1 module consists of USI1 baud rate generation register (USI1BD), USI1 data register (USI1DR), USI1 SDA hold time register (USI1SDHR), USI1 SCL high period register (USI1SCHR), USI1 SCL low period Register (USI1SCLR), USI1 slave address register (USI1SAR), USI1 control register 1/2/3/4 (USI1CR1/2/3/4), USI1 status register 1/2 (USI1ST1/2).

11.13.24 Register Description for USI1

USI1BD (USI1 Baud- Rate Generation Register: For UART and SPI mode) : F3H

7	6	5	4	3	2	1	0
USI1BD7	USI1BD6	USI1BD5	USI1BD4	USI1BD3	USI1BD2	USI1BD1	USI1BD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FFH

USI1BD[7:0] The value in this register is used to generate internal baud rate in asynchronous mode or to generate SCK1 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 USI1SAR register, USI1BD register is used for slave address register when the USI1 I2C mode.

USI1DR (USI1 Data Register: For UART, SPI, and I2C mode) : F5H

7	6	5	4	3	2	1	0
USI1DR7	USI1DR6	USI1DR5	USI1DR4	USI1DR3	USI1DR2	USI1DR1	USI1DR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

USI1DR[7:0] The USI1 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 USI1DR register. Reading the USI1DR register returns the contents of the receive buffer.

Write to this register only when the DRE1 flag is set. In SPI master mode, the SCK1 clock is generated when data are written to this register.

USI1SDHR (USI1 SDA Hold Time Register: For I2C mode) : F4H

7	6	5	4	3	2	1	0
USI1SDHR7	USI1SDHR6	USI1SDHR5	USI1SDHR4	USI1SDHR3	USI1SDHR2	USI1SDHR1	USI1SDHR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

USI1SDHR[7:0] The register is used to control SDA1 output timing from the falling edge of SCL1 in I2C mode.

NOTE) That SDA1 is changed after t_{SCLK} X USI1SDHR, in master SDA1 change in the middle of SCL1.

In slave mode, configure this register regarding the frequency of SCL1 from master.

The SDA1 is changed after tsclk X (USI1SDHR+2) in master mode. So, to insure operation in slave mode, the value

t_{SCLK} X (USI1SDHR +1) must be smaller than the period of SCL1.

USI1SCHR (USI1 SCL High Period Register: For I2C mode) : F7H

7	6	5	4	3	2	1	0
USI1SCHR7	USI1SCHR6	USI1SCHR5	USI1SCHR4	USI1SCHR3	USI1SCHR2	USI1SCHR1	USI1SCHR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

USI1SCHR[7:0] This register defines the high period of SCL1 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 USI1SCHR) +2 where

t_{SCLK} is the period of SCLK.

So, the operating frequency of I2C master mode is calculated by the following equation.

$$f_{I2C} = \frac{1}{t_{SCLK} \text{ X } (4 \text{ X } (USI1SCLR + USI1SCHR + 4))}$$

USI1SCLR (USI1 SCL Low Period Register: For I2C mode) : F6H

7	6	5	4	3	2	1	0
USI1SCLR7	USI1SCLR6	USI1SCLR5	USI1SCLR4	USI1SCLR3	USI1SCLR2	USI1SCLR1	USI1SCLR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

USI1SCLR[7:0] This register defines the high period of SCL1 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 USI1SCLR) +2 where t_{SCLK} is the period of SCLK.

USI1SAR (USI1 Slave Address Register: For I2C mode) : EDH

7	6	5	4	3	2	1	0
USI1SLA6	USI1SLA5	USI1SLA4	USI1SLA3	USI1SLA2	USI1SLA1	USI1SLA0	USI1GCE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

USI1SLA[6:0] These bits configure the slave address of I2C when it operaties in I2C slave mode.

UPM[1:0] 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

USI1CR1 (USI1 Control Register 1: For UART, SPI, and I2C mode) : E9H

7	6	5	4	3	2	1	0
USI1MS1	USI1MS0	USI1PM1	USI1PM0	USI1S2	USI1S1 ORD1	USI1S0 CPHA1	CPOL1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

USI1MS[1:0] Selects operation mode of USI1

USI1MS1	USI1MS0	Operation mode
0	0	Asynchronous Mode (UART)
0	1	Synchronous Mode
1	0	I2C mode
1	1	SPI mode

USI1PM[1:0] Selects parity generation and check methods (only UART mode)

USI1PM1	USI1PM0	Parity
0	0	No Parity
0	1	Reserved
1	0	Even Parity
1	1	Odd Parity

USI1S[2:0] When in asynchronous or synchronous mode of operation, selects the length of data bits in frame

USI1S2	USI1S1	USI1S0	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

ORD1 This bit in the same bit position with USI1S1. 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

CPOL1 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

CPHA1 This bit is in the same bit position with USI1S0. This bit determines if data are sampled on the leading or trailing edge of SCK1 (only SPI mode).

CPOL1	CPHA1	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)

USI1CR2 (USI1 Control Register 2: For UART, SPI, and I2C mode) : EAH

7	6	5	4	3	2	1	0
DRIE1	TXCIE1	RXCIE1	WAKEIE1	TXE1	RXE1	USI1EN	DBLS1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

DRIE1 Interrupt enable bit for data register empty (only UART and SPI mode).

0 Interrupt from DRE1 is inhibited (use polling)

1 When DRE1 is set, request an interrupt

TXCIE1 Interrupt enable bit for transmit complete (only UART and SPI mode).

0 Interrupt from TXC1 is inhibited (use polling)

1 When TXC1 is set, request an interrupt

RXCIE1 Interrupt enable bit for receive complete (only UART and SPI mode).

0 Interrupt from RXC1 is inhibited (use polling)

1 When RXC1 is set, request an interrupt

WAKEIE1 Interrupt enable bit for asynchronous wake in STOP mode. When device is in stop mode, if RXD1 goes to low level an interrupt can be requested to wake-up system. (only UART mode). At that time the DRIE1 bit and USI1ST1 register value should be set to '0b' and "00H", respectively.

0 Interrupt from Wake is inhibited

1 When WAKE1 is set, request an interrupt

TXE1 Enables the transmitter unit (only UART and SPI mode).

0 Transmitter is disabled

1 Transmitter is enabled

RXE1 Enables the receiver unit (only UART and SPI mode).

0 Receiver is disabled

1 Receiver is enabled

USI1EN Activate USI1 function block by supplying.

0 USI1 is disabled

1 USI1 is enabled

DBLS1 This bit selects receiver sampling rate (only UART)

0 Normal asynchronous operation

1 Double Speed asynchronous operation

USI1CR3 (USI1 Control Register 3: For UART, SPI, and I2C mode) : EBH

7	6	5	4	3	2	1	0
MASTER1	LOOPS1	DISSCK1	USI1SSEN	FXCH1	USI1SB	USI1TX8	USI1RX8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

MASTER1 Selects master or slave in SPI and synchronous mode operation and controls the direction of SCK1 pin

0 Slave mode operation (External clock for SCK1).

1 Master mode operation(Internal clock for SCK1).

LOOPS1 Controls the loop back mode of USI1 for test mode (only UART and SPI mode)

0 Normal operation

1 Loop Back mode

DISSCK1 In synchronous mode of operation, selects the waveform of SCK1 output

0 ACK is free-running while UART is enabled in synchronous master mode

1 ACK is active while any frame is on transferring

USI1SSEN This bit controls the SS1 pin operation (only SPI mode)

0 Disable

1 Enable (The SS1 pin should be a normal input)

FXCH1 SPI port function exchange control bit (only SPI mode)

0 No effect

1 Exchange MOSI1 and MISO1 function

USI1SB Selects the length of stop bit in asynchronous or synchronous mode of operation.

0 1 Stop Bit

1 2 Stop Bit

USI1TX8 The ninth bit of data frame in asynchronous or synchronous mode of operation. Write this bit first before loading the USI1DR register

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

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

USI1RX8 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’

USI1CR4 (USI1 Control Register 4: For I2C mode) : ECH

7	6	5	4	3	2	1	0
IIC1IFR	–	TXDLYENB1	IIC1IE	ACK1EN	IMASTER1	STOPC1	STARTC1
R	–	R/W	R/W	R/W	R	R/W	R/W

Initial value : 00H

IIC1IFR 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 USI1ST2.

0 I2C interrupt no generation

1 I2C interrupt generation

TXDLYENB1 USI1SDHR register control bit

0 Enable USI1SDHR register

1 Disable USI1SDHR register

IIC1IE Interrupt Enable bit for I2C mode

0 Interrupt from I2C is inhibited (use polling)

1 Enable interrupt for I2C

ACK1EN Controls ACK signal Generation at ninth SCL1 period.

0 No ACK signal is generated (SDA1 =1)

1 ACK signal is generated (SDA1 =0)

NOTES) ACK signal is output (SDA1 =0) for the following 3 cases.

1. When received address packet equals to USI1SLA bits in USI1SAR.

2. When received address packet equals to value 0x00 with GCALL1 enabled.

3. When I2C operates as a receiver (master or slave)

IMASTER1 Represent operating mode of I2C

0 I2C is in slave mode

1 I2C is in master mode

STOPC1 When I2C is master, STOP condition generation

0 No effect

1 STOP condition is to be generated

STARTC1 When I2C is master, START condition generation

0 No effect

1 START or repeated START condition is to be generated

USI1ST1 (USI1 Status Register 1: For UART and SPI mode) : F1H

7	6	5	4	3	2	1	0
DRE1	TXC1	RXC1	WAKE1	USI1RST	DOR1	FE1	PE1
R/W	R/W	R	R/W	R/W	R	R/W	R/W

Initial value : 80H

DRE1 The DRE1 flag indicates if the transmit buffer (USI1DR) is ready to receive new data. If DRE1 is '1', the buffer is empty and ready to be written. This flag can generate a DRE1 interrupt.

0 Transmit buffer is not empty.

1 Transmit buffer is empty.

TXC1 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 TXC1 interrupt is executed. This flag can generate a TXC1 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.

RXC1 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 RXC1 flag can be used to generate a RXC1 interrupt. This bit is automatically cleared.

0 There is no data unread in the receive buffer

1 There are more than 1 data in the receive buffer

WAKE1 This flag is set when the RXD1 pin is detected low while the CPU is in STOP mode. This flag can be used to generate a WAKE1 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

USI1RST This is an internal reset and only has effect on USI1. Writing '1' to this bit initializes the internal logic of USI1 and this bit is automatically cleared to '0'.

0 No operation

1 Reset USI1

DOR1 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

FE1 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

PE1 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

USI1ST2 (USI1 Status Register 2: For I2C mode) : F2H

7	6	5	4	3	2	1	0
GCALL1	TEND1	STOPD1	SSEL1	MLOST1	BUSY1	TMODE1	RXACK1
R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

GCALL1(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)

TEND1(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

STOPD1(NOTE) This bit is set when a STOP condition is detected.

0 No STOP condition is detected

1 STOP condition is detected

SSEL1(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

MLOST1(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

BUSY1 This bit reflects bus status.

0 I2C bus is idle, so a master can issue a START condition

1 I2C bus is busy

TMODE1 This bit is used to indicate whether I2C is transmitter or receiver.

0 I2C is a receiver

1 I2C is a transmitter

RXACK1 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 SCL1 line is hold LOW. To release SCL1, write rbitrary value to USI1ST2. When USI1ST2 is written, the TEND1, STOPD1, SSEL1, MLOST1, and RXACK1 bits are cleared.

11.14.1 Baud Rate setting (example)

Table 11-25 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	-	-	-	-

11.15 LCD Driver

11.15.1 Overview

The LCD driver is controlled by the LCD Control Register (LCDCRH/L). The LCLK[1:0] determines the frequency of COM signal scanning of each segment output. A RESET clears the LCD control register LCDCRH and LCDCRL values to logic ‘0’.

The LCD display can continue operating during IDLE and STOP modes if a sub-frequency clock is used as system clock source.

11.15.2 LCD Display RAM Organization

Display data are stored to the display data area in the external data memory.

The display data which stored to the display external data area (address 0000H-001AH) are read automatically and sent to the LCD driver by the hardware. The LCD driver generates the segment signals and common signals in accordance with the display data and drive method. Therefore, display patterns can be changed by only overwriting the contents of the display external data area with a program.

Figure 11-99 shows the correspondence between the display external data area and the COM/SEG pins. The LCD is turned on when the display data is “1” and turned off when “0”.

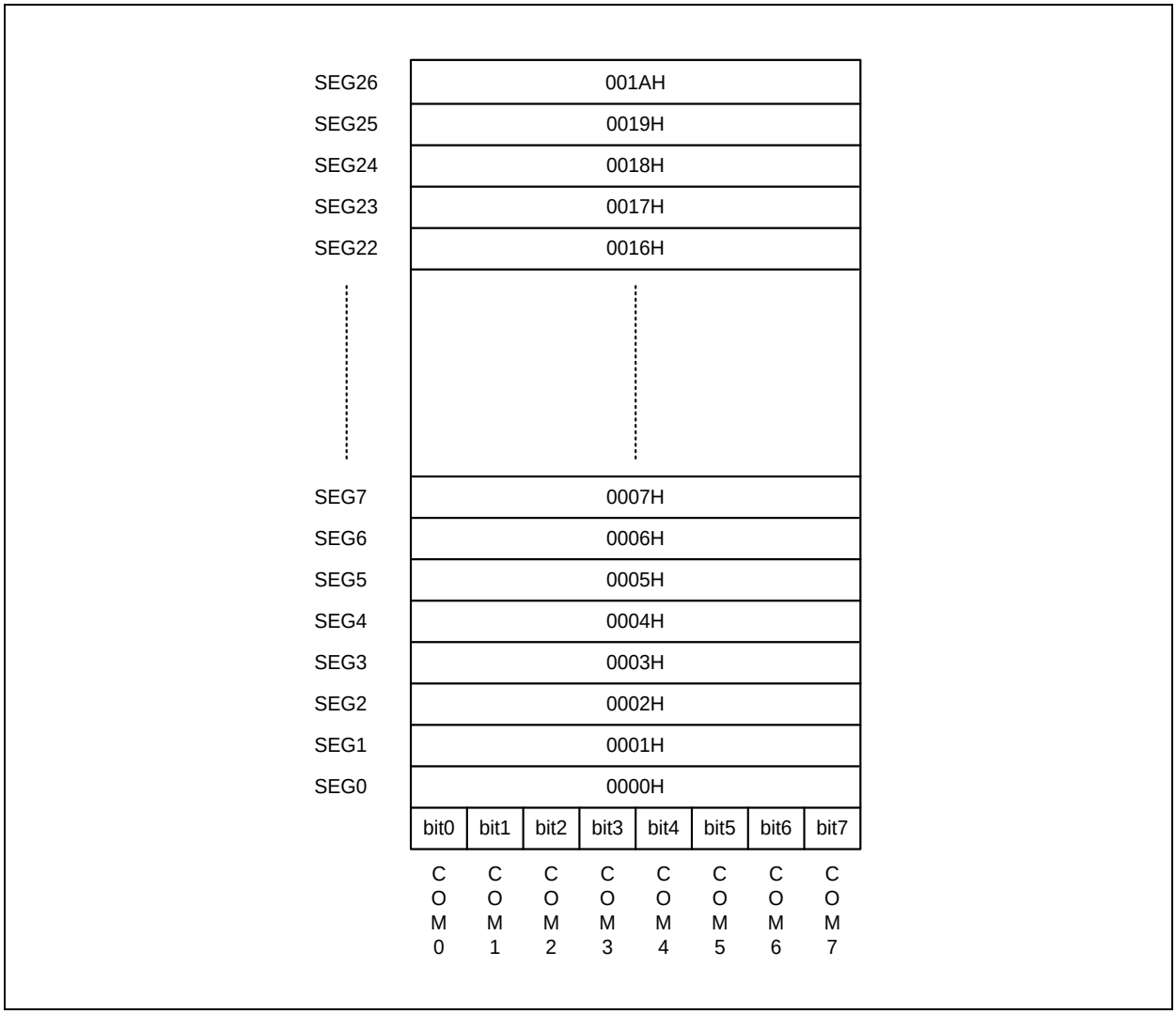


Figure 11.99 LCD Circuit Block Diagram

11.15.3 LCD Signal Waveform

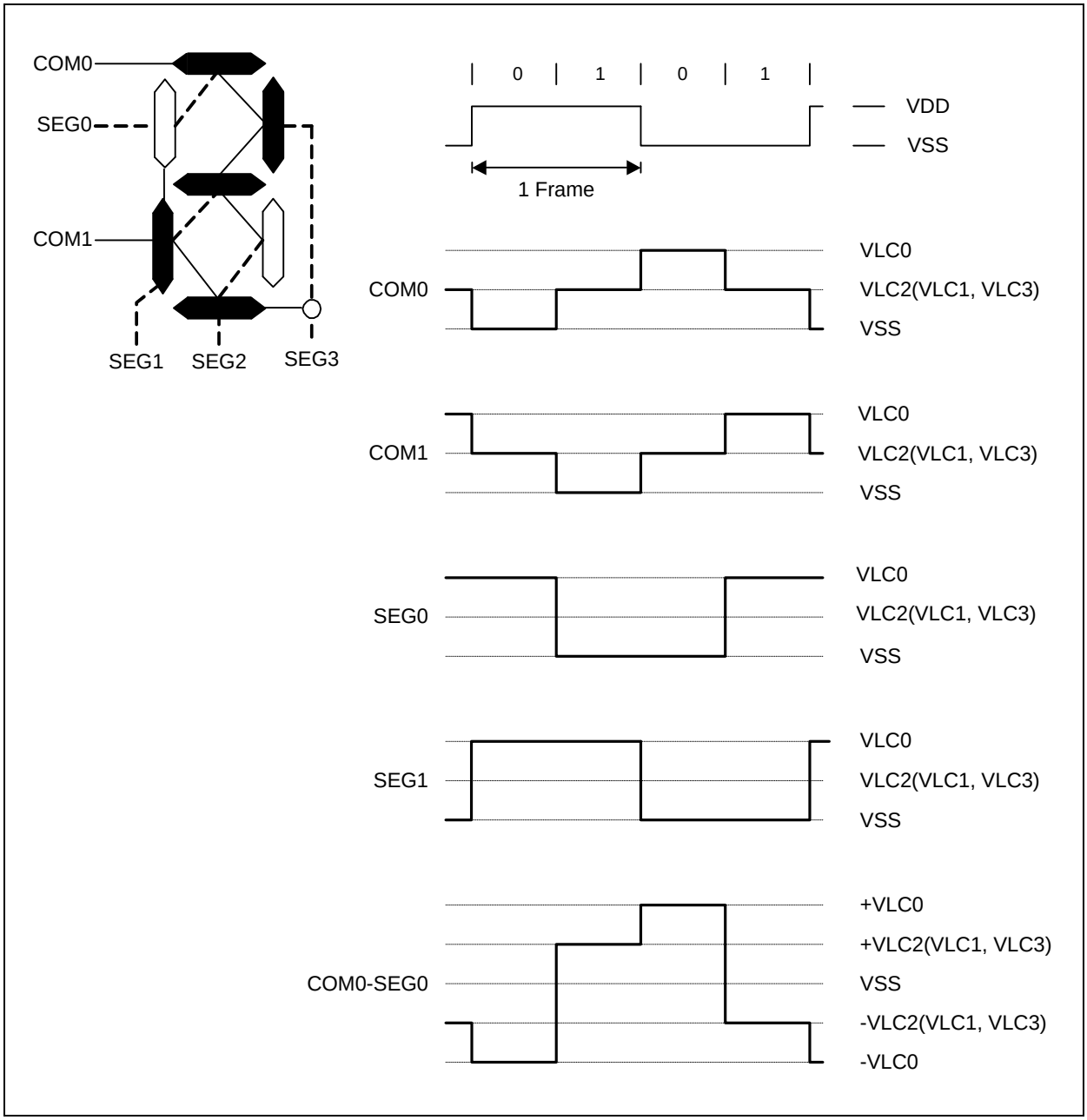


Figure 11.100 LCD Signal Waveforms (1/2Duty, 1/2Bias)

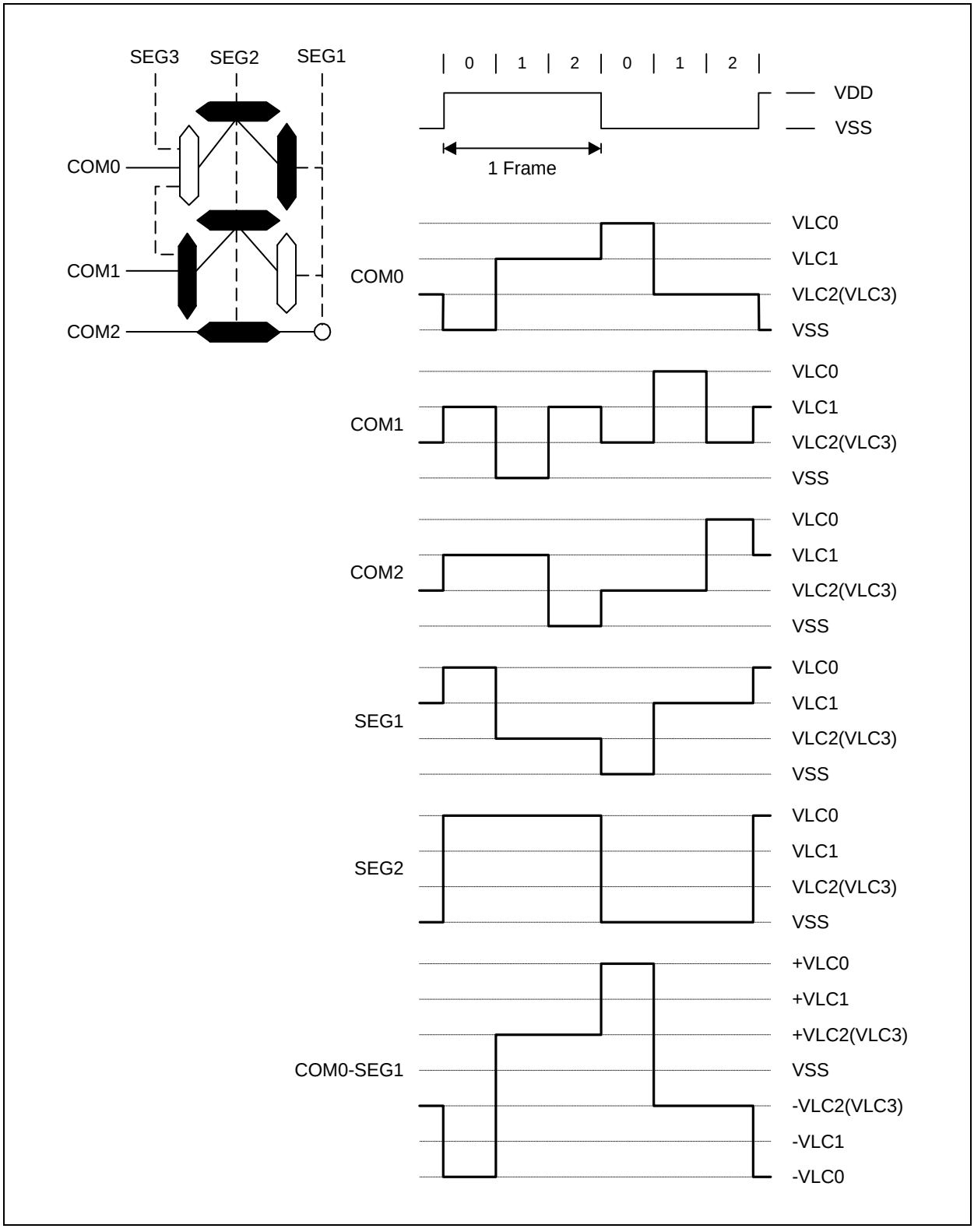


Figure 11.101 LCD Signal Waveforms (1/3Duty, 1/3Bias)

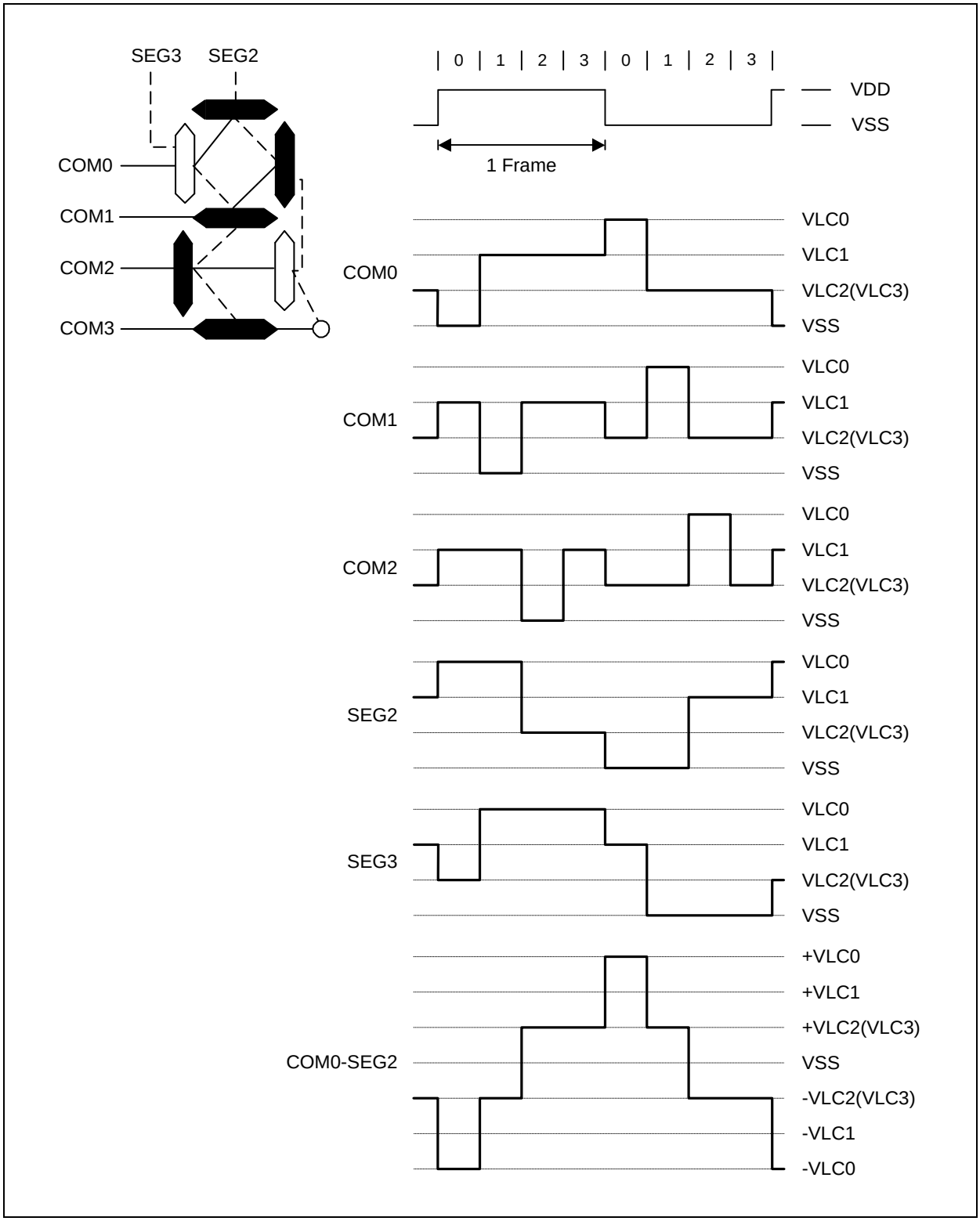


Figure 11.102 LCD Signal Waveforms (1/4Duty, 1/3Bias)

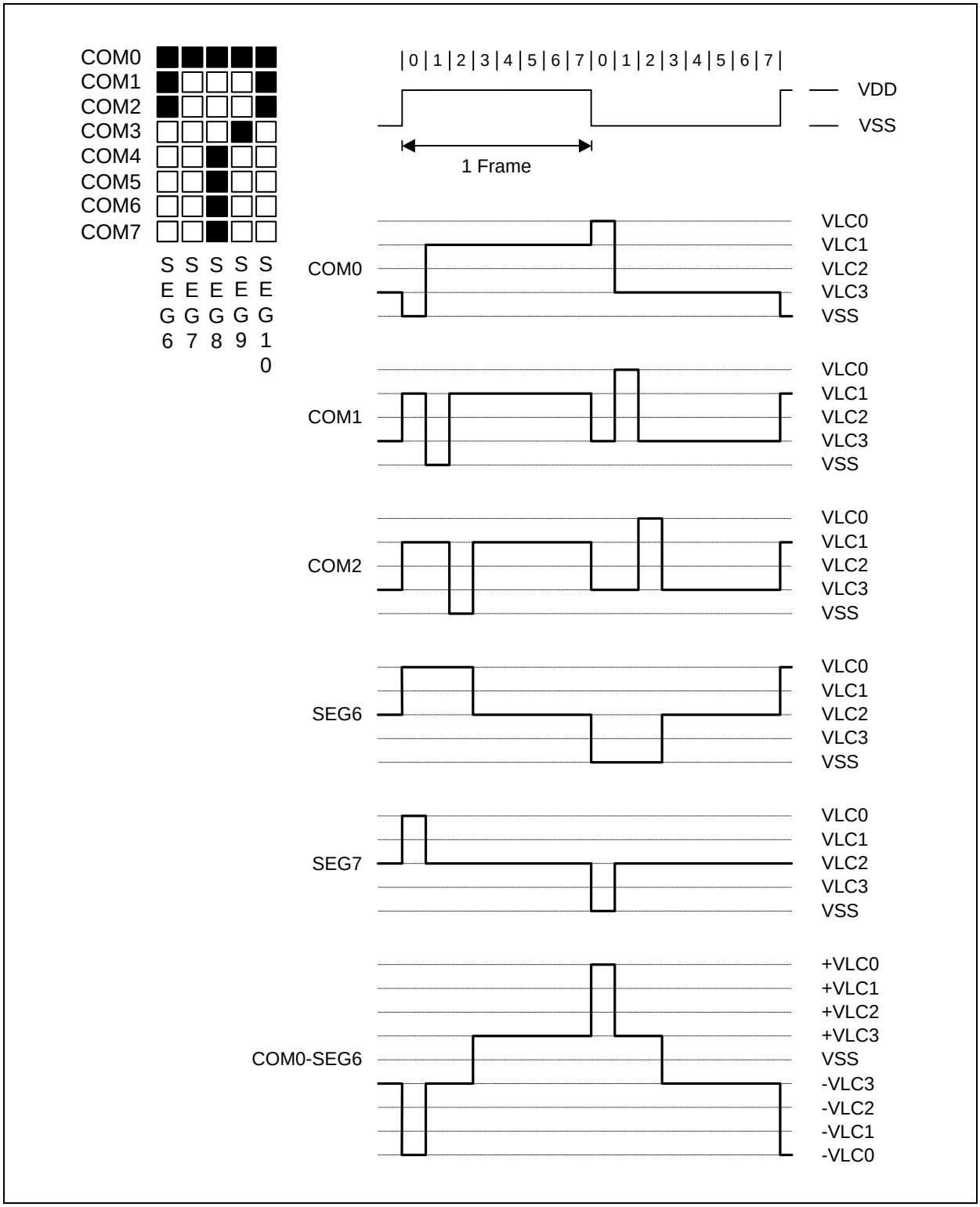
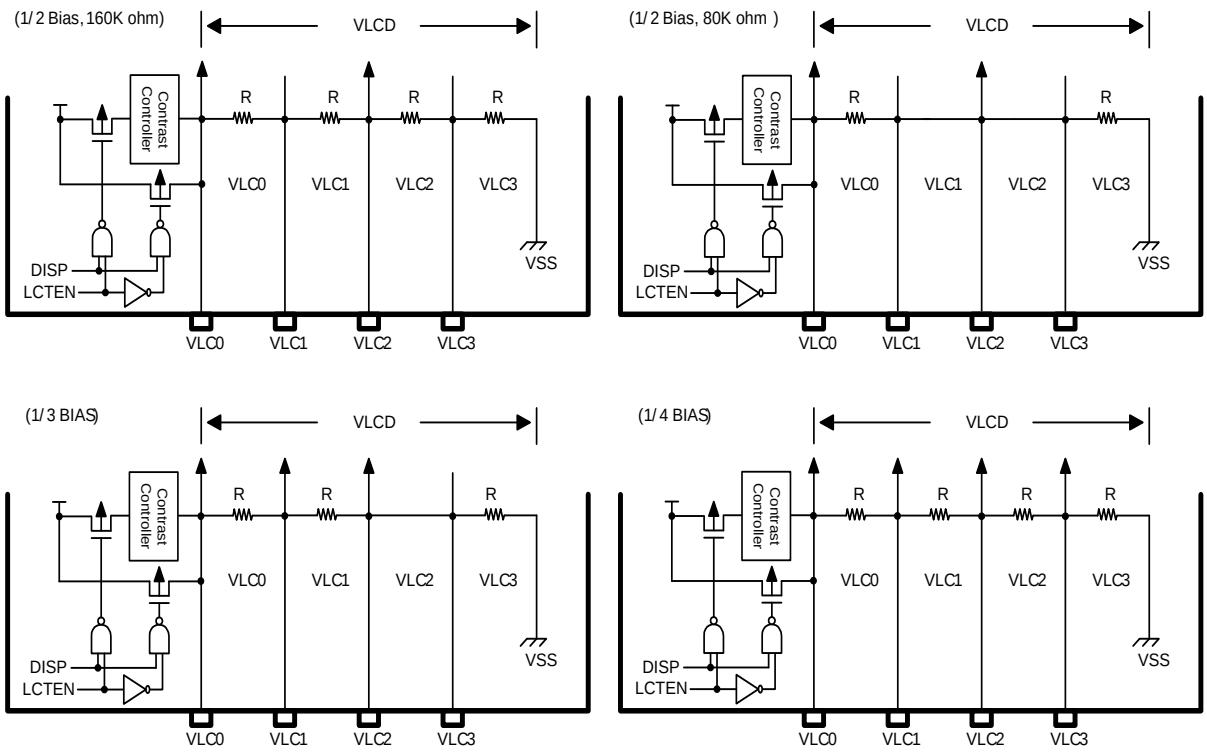


Figure 11.103 LCD Signal Waveforms (1/8Duty, 1/4Bias)

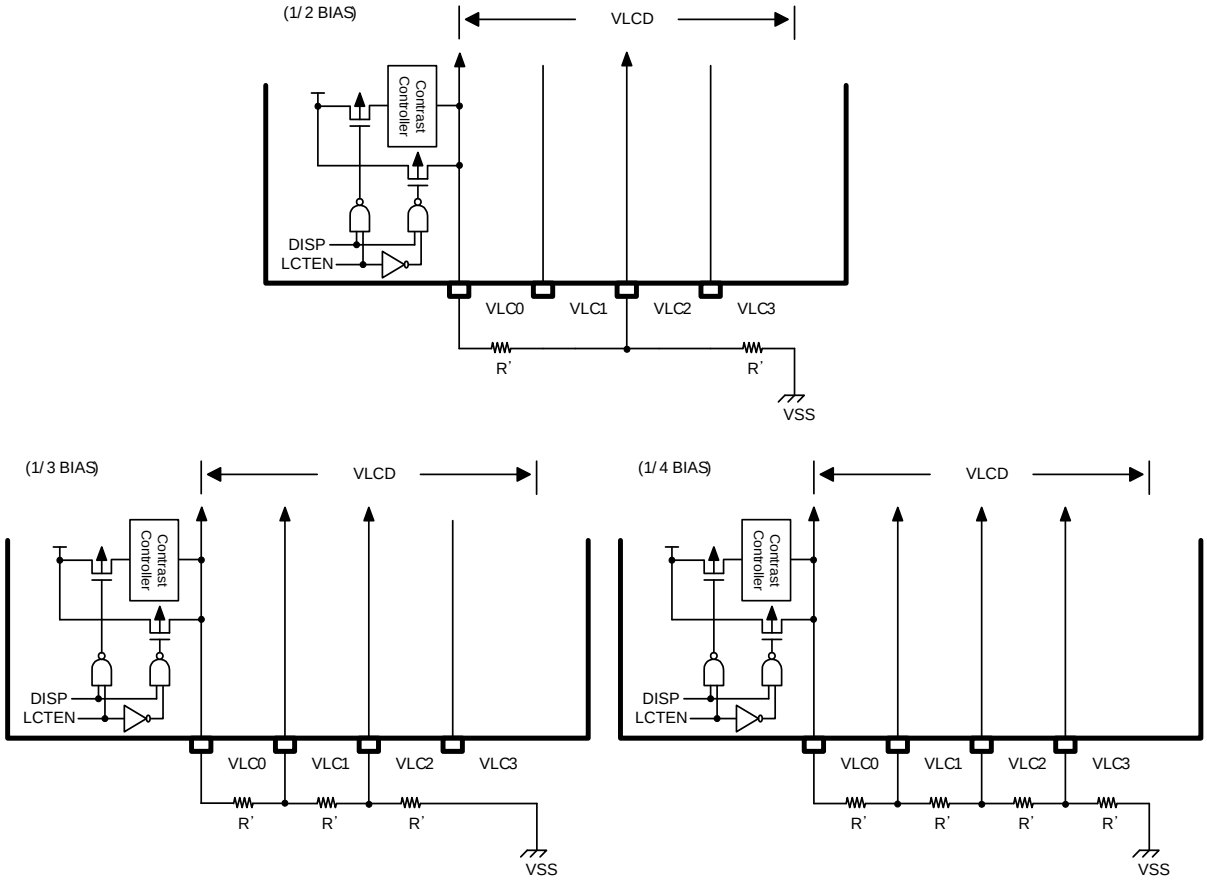
11.15.4 LCD Voltage Dividing Resistor Connection



NOTES)

1. The above figures are for the internal resistor bias connection. So, It is not needed an external connection.
2. When the internal resistors are selected, all the P40/VLC3, P41/VLC2, P42/VLC1 and P43/VLC0 pins can be used for normal I/O.

Figure 11.104 Internal Resistor Bias Connection



NOTES)

1. When the external resistor bias is selected, the internal resistors for bias are disconnected.
2. When the external resistor bias is selected, the dividing resistors should be connected like the above figure and the needed bias pins should be selected as the LCD bias function pins (VLC0, VLC1, VLC2, and VLC3) by P4FSR register.
   - When it is 1/2 bias, the P43/VLC0 and P41/VLC2 pins should be selected as VLC0 and VLC2 functions. The other pins can be used for normal I/O.
   - When it is 1/3 bias, the P43/VLC0, P42/VLC1, and P41/VLC2 pins should be selected as VLC0, VLC1, and VLC2 functions. Another pin can be used for normal I/O.
   - When it is 1/4 bias, the P43/VLC0, P42/VLC1, P41/VLC2, and P40/VLC3 pins should be selected as VLC0, VLC1, VLC2, and VLC3 functions

Figure 11.105 External Resistor Bias Connection

11.15.5 Block Diagram

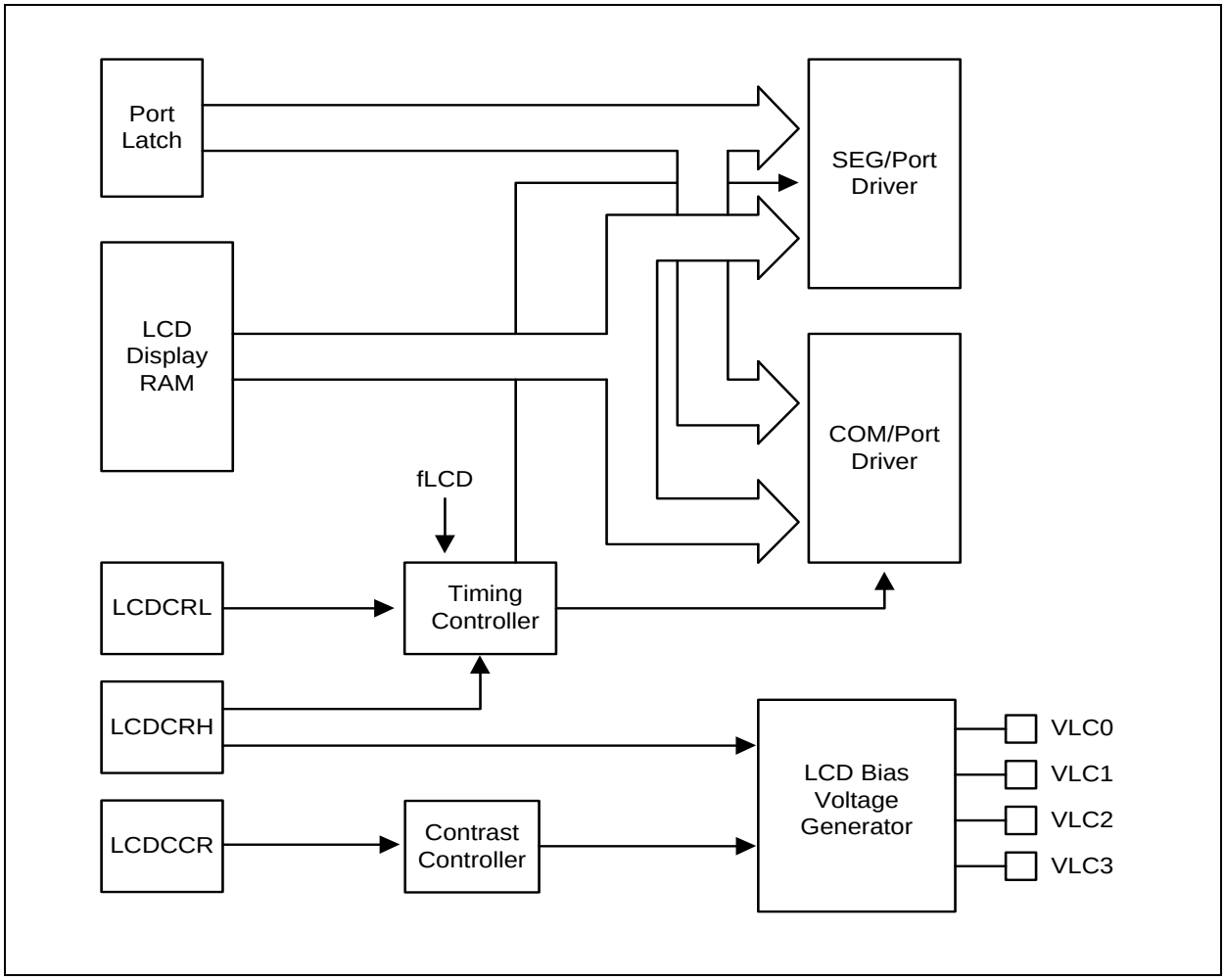


Figure 11.106 LCD Circuit Block Diagram

11.15.6 Register Map

Table 11-26 LCD Register Map

Name	Address	Dir	Default	Description
LCDCRH	9AH	R/W	00H	LCD Driver Control High Register
LCDCRL	99H	R/W	00H	LCD Driver Control Low Register
LCDCCR	9BH	R/W	00H	LCD Contrast Control Register

11.15.7 LCD Driver Register Description

LCD driver register has two control registers, LCD driver control high register (LCDCRH), LCD driver control low register (LCDCRL) and LCD contrast control register.

11.15.8 Register Description for LCD Driver

LCDCRH (LCD Driver Control High Register) : 9AH

7	6	5	4	3	2	1	0
–	–	–	COMCHG	–	–	LCDDR	DISP
–	–	–	R/W	–	–	R/W	R/W

Initial value : 00H

COMCHG Common Signal Output Port Change Control

0 COM0 – COM3 signals are outputted through the P37-P34

1 COM0 – COM3 signals are outputted through the P33-P30

NOTES)

1. The COM0/COM1/COM2/COM3 signals can be outputted through the P33/P32/P31/P30, respectively.
2. For example, the COM0 signal may be outputted to P33 pin if the P3FSR.3 is “1b” and the COMCHG bit is “1b”.
3. Refer to the port3 function selection register (P3FSR).
4. Available only below the 1/4 duty.

LCDDR LCD Driving Resistor for Bias Select

0 Internal LCD driving resistors for bias

1 External lCD driving resistors for bias

DISP LCD Display Control

0 Display off

1 Normal display on

LCDCRL (LCD Driver Control Low Register) : 99H

7	6	5	4	3	2	1	0
–	–	DBS3	DBS2	DBS1	DBS0	LCLK1	LCK0
–	–	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

DBS[3:0] LCD Duty and Bias Select (NOTE)

DBS3	DBS2	DBS1	DBS0	Description
0	0	0	0	1/8Duty, 1/4Bias (60k ohm)
0	0	0	1	1/6Duty, 1/4Bias (60k ohm)
0	0	1	0	1/5Duty, 1/3Bias (60k ohm)
0	0	1	1	1/4Duty, 1/3Bias (60k ohm)
0	1	0	0	1/3Duty, 1/3Bias (60k ohm)
0	1	0	1	1/3Duty, 1/2Bias (60k ohm)
0	1	1	0	1/3Duty, 1/2Bias (120k ohm)
0	1	1	1	1/2Duty, 1/2Bias (60k ohm)
1	0	0	0	1/2Duty, 1/2Bias (120k ohm)
Other values				Not available

LCLK[1:0] LCD Clock Select (When f_{WCK}(Watch timer clock)= 32.768 kHz)

LCLK1	LCLK0	Description
0	0	f_{LCD} = 128Hz
0	1	f_{LCD} = 256Hz
1	0	f_{LCD} = 512Hz
1	1	f_{LCD} = 1024Hz

NOTE) The LCD clock is generated by watch timer clock (f_{WCK}). So the watch timer should be enabled when the LCD display is turned on.

LCDCCR (LCD Driver Contrast Control Low Register) : 9BH

7	6	5	4	3	2	1	0
LCTEN	–	–	–	VLCD3	VLCD	VLCD1	VLCD0
R/W	–	–	–	R/W	R/W	R/W	R/W

Initial value : 00H

LCTEN Control LCD Driver Contrast

0 LCD Driver Contrast disable

1 LCD Driver Contrast enable

VLCD[3:0] VLC0 Voltage Control when the contrast is enabled

VLCD3	VLCD 2	VLCD 1	VLCD 0	Description
0	0	0	0	VLC0 = VDD x 16/31 step
0	0	0	1	VLC0 = VDD x 16/30 step
0	0	1	0	VLC0 = VDD x 16/29 step
0	0	1	1	VLC0 = VDD x 16/28 step
0	1	0	0	VLC0 = VDD x 16/27 step
0	1	0	1	VLC0 = VDD x 16/26 step
0	1	1	0	VLC0 = VDD x 16/25 step
0	1	1	1	VLC0 = VDD x 16/24 step
1	0	0	0	VLC0 = VDD x 16/23 step
1	0	0	1	VLC0 = VDD x 16/22 step
1	0	1	0	VLC0 = VDD x 16/21 step
1	0	1	1	VLC0 = VDD x 16/20 step
1	1	0	0	VLC0 = VDD x 16/19 step
1	1	0	1	VLC0 = VDD x 16/18 step
1	1	1	0	VLC0 = VDD x 16/17 step
1	1	1	1	VLC0 = VDD x 16/16 step

NOTES) The LCD contrast step is based on 1/4 bias.
1/4 bias : VDD x (16/31 – VLC[3:0])
1/3 bias : VDD x (12/27 – VLC[3:0])
1/2 bias : VDD x (8/23 – VLC[3:0])

12. Power Down Operation

12.1 Overview

The MC96F6432/F6332/F6232 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)
Timer0~4	Operates Continuously	Halted (Only when the Event Counter Mode is Enabled, Timer operates Normally)
ADC	Operates Continuously	Stop
BUZ	Operates Continuously	Stop
SPI	Operates Continuously	Only operate with external clock
USI0/1	Operates Continuously	Only operate with external clock
LCD Controller	Operates Continuously	Stop (Can be operated with sub clock)
Internal OSC (16MHz)	Oscillation	Stop when the system clock (fx) is f_{IRC}
WDTRC OSC (6kHz)	Stop	Can be operated with setting value
Main OSC (0.4~12MHz)	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, EC1, EC3), SPI (External clock), External Interrupt, UART by ACK, 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.

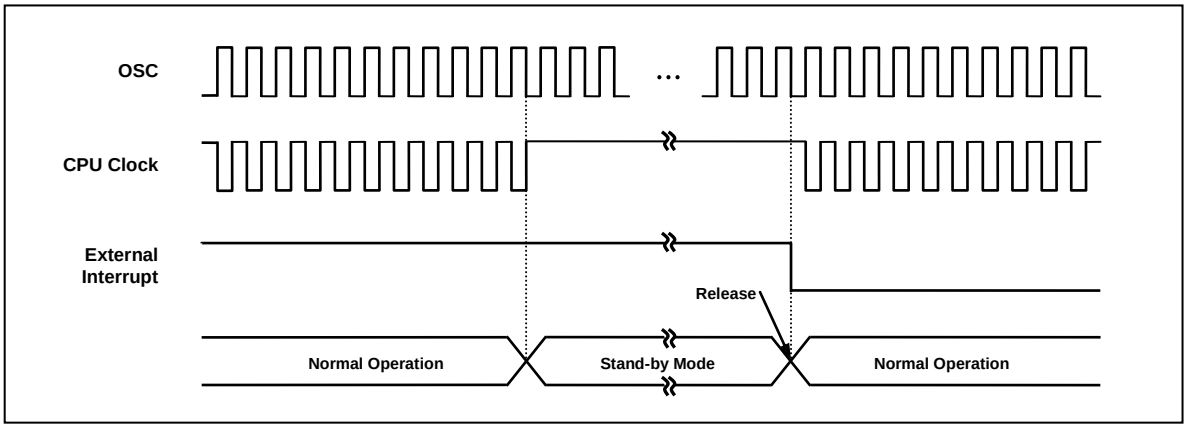


Figure 12.1 IDLE Mode Release Timing by External Interrupt

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 and LCD controller 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.

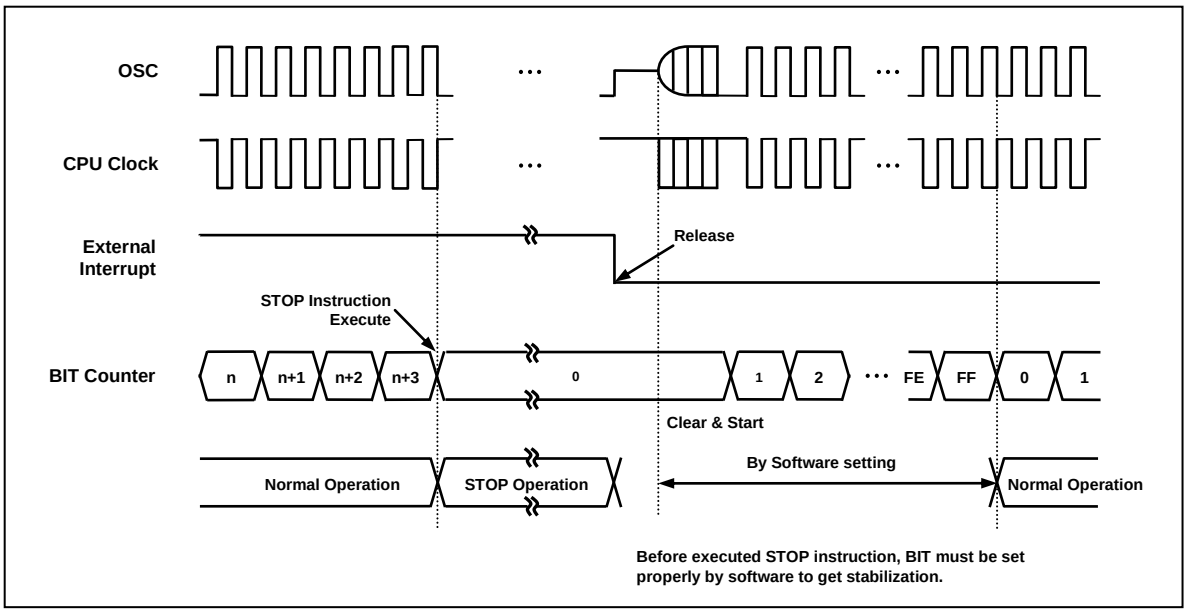


Figure 12.2 STOP Mode Release Timing by External Interrupt

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'.

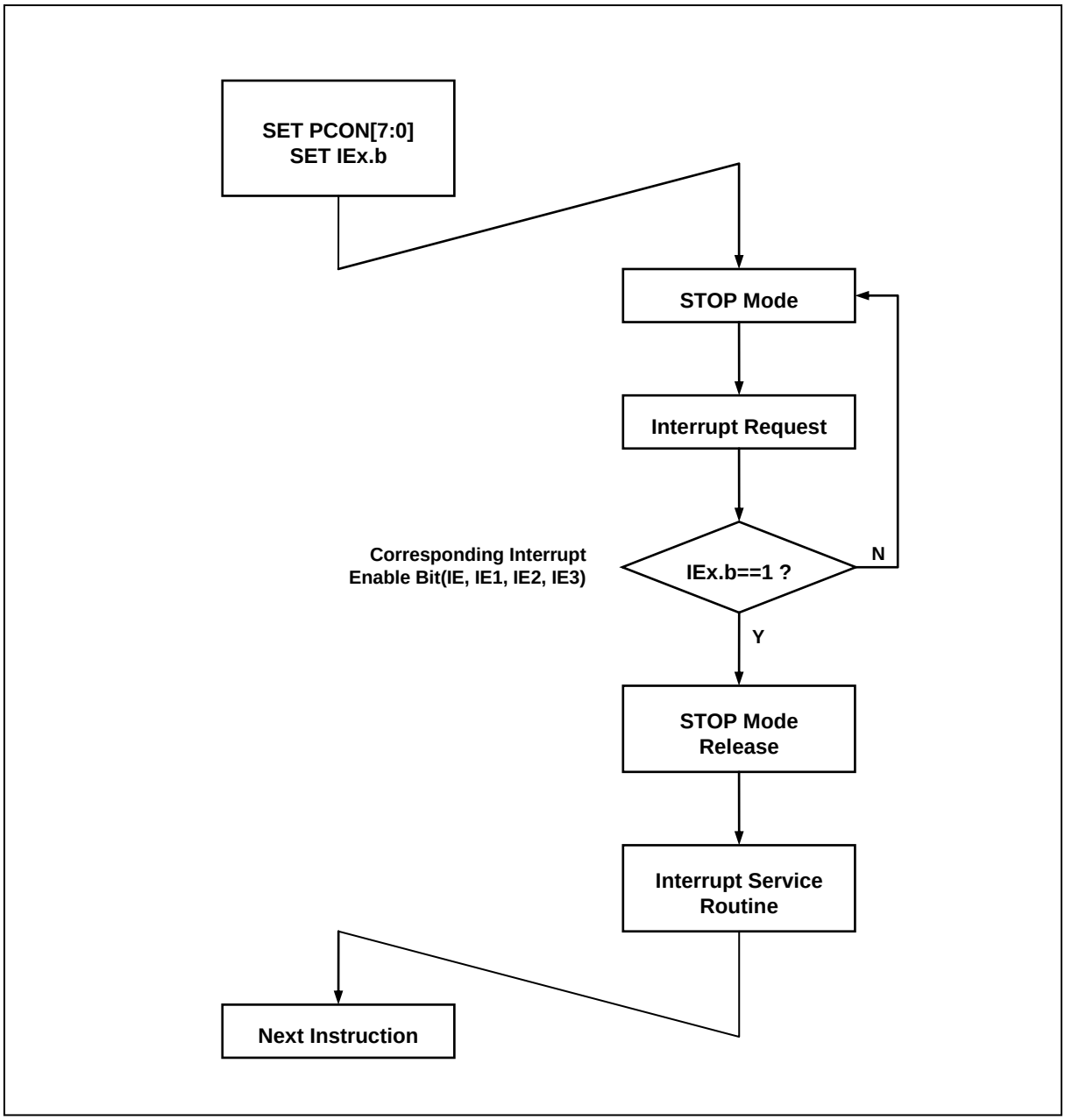


Figure 12.3 STOP Mode Release Flow

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 the instruction that make the device enter STOP/IDLE mode. Refer to the following 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 MC96F6432/F6332/F6232 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

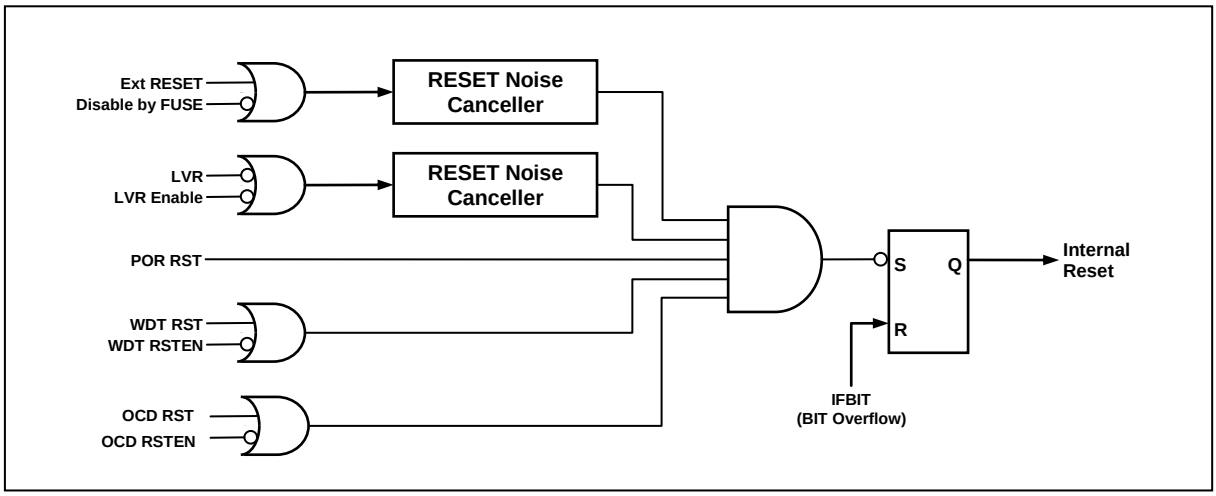


Figure 13.1 RESET Block Diagram

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.

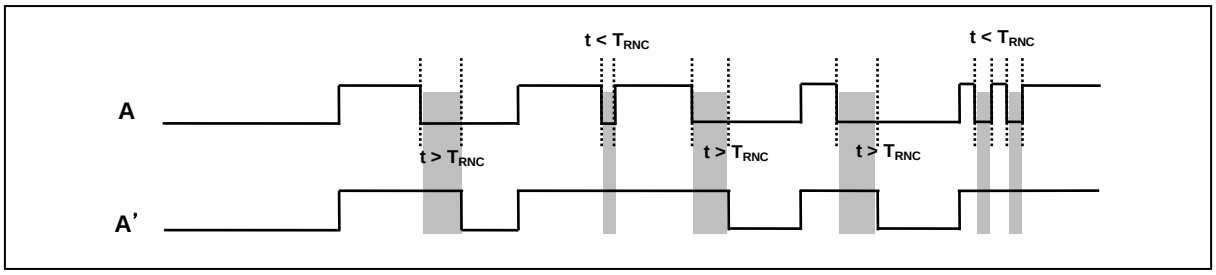


Figure 13.2 Reset noise canceller timer diagram

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.

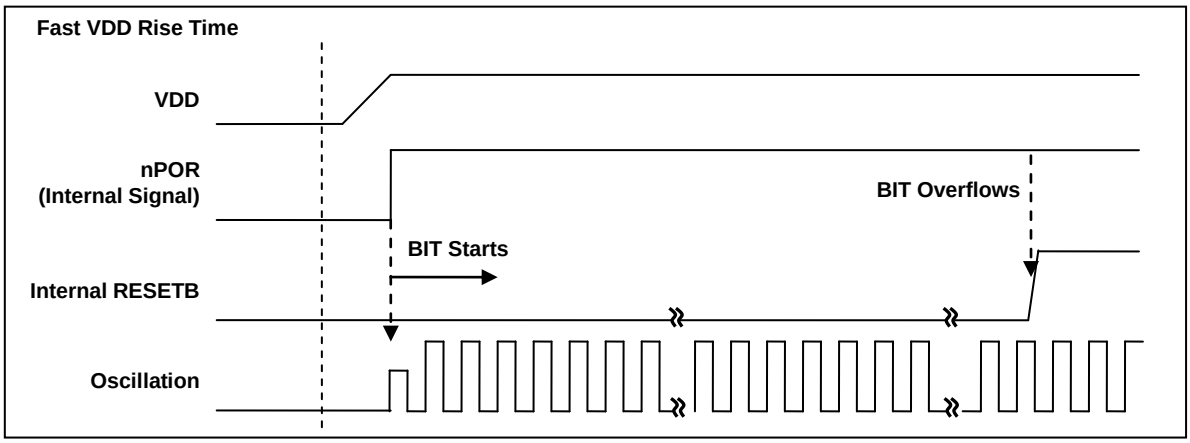


Figure 13.3 Fast VDD Rising Time

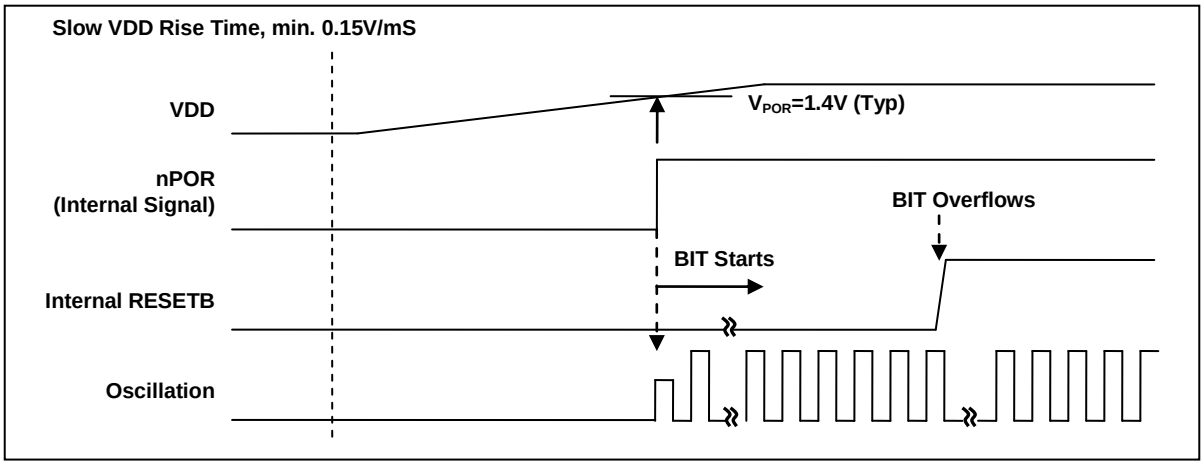


Figure 13.4 Internal RESET Release Timing On Power-Up

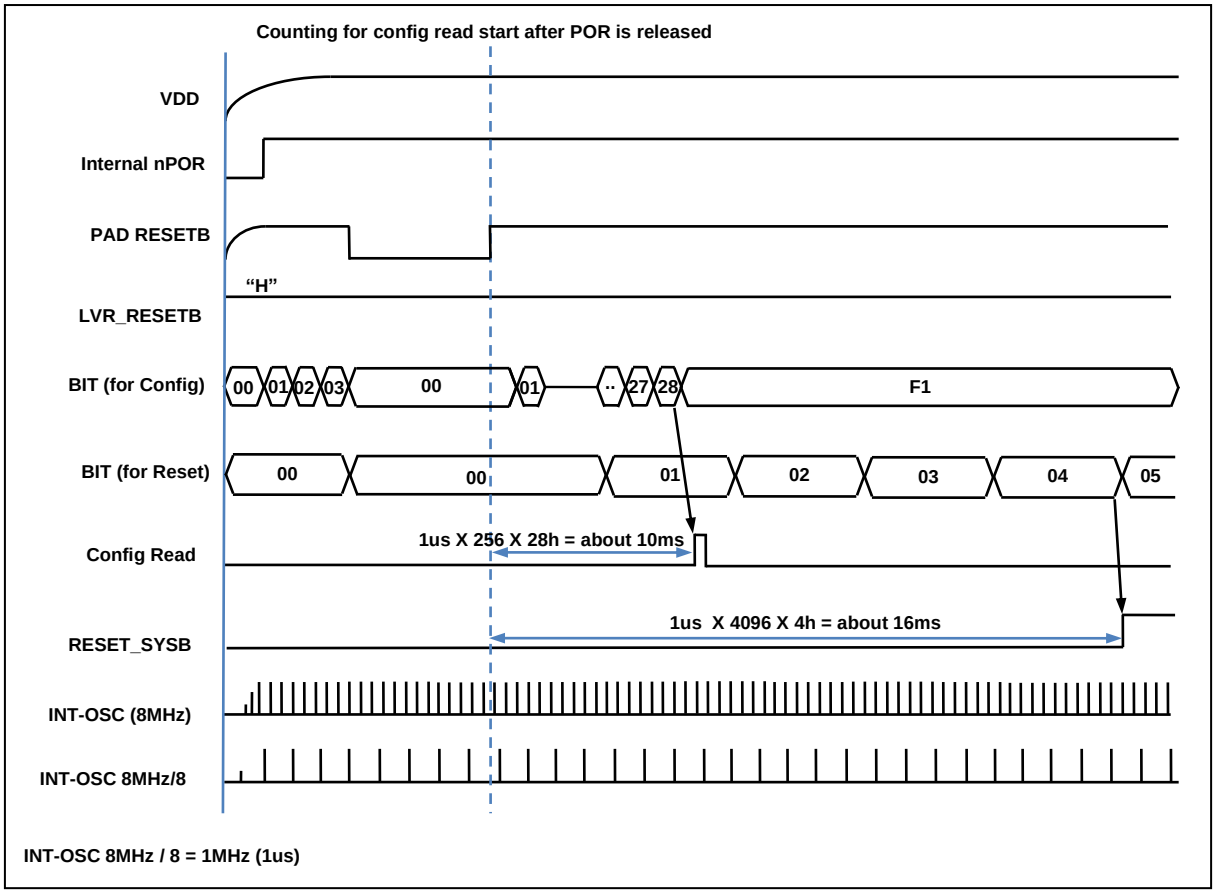


Figure 13.5 Configuration Timing when Power-on

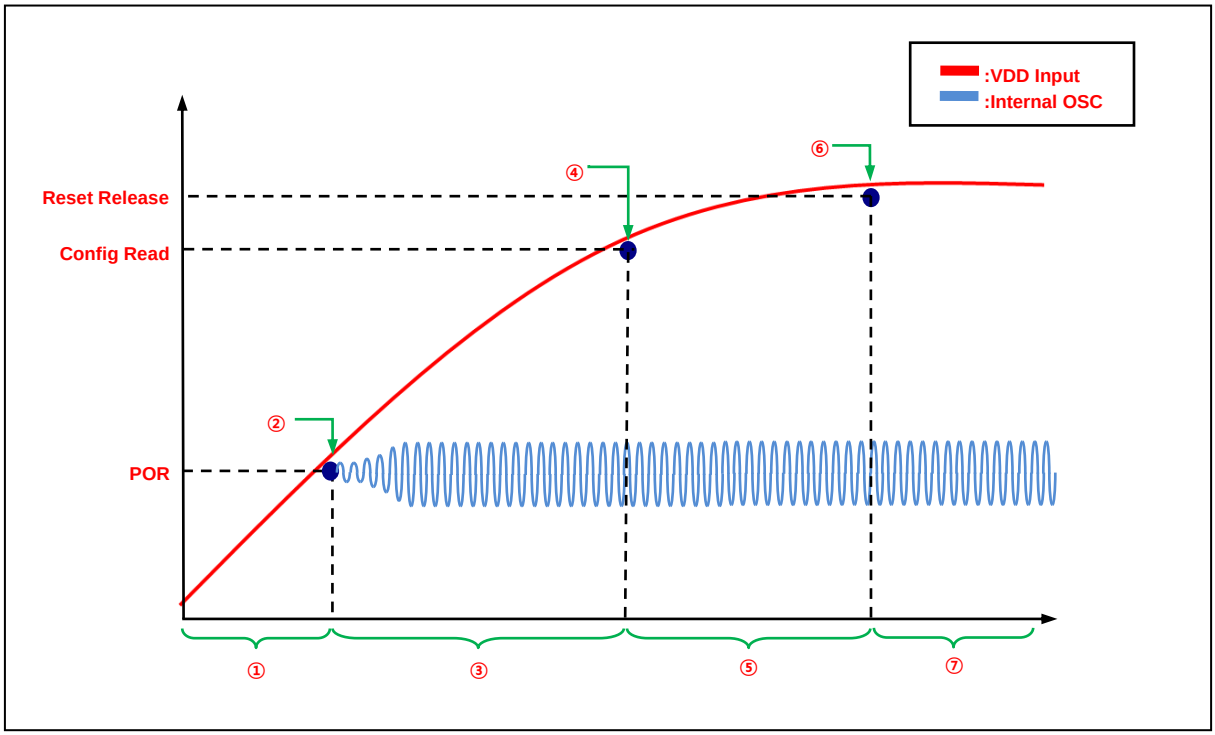


Figure 13.6 Boot Process WaveForm

Table 13-2 Boot Process Description

Process	Description	Remarks
①	-No Operation	
②	-1st POR level Detection	-about 1.4V
③	- (INT-OSC 8MHz/8)x256x28h Delay section (=10ms) -VDD input voltage must rise over than flash operating voltage for Config read	-Slew Rate >= 0.15V/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.

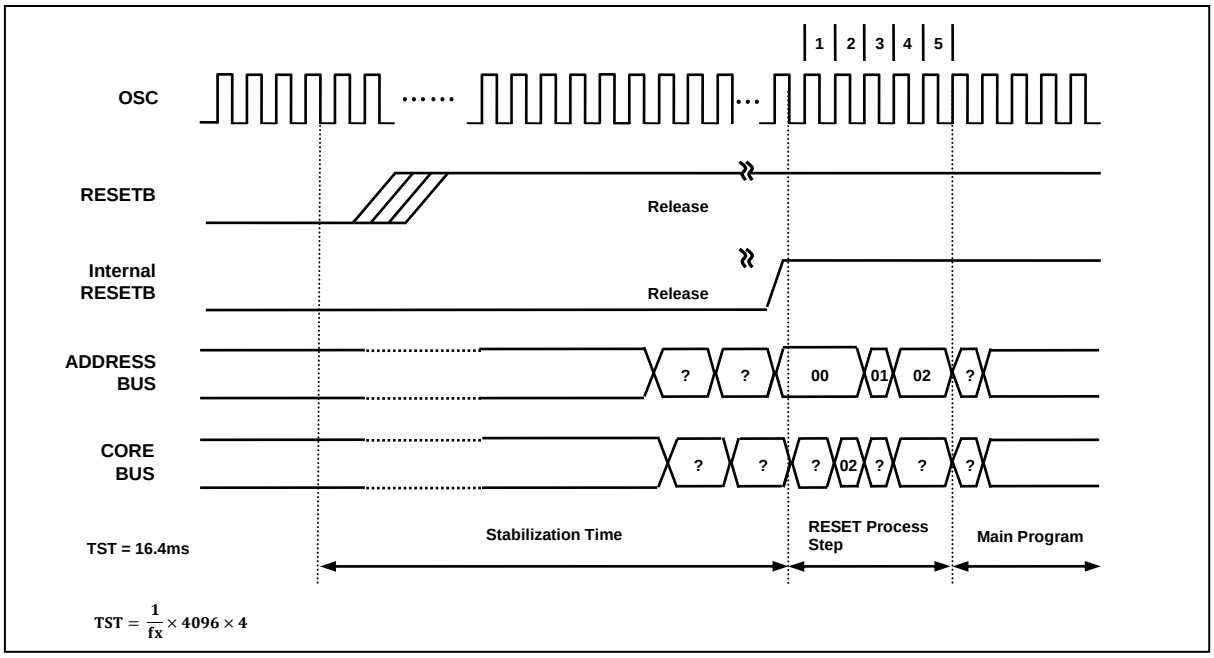


Figure 13.7 Timing Diagram after RESET

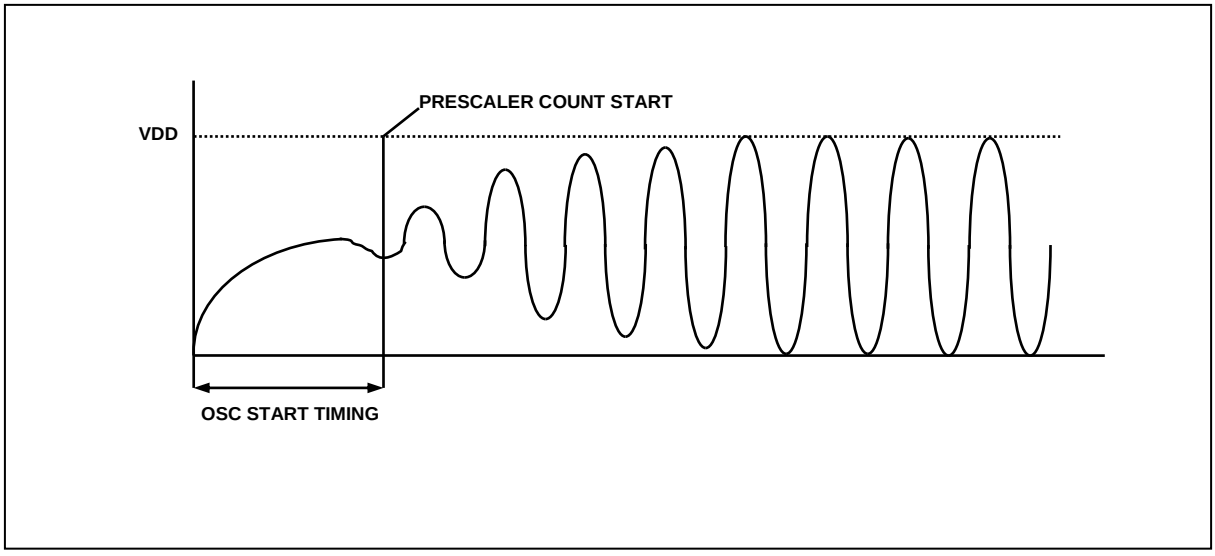


Figure 13.8 Oscillator generating waveform example

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 MC96F6432/F6332/F6232 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.

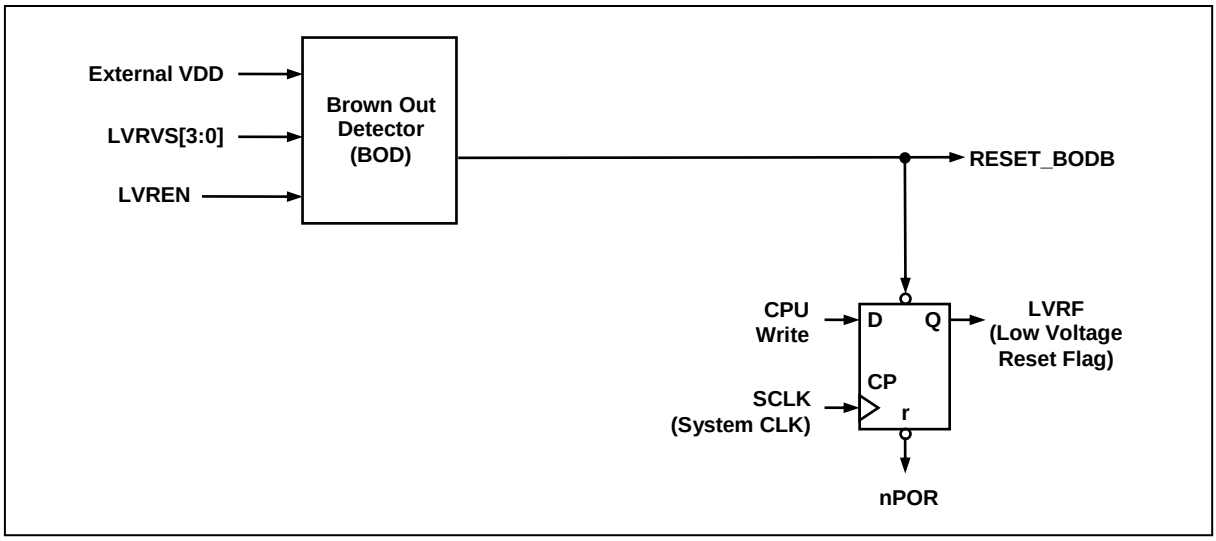


Figure 13.9 Block Diagram of BOD

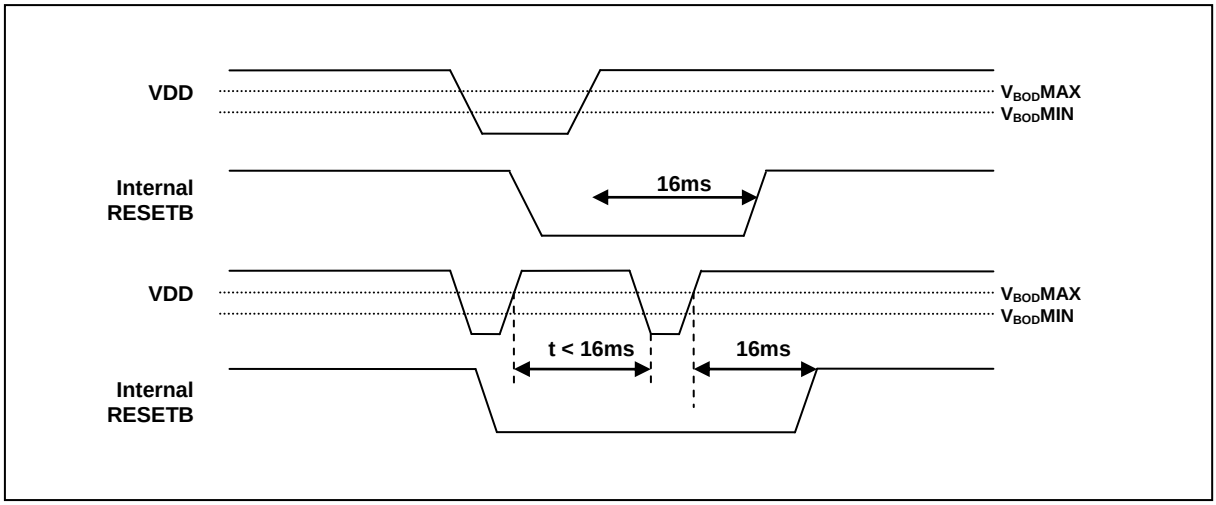


Figure 13.10 Internal Reset at the power fail situation

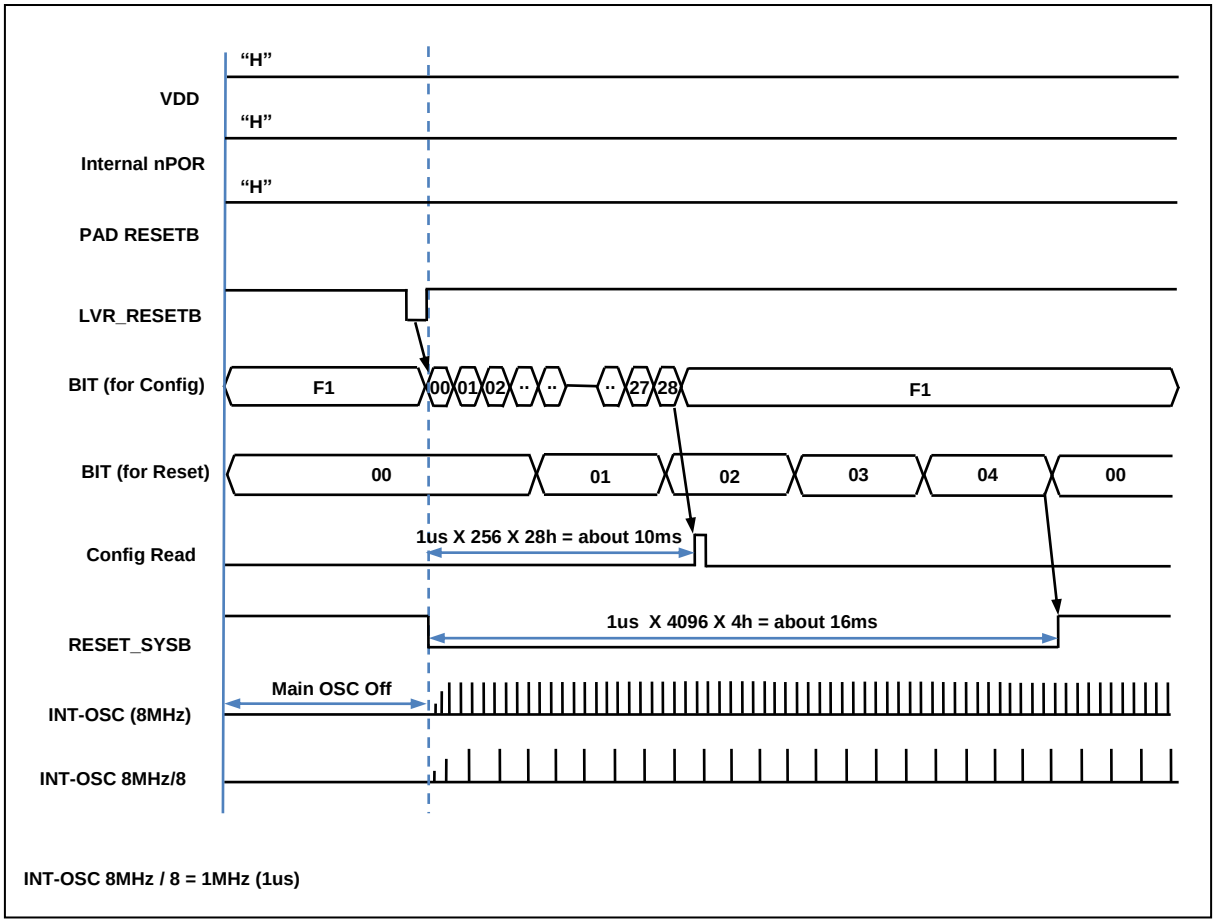


Figure 13.11 Configuration timing when BOD RESET

13.8 LVI Block Diagram

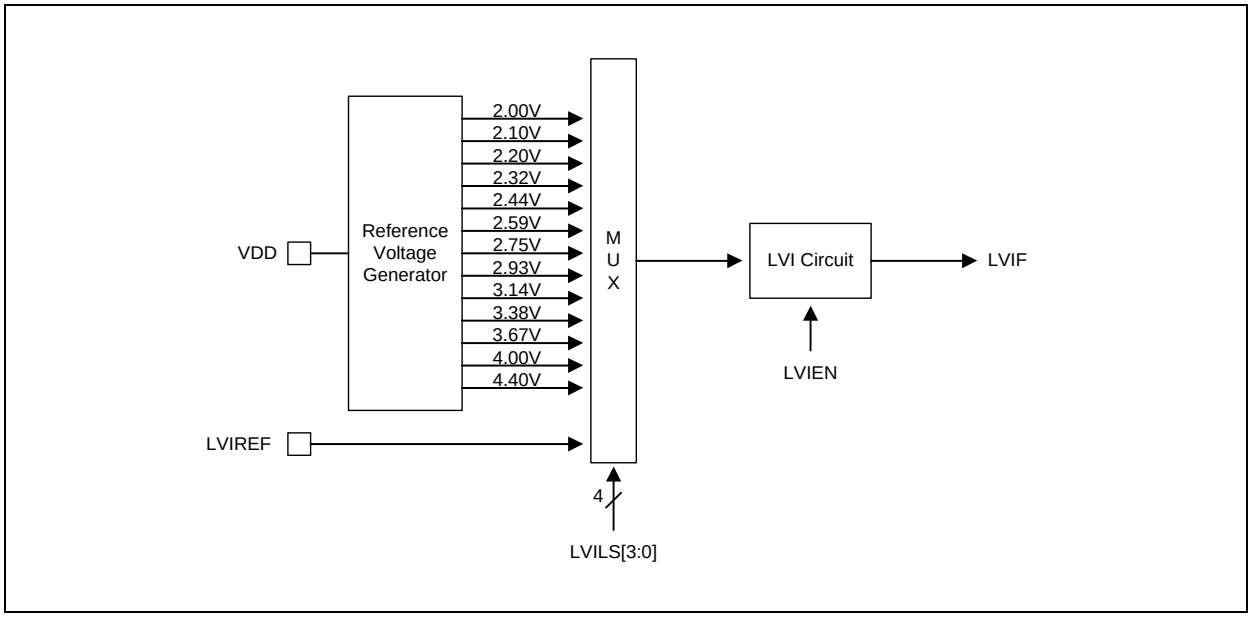


Figure 13.12 LVI Diagram

13.8.1 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

13.8.2 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).

13.8.3 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
1	1	1	0	Not available
1	1	1	1	Not available

LVREN LVR Operation

0 LVR Enable

1 LVR Disable

NOTE) The LVRVS[3:0] and LVREN bits are not retained at a power-on reset but are retained at the other reset signals.

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 (OCD) of MC96F6432/F6332/F6232 can be used for programming the non-volatile memories and on-chip debugging. Detail descriptions for programming via the OCD interface can be found in the following chapter.

Figure 14.1 shows a block diagram of the OCD 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

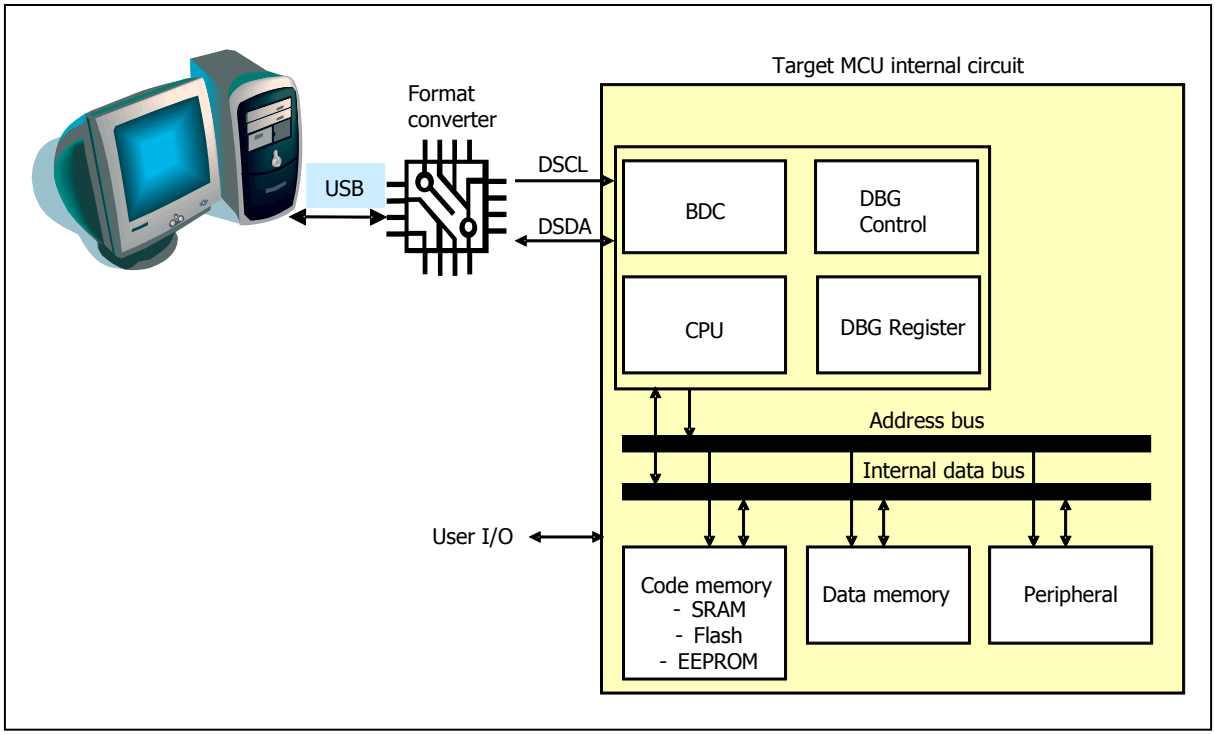


Figure 14.1 Block Diagram of On-Chip Debug System

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.

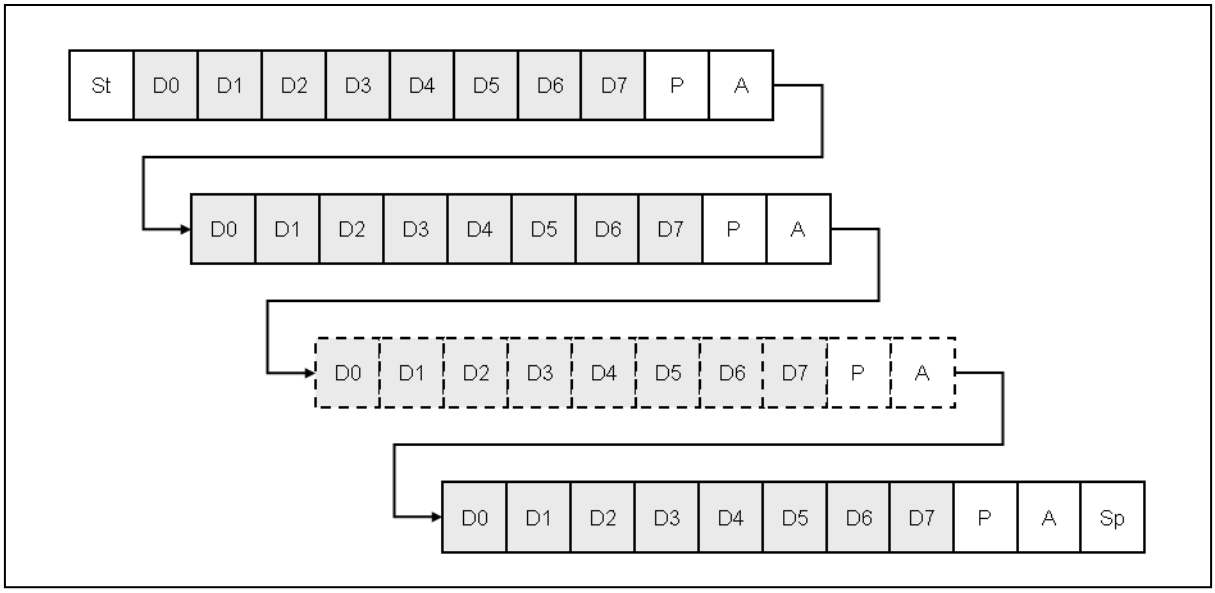


Figure 14.2 10-bit Transmission Packet

14.2.2 Packet Transmission Timing

14.2.2.1 Data Transfer

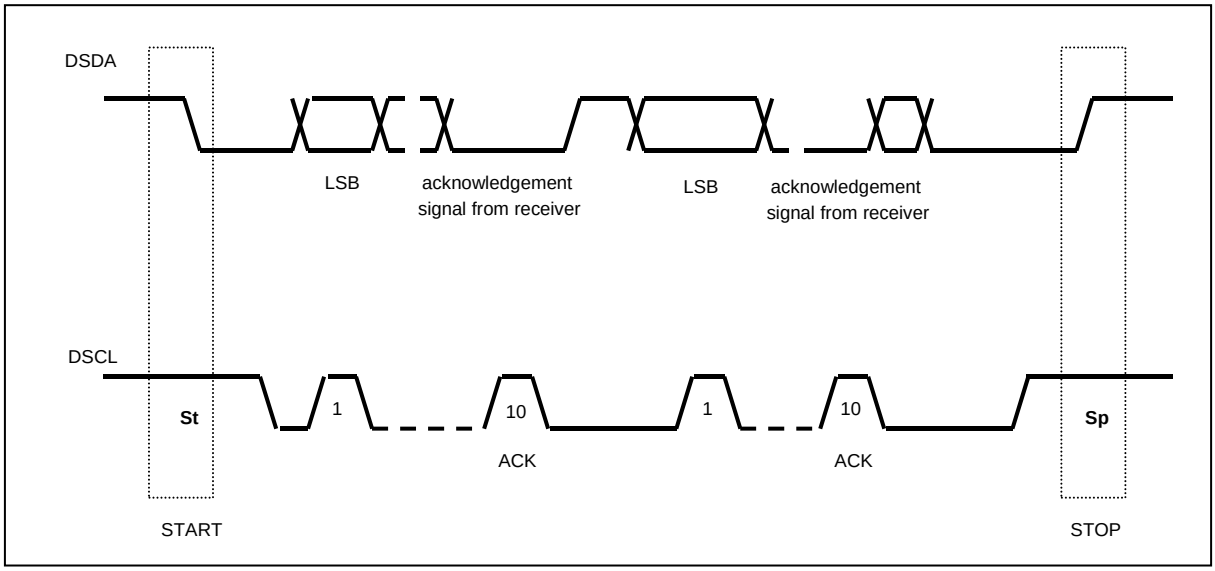


Figure 14.3 Data Transfer on the Twin Bus

14.2.2.2 Bit Transfer

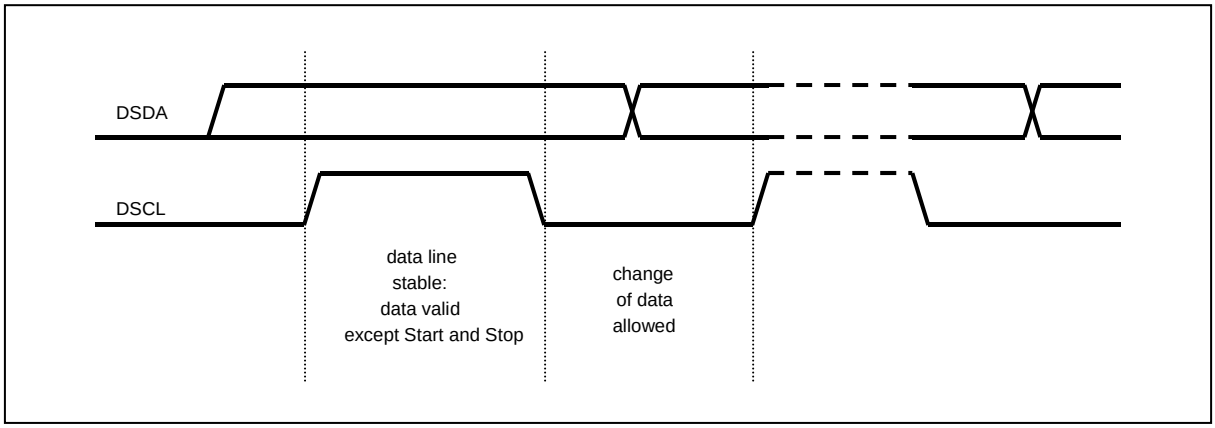


Figure 14.4 Bit Transfer on the Serial Bus

14.2.2.3 Start and Stop Condition

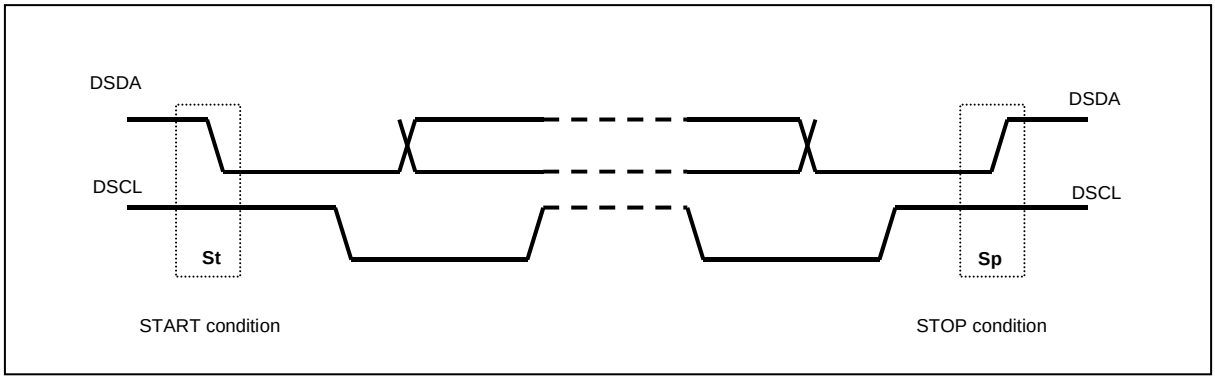


Figure 14.5 Start and Stop Condition

14.2.2.4 Acknowledge Bit

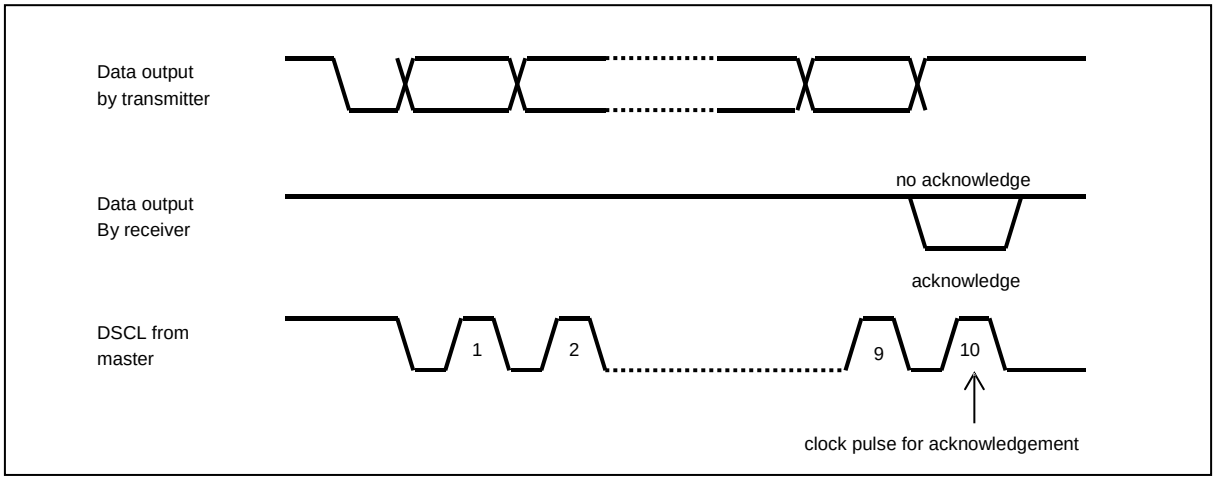


Figure 14.6 Acknowledge on the Serial Bus

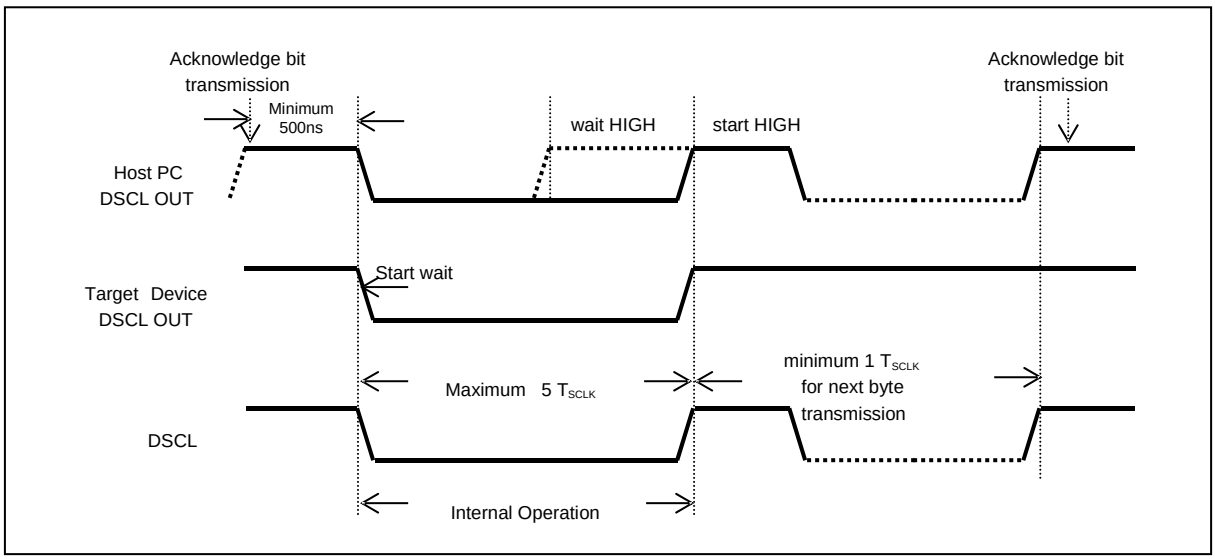


Figure 14.7 Clock Synchronization during Wait Procedure

14.2.3 Connection of Transmission

Two-pin interface connection uses open-drain (wire-AND bidirectional I/O).

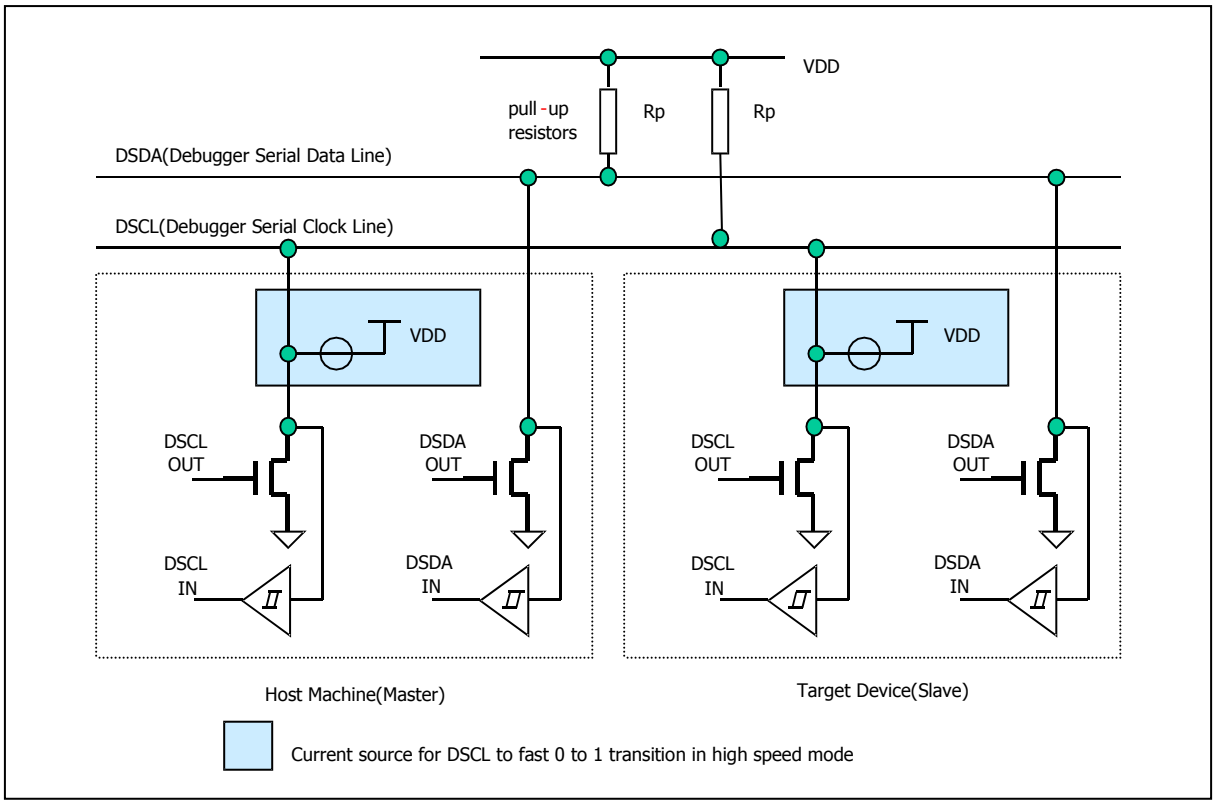


Figure 14.8 Connection of Transmission

15. Flash Memory

15.1 Overview

15.1.1 Description

MC96F6432/F6332/F6232 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 OCD, serial ISP mode or user program mode.

- Flash Size : 32kbytes
- Single power supply program and erase
- Command interface for fast program and erase operation
- Up to 100,000 program/erase cycles at typical voltage and temperature for flash memory

15.1.2 Flash Program ROM Structure

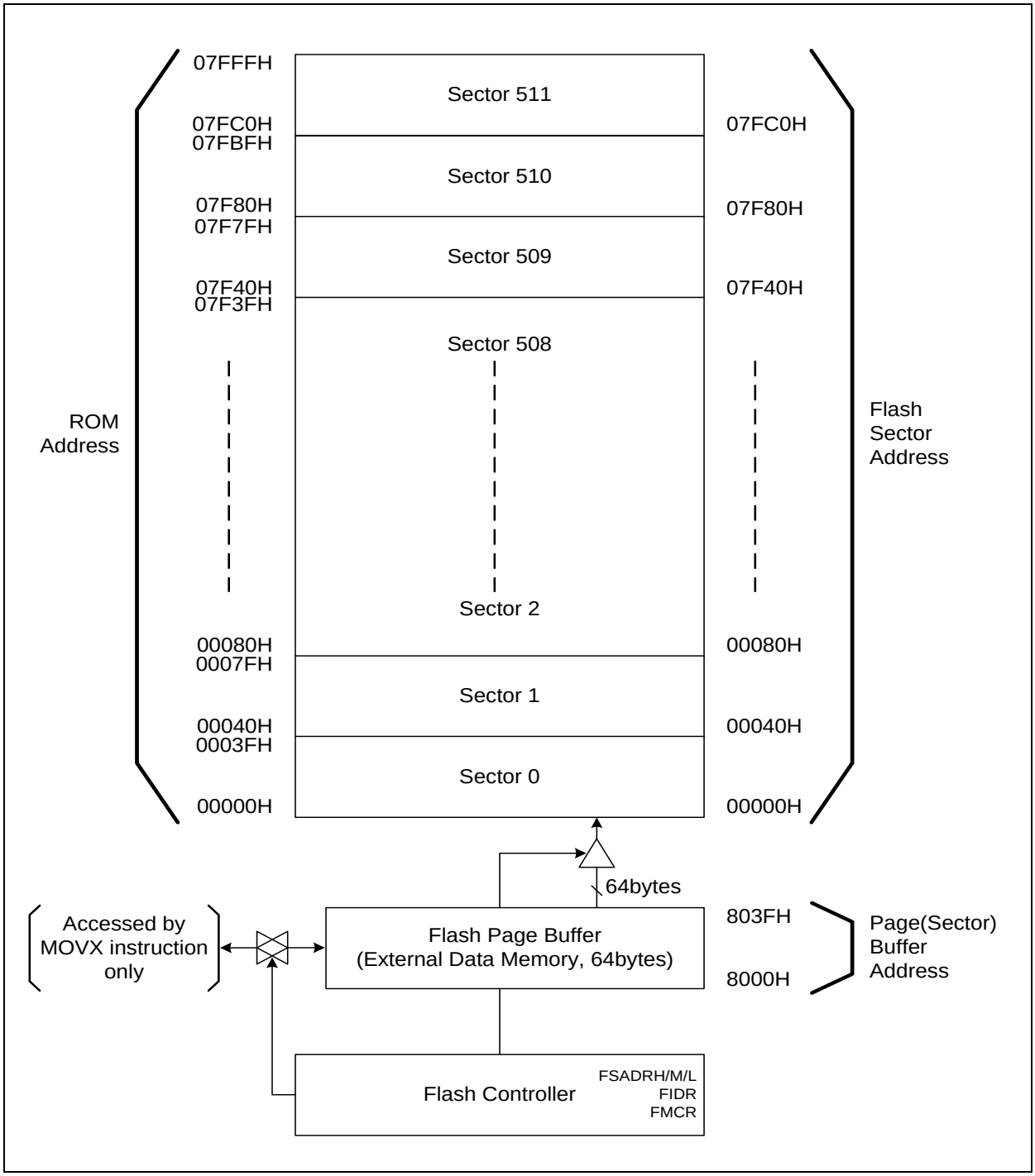


Figure 15.1 Flash Program ROM Structure

15.1.3 Register Map

Table 15-1Flash 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)

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 64bytes 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 hard lock 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)

MC96F6432/F6332/F6232 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. Erase verify

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. Erase verify

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 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
```

15.1.11 Hard Lock Mode

The Reading program procedure in user program mode

1. Set flash identification register (FIDR).
2. Set flash mode control register (FMCR).

Program Tip – reading

```
MOV     FIDR,#0xA5          ;Identification value
MOV     FMCR,#0x04          ;Start flash hard lock 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	–	–	–	–	–	RSTS

Initial value : 00H

R_P Read Protection
- 0 Disable “Read protection”
- 1 Enable “Read protection”

HL Hard-Lock
- 0 Disable “Hard-lock”
- 1 Enable “Hard-lock”

RSTS RESETB Select
- 0 P55 port
- 1 RESETB port with a pull-up resistor

CONFIGURE OPTION 2: ROM Address 003EH

7	6	5	4	3	2	1	0
–	–	–	–	–	PAEN	PASS1	PASS0

Initial value : 00H

PAEN Protection Area Enable/Disable
- 0 Disable Protection (Erasable by instruction)
- 1 Enable Protection (Not erasable by instruction)

PASS [1:0] Protection Area Size Select

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	3	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.