CMOS single-chip 8-bit MCU with 12-bit ADC, LDO and Comparator

A94B114

User's manual

V 1.09

Main features

- **Enhanced 8051 microcontroller with reduced instruction cycle time** (CM8051-S CPU)
- **Basic MCU Function**
  - 8 Kbytes Flash Code Memory
  - Code Area Protection
  - 512 bytes SRAM Data Memory
- **Built-in Analog Function**
  - Power-On Reset and Low Voltage Detect Reset
  - Internal HFO (32MHz TYP ± 1.5%, T_A= +25°C)
  - Internal LFO (256kHz ± 10%, T_A= -40 ~ +85°C@Normal/IDLE Mode)
  - Internal LFO (168kHz ± 15%, T_A= +25°C @STOP Mode)
  - Internal LFO (168kHz ± 20%, T_A= -40 ~ +85°C@STOP Mode)
  - Watchdog Timer RC Oscillator (LFO/2 =128kHz)
- **Peripheral Features**
  - 12-bit Analog to Digital Converter with 2.5V LDO
  - 16-bit PWM with inverters and dead-band control
  - USART 8-bit x 1-ch
  - I2C 8-bit x 1-ch
- **I/O and Packages**
  - Up to 18 Programmable I/O lines with 20-SOP
  - 20-SOP, 20-TSSOP, 16-SOPN
- **Operating Conditions**
  - 2.0V to 5.5V Wide Voltage Range
  - - 40°C to 85°C Temperature Range
- **Application**
  - Small Home Appliance
  - Battery charge & discharge control

Revised 17 July, 2018

Revision history

Version	Date	Revision list
1.00	2017.07.31	Primary Version.
1.01	2017.08.02	Added content of SYSCON_AR register. (11.1.5) Added note of CRC Polynomial. (11.11.4)
1.02	2017.08.15	Fixed typos.
1.03	2017.09.13	Sector 248~255 Endurance added. (7.15.1) Delete the LVIR reset function. (except LVR 1.75V)
1.04	2017.10.24	Added description of PUSH SP, POP SP, DA incompatible instruction. (17.1 APPENDIX)
1.05	2017.11.08	Update Table 7.4 Analog ADC Voltage
1.06	2018.01.22	Revised this book. Add Device Nomenclature Update Internal High Frequency Oscillator spec. Update Figure 10.8 & Page 193. Fuse_CFG1
1.07	2018.02.26	Revised this book. Added Page 56 ILVL description
1.08	2018.05.11	Added Note of EINT0~2 operation (10.2), IRQ0~2 operation(10.12.3).
1.09	2018.07.17	The LFO Spec is separated from Normal / IDLE mode and STOP mode.(7.8) Added operation description in STOP mode of BIT. (11.2.1) Added WDT precautions when using LFO clock. (11.3.1) Fixed Peripheral Operation of STOP mode. (12.2)

Version 1.09

Published by AE 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.

Contact information

For more information, please visit : http://www.abov.co.kr
E-mail : fae@abov.co.kr

1 Overview

1.1. Description

The A94B114 is an advanced CMOS 8-bit microcontroller with 8 Kbytes 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 : 8 Kbytes of FLASH, 512 bytes of SRAM, 16-bit timer/counter, 16-bit PWM with inverters and dead-band control, Watchdog timer, 12-bit ADC with LDO, On-chip POR, LVI and LVR, Internal High Frequency Oscillator, Internal Low Frequency Oscillator and clock circuitry. The A94B114 also supports Power saving modes to reduce Power Consumption.

Device Name	FLASH	XRAM	IRAM	ADC	I/O PORT	Package
A94B114FR	8 Kbytes FLASH	256 bytes	256 bytes	10 inputs	18	20-TSSOP
A94B114FD				10 inputs	18	20-SOP
A94B114AE				8 inputs	14	16-SOPN

Table 1.1 Ordering Information of A94B114

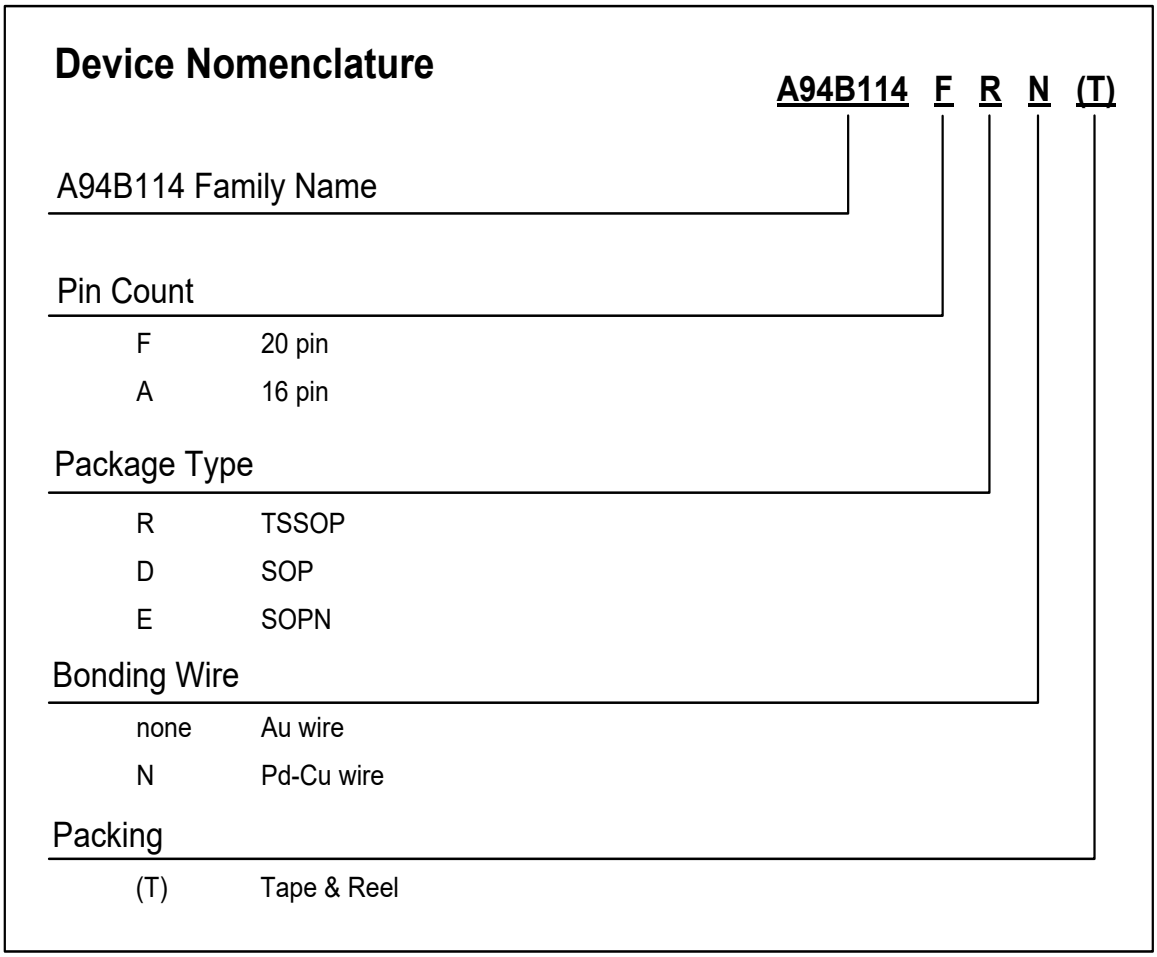


Figure 1.1 Device Nomenclature

1.2 Features

- **CPU**
  - 8-bit CISC CM8051-S core (8051 Compatible, 1 clocks per cycle)
- **8 Kbytes On-Chip FLASH**
  - Endurance : 10,000 times (Sector 0~247) 50,000 timers (Sector 248~255)
  - Retention : 10 years
  - In-System Programming (ISP)
- **256 bytes IRAM / 256 bytes XRAM**
- **General Purpose I/O (GPIO)**
  - Normal I/O : 18 Port (P0[7:0], P1[5:0], P2[3:0])
- **Timer/Counter**
  - 8-bit Timer x 1-ch (T0)
  - 16-bit Timer x 2-ch (T1, T2)
- **Programmable Pulse Generation**
  - Pulse generation (by T1/T2)
  - 16-bit PWM with inverters and Dead-band control x 1ch (by T1)
- **Comparator**
  - Selectable internal reference Comparator.
  - External (CMN0, CMN1)
  - Internal (1.6V, 2.2V)
- **Watch Timer (WDT)**
  - 8-bit x 1-ch
  - RCWDT with LFO
- **USART**
  - 8-bit× 1-ch
- **I2C**
  - 8-bit× 1-ch
- **12-bit A/D Converter**
  - 10 Input channels
- **Power Down Mode**
  - IDLE, STOP mode
- **Sub-Active mode**
  - System used by internal 128kHz Ring oscillator (LFO/2)
- **Power On Reset**
  - Reset release level (1.4V)
- **Low Voltage Reset**
  - 1 level detect (1.75)
- **Low Voltage Indicator**
  - 4 level detect (2.40/ 2.90/ 3.90/ 4.20V)
- **Interrupt Sources**
  - External Interrupts (EINT0~2, EINT10, EINT11, EINT12) (4)
  - Timer(0/1/2) (3)
  - WDT (1)
  - LVI (1)
  - USART (2)
  - I2C (1)
  - ADC (1)
  - BIT (1)
  - CRC (1)
  - Comparator (1)
- **Internal Oscillator**
  - Internal High Frequency Oscillator : 32MHz TYP ±1.5% (T_A= 25°C)
  - Internal Low Frequency Oscillator : 256kHz ±10% (T_A= -40~+85°C, Normal/IDLE) 168kHz ±15% (T_A= +25°C, STOP) 168kHz ±20% (T_A= -40~+85°C, STOP)
- **Operating Voltage and Frequency**
  - 1MHz ~ 16MHz
    - Internal Clock : T_A= -40~+85°C, V_{DD}=2.0V ~ 5.5V
    - External Crystal : T_A= -40~+85°C, V_{DD}=2.7V ~ 5.5V
- **Oscillator Type**
  - 0.4~16 MHz Crystal or Ceramic for main clock
- **Minimum Instruction Execution Time**
  - 62.5ns (@16MHz main clock)
- **Operating Temperature**
  - -40~ +85℃
- **Operating Frequency**
  - 1 / 4 / 8 / 16MHz (with HFO/2)
  - 128kHz (with LFO/2)
- **Package Type**
  - 20-SOP
  - 20-TSSOP
  - 16-SOPN

1.3 Development tools

1.3.1 Compiler

ABOV semiconductor does not provide any compiler for the A94B114. But the CPU core of A94B114 is CM8051-S core, you can use all kinds of third party's standard 8051 compiler like Keil C Compiler, Open Source SDCC (Small Device C Compiler). These compilers' output debug information can be integrated with our OCD2 emulator and debugger. Refer to OCD2 manual for more details.

1.3.2 OCD2 emulator and debugger

The OCD2 emulator supports ABOV Semiconductor's 8051 series MCU emulation.

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

The OCD2 Debugger program works on Microsoft-Windows NT, 2000, XP, Vista, 7, 8, 8.1, 10 (32-bit, 64-bit) operating system.

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

There are OCD2 mode connection.

- P01 (A94B114 DSCL pin)
- P00 (A94B114 DSDA pin)

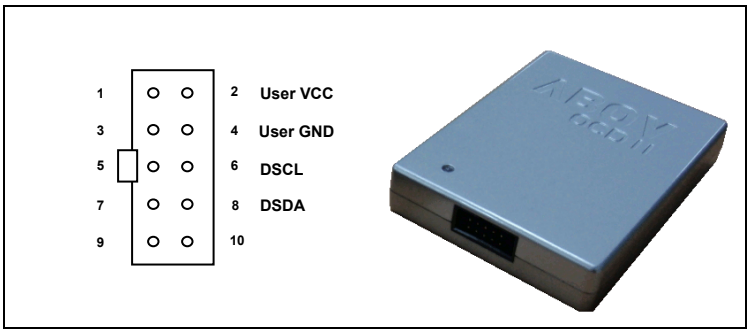


Figure 1.2 On Chip Debugger 2 and Pin description (OCD2 mode)

OCD2 (On Chip Debug) Emulator

- MCU emulation control via 2pin OCD interface.
- 2pin interface : OCD2 clock & data.
- Higher interface speed than OCD dongle.
- Support newly added debugging specifications.
- Compact size.
- Cost effective emulator.
- Emulation & debugging on the target system directly.
- Real time emulation.
- PC interface : USB.

Debugger

- Operates with OCD2 emulator H/W.
- Integrated Development Environment (IDE).
  Support docking windows and menus.
- Support Free run, Step run, auto step run.
- Support Symbolic debugging.
- Support Source level debugging.

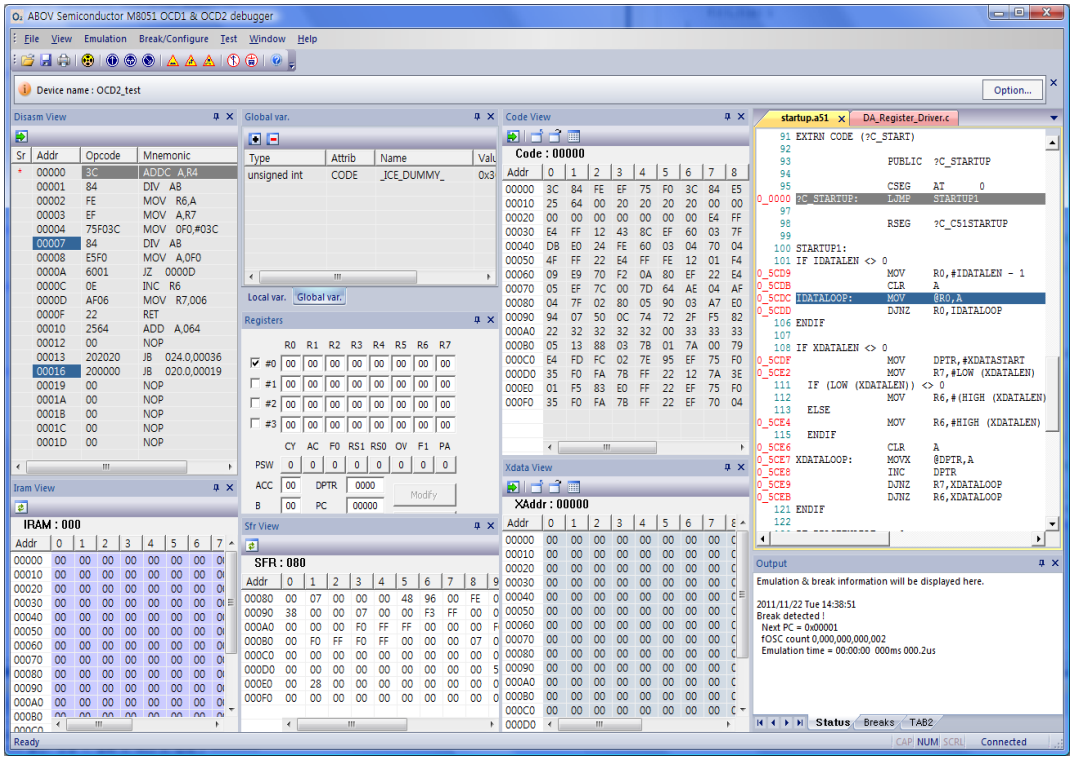


Figure 1.3 OCD2 Debugger

1.3.3 Programmer

E-PGM +

- Support ABOV / ADAM devices
- 2~5 times faster than S-PGM+
- Main controller : 32-bit MCU @ 72MHz
- Buffer memory : 1MB

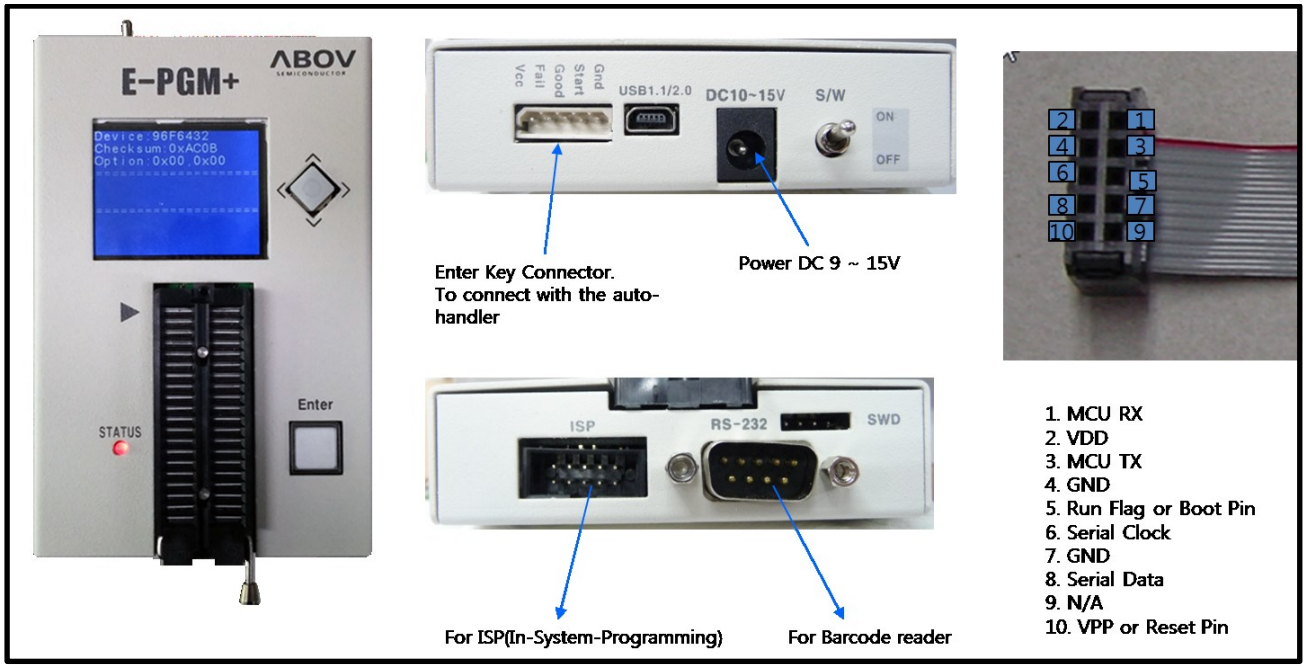


Figure 1.4 E-PGM+ component and connector

PGMPlusLC2

Description

PGMPlusLC2 is for ISP(In System Programming). It is used to write into the MCU Which is already mounted on target board using 10pin cable.

Features

- PGMplusLC2 is low cost writing Tool.
- USB interface is supported.
- Not need USB driver installation.
- Connect the external power adaptor(5V@2A).
- Supported high voltage Max 18V.
- PGMplusLC2 is based on PC environment.
- PGMplusLC2 is faster than PGMplusLC.
- Transmission speed is 64KB/s

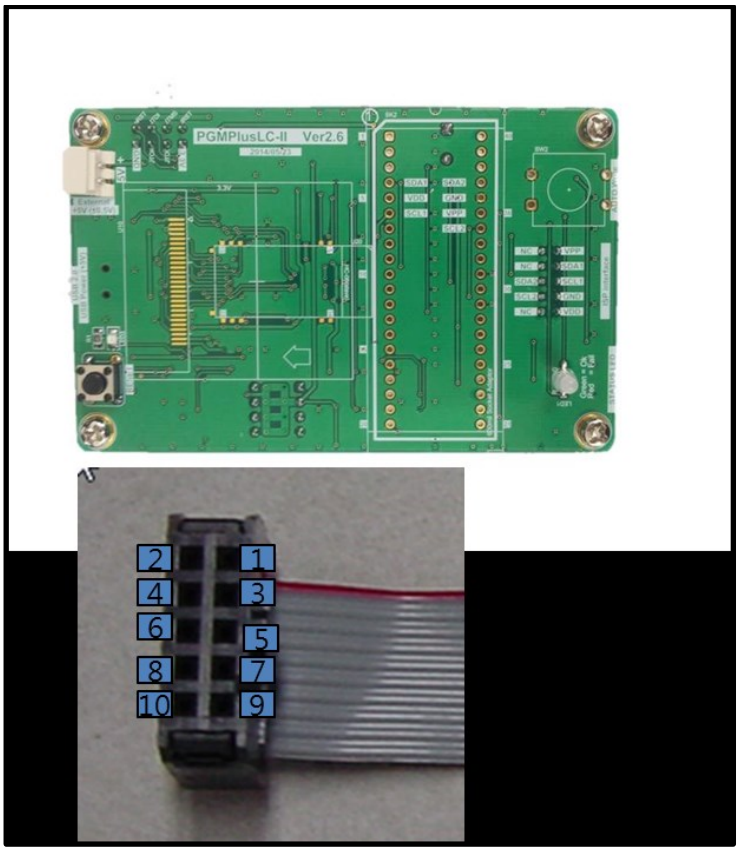


Figure 1.5 PGMplusLC Writer

E-PGM+ Gang4/6

- Product name : **E-PGM+ GANG 4**
- Dimension(x, y, h) : 33.5 x 22.5 x35mm
- Weight : 2.0kg
- Input Voltage : DC Adaptor 15V/2A
- Operating Temp : -10 ~ 40℃
- Storage Temp : -30 ~ 80℃
- Water Proof : No

- Product name : **E-PGM+ GANG 6**
- Dimension(x, y, h) : 148.2 x 22.5 x35mm
- Weight : 2.8kg
- Input Voltage : DC Adaptor 15V/2A
- Operating Temp : -10 ~ 40℃
- Storage Temp : -30 ~ 80℃
- Water Proof : No

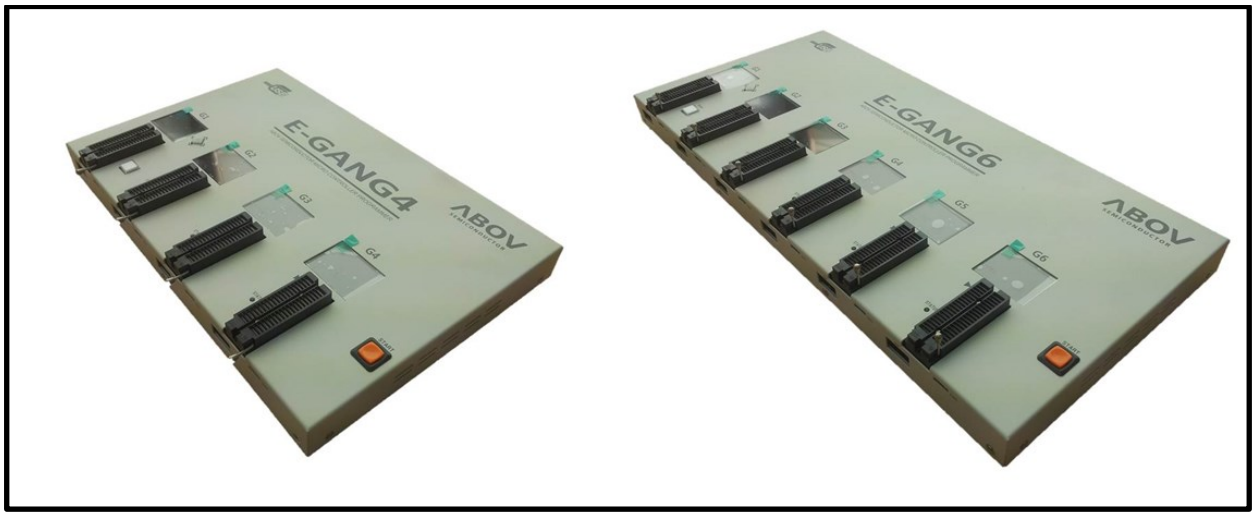
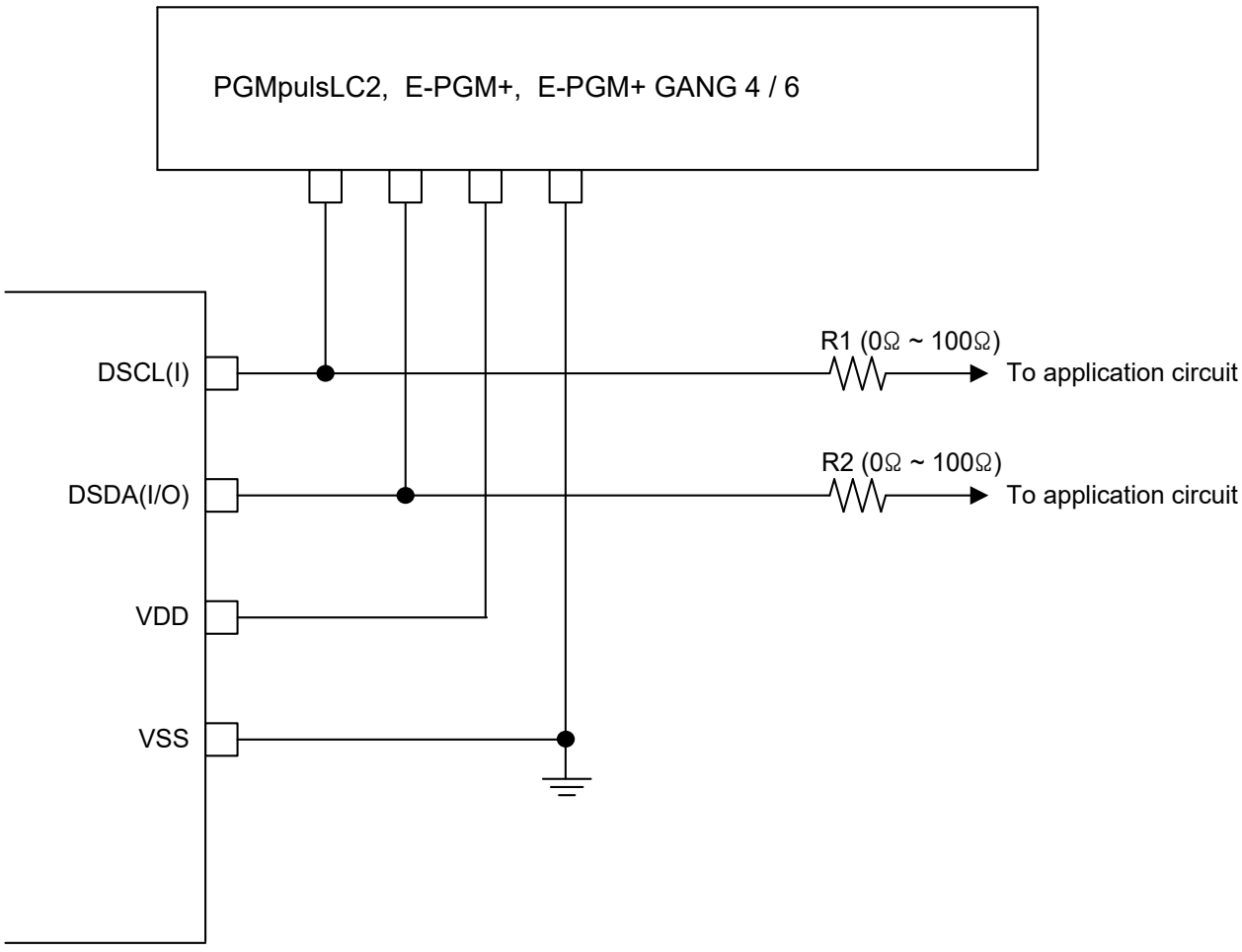


Figure 1.6 Gang Programmer

1.3.4 Circuit Design Guide

At the FLASH programming, the programming tool needs 4 signal lines that are DSCL, DSDA, VDD, and VSS. When you design the PCB circuits, you should consider the usage of these signal lines for the on-board programming. Please be careful to design the related circuit of these signal pins because rising/falling timing of DSCL and DSDA is very important for proper programming.



NOTE)

1. In on-board programming mode, very high-speed signal will be provided to pin DSCL and DSDA. And it will cause some damages to the application circuits connected to DSCL or DSDA port if the application circuit is designed as high speed response such as relay control circuit. If possible, the I/O configuration of DSDA, DSCL pins had better be set to input mode.
2. The value of R1 and R2 is recommended value. It varies with circuit of system.
3. When using LFO (Low Frequency Oscillator), it is recommended that the resistance value of R2 is 100 ohms. Because the internal De-bounce circuit does not operate normally at low clocks, this prevents Glitch from corrupting OCD communication.

Figure 1.7 PCB design guide for on board programming

2 Block diagram

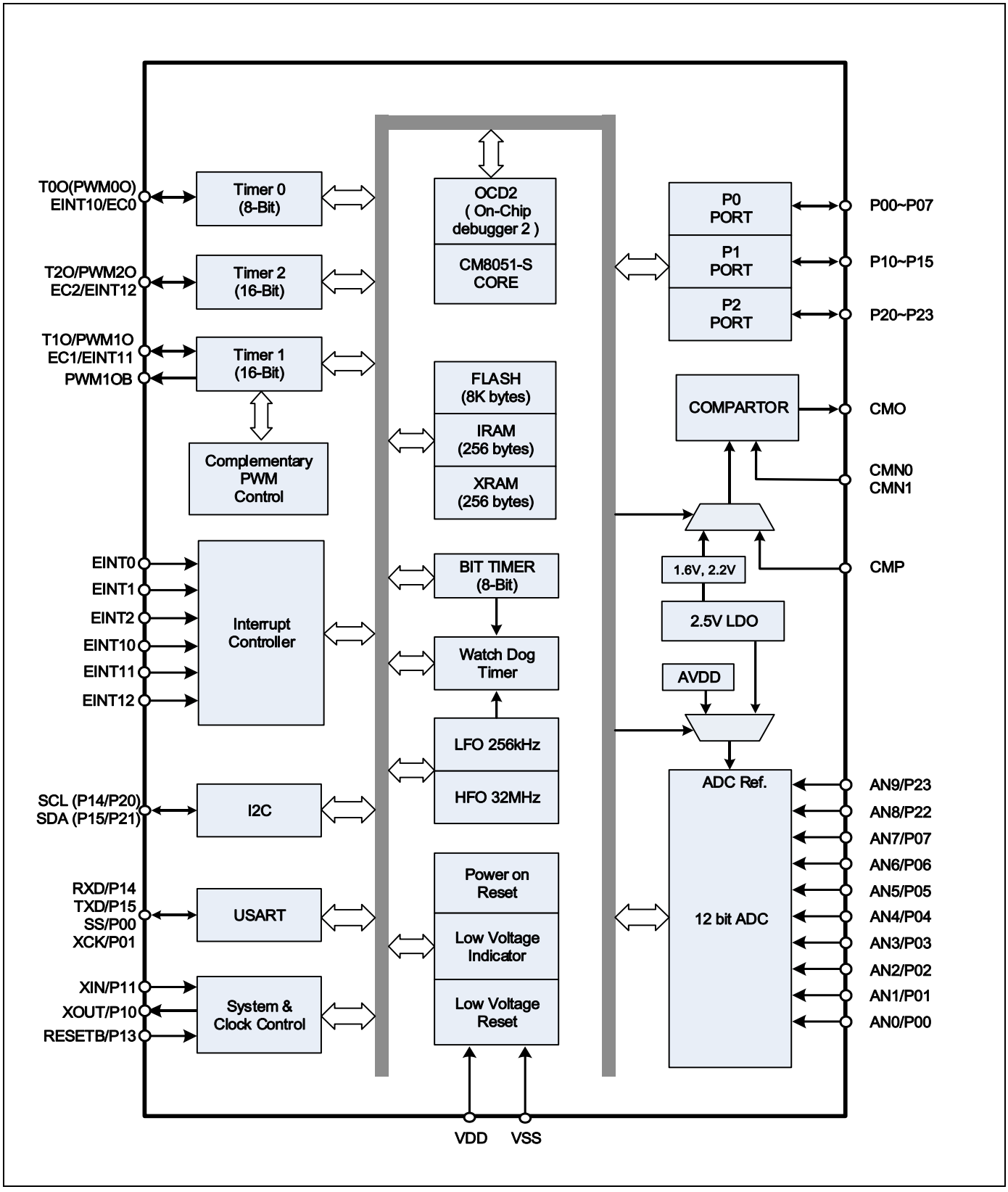
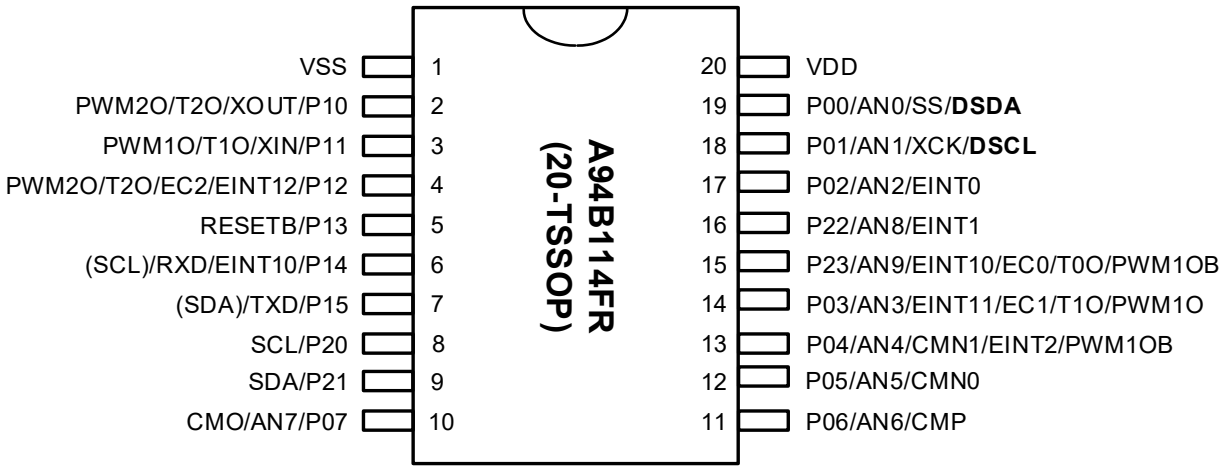


Figure 2.1 Block diagram of A94B114

NOTE) The P20, P21, P22 and P23 are not in the 16-Pin package.

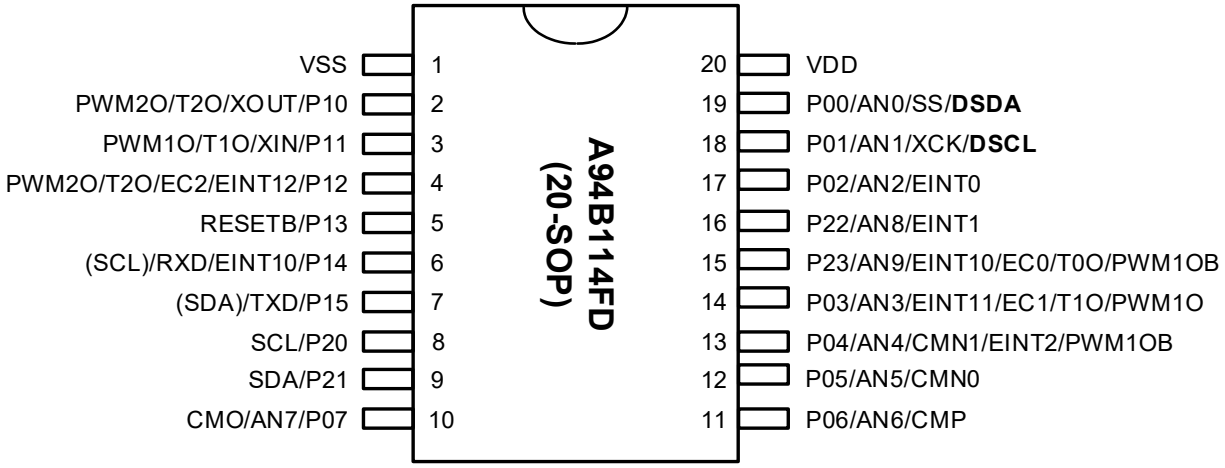
3 Pin assignment



NOTE)

1. The programmer (PGMplusLC, E-PGM+) uses P0[1:0] pin as DSCL, DSDA.
2. The SDA/SCL lines for I2C can be configured on the P20/P21 (P14/P15) pins by software control.
3. To use T1O/PWM1O as pin3(P11), set Sub-function and set it to Output High.
4. To use T2O/PWM2O as pin2(P10), set Sub-function and set it to Output Low.

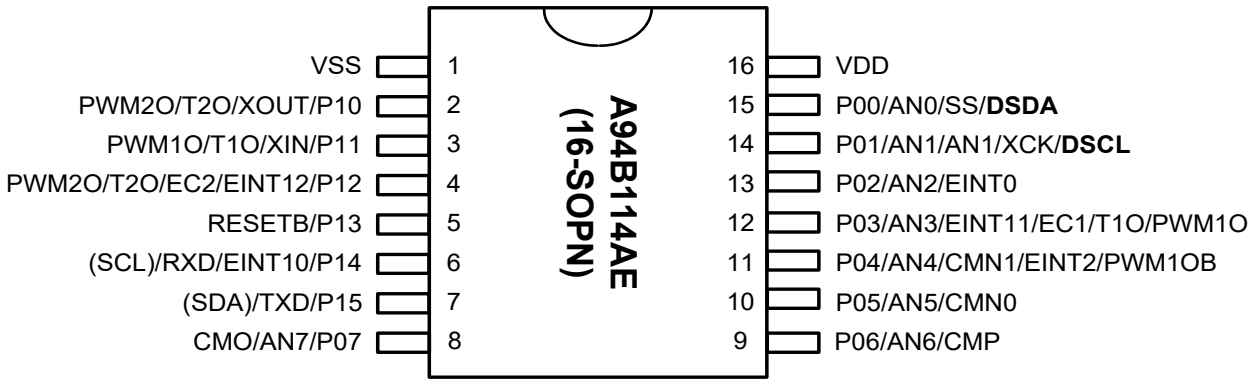
Figure 3.1 A94B114FR 20-TSSOP Pin Assignment



NOTE)

1. The programmer (PGMplusLC, E-PGM+) uses P0[1:0] pin as DSCL, DSDA.
2. The SDA/SCL lines for I2C can be configured on the P20/P21 (P14/P15) pins by software control.
3. To use T1O/PWM1O as pin3(P11), set Sub-function and set it to Output High.
4. To use T2O/PWM2O as pin2(P10), set Sub-function and set it to Output Low.

Figure 3.2 A94B114FD 20-SOP Pin Assignment



NOTE)

1. The programmer (PGMplusLC, E-PGM+) uses P0[1:0] pin as DSCL, DSDA.
2. The P20, P21, P22 and P23 pins should be selected as a push-pull output or an input with pull-up resistor by software control when the 16-pin package is used.
3. The SDA/SCL lines for I2C can be configured on the P20/P21 (P14/P15) pins by software control.
4. To use T1O/PWM1O as pin3(P11), set Sub-function and set it to Output High.
5. To use T2O/PWM2O as pin2(P10), set Sub-function and set it to Output Low.

Figure 3.3 A94B114AE 16-SOPN Pin Assignment

4 Package Diagram

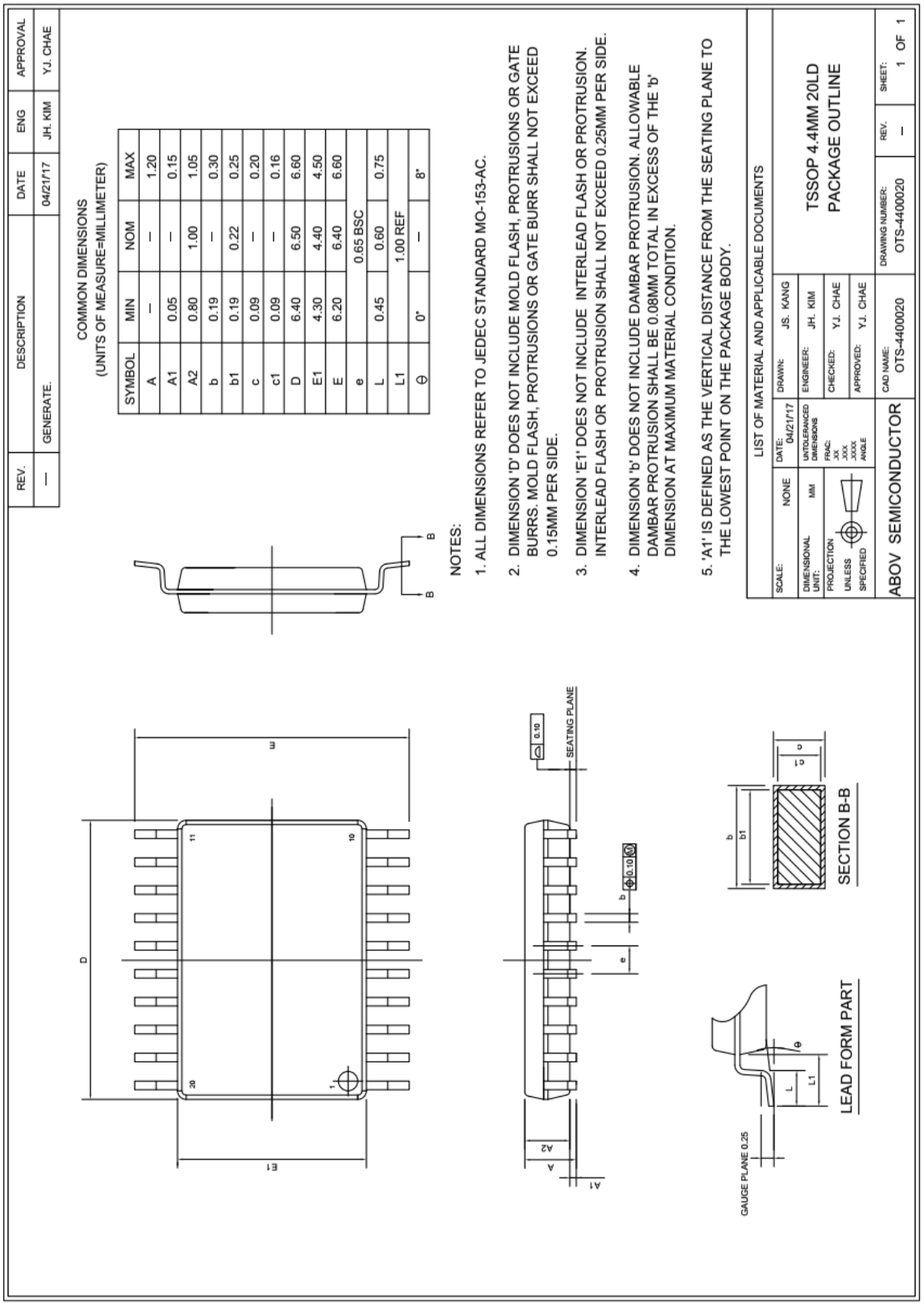


Figure 4.1 A94B114FR (20-TSSOP) Package Dimention

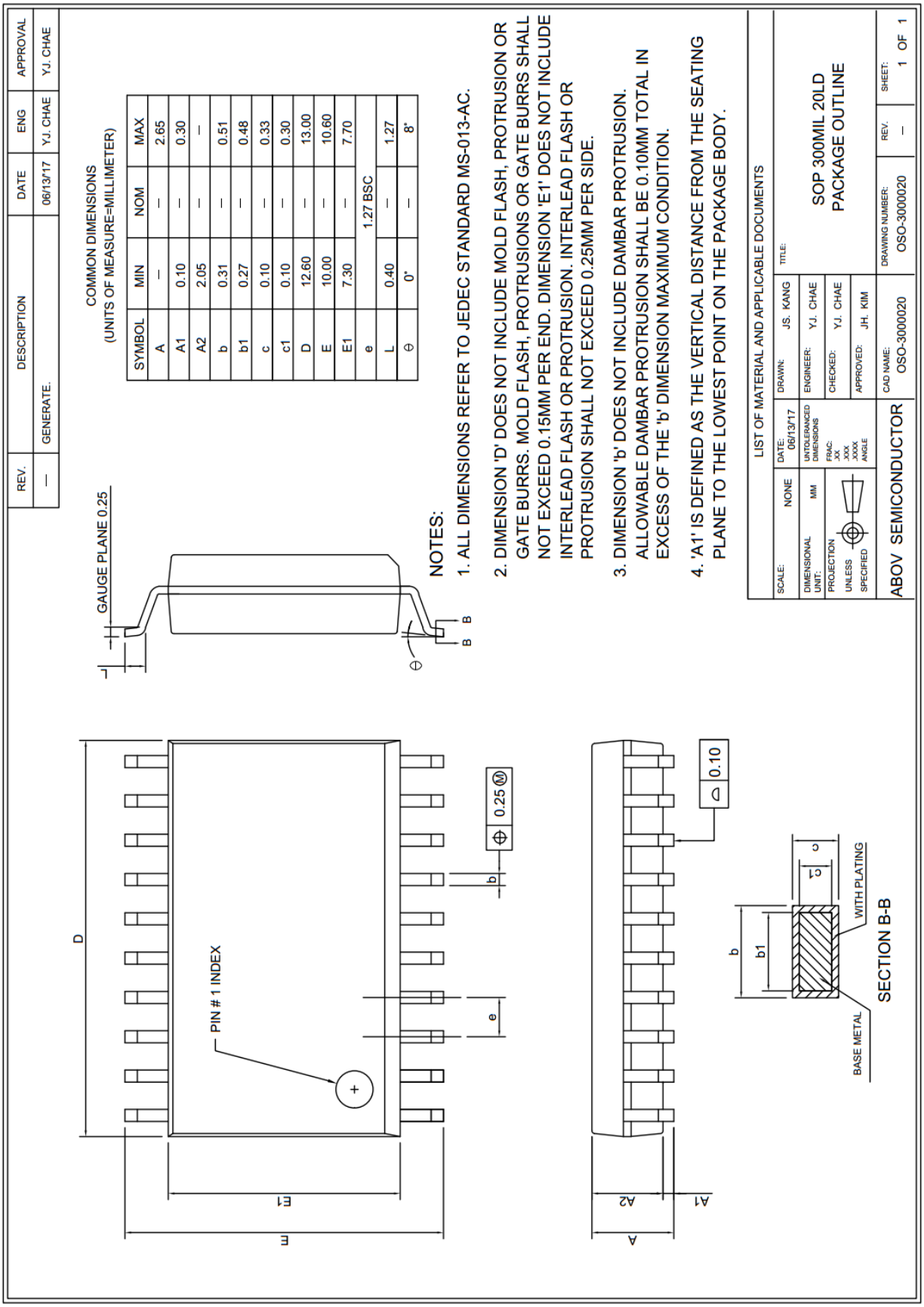


Figure 4.2 A94B114FD (20-SOP) Package Dimention

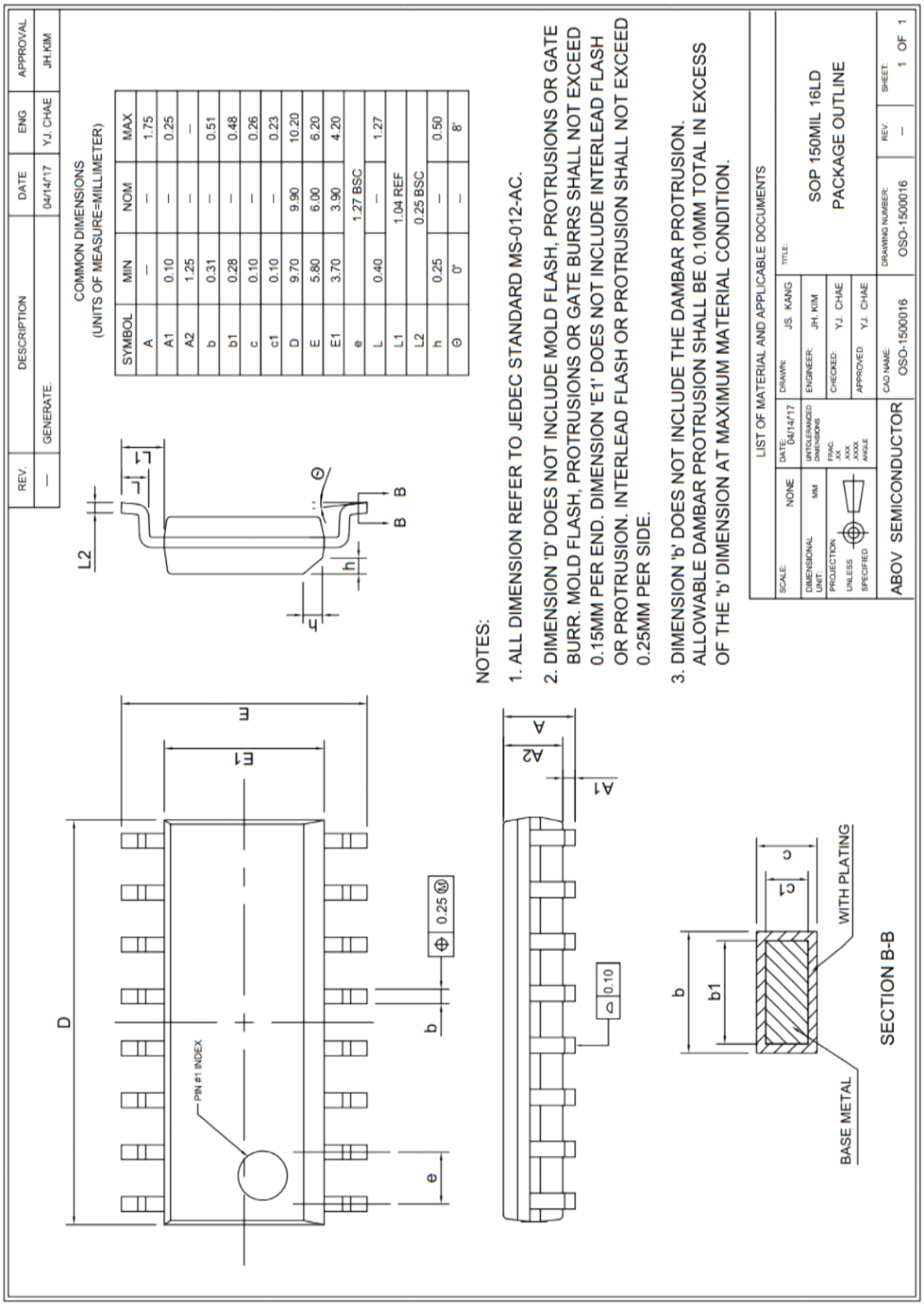


Figure 4.3 A94B114AE (16-SOPN) Package Dimention

5 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	AN0/SS/DSDA
P01				AN1/XCK/DSCL
P02				AN2//EINT0
P03				AN3/EINT11/EC1/T1O/PWM1O
P04				AN4/EINT2/CMN1/PWM1OB
P05				AN5/CMN0
P06				AN6/CMP
P07				AN7/CMO
P10	I/O	Port 1 is a bit-programmable I/O port which can be configured as a schmitt-trigger input, a push-pull output, or an open-drain output. A pull-up resistor can be specified in 1-bit unit.	Input	XOUT/T2O/PWM2O
P11				XIN/T1O/PWM1O
P12				EINT12/EC2/T2O/PWM2O
P13				RESETB
P14				RXD/(SCL)/EINT10
P15				TXD/(SDA)
P20	I/O	Port 2 is a bit-programmable I/O port which can be configured as a schmitt-trigger input, a push-pull output, or an open-drain output. A pull-up resistor can be specified in 1-bit unit. The P20 - P23 are only in the 20-Pin package.	Input	SCL
P21				SDA
P22				AN8/EINT1
P23				AN9/EINT10/EC0/ECO/T0O/PWM1OB
EINT0	I	External interrupt inputs	Input	P02/AN2/SS
EINT1				P22/AN8
EINT2				P04/AN4/CMN1/PWM1OB
EINT10		External interrupt inputs, Timer Capture and timer event counter inputs. (Timer 0, 1, 2)	Input	P14/RXD/(SCL) P23/AN9/EC0/T0O/PWM1OB
EINT11			Input	P03/AN3/EC1/T1O/PWM1O
EINT12			Input	P12/EC2/T2O/PWM2O
T0O	O	Timer 0 interval output	Input	P23/AN9/EINT10
T1O		Timer 1 interval output	Input	P03/AN3/EINT11/EC1/PWM1O
T2O		Timer 2 interval output	Input	P12/EINT12/ PWM2O
PWM0O		Timer 0 PWM output	Input	P23/AN9/EINT10/T0O
PWM1O		Timer 1 PWM output	Input	P03/AN3/EINT11/EC1/T1O
PWM1OB		Timer 1 PWM Complementary output	Input	P04/AN4/EINT2/CMN1 P23/AN9/EINT10/EC0/T0O
PWM2O		Timer 2 PWM output	Input	P12/EINT12/T2O
SS	I/O	USART external clock input/output	Input	P00/AN0/DSDA
XCK	I/O	USART slave selection input/output	input	P01/AN1/DSCL
TXD	O	USART data output	Input	P15/SDA (Second Select Pin)
RXD	I	USART data input	Input	P14/SCL (Second Select Pin)/EINT10
SCL	I/O	I2C clock input/output	Input	P20, P14/(SCL)
SDA	I/O	I2C data input/output	Input	P21, P15/(SDA)

Table 5.1 Normal Pin Description

PIN Name	I/O	Function	@RESET	Shared with
AN0	I	A/D converter input channels	Input	P00/SS/DSDA
AN1				P01/XCK/DSCL
AN2				P02//EINT0
AN3				P03/EINT11/EC1/T1O/PWM1O
AN4				P04/EINT2/CMN1/PWM1OB
AN5				P05/CMN0
AN6				P06/CMP
AN7				P07/CMO
AN8				P22/EINT1
AN9				P23/EIN10/EC0/T0O/PWM1OB
CMP	I	Comparator input channel (+)	Input	P06/AN6
CMN0	I	Comparator input channel (-)	input	P05/AN5
CMN1	I	Comparator input channel (-)	input	P04/AN4/EINT2/PWM1OB
CMO	O	Comparator output	input	P07/ AN7
RESETB	I	System reset pin with a pull-up resistor when it is selected as the RESETB by CONFIGURE OPTION	Input	P13
DSDA	I/O	Programmer data input/output (NOTE4,5)	Input	P00/AN0/SS
DSCL	I	Programmer clock input (NOTE4,5)	Input	P01/AN1/XCK
XIN	I/O	Main oscillator pins	Input	P11/T2O/PWM2O
XOUT				P10/T1O/PWM1O
VDD, VSS	–	Power input pins	–	–

Table 5.2 Normal Pin Description (Concluded)

NOTE)

1. The P20, P21, P22 and P23 are not in the 16-Pin package.
2. The P13/RESETB pin is configured as one of the P13 and the RESETB pin by the “CONFIGURE OPTION”.
3. If the P00 and P01 pins are connected to the programmer during power-on reset, the pins are automatically configured as In-System programming pins.
4. The P00 and P01 pins are configured as inputs with internal pull-up resistor only during the reset or power-on reset.

6 Port Structures

6.1 General Purpose I/O Port

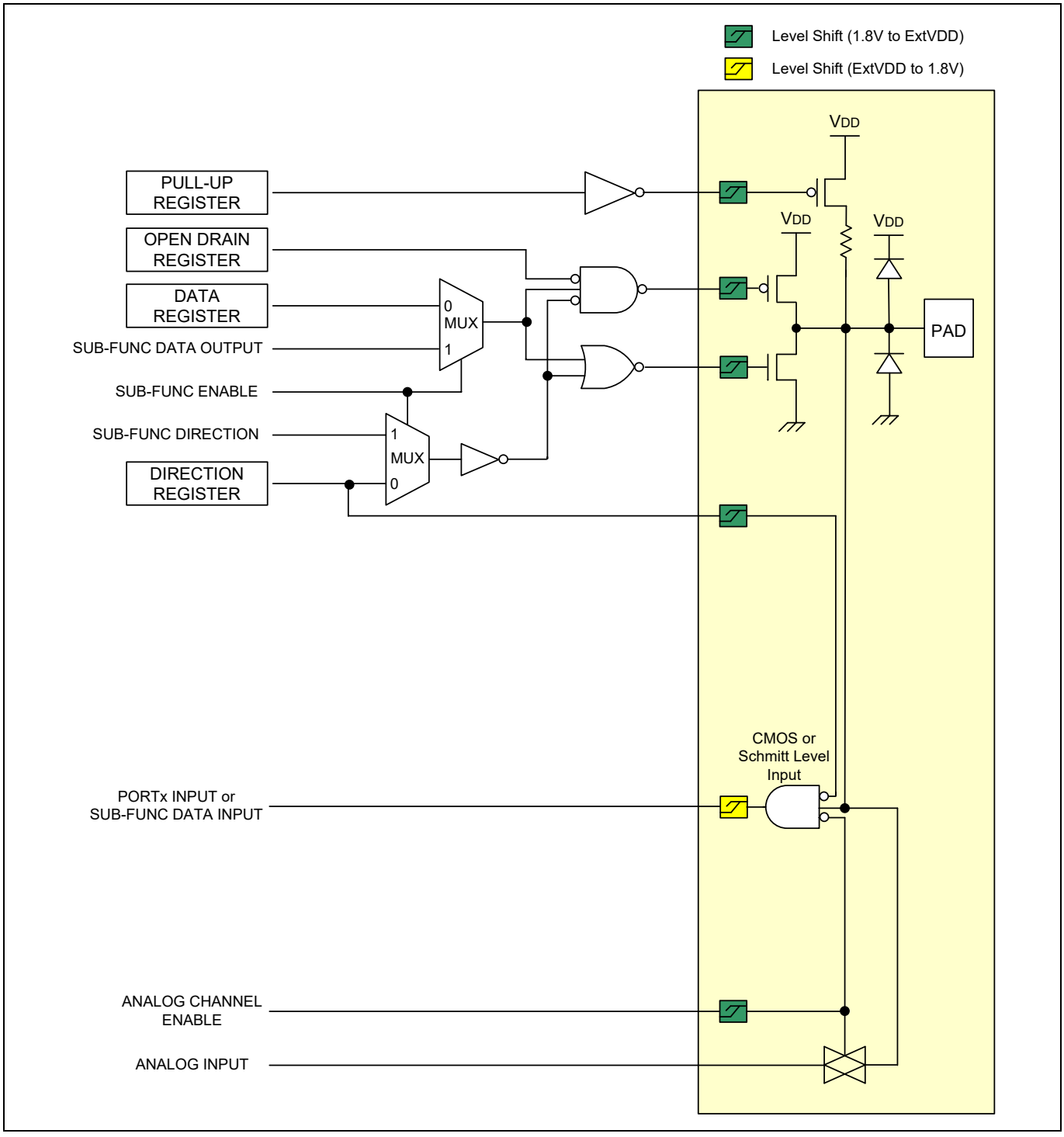


Figure 6.1 General Purpose I/O Port

6.2 External Interrupt I/O Port

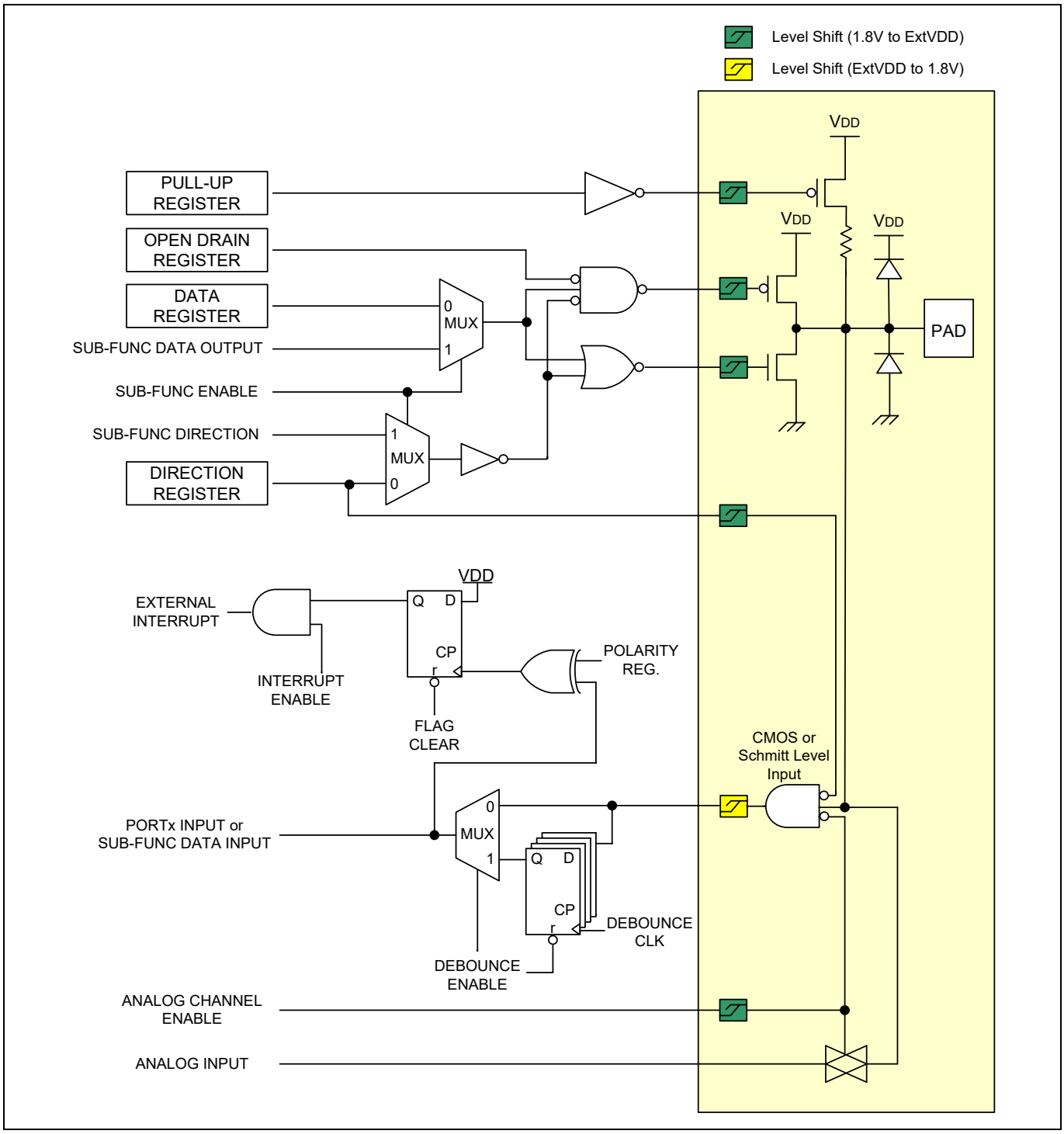


Figure 6.2 External Interrupt I/O Port

7 Electrical Characteristics

7.1 Absolute Maximum Ratings

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}	-15	mA	Maximum current output sourced by (I_{OH} per I/O pin)
	$\sum I_{OH}$	-80	mA	Maximum current ($\sum I_{OH}$)
	I_{OL}	30	mA	Maximum current sunk by (I_{OL} per I/O pin)
	$\sum I_{OL}$	160	mA	Maximum current ($\sum I_{OL}$)
Total Power Dissipation	P_T	400	mW	–
Storage Temperature	T_{STG}	-65~+150	°C	–

Table 7.1 Absolute Maximum Ratings

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

(T_A= -40°C ~ +85°C)

Parameter	Symbol	Conditions		MIN	TYP	MAX	Unit
Operating Voltage	VDD	f_X= 1, 4, 8, 16MHz	HFO	2.0	–	5.5	V
		f_X= 256kHz	LFO				
		f_X= 0.4 ~ 16MHz	Crystal	2.7	–		
Operating Temperature	T_{OPR}	VDD=2.0~5.5V (Internal RC) VDD=2.7~5.5V (Crystal)		-40	–	85	°C

Table 7.2 Recommended Operating Conditions

7.3 Low Drop Out Characteristics

(T_A= -40 ~ +85°C, AVDD=2.7~5.5V, VSS=0V)

Parameter	Symbol	Condition	MIN	TYP	MAX	Unit
LDO Output Voltage	V_{LDO}	T_A=-40~85℃	2.450	2.5	2.550	
		T_A=25℃	2.470	2.5	2.530	
Quiescent Current	I_Q	AVDD=5.5V	-	300	400	uA
Load Current	I_L	Dropout Voltage=0.2V	-	-	1	mA

Table 7.3 Low Drop Out Characteristics

7.4 A/D Converter Characteristics

(T_A= -40℃ ~ +85℃, VDD= 2.5V ~ 5.5V, VSS=0V)

Parameter	Symbol	Conditions		MIN	TYP	MAX	Unit
Resolution	–	–		–-	12	–	bit
Integral Non-Linearity	INL	Analog Reference NOTE Voltage = 2.7V ~ 5.5V (VDD ref) ADC Clock= 2MHz (VDD ref)		–	–	±5	LSB
Differential Non-Linearity	DNL			–	–	±3	
Offset Error of Top	EOT			–	±4	±8	
Offset Error of Bottom	EOB			–	±2	±4	
LDO Ref. Integral Non-Linearity	LDO_INL	Voltage = 1.25V (LDO ref) ADC Clock= 500kHz (LDO ref)		–	±15	–	LSB
		Voltage = 2.50V (LDO ref) ADC Clock= 500kHz (LDO ref)		–	±20	–	
LDO Ref. Differential Non-Linearity	LDO_DNL	Voltage = 2.5V ~ 5.5V (LDO ref) ADC Clock= 500kHz (LDO ref)		–	–	±3	
LDO Ref. Offset Error of Top	LDO_EOT			–	±4	±8	
LDO Ref. Offset Error of Bottom	LDO_EOB			–	±2	±3	
Conversion Time	t_{CON}	AVREF = 3.0V ~ 5.5V		20	–	–	us
		AVREF = 2.7V ~ 5.5V		30	–	–	
		AVREF = 2.4V ~ 5.5V		60	–	–	
Analog Input Voltage	V_{AN}	–		VSS	–	VDD	V
Analog Reference Voltage	VDDREF	*NOTE		2.2	–	VDD	V
	LDOREF	–		-	2.5	-	
Analog Input Leakage Current	I_{AN}	VDDREF=5.12V		–	–	2	uA
ADC Operating Current	I_{ADC}	Enable	VDD=5.12V	–	1	2	mA
		Disable		–	–	0.1	uA

Table 7.4 A/D Converter Characteristics

***NOTE)** If the Analog reference voltage is lower than 2.7V and the ADC Clock is faster than 2MHz, ADC resolution will be degraded. (Analog Reference = VDDREF) If the Analog reference voltage is lower than 2.5V and the ADC Clock is faster than 500kHz, ADC resolution will be degraded. (Analog Reference = LDOREF)
When LDO Reference is used, ADC gain can be affected by LDO output variation.

7.5 Low Voltage Reset and Low Voltage Indicator Characteristics

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

Parameter	Symbol	Conditions		MIN	TYP	MAX	Unit
Detection Level	V_{LVR} V_{LVI}	The LVR can select 1.75V but LVI can select other levels except 1.75V		–	1.75	1.90	V
				2.2	2.4	2.6	
				2.7	2.9	3.1	
				3.6	3.9	4.2	
				3.9	4.2	4.5	
Hysteresis	△V	–		–	50	–	mV
Minimum Pulse Width	t_{LW}	–		–	500	–	us
LVR and LVI Current	I_{LVR}	LVR 1.75V	VDD=5V	–	1	–	uA
		LVI except 1.75V		–	–	50	

Table 7.5 LVR and LVI Characteristics

7.6 Power-On Reset Characteristics

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

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
RESET Release Level	V_{POR}	–	–	1.25	–	V
VDD Voltage Rising Time	t_R	–	0.05	–	5.0	V/ms
POR Current	I_{POR}	–	–	0.1	–	uA

Table 7.6 Power-on Reset Characteristics

7.7 Internal High Frequency Oscillator Characteristics

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

Parameter	Symbol	Conditions		MIN	TYP	MAX	Unit
Frequency	f_{IRC}	V_{DD} = 2.0 ~ 5.5V		–	32	–	MHz
Tolerance	–	T_A = 25°C	With 0.1uF Bypass capacitor	–	±1.5	–	%
		T_A = 0°C to +50°C		-2.5	–	2.5	
		T_A = -40°C to +85°C		-3.0	–	3.0	
Stabilization Time	T_{HFS}	–		–	1	–	ms
Operating Current	I_{HFO}	Enable		–	0.4	–	mA

Table 7.7 High Internal RC Oscillator Characteristics

7.8 Internal Low Frequency Oscillator Characteristics

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

Parameter	Symbol	Conditions		MIN	TYP	MAX	Unit
Frequency	f_{LFON}(Normal/IDLE)	V_{DD} = 2.0 ~ 5.5V		–	256	–	kHz
	f_{LFOS}(STOP)			–	168	–	kHz
Tolerance	T_{LFON}(Normal/IDLE)	T_A = -40°C to +85°C	With 0.1uF Bypass capacitor	-10	–	+10	%
	T_{LFOS}(STOP)	T_A = 25°C		-15	–	+15	%
		T_A = -40°C to +85°C		-20	–	+20	%
Stabilization Time	T_{LFS}	–		–	1	–	ms
Operating Current	I_{LFO}	Enable		–	1.5	–	uA

Table 7.8 Ring-Oscillator Characteristics

7.9 Comparator Characteristics

(T_J=-40℃ ~ +125℃, AVDD=2.2V ~ 5.5V)

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
Operating Current	$IDD_{(RMS)}$		-	100	200	uA
Stop Current	$ISTOP_{(RMS)}$			12	100	nA
Comparator Input Offset Voltage	V_{OS}	@Trim Comparator	-	±10	-	mV
		@Trim Comparator + LDO Internal Ref	-	±25	-	
		@Trim Comparator + AVDD Internal Ref	-	±25	-	
		No Trim, Comparator Only	-	±40	-	
Propagation Delay	$t_{PD}(t_{PHL})$		-	0.08	0.15	us
Hysteresis	t_{HYS}	HYS_EN_I='High' (Except input offset)	-	±16	±20	mV
Internal VREF Stabilization Time	t_{WKUP}		4	-		us

Table 7.9 Comparator Characteristics

7.10 DC Characteristics

(T_A= -40°C ~ +85°C, VDD= 2.0V ~ 5.5V, VSS= 0V, f_{XIN}= 16MHz)

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
Input High Voltage	V_{IH1}	P0, P1, P2	0.7VDD	–	VDD	V
Input Low Voltage	V_{IL1}	P0, P1, P2	–	–	0.3VDD	V
Output High Voltage	V_{OH1}	VDD=3.3V, I_{OH}=-4mA, All output ports	VDD-1.0	–	–	V
	V_{OH2}	VDD=5V, I_{OH}=-8mA, All output ports	VDD-1.0	–	–	V
Output Low Voltage	V_{OL1}	VDD=3.3V, IOL= 5mA, All output ports	–	–	1.0	V
	V_{OL2}	VDD=5V, IOL= 10mA, All output ports	–	–	1.0	V
Input High Leakage Current	I_{IH}	All input ports	-	–	1	uA
Input Low Leakage Current	I_{IL}	All input ports	-1	–	-	uA
Pull-Up Resistor	R_{PU1}	V_I=0V, T_A= 25°C All Input ports	25	50	100	kΩ
Power Supply Current	I_{DD1} (RUN)	Run Mode, f_X=16MHz (HFO/2)	–	3	6	mA
	I_{IDD2} (IDLE)	IDLE Mode, f_X=16MHz	–	0.8	1.6	mA
	I_{DD3} (RUN_LFO)	Run Mode, f_X=128kHz (LFO/2)	–	1	2.5	mA
	I_{DD4} (STOP1)	STOP1 Mode, LFO Enable	–	5	20	uA
	I_{DD5} (STOP2)	STOP2 Mode, LFO Disable, LVR Disable	–	2.5	5	uA

Table 7.10 DC Characteristics

7.11 AC Characteristics

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

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
RESETB input low width	t_{RST}	Input, VDD= 5V	–	500	–	us
Interrupt input high, low width	t_{IWH}, t_{IWL}	All interrupt, VDD= 5V	125	–	–	ns
External Counter Input High, Low Pulse Width	t_{ECWH}, t_{ECWL}	ECn, VDD = 5V	125	–	–	
External Counter Transition Time	t_{REC}, t_{FEC}	ECn, VDD = 5V	–	–	20	

Table 7.11 AC Characteristics

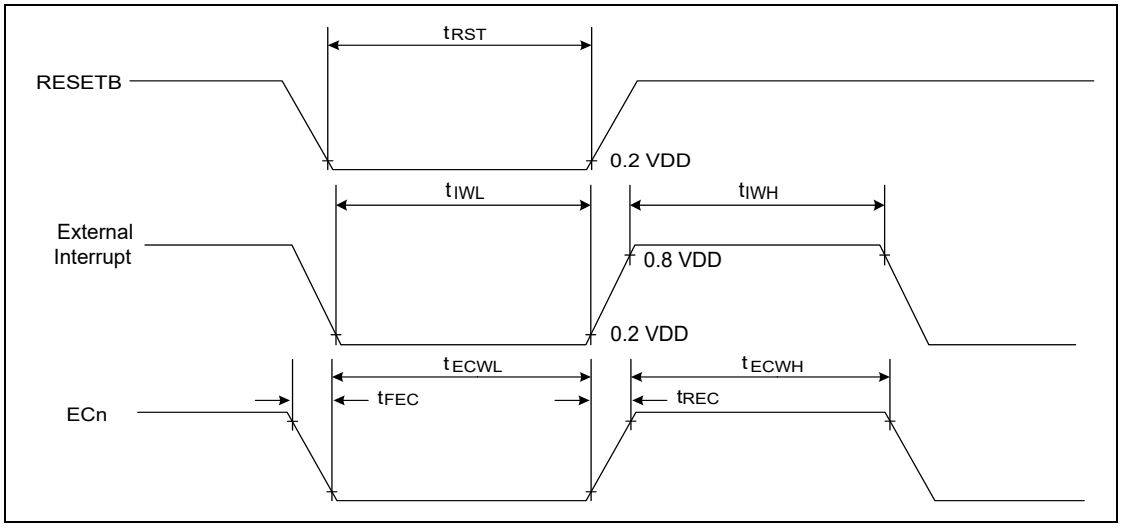


Figure 7.1 AC Timing

7.12 USART Characteristics

(T_A= -40°C ~ +85°C, VDD=2.0V ~ 5.5V, f_{XIN}=16MHz)

Parameter	Symbol	MIN	TYP	MAX	Unit
Serial port clock cycle time	t_{SCK}	1800	t_{CPU} x 16	2200	ns
Output data setup to clock rising edge	t_{S1}	8100	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}	720	t_{CPU} x 8	1280	ns

Table 7.12 USART Characteristics

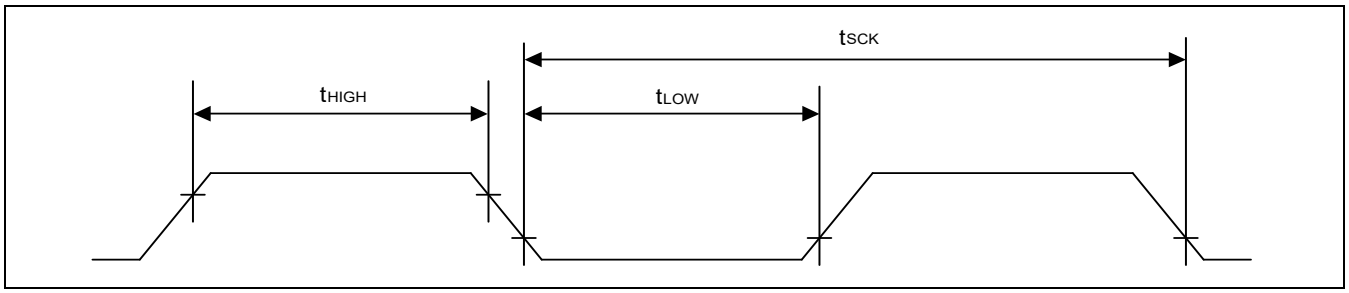


Figure 7.2 Waveform for USART Timing Characteristics

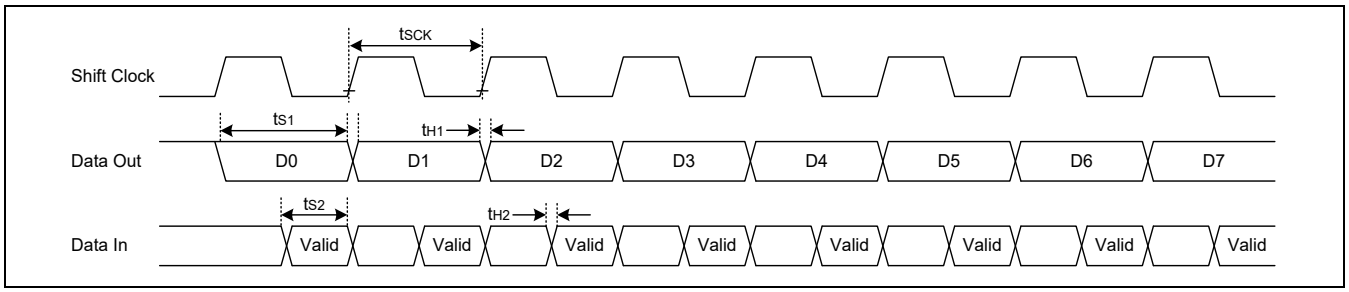


Figure 7.3 Timing Waveform for the USART Module

7.13 I2C Characteristics

(T_A= -40°C ~ +85°C, VDD=2.0V ~ 5.5V)

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

Table 7.13 I2C Characteristics

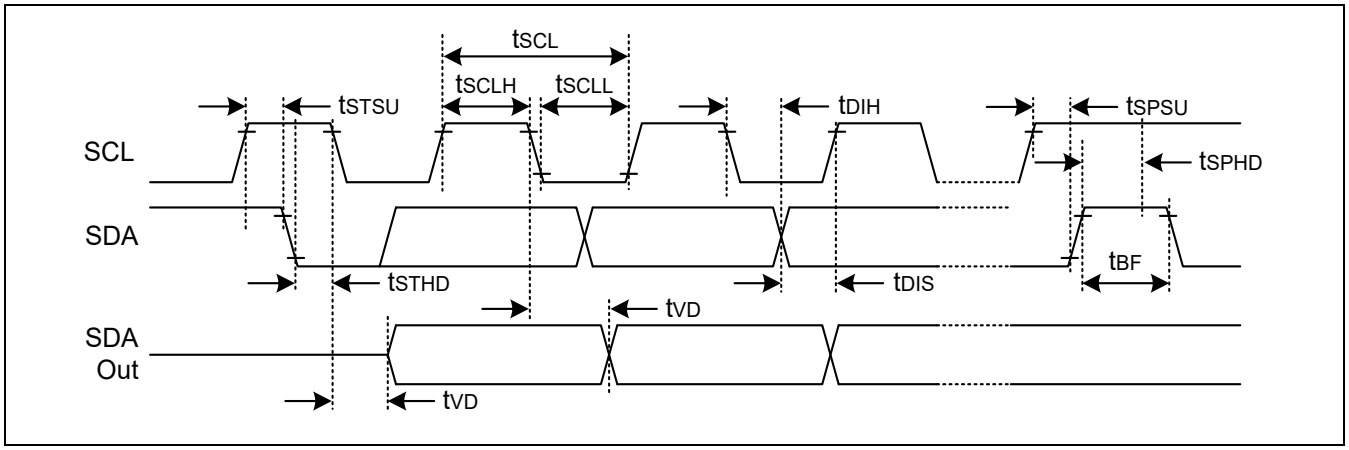


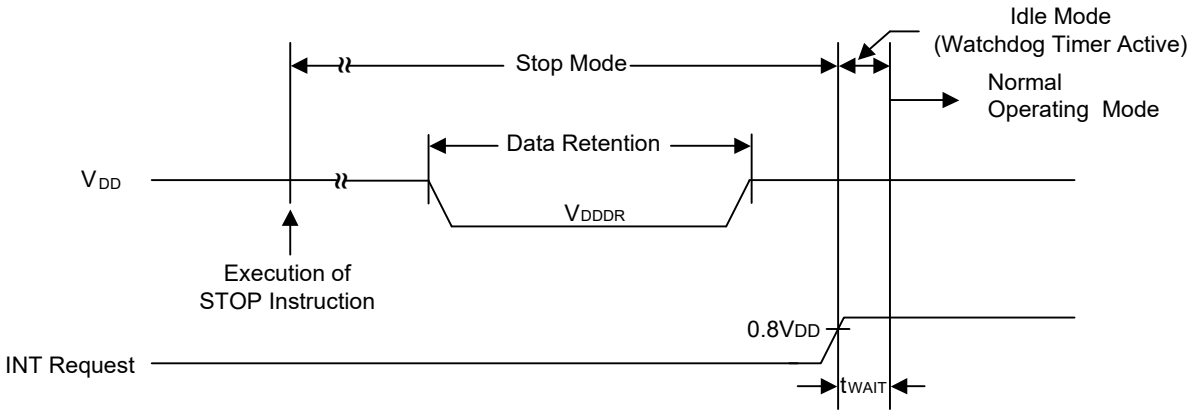
Figure 7.4 I2C Timing

7.14 Data Retention Voltage in Stop Mode

(T_A=-40°C ~ +85°C)

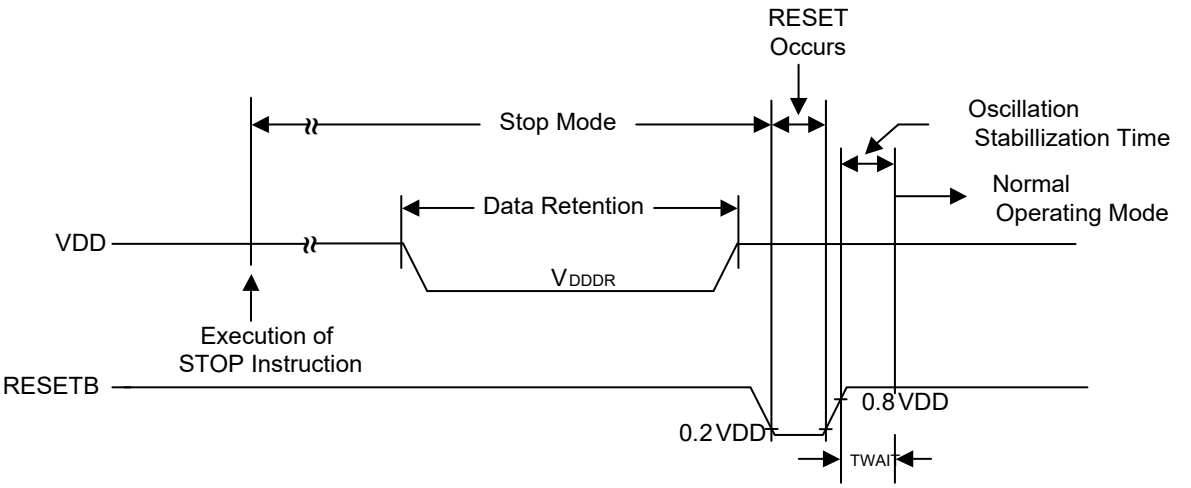
Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
Data retention supply voltage	V_{DDDR}	–	1.8	–	5.5	V
Data retention supply current	I_{DDDR}	VDDR= 1.8V, (T_A= 25°C), Stop mode	–	–	1	uA

Table 7.14 Data Retention Voltage in Stop Mode



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

Figure 7.5 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 7.6 Stop Mode Release Timing when Initiated by RESETB

7.15 Internal Flash Rom Characteristics

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

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

Table 7.15 Internal Flash Rom Characteristics

7.15.1 Internal Flash Rom Characteristics (Sector 248~255)

(T_A= 25°C, VDD=2.0V ~ 5.5V, VSS= 0V)

Parameter	Symbol	Condition	MIN	TYP	MAX	Unit
Sector Write Time	t_{FSW}	–	–	2.5	2.7	ms
Sector Erase Time	t_{FSE}	–	–	2.5	2.7	
Code Write Protection Time	t_{FHL}	–	–	2.5	2.7	
Page Buffer Reset Time	t_{FBR}	–	–	–	5	us
Flash Programming Frequency	f_{PGM}	–	0.4	–	–	MHz
Endurance of Write/Erase	NF_{WE}	–	–	–	50,000	cycles

Table 7.16 Internal Flash Rom Characteristics

NOTE) During a flash operation, SCLK[1:0] of SCCR must be set to “00” or “01” (HFO or Main Crystal for system clock).

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}					

Table 7.17 Input / Output Capacitance

7.17 Main Clock Oscillator Characteristics

(T_A=-40°C ~ +85°C, VDD=2.0V ~ 5.5V)

Oscillator	Parameter	Condition	MIN	TYP	MAX	Unit
Crystal	Main oscillation frequency	2.0V – 5.5V	0.4	–	4.2	MHz
		2.7V – 5.5V	0.4	–	16.0	
Ceramic Oscillator	Main oscillation frequency	2.0V – 5.5V	0.4	–	4.2	MHz
		2.7V – 5.5V	0.4	–	16.0	
External Clock	XIN input frequency	2.0V – 5.5V	0.4	–	4.2	MHz
		2.7V – 5.5V	0.4	–	16.0	

Table 7.18 Main Clock Oscillator Characteristics

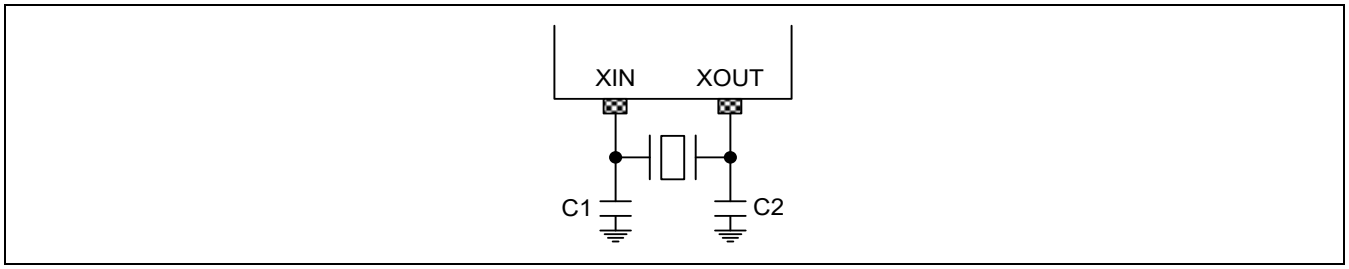


Figure 7.7 Crystal/Ceramic Oscillator

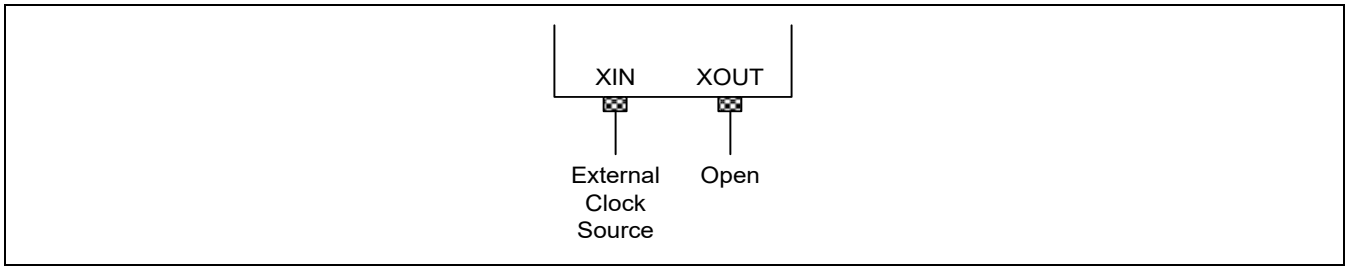


Figure 7.8 External Clock

7.18 Main Oscillation Stabilization Characteristics

(T_A=-40°C ~ +85°C, VDD=2.0V ~ 5.5V)

Oscillator	Parameter	MIN	TYP	MAX	Unit
Crystal	fx > 1MHz Oscillation stabilization occurs when VDD is equal to the minimum oscillator voltage range.	–	–	60	ms
Ceramic		–	–	10	ms
External Clock	f_{XIN} = 0.4 to 16MHz XIN input high and low width (t_{XH}, t_{XL})	42	–	1250	ns

Table 7.19 Main Oscillation Stabilization Characteristics

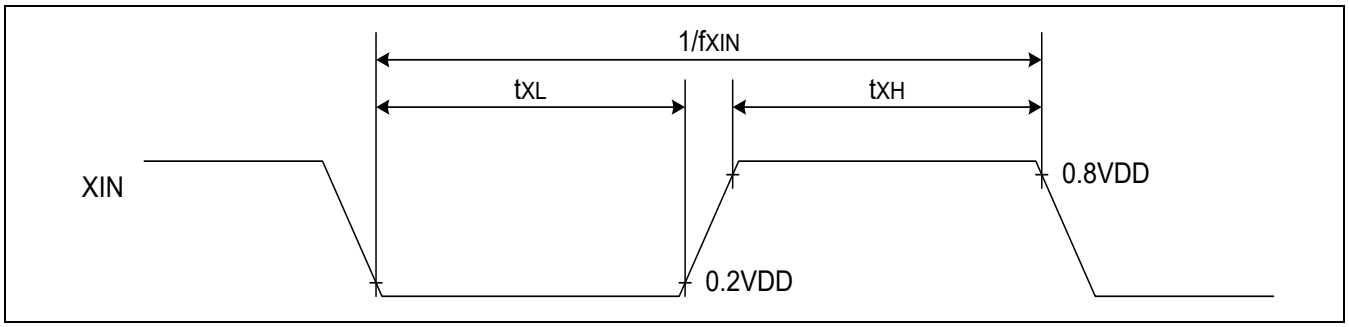


Figure 7.9 Clock Timing Measurement at XIN

7.19 Operating Voltage Range

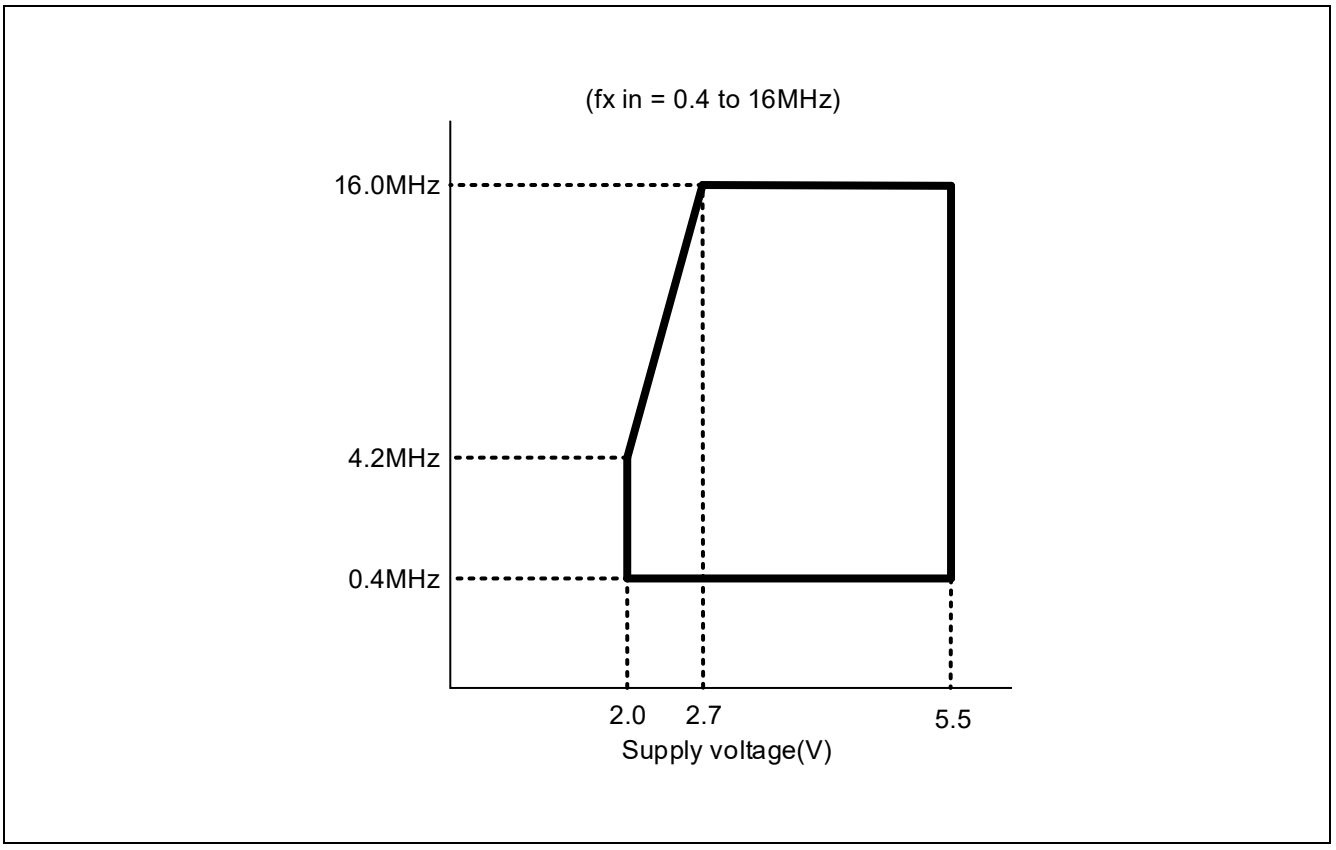


Figure 7.10 Operating Voltage Range

7.20 Recommended Circuit and Layout

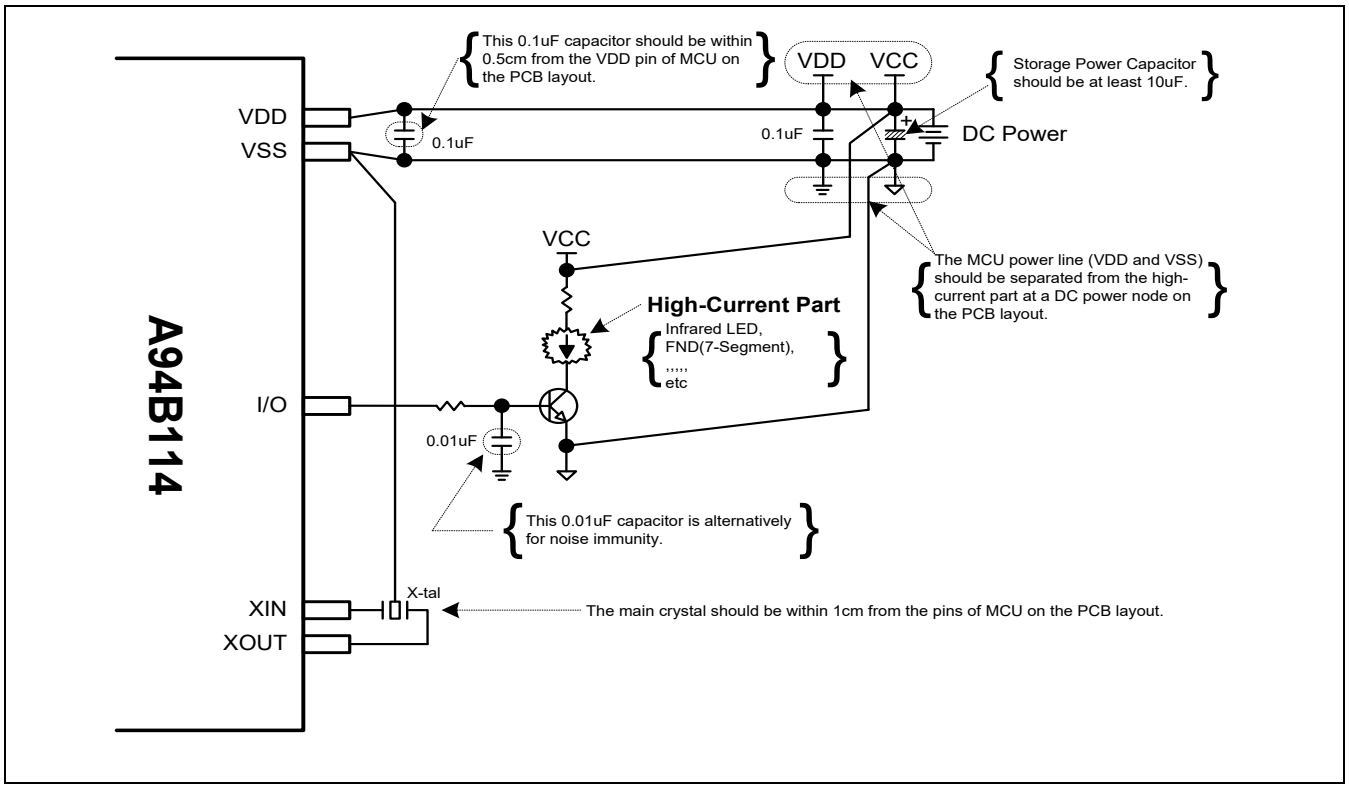


Figure 7.11 Recommended Circuit and Layout

7.21 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.

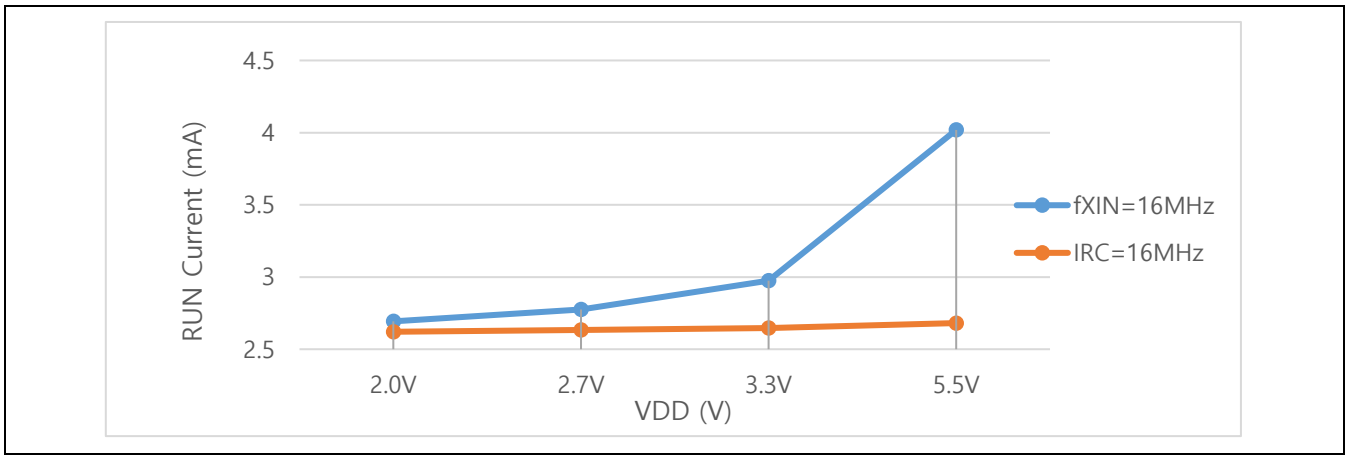


Figure 7.12 RUN (IDD1) Current

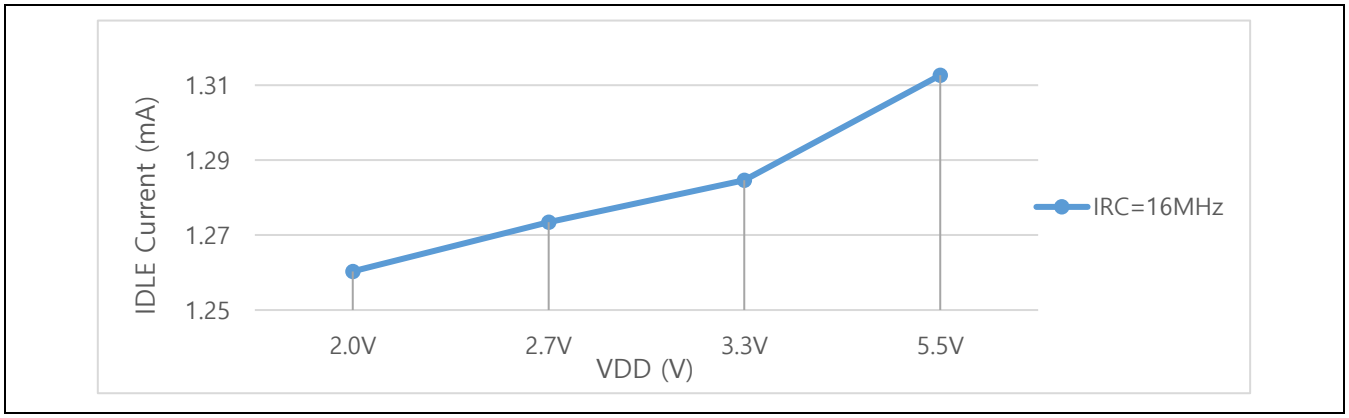


Figure 7.13 IDLE (IDD2) Current

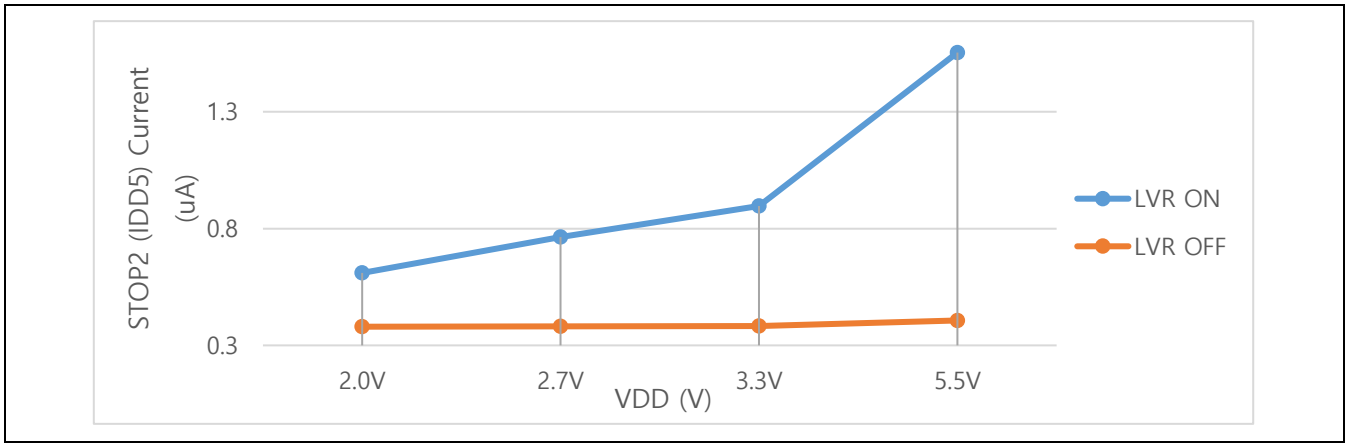


Figure 7.14 STOP (IDD5) Current

8 Memory

The A94B114 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.

A94B114 provides on-chip 8 Kbytes 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 256 bytes.

8.1 Program Memory

A 16-bit program counter is capable of addressing up to 64 Kbytes, but this device has just 8 Kbytes 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 10, for example, is assigned to location 000BH. If external interrupt 10 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.

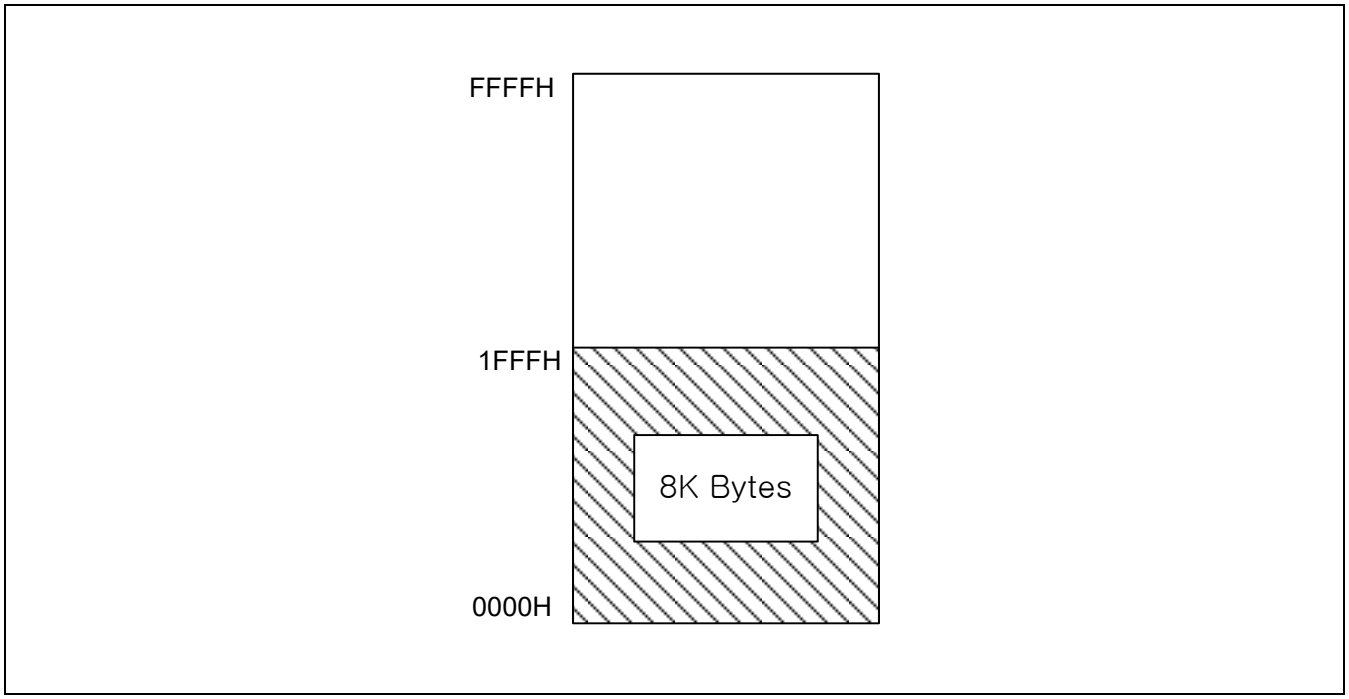


Figure 8.1 Program Memory

NOTE)

1. 8k bytes Including Interrupt Vector Region

8.2 Data Memory

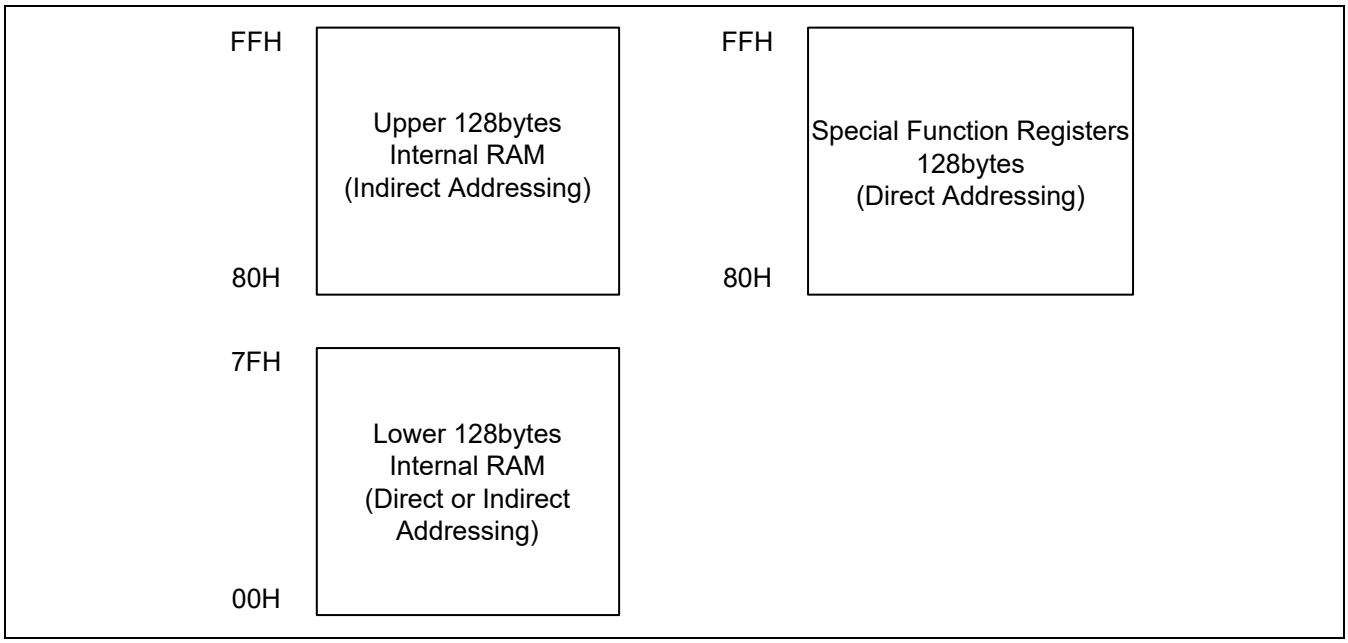


Figure 8.2 Data Memory Map

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.

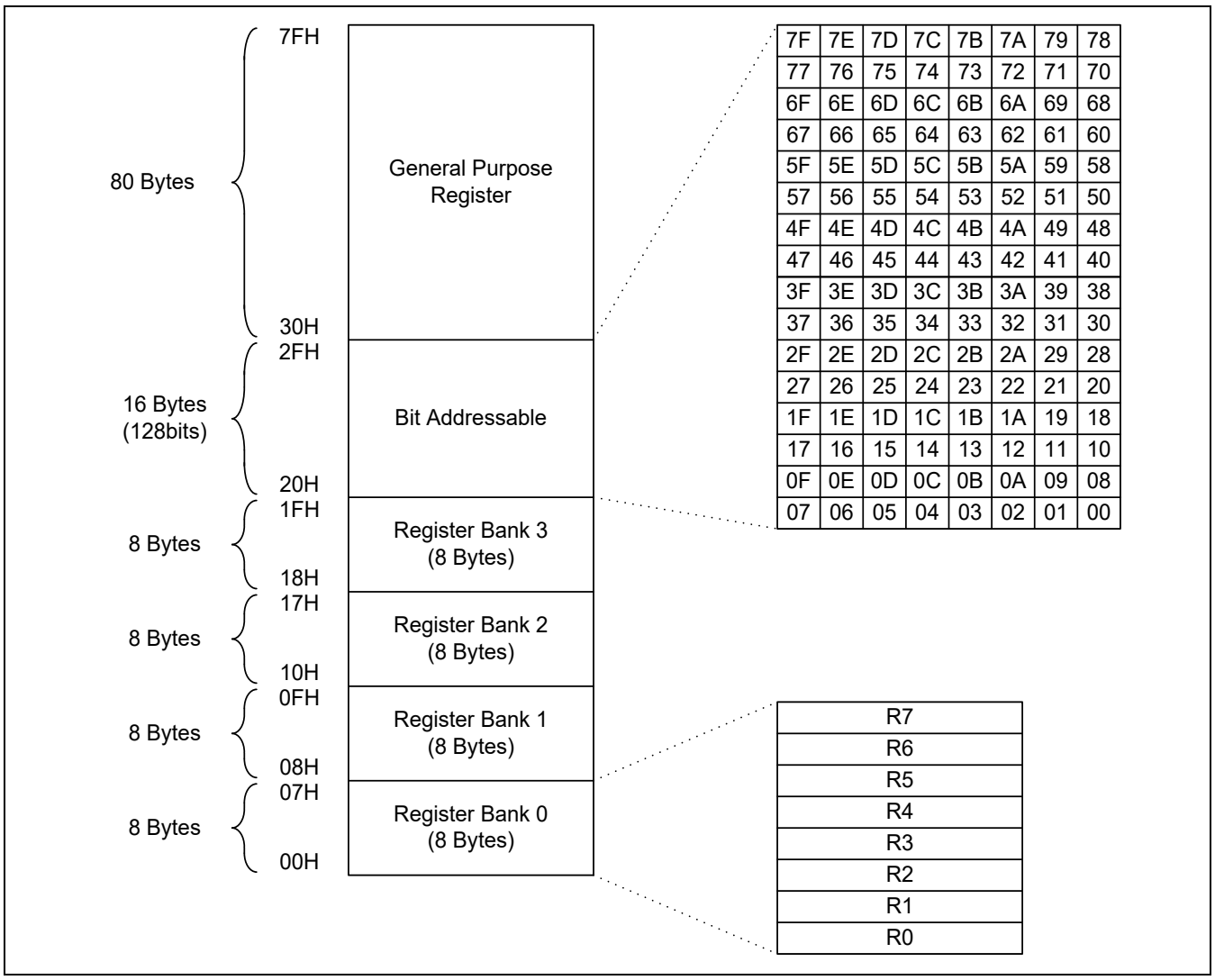


Figure 8.3 Lower 128 bytes RAM

8.3 External Data Memory

A94B114 has 256 bytes XRAM and 32 byte XSFR. This area has no relation with RAM/FLASH. It can be read and written to through SFR with 8-bit unit.

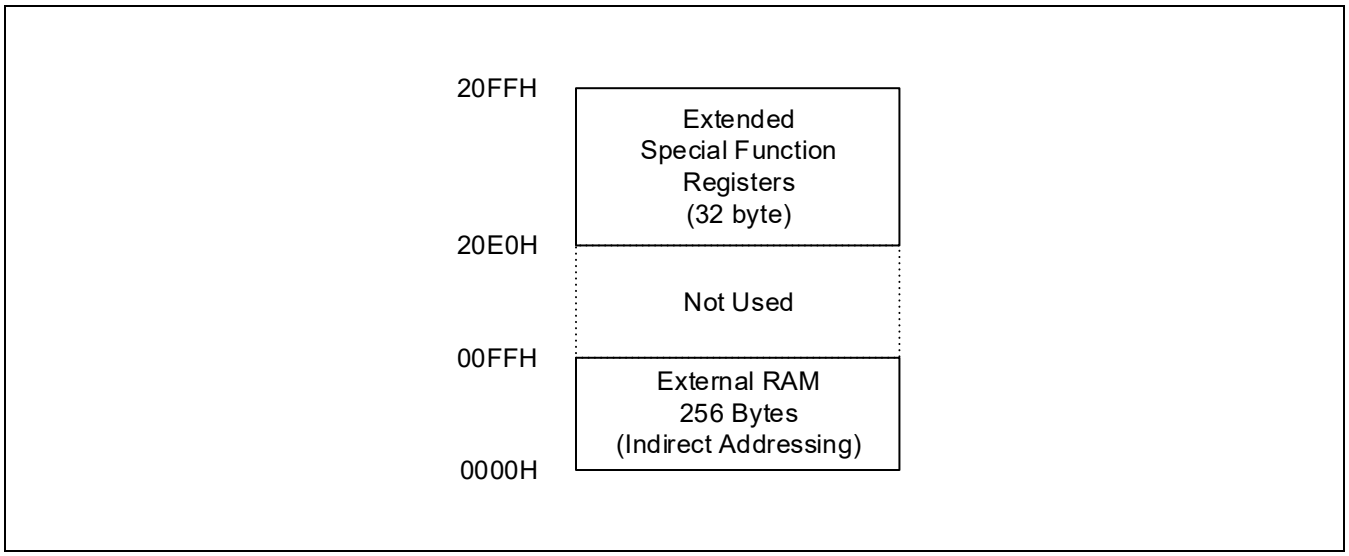


Figure 8.4 XDATA Memory Area

NOTE) XSFR can be declared and used as follows.

```
#define     CRC_CON     *(volatile unsigned char xdata *) 0x20E0
#define     CRC_H       *(volatile unsigned char xdata *) 0x20E3
#define     CRC_L       *(volatile unsigned char xdata *) 0x20E4
…
```

8.4 SFR Map

8.4.1 SFR Map Summary

- Reserved
- M8051 compatible

	00H/8H(1)	01H/9H	02H/0AH	03H/0BH	04H/0CH	05H/0DH	06H/0EH	07H/0FH
0F8H	I2CSR	I2CMR	I2CSCLLR	I2CSCLHR	I2CSDAHR	I2CDR	I2CSAR	I2CSAR1
0F0H	B	FEMR	FECR	FESR	FETCR	FEARL	FEARM	FEARH
0E8H	IP	IP1	IP2	UCTRL4	FPCR	-	I2CMR1	-
0E0H	ACC	UCTRL1	UCTRL2	UCTRL3	USTAT	UBAUD	UDATA	-
0D8H	OSCCR	SCCR	-	-	-	-	CMPDBT	-
0D0H	PSW	-	-	-	-	-	CMPCR	CMPTR
0C8H	IE2	EIFLAG	-	-	T1CDRL	T1CDRH	T1DDRL	T1DDRH
0C0H	IE1	-	T2CRL	T2CRH	T2ADRL	T2ADRH	T2BDRL	T2BDRH
0B8H	IE	-	T1CRL	T1CRH	T1ADRL	T1ADRH	T1BDRL	T1BDRH
0B0H	IRQ2	-	-	P0FSR_L	P0FSR_H	P1FSR_L	P1FSR_H	P2FSR
0A8H	IRQ1	-	LDOCR	P2IO	P2PU	P2OD	P2DB	-
0A0H	IRQ0	-	EO	P1IO	P1PU	P1OD	P1DB	-
98H	RSFR	-	ILVL	P0IO	P0PU	P0OD	P0DB	-
90H	P2	-	T0CR	T0CNT	T0DR/ T0CDR	ADCM	ADCM1/ ADCRL	ADCRH
88H	P1	LVIR	IOFFSET	EIPOL0	EIPOL1	WDTMR	WDTR/ WDTCR	SYSCON_AR
80H	P0	SP	DPL/ DPL1	DPH/ DPH1	DBTSR	BITCNT	BITCR	PCON

Table 8.1 SFR Map Summary

NOTE) 00H/8H(1), These registers are bit-addressable.

8.4.2 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
82H	Data Pointer Register Low 1	DPL1	R/W	0	0	0	0	0	0	0	0
83H	Data Pointer Register High 1	DPH1	R/W	0	0	0	0	0	0	0	0
84H	De-bounce Time Selection Register	DBTSR	R/W	0	0	0	0	0	0	0	0
85H	Basic Interval Timer Counter Register	BITCNT	R	0	0	0	0	0	0	0	0
86H	Basic Interval Timer Control Register	BITCR	R/W	0	0	0	0	0	1	0	0
87H	Power Control Register	PCON	R/W	-	–	–	–	0	0	0	0
88H	P1 Data Register	P1	R/W	-	-	0	0	0	0	0	0
89H	Low Voltage Indicator Control Register	LVIR	R/W	0	0	0	0	0	0	0	0
8AH	Interrupt Offset Register	IOFFSET	W	0	0	0	0	0	0	0	0
8BH	External Interrupt Polarity Register 0	EIPOL0	R/W	0	0	0	0	0	0	0	0
8CH	External Interrupt Polarity Register 1	EIPOL1	R/W	0	0	0	0	0	0	0	0
8DH	Watch Dog Timer Mode Register	WDTMR	R/W	0	0	1	1	1	0	1	1
8EH	Watch Dog Timer Data Register	WTDR	W	1	1	1	1	1	1	1	1
	Watch Dog Timer Counter Register	WTCR	R	0	0	0	0	0	0	0	0
8FH	System Control Access Register	SYSCON_AR	R/W	0	0	0	0	0	0	0	0
90H	P2 Data Register	P2	R/W	–	–	–	–	0	0	0	0
91H	Reserved	–	–	–							
92H	Timer 0 Control Register	T0CR	R/W	0	0	0	0	0	0	0	0
93H	Timer 0 Counter Register	T0CNT	R	0	0	0	0	0	0	0	0
94H	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
95H	A/D Converter Mode Register	ADCM	R/W	1	0	0	0	1	1	1	1
96H	A/D Converter Mode 1 Register	ADCM1	W	0	0	0	0	0	0	0	1
	A/D Converter Data Low Register	ADCRL	R	0	0	0	0	0	0	0	0
97H	A/D Converter Data High Register	ADCRH	R	0	0	0	0	0	0	0	0
98H	Reset Source Flag Register	RSFR	R/W	1	0	0	0	0	1	0	0
99H	Reserved	–	–	–							
9AH	Interrupt nesting level Register	ILVL	R/W	0	0	0	0	0	0	0	0
9BH	P0 Direction Register	P0IO	R/W	0	0	0	0	0	0	0	0
9CH	P0 Pull-up Resistor Selection Register	P0PU	R/W	0	0	0	0	0	0	0	0
9DH	P0 Open-drain Selection Register	P0OD	R/W	0	0	0	0	0	0	0	0
9EH	P0 De-bounce Time Selection Register	P0DB	R/W	0	0	0	0	0	0	0	0
9FH	Reserved	–	–	–							
A0H	Interrupt Request Register 0	IRQ0	0	0	0	0	0	0	0	0	0
A1H	Reserved	–	–	–							
A2H	Extended Operation Register	EO	R/W	–	–	–	–	–	–	–	0
A3H	P1 Direction Register	P1IO	R/W	0	0	0	0	0	0	0	0
A4H	P1 Pull-up Resistor Selection Register	P1PU	R/W	0	0	0	0	0	0	0	0

Table 8.2 SFR Map

Address	Function	Symbol	R/W	@Reset							
				7	6	5	4	3	2	1	0
A5H	P1 Open-drain Selection Register	P1OD	R/W	0	0	0	0	0	0	0	0
A6H	P1 De-bounce Time Selection Register	P1DB	R/W	0	0	0	0	0	0	0	0
A7H	Reserved	–	–	–							
A8H	Interrupt Request Register 1	IRQ1	0	0	0	0	0	0	0	0	0
A9H	Reserved	–	–	–							
AAH	LDO Control Register	LDOCR	R/W	0	0	0	0	0	0	0	0
ABH	P2 Direction Register	P2IO	R/W	0	0	0	0	0	0	0	0
ACH	P2 Pull-up Resistor Selection Register	P2PU	R/W	0	0	0	0	0	0	0	0
ADH	P2 Open-drain Selection Register	P2OD	R/W	0	0	0	0	0	0	0	0
AEH	P2 De-bounce Time Selection Register	P2DB	R/W	0	0	0	0	0	0	0	0
AFH	Reserved	–	–	–							
B0H	Interrupt Request Register 2	IRQ2	0	0	0	0	0	0	0	0	0
B1H	Reserved	–	–	–							
B2H	Reserved	–	–	–							
B3H	P0 Function Selection Low Register	P0FSRL	R/W	0	0	0	0	0	0	0	0
B4H	P0 Function Selection High Register	P0FSRH	R/W	0	0	0	0	0	0	0	0
B5H	P1 Function Selection Low Register	P1FSRL	R/W	0	0	0	0	0	0	0	0
B6H	P1 Function Selection High Register	P1FSRH	R/W	0	0	0	0	0	0	0	0
B7H	P2 Function Selection Register	P2FSR	R/W	0	0	0	0	0	0	0	0
B8H	Interrupt Enable Register	IE	R/W	0	–	–	0	0	0	0	0
B9H	Reserved	–	–	–							
BAH	Timer 1 Control Low Register	T1CRL	R/W	0	0	0	0	0	0	0	0
BBH	Timer 1 Control High Register	T1CRH	R/W	0	0	0	0	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 B Data High Register	T1BDRH	R/W	1	1	1	1	1	1	1	1
C0H	Interrupt Enable Register 1	IE1	R/W	–	0	0	0	0	0	0	0
C1H	Reserved	–	–	–							
C2H	Timer 2 Control Low Register	T2CRL	R/W	0	0	0	0	0	0	0	0
C3H	Timer 2 Control High Register	T2CRH	R/W	0	0	0	0	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 B Data High Register	T2BDRH	R/W	1	1	1	1	1	1	1	1
C8H	Interrupt Enable Register 2	IE2	R/W	–	–	–	–	0	0	0	0
C9H	External Interrupt Flag Register	EIFLAG	R/W	–	–	–	–	–	0	0	0
CAH	Reserved	–	–	–							
CBH	Reserved	–	–	–							
CCH	Timer 1 C Data Low Register	T1CDRL	R/W	1	1	1	1	1	1	1	1
CDH	Timer 1 C Data High Register	T1CDRH	R/W	1	1	1	1	1	1	1	1
CEH	Timer 1 D Data Low Register	T1DDRL	R/W	1	1	1	1	1	1	1	1
CFH	Timer 1 D Data High Register	T1DDRH	R/W	1	1	1	1	1	1	1	1

Table 8.3 SFR Map (Continue)

Address	Function	Symbol	R/W	@Reset							
				7	6	5	4	3	2	1	0
D0H	Program Status Word Register	PSW	R/W	0	0	0	0	0	0	0	0
D1H	Reserved	–	–	–							
D2H	Reserved	–	–	–							
D3H	Reserved	–	–	–							
D4H	Reserved	–	–	–							
D5H	Reserved	–	–	–							
D6H	Comparator Control Register	CMPCR	R/W	0	0	0	0	0	0	0	1
D7H	Comparator Trigger Control Register	CMPTR	R/W	0	0	0	0	0	0	0	0
D8H	Oscillator Control Register	OSCCR	R/W	0	0	0	0	0	1	0	0
D9H	System Clock Control Register	SCCR	R/W	0	1	0	0	0	0	0	0
DAH	Reserved	–	–	–							
DBH	Reserved	–	–	–							
DCH	Reserved	–	–	–							
DDH	Reserved	–	–	–							
DEH	Comparator De-bounce Time Register	CMPDBT	R/W	0	0	0	0	1	1	1	1
DFH	Reserved	–	–	–							
E0H	Accumulator A Register	ACC	R/W	0	0	0	0	0	0	0	0
E1H	USART Control Register 1	UCTRL1	R/W	0	0	0	0	0	0	0	0
E2H	USART Control Register 2	UCTRL2	R/W	0	0	0	0	0	0	0	0
E3H	USART Control Register 3	UCTRL3	R/W	0	0	0	0	–	0	0	0
E4H	USART Status Register	USTAT	R/W	1	0	0	0	0	0	0	0
E5H	USART Baud Rate Generation Register	UBAUD	R/W	1	1	1	1	1	1	1	1
E6H	USART Data Register	UDATA	R/W	0	0	0	0	0	0	0	0
E7H	Reserved	–	–	–							
E8H	Interrupt Priority Register	IP	R/W	0	0	0	0	0	0	0	0
E9H	Interrupt Priority Register 1	IP1	R/W	0	0	0	0	0	0	0	0
EAH	Interrupt Priority Register 2	IP2	R/W	0	0	0	0	0	0	0	0
EBH	USART Control Register 4	UCTRL4	R/W	0	0	0	0	–	0	0	0
ECH	USART Floating Point Counter	FPCR	R/W	0	0	0	0	0	0	0	0
EDH	Reserved	–	–	–							
EEH	I2C Mode Register 1	I2CMR1	R/W	0	0	0	0	0	0	0	0
EFH	Reserved	–	–	–							
F0H	B Register	B	R/W	0	0	0	0	0	0	0	0
F1H	Flash Mode Register	FEMR	R/W	0	0	0	0	0	0	0	0
F2H	Flash Control Register	FECR	R/W	0	0	0	0	0	0	1	1
F3H	Flash Status Register	FESR	R/W	1	0	0	0	0	0	0	0
F4H	Flash Time Control Register	FETCR	R/W	0	0	0	0	0	0	0	0
F5H	Flash Address Low Register	FEARL	R/W	0	0	0	0	0	0	0	0
F6H	Flash Address Middle Register	FEARM	R/W	0	0	0	0	0	0	0	0
F7H	Flash Address High Register	FEARH	R/W	0	0	0	0	0	0	0	0

Table 8.4 SFR Map (Continue)

Address	Function	Symbol	R/W	@Reset							
				7	6	5	4	3	2	1	0
F8H	I2C Status Register	I2CSR	R/W	0	0	0	0	0	0	0	0
F9H	I2C Mode Register	I2CMR	R/W	0	0	0	0	0	0	0	0
FAH	I2C SCL Low Period Register	I2CSCLLR	R/W	0	0	1	1	1	1	1	1
FBH	I2C SCL High Period Register	I2CSCLHR	R/W	0	0	1	1	1	1	1	1
FCH	I2C SDA Hold Time Register	I2CSDAHR	R/W	0	0	0	0	0	0	0	1
FDH	I2C Data Register	I2CDR	R/W	1	1	1	1	1	1	1	1
FEH	I2C Slave Address Register	I2CSAR	R/W	0	0	0	0	0	0	0	0
FFH	I2C Slave Address 1 Register	I2CSAR1	R/W	0	0	0	0	0	0	0	0

Table 8.5 SFR Map (Concluded)

8.4.3 8051 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

PSW (Program Status Word Register) : D0H

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

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
–	–	–	–	–	–	–	DPSEL0
–	–	–	–	–	–	–	RW

Initial value : 00H

DPSEL[2:0] Select Banked Data Pointer Register

DPSEL0	Description
0	DPTR0
1	DPTR1

Reserved

9 I/O Ports

9.1 I/O Ports

The A94B114 has four groups of I/O ports (P0 ~ P2). 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, P1 and P2 include function that can generate interrupt according to change of state of the pin.

9.2 Port Register

9.2.1 Data Register (Px)

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

9.2.2 Direction Register (PxIO)

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

9.2.3 Pull-up Resistor Selection Register (PxPU)

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

9.2.4 Open-drain Selection Register (PxOD)

There are internally open-drain selection registers (PxOD) for P0 ~ P2. 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 De-bounce Enable Register (PxDB)

P0[6:2], P1[3:0], P2[3:0] support debounce function. Debounce clocks of each ports are fx/4, fx/8, fx/16 and fx/32.

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 De-bounce Time Selection Register (DBTSR)

The de-bounce time selection register (DBTSR) select the de-bounce time of all ports and external reset.

9.2.8 Register Map

Name	Address	Direction	Default	Description
P0	80H	R/W	00H	P0 Data Register
P0IO	9BH	R/W	00H	P0 Direction Register
P0PU	9CH	R/W	00H	P0 Pull-up Resistor Selection Register
P0OD	9DH	R/W	00H	P0 Open-drain Selection Register
P0DB	9EH	R/W	00H	P0 De-bounce Enable Register
P0FSRL	B3H	R/W	00H	P0 Function Selection Low Register
P0FSRH	B4H	R/W	00H	P0 Function Selection High Register
P1	88H	R/W	00H	P1 Data Register
P1IO	A3H	R/W	00H	P1 Direction Register
P1PU	A4H	R/W	00H	P1 Pull-up Resistor Selection Register
P1OD	A5H	R/W	00H	P1 Open-drain Selection Register
P1DB	A6H	R/W	00H	P1 De-bounce Enable Register
P1FSRL	B5H	R/W	00H	P1 Function Selection Low Register
P1FSRH	B6H	R/W	00H	P1 Function Selection High Register
P2	90H	R/W	00H	P2 Data Register
P2IO	ABH	R/W	00H	P2 Direction Register
P2PU	ACH	R/W	00H	P2 Pull-up Resistor Selection Register
P2OD	ADH	R/W	00H	P2 Open-drain Selection Register
P2DB	AEH	R/W	00H	P2 De-bounce Enable Register
P2FSR	B7H	R/W	00H	P2 Function Selection Register
DBTSR	84H	R/W	00H	De-bounce Time Selection Register

Table 9.1 Port Register Map

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) : 9BH

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) EINT0, EINT2, EINT11 function possible when input

P0PU (P0 Pull-up Resistor Selection Register) : 9CH

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) : 9DH

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

Initial value : 00H

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

0 Push-pull output

1 Open-drain output

P0DB (P0 De-bounce Enable Register) : 9EH

7	6	5	4	3	2	1	0
P07DB	P06DB	P05DB	P04DB	P03DB	P02DB	P01DB	P00DB
RW	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

P0DB[7:0] Configure De-bounce of P0 Port

0 Disable

1 Enable

NOTE)

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 de-bounce is automatically disabled at stop mode and recovered after stop mode release.
4. This can not work unless the system clock is HFO IRC.
   (HFO : Internal High Frequency Oscillator)

P0FSR_L (Port 0 Function Selection Register Low) : B3H

7	6	5	4	3	2	1	0
P0FSR_L7	P0FSR_L6	P0FSR_L5	P0FSR_L4	P0FSR_L3	P0FSR_L2	P0FSR_L1	P0FSR_L0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

P0FSR_L[7:6] P03 Function Select

P0FSR_L7	P0FSR_L6	Description
0	0	I/O Port (EINT11 function possible when input)
0	1	EINT11 / EC1
1	0	T1O / PWM1O
1	1	AN3

P0FSR_L[5:4] P02 Function Select

P0FSR_L5	P0FSR_L4	Description
0	0	I/O Port (EINT0 function possible when input)
0	1	EINT0
1	0	Not used
1	1	AN2

P0FSR_L[3:2] P01 Function Select

P0FSR_L3	P0FSR_L2	Description
0	0	I/O Port
0	1	XCK
1	0	Not used
1	1	AN1

P0FSR_L[1:0] P00 Function Select

P0FSR_L1	P0FSR_L0	Description
0	0	I/O Port
0	1	SS
1	0	Not used
1	1	AN0

P0FSR_H (Port 0 Function Selection Register High) : B4H

7	6	5	4	3	2	1	0
P0FSR_H7	P0FSR_H6	P0FSR_H5	P0FSR_H4	P0FSR_H3	P0FSR_H2	P0FSR_H1	P0FSR_H0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

P0FSR_H[7:6] P07 Function Select

P0FSR_H7	P0FSR_H6	Description
0	0	I/O Port
0	1	Not used
1	0	CMO
1	1	AN7

P0FSR_H[5:4] P06 Function Select

P0FSR_H5	P0FSR_H4	Description
0	0	I/O Port
0	1	Not used
1	0	CMP
1	1	AN6

P0FSR_H[3:2] P05 Function Select

P0FSR_H3	P0FSR_H2	Description
0	0	I/O Port
0	1	Not used
1	0	CMN0
1	1	AN5

P0FSR_H[1:0] P04 Function Select

P0FSR_H1	P0FSR_H0	Description
0	0	I/O Port (EINT2 function possible when input)
0	1	EINT2
1	0	PWM1OB/CMN1
1	1	AN4

NOTE) P0SFR_H[1:0] = 10’b Function requires caution. The PWM1OB function is the output, and the CMN1 Function is the input. This feature is not compatible at the same time and should not be used at the same time. Each function can be set using different pins. (PWM1OB P04/P23, CMN P04/P05)

9.4 P1 Port

9.4.1 P1 Port Description

P1 is 6-bit I/O port. P1 control registers consist of P1 data register (P1), P1 direction register (P1IO), debounce enable register (P12DB), 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
–	–	P15	P14	P13	P12	P11	P10
–	–	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P1[5:0] I/O Data

P1IO (P1 Direction Register) : A3H

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

Initial value : 00H

P1IO[5:0] P1 Data I/O Direction

0 Input

1 Output

NOTE) ENINT10/EINT12 function possible when input

P1PU (P1 Pull-up Resistor Selection Register) : A4H

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

Initial value : 00H

P1PU[5:0] Configure Pull-up Resistor of P1 Port

0 Disable

1 Enable

P1OD (P1 Open-drain Selection Register) : A5H

7	6	5	4	3	2	1	0
–	–	P15OD	P14OD	P13OD	P12OD	P11OD	P10OD
–	–	RW	RW	RW	RW	RW	RW

Initial value : 00H

P1OD[5:0]	Configure Open-drain of P1 Port
0	Push-pull output
1	Open-drain output

P1DB (P1 De-bounce Enable Register) : A6H

7	6	5	4	3	2	1	0
–	–	P15DB	P14DB	P13DB	P12DB	P11DB	P10DB
–	–	RW	RW	RW	RW	RW	RW

Initial value : 00H

P1DB[5:0]	Configure De-bounce of P1 Port
0	Disable
1	Enable

NOTE)

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 de-bounce is automatically disabled at stop mode and recovered after stop mode release.
4. This can not work unless the system clock is HFO IRC.
   (HFO : Internal High Frequency Oscillator)

P1FSR_L (Port 1 Function Selection Register Low) : B5H

7	6	5	4	3	2	1	0
P1FSR_L7	P1FSR_L6	P1FSR_L5	P1FSR_L4	P1FSR_L3	P1FSR_L2	P1FSR_L1	P1FSR_L0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

P1FSR_L[7:6] P13 Function Select

P1FSR_L7	P1FSR_L6	Description
0	0	I/O Port
0	1	Not used
1	0	Not used
1	1	Not used

P1FSR_L[5:4] P12 Function Select

P1FSR_L5	P1FSR_L4	Description
0	0	I/O Port (EINT12 function possible when input)
0	1	EINT12 / EC2
1	0	T2O / PWM2O
1	1	Not used

P1FSR_L[3:2] P11 Function Select

P1FSR_L3	P1FSR_L2	Description
0	0	I/O Port
0	1	Not used
1	0	T1O / PWM1O
1	1	Not used

P1FSR_L[1:0] P10 Function Select

P1FSR_L1	P1FSR_L0	Description
0	0	I/O Port
0	1	Not used
1	0	T2O / PWM2O
1	1	Not used

****NOTE)**

1. To use T1O/PWM1O as pin3(P11), set Sub-function and set it to Output High.
2. To use T2O/PWM2O as pin2(P10), set Sub-function and set it to Output Low.

P1FSR_H (Port 1 Function Selection Register High) : B6H

7	6	5	4	3	2	1	0
–	–	P1FSR_H5	P1FSR_H4	P1FSR_H3	P1FSR_H2	P1FSR_H1	P1FSR_H0
–	–	RW	RW	RW	RW	RW	RW

Initial value : 00H

P1FSR_H[5:4] I2C Port Select

P1FSR_H5	P1FSR_H4	Description
0	0	Not used
0	1	Use P20, P21 for I2C
1	0	Use P14, P15 for I2C
1	1	Not used

P1FSR_H[3:2] P15 Function Select

P1FSR_H3	P1FSR_H2	Description
0	0	I/O Port
0	1	TXD
1	0	Not used
1	1	Not used

P1FSR_H[1:0] P14 Function Select

P1FSR_H1	P1FSR_H0	Description
0	0	I/O Port (EINT10 function possible when input)
0	1	RXD
1	0	Not used
1	1	EINT10 / EC0

9.5 P2 Port

9.5.1 P2 Port Description

P2 is 4-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
–	–	–	–	P23	P22	P21	P20
–	–	–	–	R/W	R/W	R/W	R/W

Initial value : 00H

P2[3:0] I/O Data

P2IO (P2 Direction Register) : ABH

7	6	5	4	3	2	1	0
–	–	–	–	P23IO	P22IO	P21IO	P20IO
–	–	–	–	R/W	R/W	R/W	R/W

Initial value : 00H

P2IO[3:0]	P2 Data I/O Direction
0	Input
1	Output

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

7	6	5	4	3	2	1	0
–	–	–	–	P23PU	P22PU	P21PU	P20PU
–	–	–	–	R/W	R/W	R/W	R/W

Initial value : 00H

P2PU[3:0]	Configure Pull-up Resistor of P2 Port
0	Disable
1	Enable

P2OD (P2 Open-drain Selection Register) : ADH

7	6	5	4	3	2	1	0
–	–	–	–	P23OD	P22OD	P21OD	P20OD
–	–	–	–	R/W	R/W	R/W	R/W

Initial value : 00H

P2OD[3:0]	Configure Open-drain of P2 Port
0	Push-pull output
1	Open-drain output

P2DB (P2 De-bounce Enable Register) : AEH

7	6	5	4	3	2	1	0
–	–	–	–	P23DB	P22DB	P21DB	P20DB
–	–	–	–	RW	RW	RW	RW

Initial value : 00H

P2DB	Configure De-bounce of P2 Port	
	0	Disable
	1	Enable

NOTE)

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 de-bounce is automatically disabled at stop mode and recovered after stop mode release.
4. This can not work unless the system clock is HFO IRC.
   (HFO : Internal High Frequency Oscillator)

P2FSR (Port 2 Function Selection Register) : B7H

7	6	5	4	3	2	1	0
P2FSR7	P2FSR6	P2FSR5	P2FSR4	P2FSR3	P2FSR2	P2FSR1	P2FSR0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

P2FSR[7:6]	P23 Function Select		
	P2FSR7	P2FSR6	Description
	0	0	I/O Port (EINT10 function possible when input)
	0	1	EINT10 / EC0
	1	0	T0O / PWM1OB
	1	1	AN9
P2FSR[5:4]	P22 Function Select		
	P2FSR5	P2FSR4	Description
	0	0	I/O Port (EINT1 function possible when input)
	0	1	EINT1
	1	0	Not used
	1	1	AN8
P2FSR[3:2]	P21 Function Select		
	P2FSR3	P2FSR2	Description
	0	0	I/O Port
	0	1	Not used
	1	0	Not used
	1	1	Not used
P2FSR[1:0]	P20 Function Select		
	P2FSR1	P2FSR0	Description
	0	0	I/O Port
	0	1	Not used
	1	0	Not used
	1	1	Not used

10 Interrupt Controller

10.1 Overview

The A94B114 supports up to 16 interrupt sources. The interrupts have separate enable register bits associated with them, allowing software control. They can also have two 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 16 interrupt source
- 2 priority levels
- Multi Interrupt possibility
- 4 interrrupt nesting levels
- 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 two pair of interrupt enable registers (IE, IE1, IE2). Each bit of IE, IE1, IE2 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 A94B114 supports a two-level priority scheme. Each maskable interrupt is individually assigned to one of four priority levels according to IP, IP1 and IP2.

Figure 10.1 shows th the Interrupt Priority Level. Priority can be sets by writing to a bit of IP0, IP1 and IP2 regisgter. Each bit of IP0, IP1 and IP2 corresponds to each interrupt and desiges one of 2 priority levels of each interrupt. High level interrupt always has higher priority than low level interrupt.

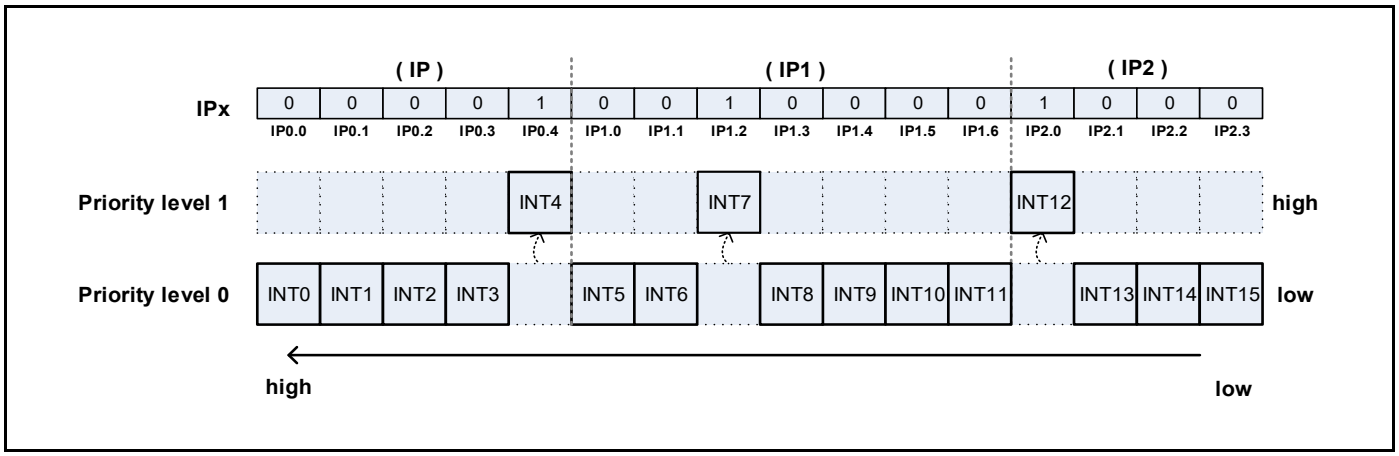


Figure 10.1 Interrupt Priority Level

This device support only 4 level interrupt nesting. Interrupt nesting level register (ILVL) has the current nesting level. If current nesting level is 4 and other interrupt having higher priority occur, it might cause a malfunction.

10.2 External Interrupt

The external interrupt on INT0, INT1, INT2, INT10, INT11 and INT12 pins receive various interrupt request depending on the external interrupt polarity 0 register (EIPOL0) and external interrupt polarity 1 register (EIPOL1) as shown in Figure 10.2. Also each external interrupt source has enable/disable bits. The external interrupt flag register (EIFLAG) and provides the status of external interrupts.

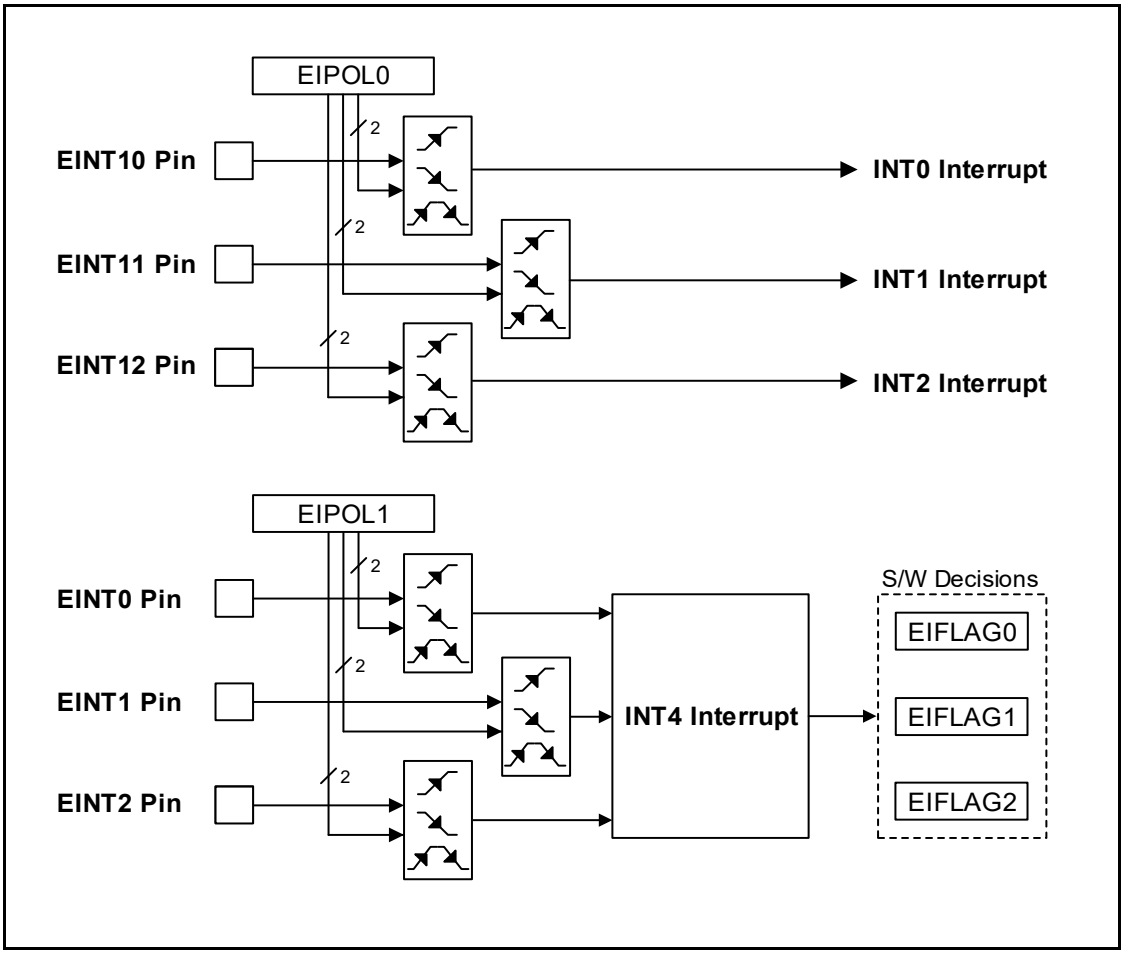


Figure 10.2 External Interrupt Description

NOTE) EINT0~EINT2 can be ignored if interrupts are requested at the same time. If an interrupt request is requested and another interrupt is requested immediately before clearing (sysclk1-2), the later requested interrupt is ignored.

10.3 Block Diagram

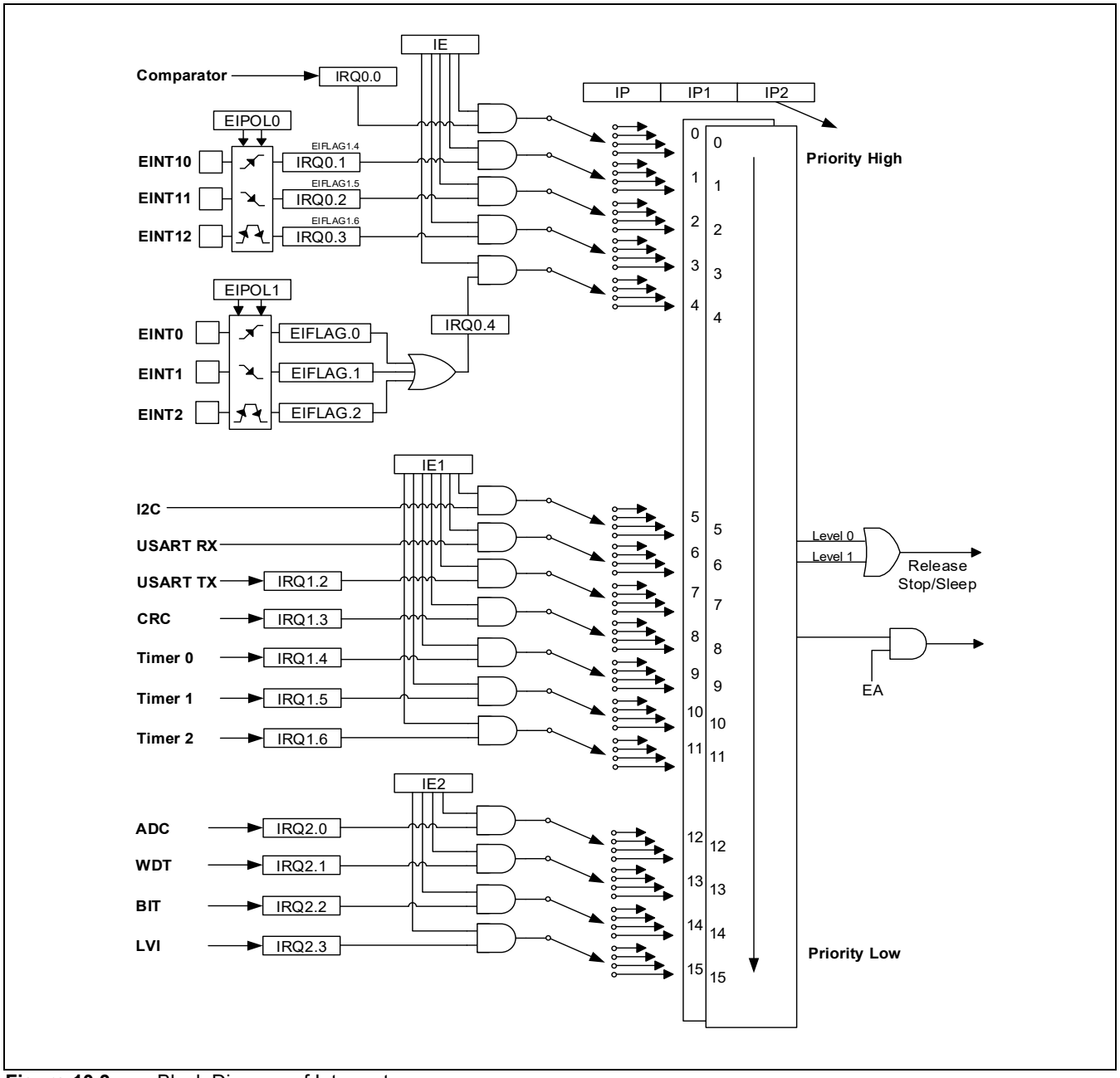


Figure 10.3 Block Diagram of Interrupt

NOTE)

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, IP, IP1, IP2 and PCON register.

10.4 Interrupt Vector Table

The interrupt controller supports 16 interrupt sources as shown in the Table 10-1. 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.

Interrupt Source	Symbol	Interrupt Enable Bit	Polarity	Mask	Vector Address
Hardware Reset	RESETB	-	0	Non-Maskable	0000H
Comparator Interrupt	INT0	IE.0	1	Maskable	0003H
External Interrupt 10	INT1	IE.1	2	Maskable	000BH
External Interrupt 11	INT2	IE.2	3	Maskable	0013H
External Interrupt 12	INT3	IE.3	4	Maskable	001BH
External Interrupt 0/1/2	INT4	IE.4	5	Maskable	0023H
I2C Interrupt	INT5	IE1.0	6	Maskable	002BH
USART Rx Interrupt	INT6	IE1.1	7	Maskable	0033H
USART Tx Interrupt	INT7	IE1.2	8	Maskable	003BH
CRC Interrupt	INT8	IE1.3	9	Maskable	0043H
T0 Match Interrupt	INT9	IE1.4	10	Maskable	004BH
T1 Match Interrupt	INT10	IE1.5	11	Maskable	0053H
T2 Match Interrupt	INT11	IE1.6	12	Maskable	005BH
ADC Interrupt	INT12	IE2.0	13	Maskable	0063H
WDT Interrupt	INT13	IE2.1	14	Maskable	006BH
BIT Interrupt	INT14	IE2.2	15	Maskable	0073H
LVI Interrupt	INT15	IE2.3	16	Maskable	007BH

Table 10.1 Interrupt Vector Address Table

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.

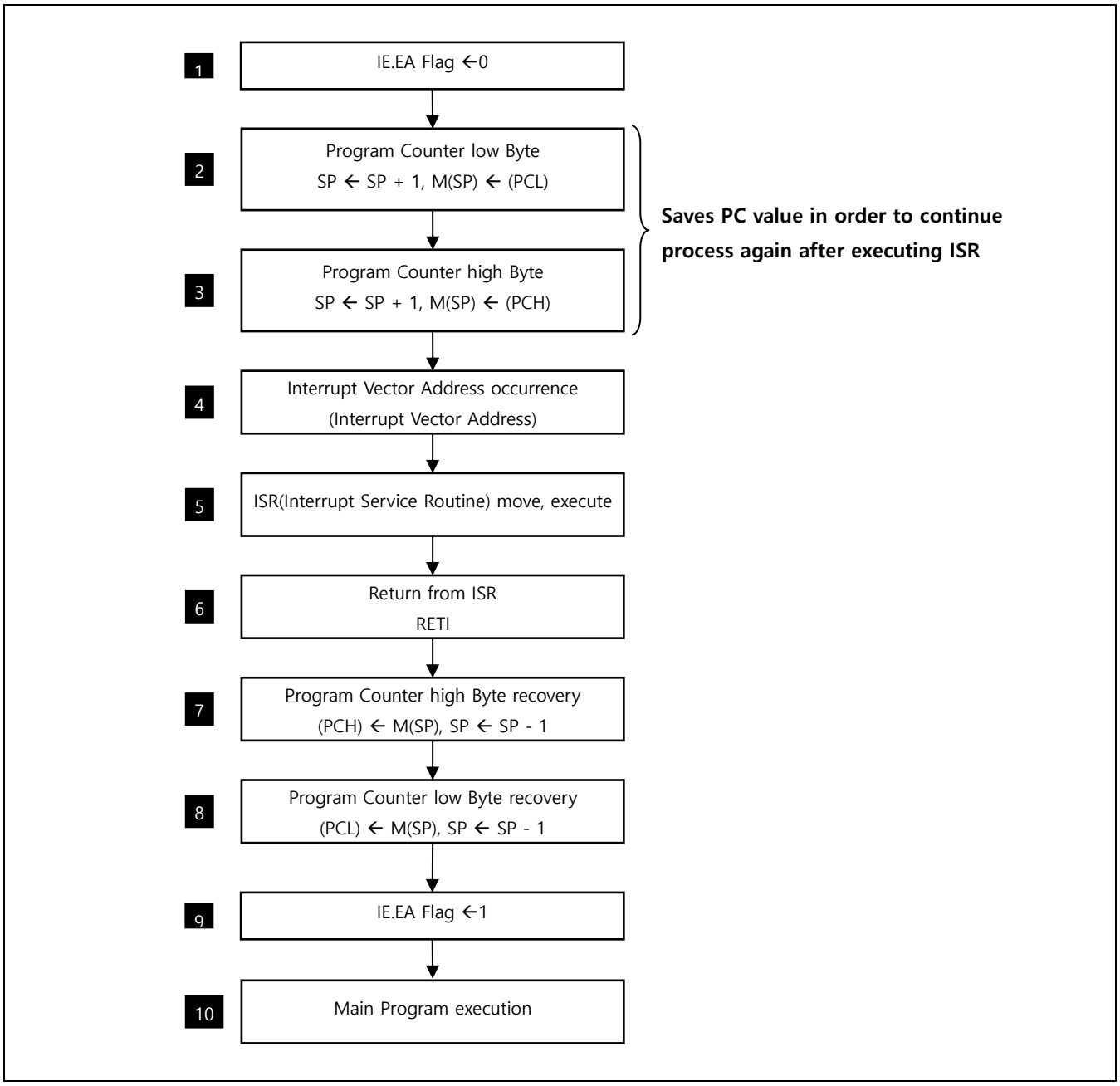


Figure 10.4 Interrupt Sequence Flow

10.6 Effective Timing after Controlling Interrupt Bit

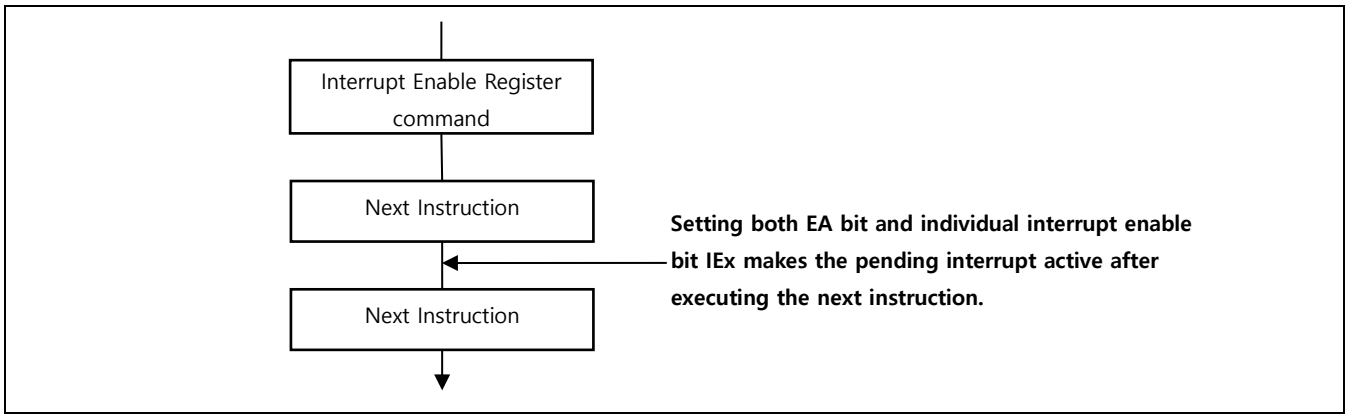
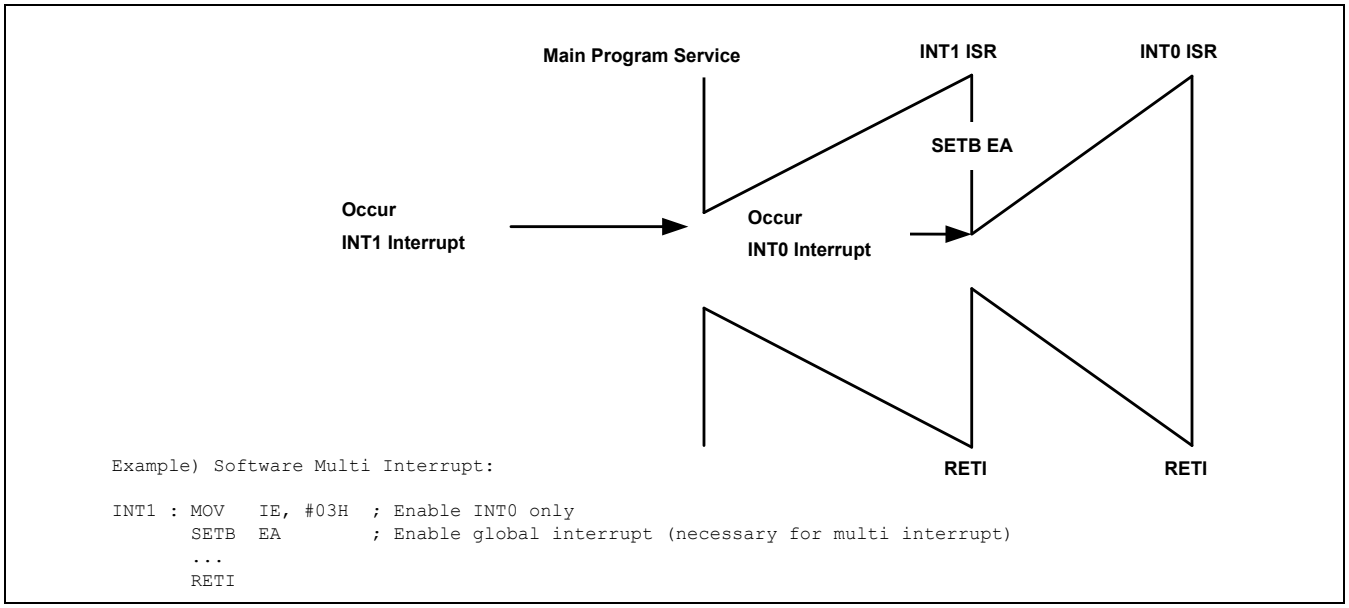


Figure 10.5 Effective Timing of Interrupt Enable Register

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.




```
Example) Software Multi Interrupt:

INT1 : MOV   IE, #03H  ; Enable INT0 only
       SETB  EA        ; Enable global interrupt (necessary for multi interrupt)
       ...
       RETI
```

Figure 10.6 Effective Timing of Multi-Interrupt

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

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

10.8 Interrupt Enable Accept Timing

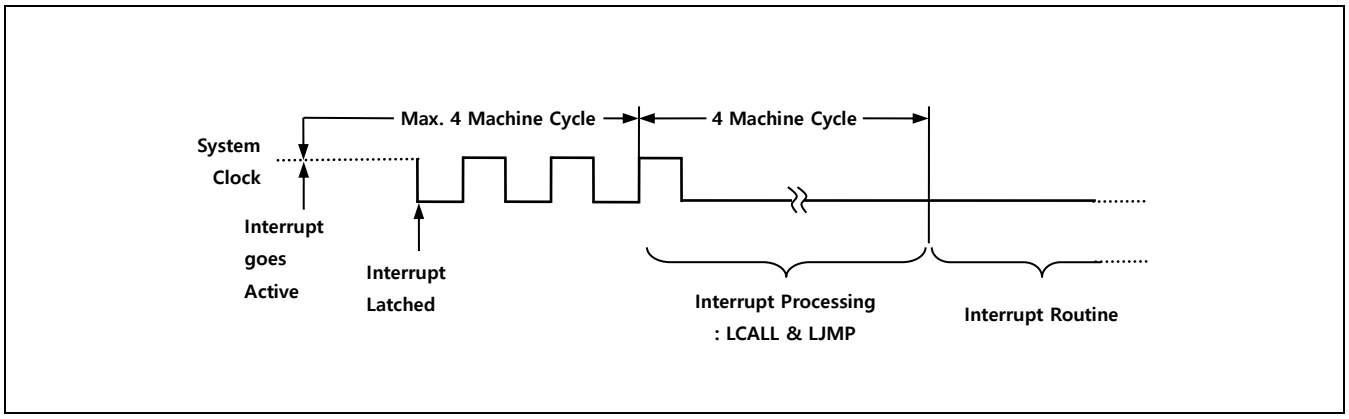


Figure 10.7 Interrupt Response Timing Diagram

10.9 Interrupt Service Routine Address

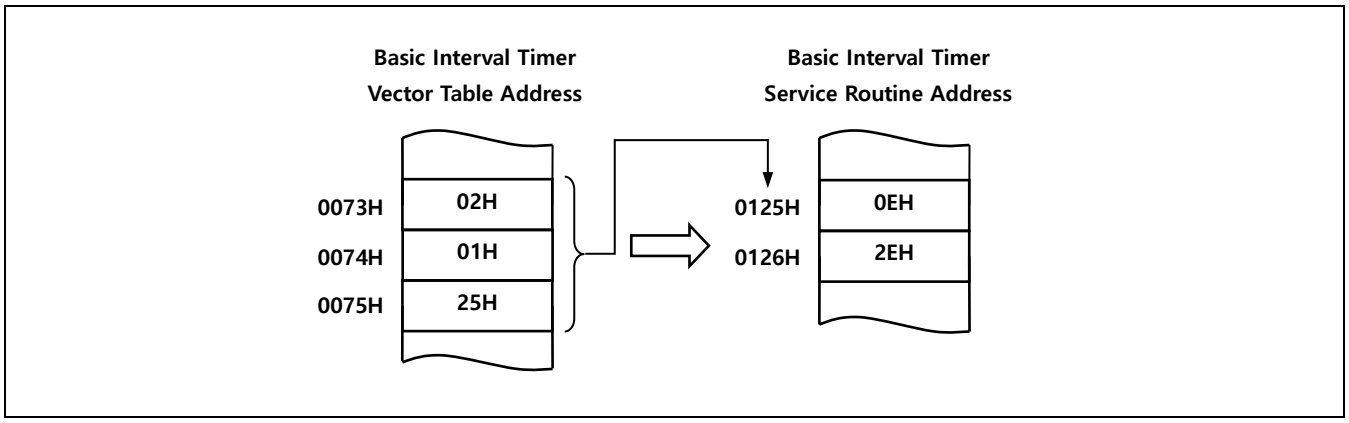


Figure 10.8 Correspondence between Vector Table Address and the Entry Address of ISR

10.10 Saving/Restore General-Purpose Registers

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


Interrupt_Processing:
          .
          .
         POP  ACC
         POP  B
         POP  DPH
         POP  DPL
         POP  PSW
         RETI
```

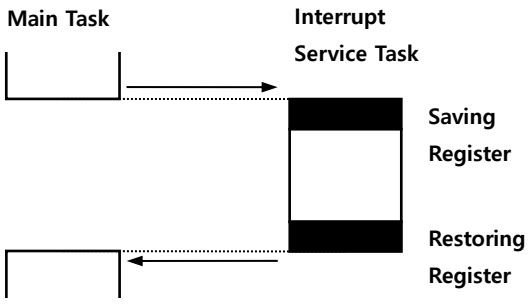


Figure 10.9 Saving/Restore Process Diagram and Sample Source

10.11 Interrupt Timing

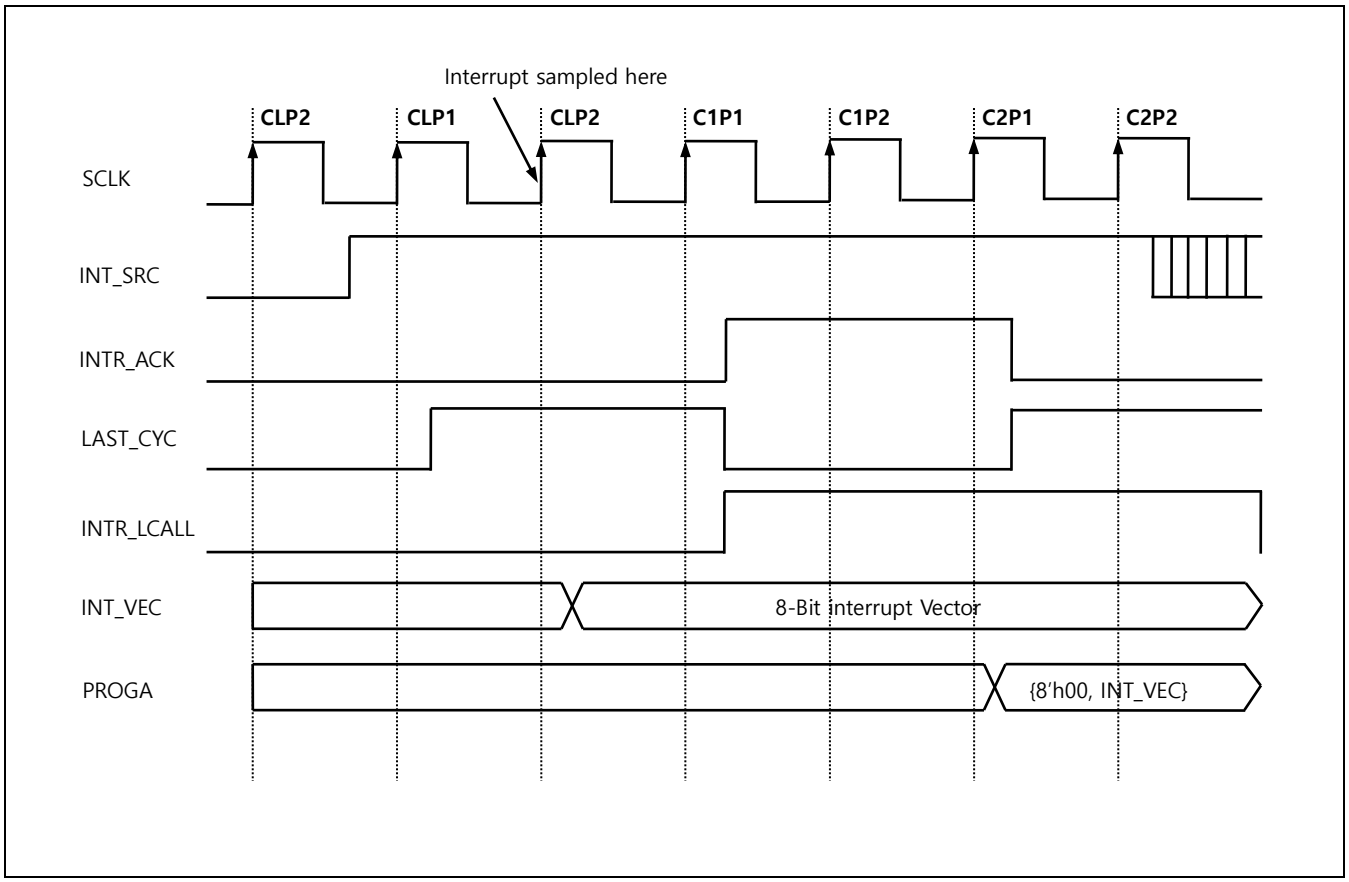


Figure 10.10 Timing Chart of Interrupt Acceptance and Interrupt Return Instruction

Interrupt sources are sampled at the last cycle of a command. When sampling interrupt source, it is decieded to low 8-bit of interrupt vector. CM8051-S core makes interrupt acknowleadge at first cycle of command, executes long call to jump interrupt routine as INT_VEC.

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)

Interrupt enable register consists of global interrupt control bit (EA) and peripheral interrupt control bits. Total 16 peripherals are able to control interrupt.

10.12.2 Interrupt Priority Register (IP, IP1, IP2)

The 15 interrupts can decide 2 levels interrupt priority using interrupt priority register. Level 1 is the high priority, while level 0 is the lowest priority. After a reset IP, IP1 and IP2 are cleared to '00H'. At that initialization, low interrupt number has a higher priority than high interrupt number. If decided the priority, low interrupt number has a higher priority than high interrupt number in the same level.

10.12.3 Interrupt Request Register (IRQ0, IRQ1, IRQ2)

Interrupt requeset register is set by 'Peripherals'(H/W) when some event is occurred. The interrupt function is called when both interrupt enable and request bits are set.

NOTE) In polling mode using IRQ, if there is another interrupt request after reading the IRQ and before writing it, the second requested IRQ may be cleared.

10.12.4 Interrupt Offset Register (IOFFSET)

Interrupt Offset changing is possible using IOFFSET register.

10.12.5 Interrupt Nesting level Register (ILVL)

Interrupt nesting level register has current interupt nesting level.

10.12.6 External Interrupt Flag Register (EIFLAG)

The external interrupt flag register (EIFLAG) is set to '1' when the external interrupt0, 1 and 2 generating condition is satisfied. The external interrupt 0, 1, 2 share the same interrupt vector address and can check the interrupt source through the EIFLAG register. The flag can be cleared by writing '0' to it.

10.12.7 External Interrupt Polarity Register (EIPOL0, EIPOL1)

The external interrupt polarity 0 high/low register (EIPOL0), external interrupt polarity 1 register (EIPOL1) determines which type of rising/falling/both edge interrupt. Initially, default value is no interrupt at any edge.

10.12.8 Register Map

Name	Address	Direction	Default	Description
IE	B8H	R/W	00H	Interrupt Enable Register
IE1	C0H	R/W	00H	Interrupt Enable Register 1
IE2	C8H	R/W	00H	Interrupt Enable Register 2
IP	E8H	R/W	00H	Interrupt Priority Register
IP1	E9H	R/W	00H	Interrupt Priority Register 1
IP2	EAH	R/W	00H	Interrupt Priority Register 2
IRQ	A0H	R/W	00H	Interrupt Request Register
IRQ1	A8H	R/W	00H	Interrupt Request Register 1
IRQ2	B0H	R/W	00H	Interrupt Request Register 2
IOFFSET	8AH	R/W	00H	Interrupt Offset Register
ILVL	9AH	R/W	00H	Interrupt Level Register
EIFLAG	C9H	R/W	00H	External Interrupt Flag Register
EIPOL0	8BH	R/W	00H	External Interrupt Polarity 0 Register
EIPOL1	8CH	R/W	00H	External Interrupt Polarity 1 Register

Table 10.2 Interrupt Register Map

10.12.9 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) and interrupt enable register 2 (IE2). For external interrupt, it consists external interrupt polarity 0 register (EIPOL0) and external interrupt polarity 1 register (EIPOL1) .

IE (Interrupt Enable Register) : B8H

7	6	5	4	3	2	1	0
EA	–	–	INT4E	INT3E	INT2E	INT1E	INT0E
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

INT4E Enable or Disable External Interrupt 0/1/2 (EINT0/EINT1/EINT2)
- 0 Disable
- 1 Enable

INT3E Enable or Disable External Interrupt 12 (EINT12)
- 0 Disable
- 1 Enable

INT2E Enable or Disable External Interrupt 11(EINT11)
- 0 Disable
- 1 Enable

INT1E Enable or Disable External Interrupt 10(EINT10)
- 0 Disable
- 1 Enable

INT0E Enable or Disable Comparator Interrupt
- 0 Disable
- 1 Enable

IE1 (Interrupt Enable Register 1): C0H

7	6	5	4	3	2	1	0
–	INT11E	INT10E	INT9E	INT8E	INT7E	INT6E	INT5E
–	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

INT11E Enable or Disable Timer2 Interrupt
- 0 Disable
- 1 Enable

INT10E Enable or Disable Timer1 Interrupt
- 0 Disable
- 1 Enable

INT9E Enable or Disable Timer 0 Interrupt
- 0 Disable
- 1 Enable

INT8E Enable or Disable CRC Interrupt
- 0 Disable
- 1 Enable

INT7E Enable or Disable USART Tx Interrupt
- 0 Disable
- 1 Enable

INT6E Enable or Disable USART Rx Interrupt
- 0 Disable
- 1 Enable

INT5E Enable or Disable I2C Interrupt
- 0 Disable
- 1 Enable

IE2 (Interrupt Enable Register 2) : C8H

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

Initial value : 00H

INT15E Enable or Disable LVI Interrupt
- 0 Disable
- 1 Enable

INT14E Enable or Disable BIT Interrupt
- 0 Disable
- 1 Enable

INT13E Enable or Disable WDT Interrupt
- 0 Disable
- 1 Enable

INT12E Enable or Disable ADC Interrupt
- 0 Disable
- 1 Enable

IP (Interrupt Priority Register) : E8H

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

Initial value : 00H

IP4 Select EINT0/1/2 Interrupt Priority
- 0 Level 0
- 1 Level 1

IP3 Select EINT12 Interrupt Priority
- 0 Level 0
- 1 Level 1

IP2 Select EINT11 Interrupt Priority
- 0 Level 0
- 1 Level 1

IP1 Select EINT10 Interrupt Priority
- 0 Level 0
- 1 Level 1

IP0 Select Comparator Interrupt Priority
- 0 Level 0
- 1 Level 1

IP1 (Interrupt Priority Register 1) : E9H

7	6	5	4	3	2	1	0
–	IP16	IP15	IP14	IP13	IP12	IP11	IP10
–	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

IP16 Select Timer2 Interrupt Priority
- 0 Level 0
- 1 Level 1

IP15 Select Timer1 Interrupt Priority
- 0 Level 0
- 1 Level 1

IP14 Select Timer0 Interrupt Priority
- 0 Level 0
- 1 Level 1

IP13 Select CRC Interrupt Priority
- 0 Level 0
- 1 Level 1

IP12 Select USART Tx Interrupt Priority
- 0 Level 0
- 1 Level 1

IP11 Select USART Rx Interrupt Priority
- 0 Level 0
- 1 Level 1

IP10 Select I2C Interrupt Priority
- 0 Level 0
- 1 Level 1

IP2 (Interrupt Priority Register 2) : EAH

7	6	5	4	3	2	1	0
–	–	–	–	IP23	IP22	IP21	IP20
–	–	–	–	RW	RW	RW	RW

Initial value : 00H

IP23 Select LVI Interrupt Priority
- 0 Level 0
- 1 Level 1

IP22 Select BIT Interrupt Priority
- 0 Level 0
- 1 Level 1

IP21 Select WDT Interrupt Priority
- 0 Level 0
- 1 Level 1

IP20 Select ADC Interrupt Priority
- 0 Level 0
- 1 Level 1

IRQ (Interrupt Request Register) : A0H

7	6	5	4	3	2	1	0
–	–	–	IRQ4	IRQ3	IRQ2	IRQ1	IRQ0
–	–	–	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

IRQ4 If EINT0/1/2 interrupt occurs, the flag is set '1'
Flag is cleared when ISR occurs or by writing '0' to bit.
0 EINT0/1/2 Interrupt doesn't occur
1 EINT0/1/2 Interrupt occur

IRQ3 If EINT12 interrupt occurs, the flag is set '1'
Flag is cleared when ISR occurs or by writing '0' to bit.
0 EINT12 Interrupt doesn't occur
1 EINT12 Interrupt occur

IRQ2 If EINT11 interrupt occurs, the flag is set '1'
Flag is cleared when ISR occurs or by writing '0' to bit.
0 EINT11 Interrupt doesn't occur
1 EINT11 Interrupt occur

IRQ1 If EINT10 interrupt occurs, the flag is set '1'
Flag is cleared when ISR occurs or by writing '0' to bit.
0 EINT10 Interrupt doesn't occur
1 EINT10 Interrupt occur

IRQ0 If Comparator interrupt occurs, the flag is set '1'
Flag is cleared when ISR occurs or by writing '0' to bit.
0 Comparator Interrupt doesn't occur
1 Comparator Interrupt occur

IRQ1 (Interrupt Request Register 1) : A8H

7	6	5	4	3	2	1	0
–	IRQ16	IRQ15	IRQ14	IRQ13	IRQ12	IRQ11	IRQ10
–	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

IRQ16 If Timer2 interrupt occurs, the flag is set '1'
Flag is cleared when ISR occurs or by writing '0' to bit.
0 Timer2 Interrupt doesn't occur
1 Timer2 Interrupt occur

IRQ15 If Timer1 interrupt occurs, the flag is set '1'
Flag is cleared when ISR occurs or by writing '0' to bit.
0 Timer1 Interrupt doesn't occur
1 Timer1 Interrupt occur

IRQ14 If Timer0 interrupt occurs, the flag is set '1'
Flag is cleared when ISR occurs or by writing '0' to bit.
0 Timer0 Interrupt doesn't occur
1 Timer0 Interrupt occur

IRQ13 If CRC interrupt occurs, the flag is set '1'
Flag is cleared when ISR occurs or by writing '0' to bit.
0 CRC Interrupt doesn't occur
1 CRC Interrupt occur

IRQ12 If USART Tx interrupt occurs, the flag is set '1'
Flag is cleared when ISR occurs or by writing '0' to bit.
0 USART Tx Interrupt doesn't occur
1 USART Tx Interrupt occur

IRQ11 If USART Rx interrupt occurs, the flag is set '1'
Flag is cleared when ISR occurs or by writing '0' to bit.
0 USART Rx Interrupt doesn't occur
1 USART Rx Interrupt occur

IRQ10 If I2C interrupt occurs, the flag is set '1'
Flag is cleared when ISR occurs or by writing '0' to bit.
0 I2C Interrupt doesn't occur
1 I2C Interrupt occur

IRQ2 (Interrupt Request Register 2) : B0H

7	6	5	4	3	2	1	0
–	–	–	–	IRQ23	IRQ22	IRQ21	IRQ20
–	–	–	–	R/W	R/W	R/W	R/W

Initial value : 00H

IRQ23 If LVI interrupt occurs, the flag is set ‘1’
Flag is cleared when ISR occurs or by writing ‘0’ to bit.
0 LVI Interrupt doesn’t occur
1 LVI Interrupt occur

IRQ22 If BIT interrupt occurs, the flag is set ‘1’
Flag is cleared when ISR occurs or by writing ‘0’ to bit.
0 BIT Interrupt doesn’t occur
1 BIT Interrupt occur

IRQ21 If WDT interrupt occurs, the flag is set ‘1’
Flag is cleared when ISR occurs or by writing ‘0’ to bit.
0 WDT Interrupt doesn’t occur
1 WDT Interrupt occur

IRQ20 If ADC interrupt occurs, the flag is set ‘1’
Flag is cleared when ISR occurs or by writing ‘0’ to bit.
0 ADC Interrupt doesn’t occur
1 ADC Interrupt occur

IOFFSET (Interrupt Offset Register) : 8AH

7	6	5	4	3	2	1	0
IOFFSET7	IOFFSET6	IOFFSET5	IOFFSET4	IOFFSET3	IOFFSET2	IOFFSET1	IOFFSET0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

IOFFSET[7:0] Interrupt Offset Value
Interrupt Vector Address = 256*INT_OFFSET
Interrupt offset changing is possible when only the SYSCON_AR value is 5AH.
So. User must set it with 5Ah before changing interrupt offset value and make sure set it with 00h after changing interrupt offset.

ILVL (Interrupt Level Register) : 9AH

7	6	5	4	3	2	1	0
–	–	–	–	–	ILVL2	ILVL1	ILVL0
–	–	–	–	–	R	R	R

Initial value : 00H

ILVL[7:0] Interrupt Level Value
Interrupt level is saved when using multi-interrupt.
It has a value of 4level from 0H to 4H.

EIFLAG (External Interrupt Flag Register) : C9H

7	6	5	4	3	2	1	0
–	–	–	–	–	EIFLAG2	EIFLAG1	EIFLAG0
–	–	–	–	–	R/W	R/W	R/W

Initial value : 00H

EIFLAG2 When an External Interrupt 2 is occurred, the flag becomes ‘1’.
The flag is clear only by writing ‘0’ to the bit. The flag should clear by software. Writing “1” has no effect.
0 External Interrupt 2 not occurred
1 External Interrupt 2 occurred

EIFLAG1 When an External Interrupt 1 is occurred, the flag becomes ‘1’.
The flag is clear only by writing ‘0’ to the bit. The flag should clear by software. Writing “1” has no effect.
0 External Interrupt 1 not occurred
1 External Interrupt 1 occurred

EIFLAG0 When an External Interrupt 0 is occurred, the flag becomes ‘1’.
The flag is clear only by writing ‘0’ to the bit. The flag should clear by software. Writing “1” has no effect.
0 External Interrupt 0 not occurred
1 External Interrupt 0 occurred

NOTE) EINT0~EINT2 can be ignored if interrupts are requested at the same time. If an interrupt request is requested and another interrupt is requested immediately before clearing (sysclk1-2), the later requested interrupt is ignored.

EIPOL0 (External Interrupt Polarity High Register): 8BH

7	6	5	4	3	2	1	0
–	–	EIPOL05	EIPOL04	EIPOL03	EIPOL02	EIPOL01	EIPOL00
–	–	RW	RW	RW	RW	RW	RW

Initial value: 00H

EIPOL0[5:4] External interrupt (EINT12) polarity selection

EIPOL0[5:4]		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

EIPOL0[3:2] External interrupt (EINT11) polarity selection

EIPOL0[3:2]		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

EIPOL0[1:0] External interrupt (EINT10) polarity selection

EIPOL0[1:2]		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

EIPOL1 (External Interrupt Polarity Register 1): 8CH

7	6	5	4	3	2	1	0
–	–	EIPOL15	EIPOL14	EIPOL13	EIPOL12	EIPOL11	EIPOL10
–	–	RW	RW	RW	RW	RW	RW

Initial value: 00H

EIPOL1[5:4] External interrupt (EINT2) polarity selection

EIPOL1[5:4]		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

EIPOL1[3:2] External interrupt (EINT1) polarity selection

EIPOL1[3:2]		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

EIPOL1[1:0] External interrupt (EINT0) polarity selection

EIPOL1[1:2]		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

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-frequency clock oscillator. The main clock operation can be easily obtained by attaching a crystal between the XIN and XOUT pin, respectively.

The main clock can be also obtained from the external oscillator. In this case, it is necessary to put the external clock signal into the XIN pin and open the XOUT pin. The default system clock is 32MHz INT-RC Oscillator and the default division rate is two. In order to stabilize system internally, it is used 32MHz INT-RC oscillator on POR.

- Calibrated Internal RC Oscillator (HFO) : 32MHz
  - INT-RC OSC/2 (16 MHz, Default system clock)
  - INT-RC OSC/2 (8 MHz)
  - INT-RC OSC/4 (4 MHz)
  - INT-RC OSC/16 (1 MHz)
- Main Crystal Oscillator (0.4~16 MHz)
- Internal Ring Oscillator (LFO) : 256 kHz

11.1.2 Block Diagram

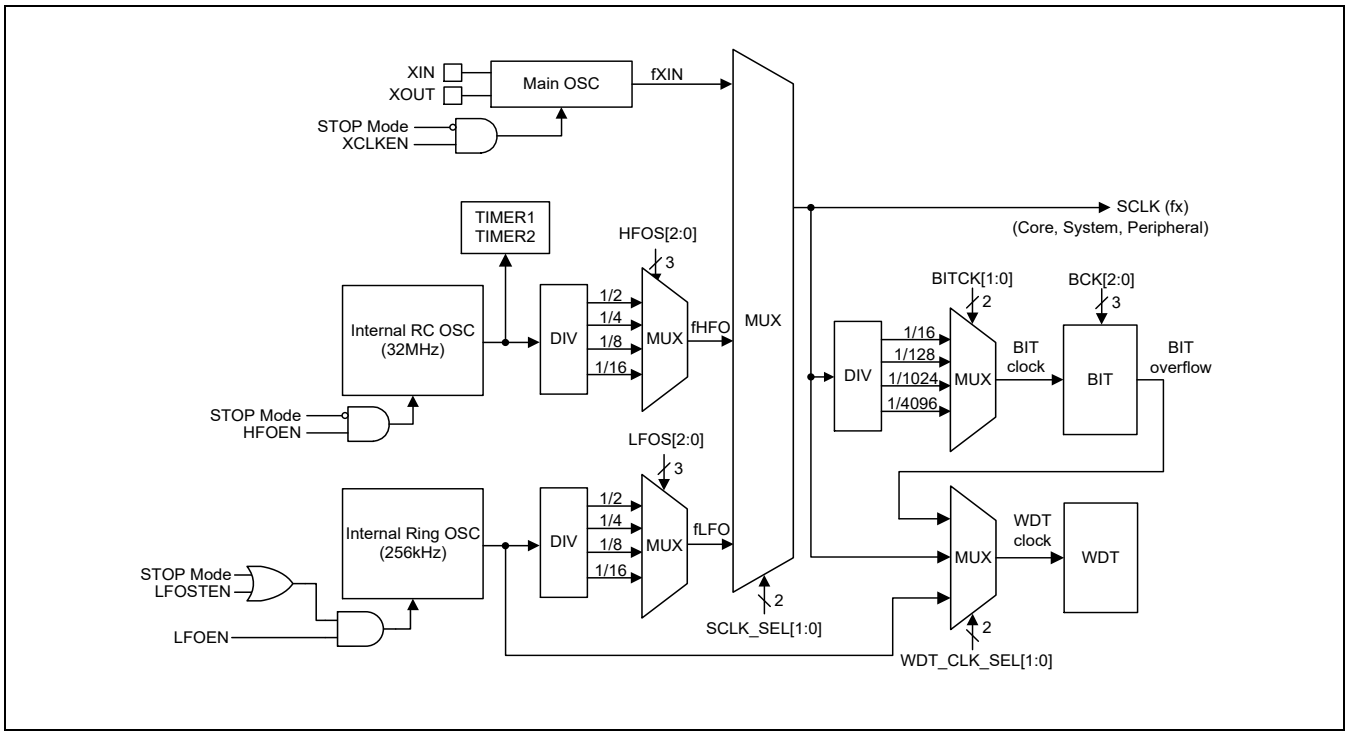


Figure 11.1 Clock Generator Block Diagram

11.1.3 Register Map

Name	Address	Direction	Default	Description
SCCR	D9H	R/W	40H	System and Clock Control Register
OSCCR	D8H	R/W	04H	Oscillator Control Register
SYSCON_AR	8FH	R/W	00H	System control access authorization register.

Table 11.1 Clock Generator Register Map

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) : D9H

7	6	5	4	3	2	1	0
–	FWKTIME2	FWKTIME1	FWKTIME0	–	LFOSTEN	SCLK1	SCLK0
–	R/W	R/W	R/W	–	R/W	R/W	R/W

Initial value : 40H

FWKTIME[2:0] Select stop release time for fast wakeup (fx=16MHz)
Release time varies depending on the system clock.

FWKTIEM[2:0]			Description
0	0	0	128us
0	0	1	256us
0	1	0	512us
0	1	1	4ms
1	0	0	8ms (Default)
1	0	1	16ms
1	1	0	32ms
1	1	1	64ms

LFOSTEN Internal Ring OSC (LFO) enable at STOP mode.

0	Disable at STOP mode.
1	Enable at STOP mode.

SCLK [1:0] System Clock Selection Bit

SCLK1	SCLK0	Description
0	0	INT RC OSC (f_{HFO}) for system clock
0	1	External Main OSC (f_{XIN}) for system clock
1	0	INT Ring OSC (f_{LFO}) for system clock
1	1	Not used

OSCCR (Oscillator Control Register) : D8H

7	6	5	4	3	2	1	0
LFOS1	LFOS0	HFOS1	HFOS0	–	HFOEN	XCLKEN	LFOEN
R/W	R/W	R/W	R/W	–	R/W	R/W	R/W

Initial value : 04H

LFOS[1:0] Internal Ring OSC (LFO) Post-divider Selection

LFOS1	LFOS0	Description
0	0	f_{LFO} / 2 (128kHz - Default)
0	1	f_{LFO} / 4 (64kHz)
1	0	f_{LFO} / 8 (32kHz)
1	1	f_{LFO} / 16 (16kHz)

HFOS[1:0] Internal RC OSC (HFO) Post-divider Selection

HFOS1	HFOS0	Description
0	0	f_{HFO} / 2 (16MHz - Default)
0	1	f_{HFO} / 4 (8MHz)
1	0	f_{HFO} / 8 (4MHz)
1	1	f_{HFO} / 16 (1MHz)

HFOEN Control the Operation of the Internal RC Oscillator (HFO)

0	Disable operation of INT-RC OSC
1	Enable operation of INT-RC OSC

XCLKE Control the Operation of the External Main Oscillator

0	Disable operation of Crystal
1	Enable operation of Crystal

LFOEN Control the Operation of the Internal Ring Oscillator (LFO)

0	Disable operation of INT-Ring OSC
1	Enable operation of INT-Rring OSC

SYSCON_AR (System control access authorization register) : 8FH

7	6	5	4	3	2	1	0
BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

The following registers can be written only when authorized.
The registers are SCCR, OSCCR, IOFFSET. These register changing are possible when only SYSCON_AR value is 5AH. So, User must set it with 5AH before changing value and make sure set it with 00h after changing value.

11.2 Basic Interval Timer

11.2.1 Overview

The A94B114 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.

The A94B114 has these basic interval timer (BIT) features:

- As timer function, timer interrupt occurrence

NOTE) Since BIT Clock uses system clock, even if LFO (256kHz) is used as system clock, it can not be used in STOP Mode

11.2.2 Block Diagram

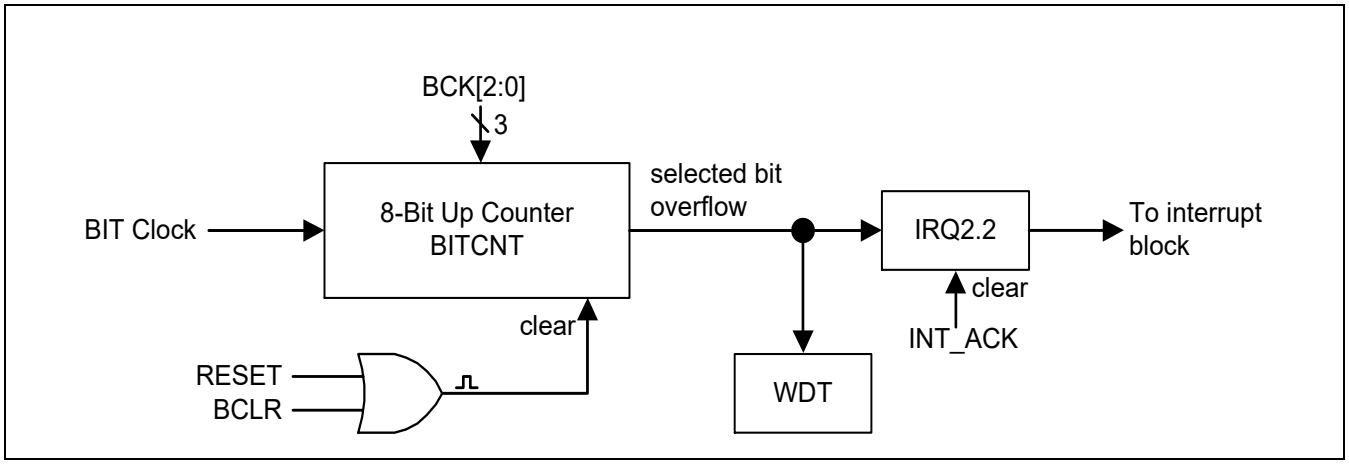


Figure 11.2 Basic Interval Timer Block Diagram

11.2.3 Register Map

Name	Address	Direction	Default	Description
BITCNT	85H	R	00H	Basic Interval Timer Counter Register
BITCR	86H	R/W	04H	Basic Interval Timer Control Register

Table 11.2 Basic Interval Timer Register Map

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 : 04H

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

0 BIT interrupt no generation

1 BIT interrupt generation

BITCK[1:0] Select BIT clock source

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

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

0 Free Running

1 Clear Counter

BCK[2:0] Select BIT overflow period

BCK2	BCK1	BCK0	Description
0	0	0	Bit 0 overflow (BIT Clock * 2)
0	0	1	Bit 1 overflow (BIT Clock * 4)
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) (Default)
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 or watch dog timer mode as setting WDT_RESET_EN bit. If WDT_CLR 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 WDTR, the interrupt request flag is generated. This can be used as Watchdog timer interrupt or reset of CPU in accordance with the bit WDT_RESET_EN.

The clock source of the watchdog timer selects one of the system clock, ring oscillator, and bit overflow. The interval of watchdog timer interrupt is decided by WDT_CLK_DIV and WDTR set value. The equation can be described as

$$WDT\_TIME[s] = \frac{WDT\_CLK\_DIV \times (WDTR\ value + 1)}{fx\ or\ fLFO\ or\ fBIT}$$

NOTE1) LFO Clock is different between Normal / IDLE mode and Stop mode.
NOTE2) LFO Clock is 256kHz for Normal / IDLE mode and 168kHz for Stop Mode. (Typical)

11.3.2 Block Diagram

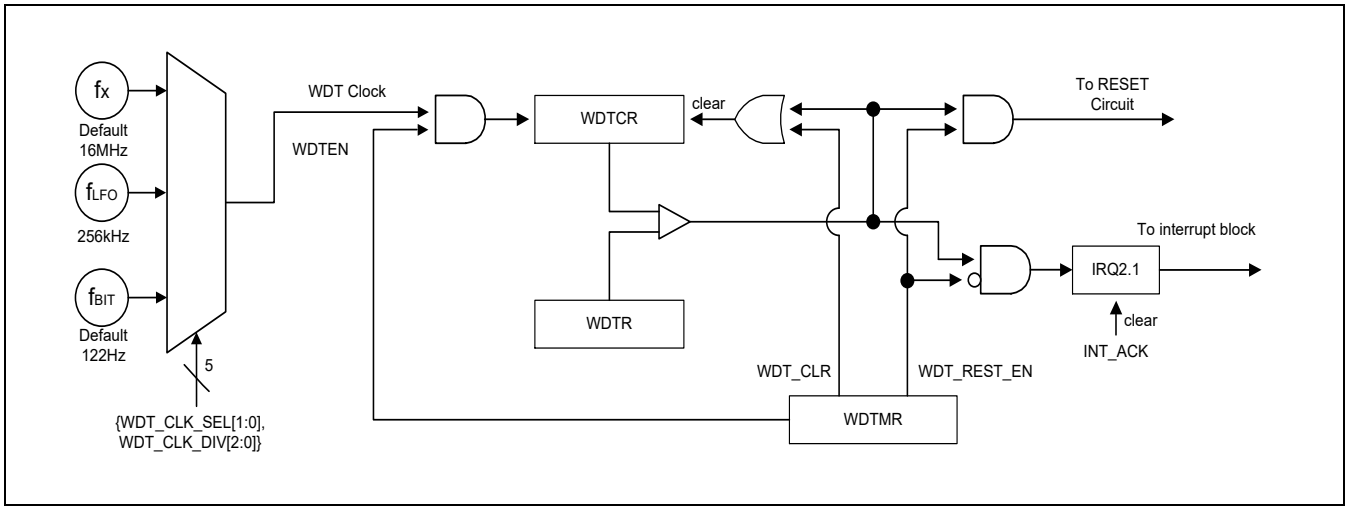


Figure 11.3 Watch Dog Timer Block Diagram

11.3.3 WDT Interrupt Timing Waveform

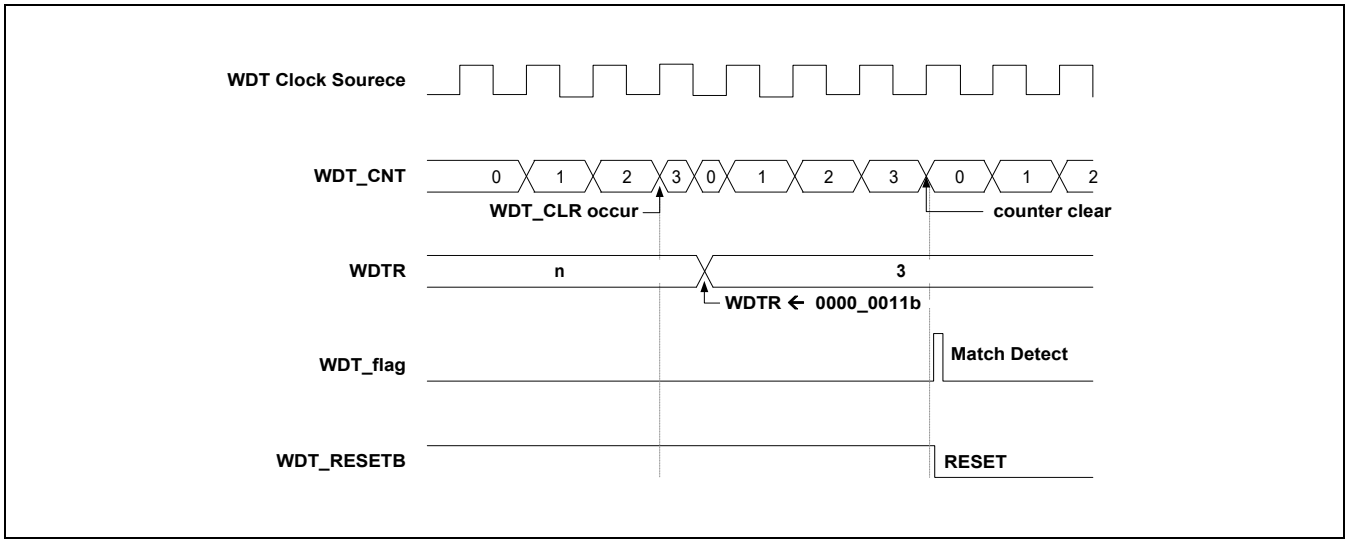


Figure 11.4 Watch Dog Timer Interrupt Timing Waveform

11.3.4 Register Map

Name	Address	Direction	Default	Description
WDTMR	8DH	R/W	3BH	Watch Dog Timer Control Register
WDTR	8EH	W	FFH	Watch Dog Timer Data Register
WDTCR	8EH	R	00H	Watch Dog Timer Counter Register

Table 11.3 Watch Dog Timer Register Map

11.3.5 Watch Dog Timer Register Description

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

11.3.6 Register Description for Watch Dog Timer

WDTMR (Watch Dog Timer Control Register) : 8DH

7	6	5	4	3	2	1	0
WDT_CLK_SEL1	WDT_CLK_SEL0	WDT_CLK_DIV2	WDT_CLK_DIV1	WDT_CLK_DIV0	WDT_CLR	WDT_RESET_EN	WDT_EN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 3BH

WDT_CLK_SEL[1:0] WDT Clock Source Selection

CLK_SEL1	CLK_SEL0	Description
0	0	Divide HFO clock (fx : Default 16MHz)
0	1	LFO clock (f_{LFO} : 256kHz)
1	0	BIT overflow (f_{BIT} : 122Hz)
1	1	Not used

WDT_CLK_DIV[2:0] WDT Clock Source Selection

f_{WDT} : WDT Clock Source set by WDT_CLK_SEL[1:0]

CLK_DIV2	CLK_DIV1	CLK_DIV0	Description
0	0	0	f_{WDT} / 2
0	0	1	f_{WDT} / 4
0	1	0	f_{WDT} / 8
0	1	1	f_{WDT} / 16
1	0	0	f_{WDT} / 256
1	0	1	f_{WDT} / 512
1	1	0	f_{WDT} / 1024
1	1	1	f_{WDT} / 2048

WDT_CLR Clear WDT Counter

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

WDT_RESET_EN WDT Reset Enable bit

0	Disable
1	Enable

WDT_EN WDT Enable bit

0	Disable
1	Enable

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

7	6	5	4	3	2	1	0
WDTCR7	WDTCR6	WDTCR5	WDTCR4	WDTCR3	WDTCR2	WDTCR1	WDTCR0
R	R	R	R	R	R	R	R

Initial value : 00H

WDTCR[7:0] WDT Counter

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

7	6	5	4	3	2	1	0
WDTR7	WDTR6	WDTR5	WDTR4	WDTR3	WDTR2	WDTR1	WDTR0
W	W	W	W	W	W	W	W

Initial value : FFH

WDTR[7:0] Set a period

WDT Interrupt Interval =

WDT_CLK_DIV x (WDTDR Value+1) / (WDT Clock Source Period)

NOTE) To guarantee proper operation, the data should be greater than 01H.

11.4 Timer 0

11.4.1 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.

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

Table 11.4 Timer 0 Operating Modes

11.4.2 8-bit Timer/Counter Mode

The 8-bit timer/counter mode is selected by control register as shown in Figure 11.5.

The 8-bit timer have counter and data register. The counter register is increased by internal. 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 time 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 P2FSR[7:6].

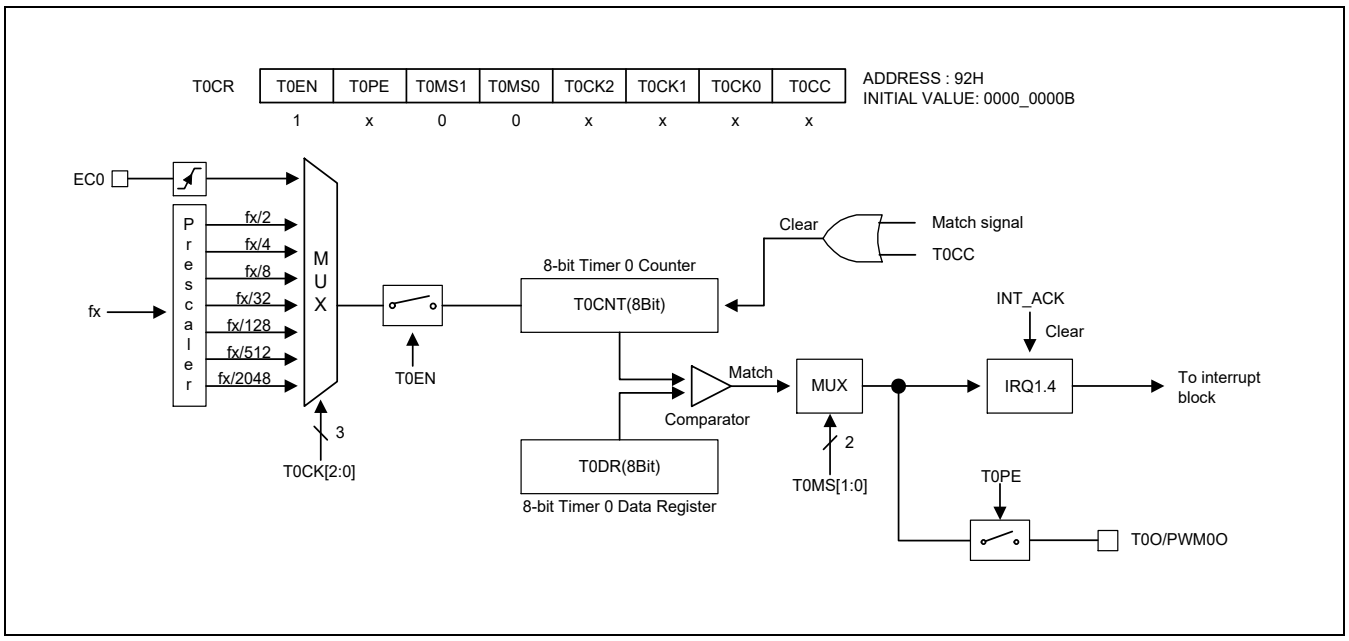


Figure 11.5 8-bit Timer/Counter Mode for Timer 0

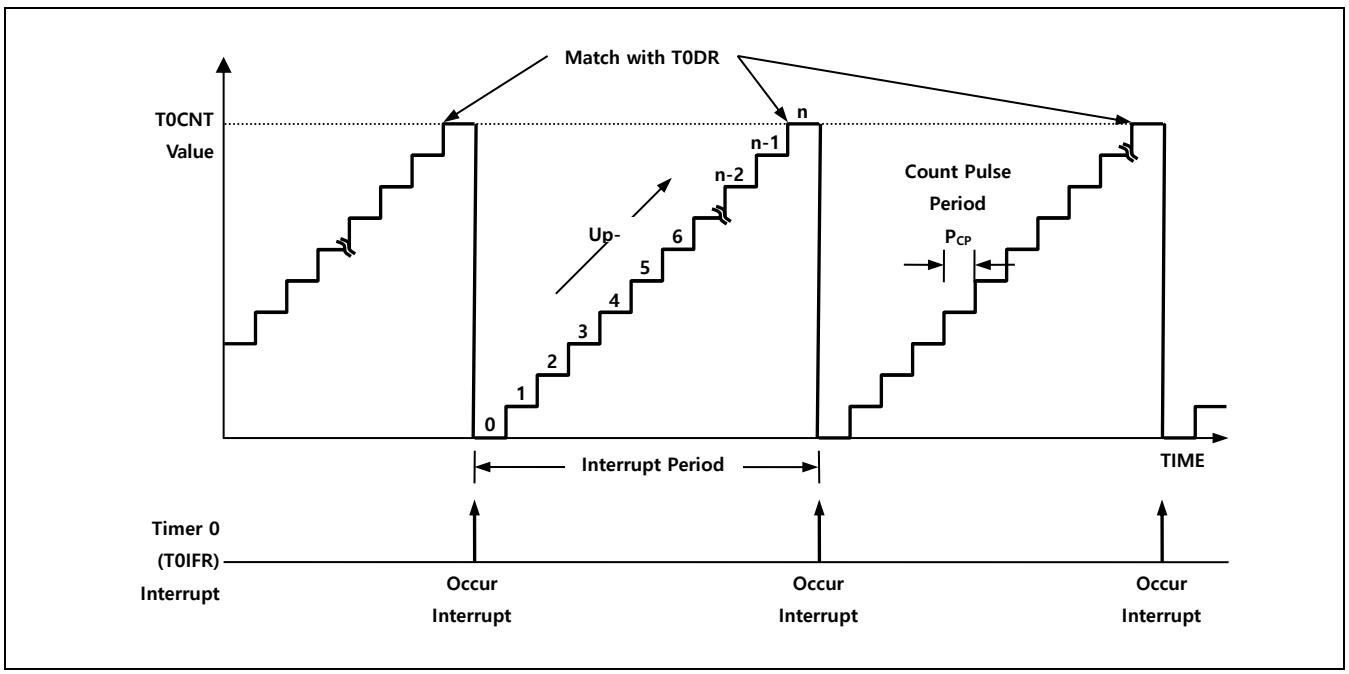


Figure 11.6 8-bit Timer/Counter 0 Example

11.4.3 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 P2FSR[7:6] to ’10’. 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.

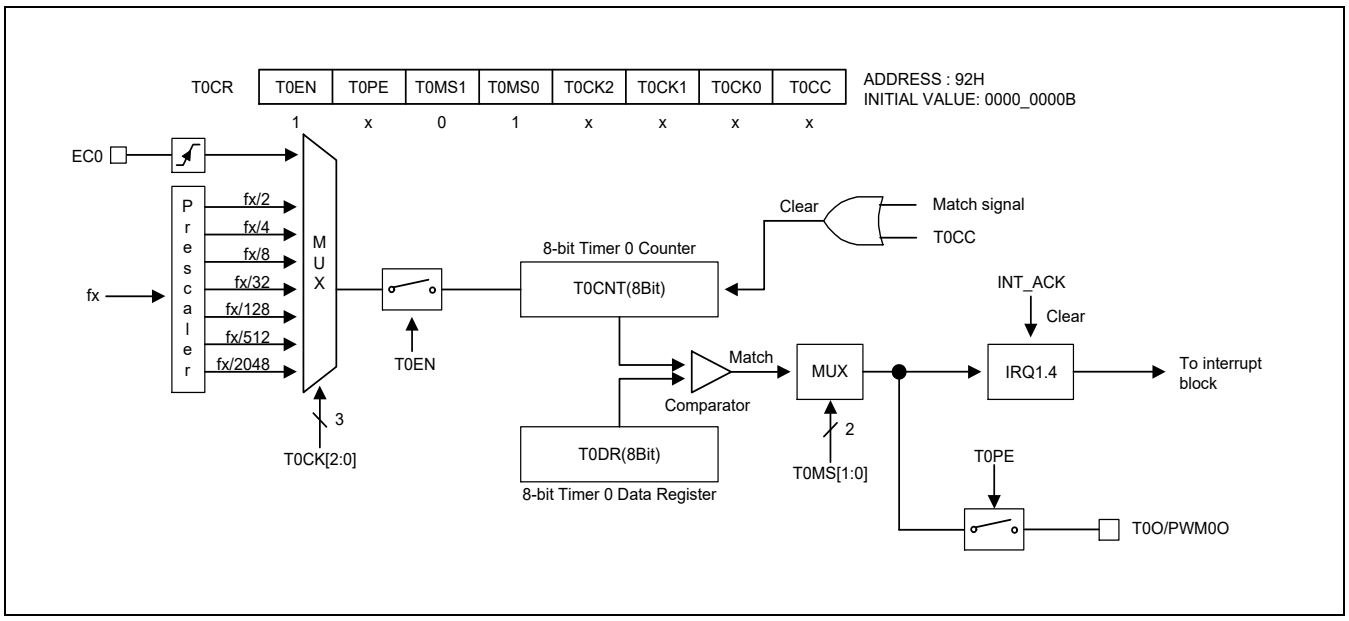


Figure 11.7 8-bit PWM Mode for Timer 0

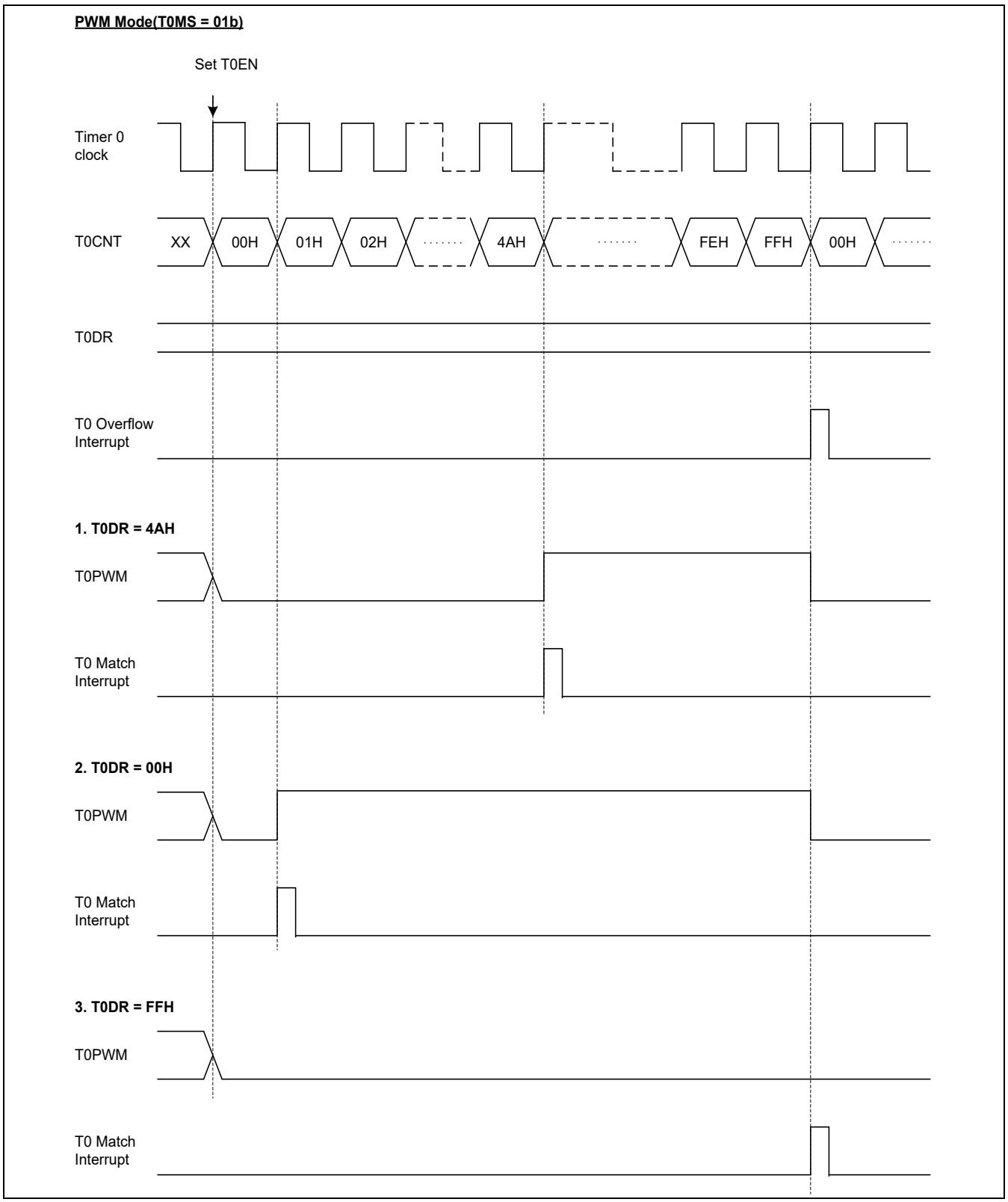


Figure 11.8 PWM Output Waveforms in PWM Mode for Timer 0

11.4.4 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 EIPOL0 registers setting, the external interrupt EINT10 function is chosen. Of course, 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.

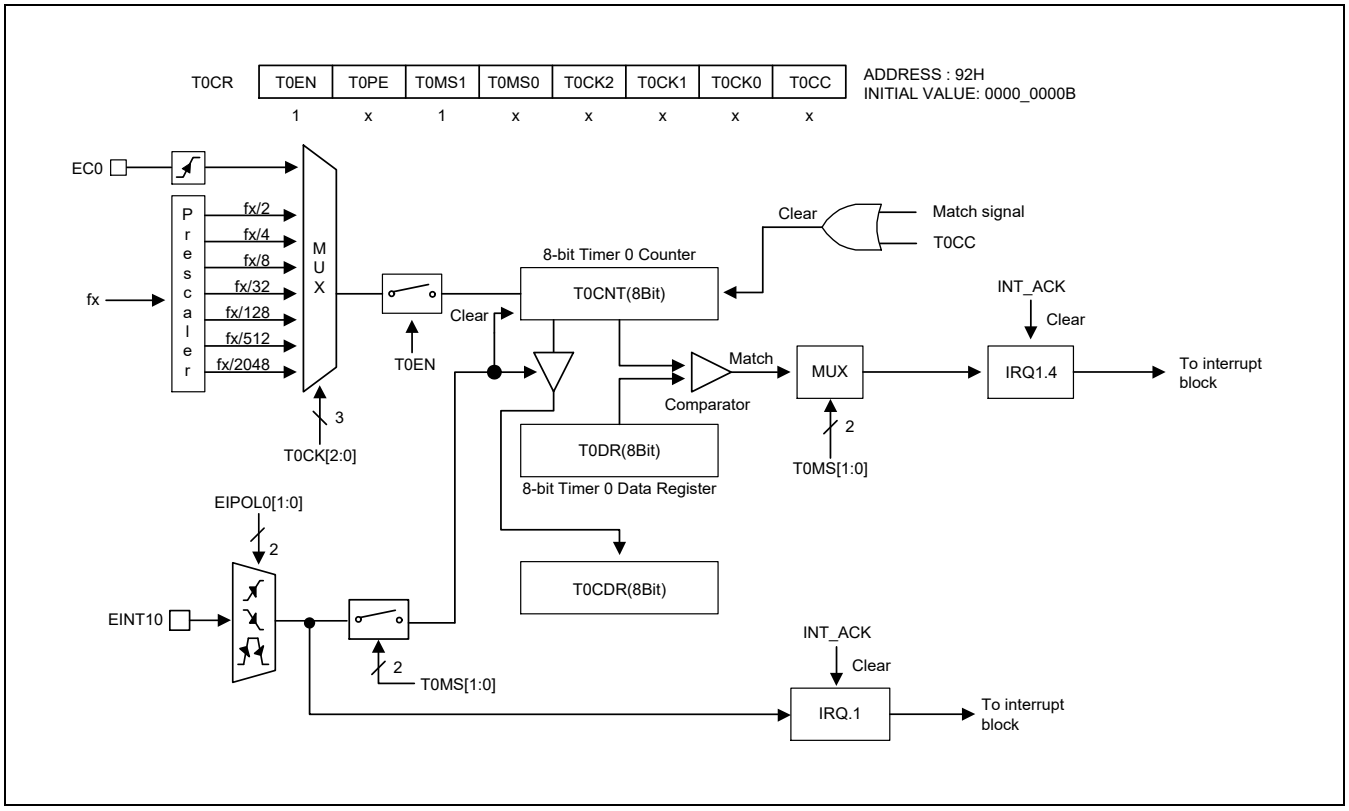


Figure 11.9 8-bit Capture Mode for Timer 0

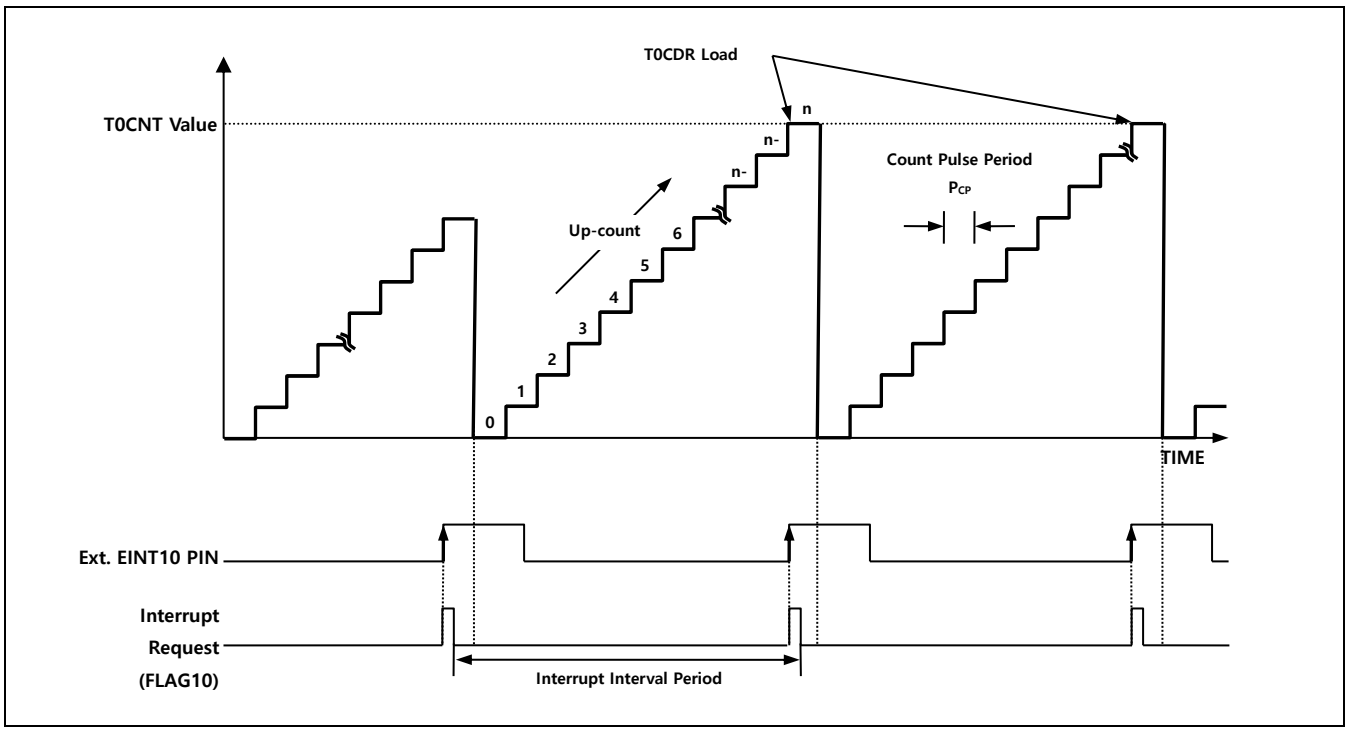


Figure 11.10 Input Capture Mode Operation for Timer 0

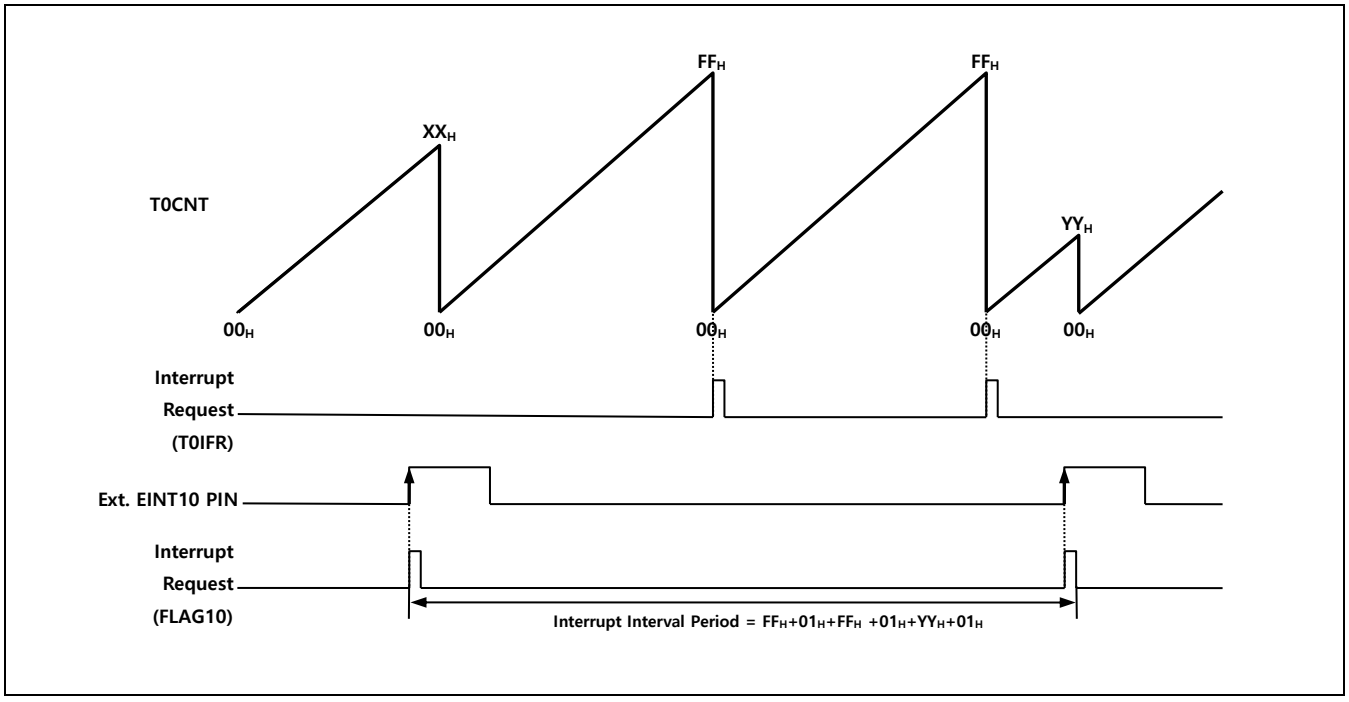


Figure 11.11 Express Timer Overflow in Capture Mode

11.4.5 Block Diagram

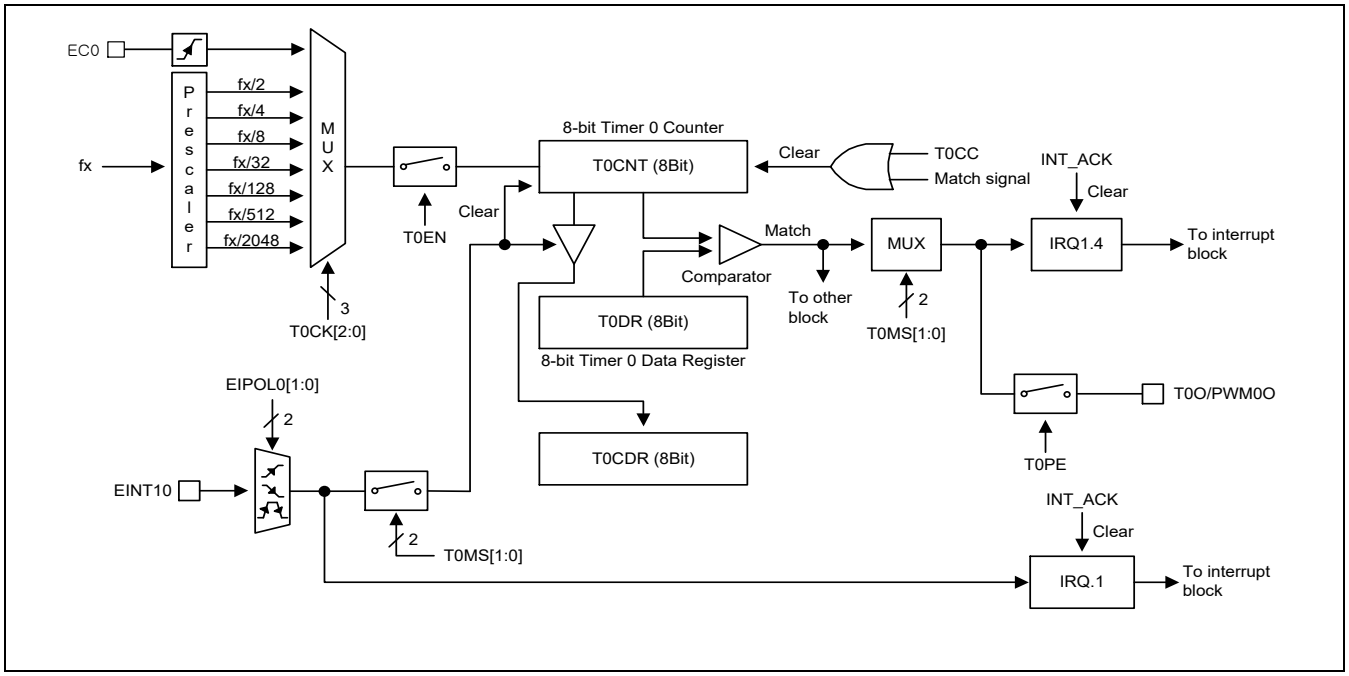


Figure 11.12 8-bit Timer 0 Block Diagram

11.4.6 Register Map

Name	Address	Direction	Default	Description
T0CR	92H	R/W	00H	Timer 0 Control Register
T0CNT	93H	R	00H	Timer 0 Counter Register
T0DR	94H	R/W	FFH	Timer 0 Data Register
T0CDR	94H	R	00H	Timer 0 Capture Data Register

Table 11.5 Timer 0 Register Map

11.4.7 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).

11.4.8 Register Description for Timer/Counter 0

T0CR (Timer 0 Control Register) : 92H

7	6	5	4	3	2	1	0
T0EN	T0PE	T0MS1	T0MS0	T0CK2	T0CK1	T0CK0	T0CC
R/W	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

T0PE Port output Control Timer0

0	Port output disable
1	Port output enable

T0MS[1:0] Control Timer 0 Operation Mode

T0MS1	T0MS0	Description
0	0	Timer/counter mode (T0O: toggle at match)
0	1	PWM mode (The overflow interrupt can occur)
1	x	Capture mode (The match interrupt can occur)

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)

T0CNT (Timer 0 Counter Register) : 93H

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) : 94H

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) : 94H

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

11.5 Timer 1

11.5.1 Overview

The 16-bit timer 1 consists of multiplexer, timer 1 A data register high/low, timer 1 B data register high/low, timer1 C data register high/low timer1 D data register high/low and timer 1 control register high/low (T1ADRH, T1ADRL, T1BDRH, T1BDRL, T1CRH, T1CRL, T1DRH, T1DRL).

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, 2048, HFO and EC1

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

T1EN	P1FSRL[4:3]	T1MS[1:0]	T1CK[2:0]	Timer 1
1	01	00	XXX	16 Bit Timer/Counter Mode
1	00	01	XXX	16 Bit Capture Mode
1	01	10	XXX	16 Bit PPG Mode(one-shot mode)
1	01	11	XXX	16 Bit PPG Mode(repeat mode)

Table 11.6 Timer 1 Operating Modes

11.5.2 16-bit Timer/Counter Mode

The 16-bit timer/counter mode is selected by control register as shown in Figure 11.13.

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, 2048 and Internal High Frequency Oscillator (HFO) 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 P0FSR_L[7:6] to '01'.

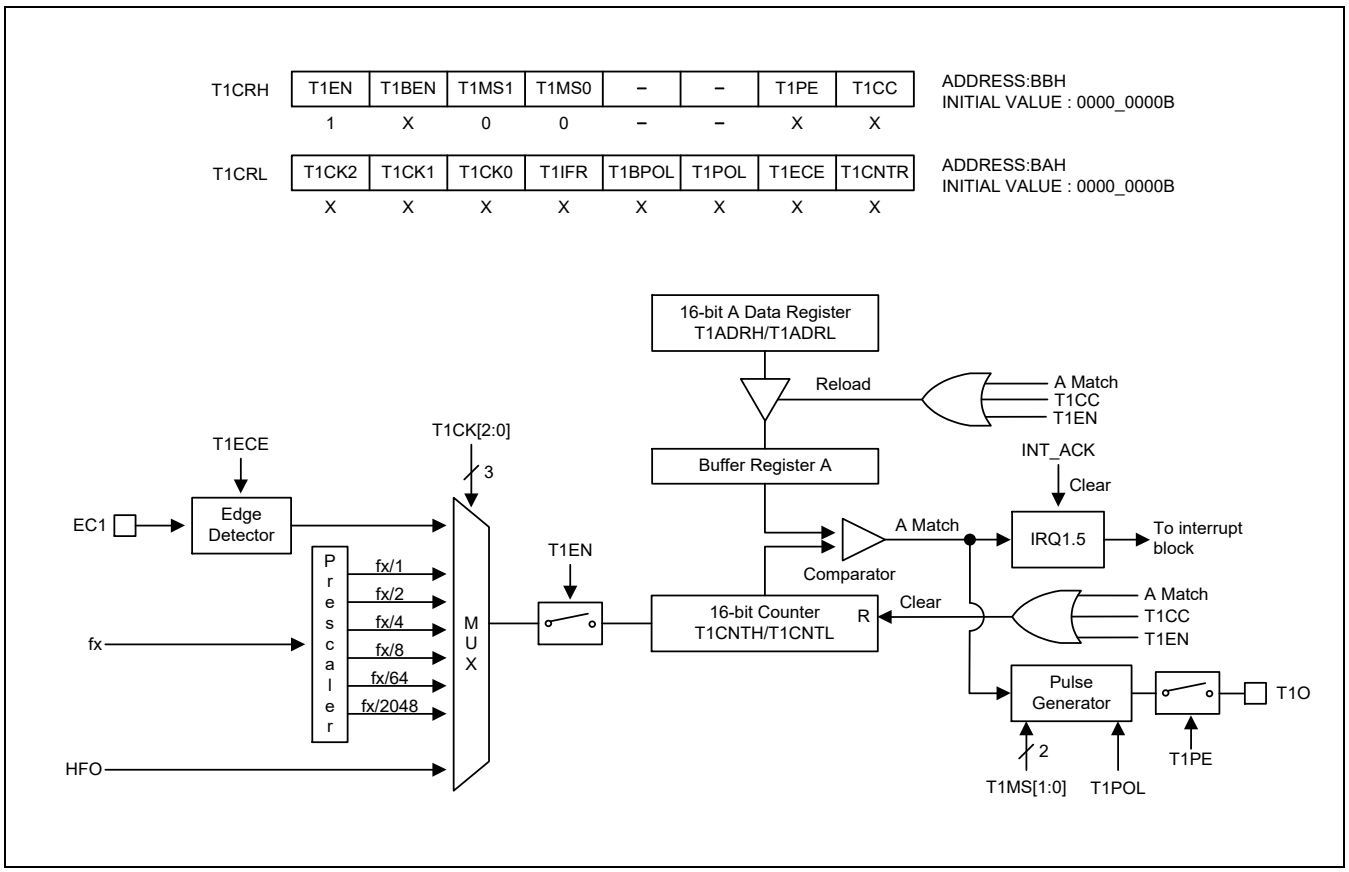


Figure 11.13 16-bit Timer/Counter Mode for Timer 1

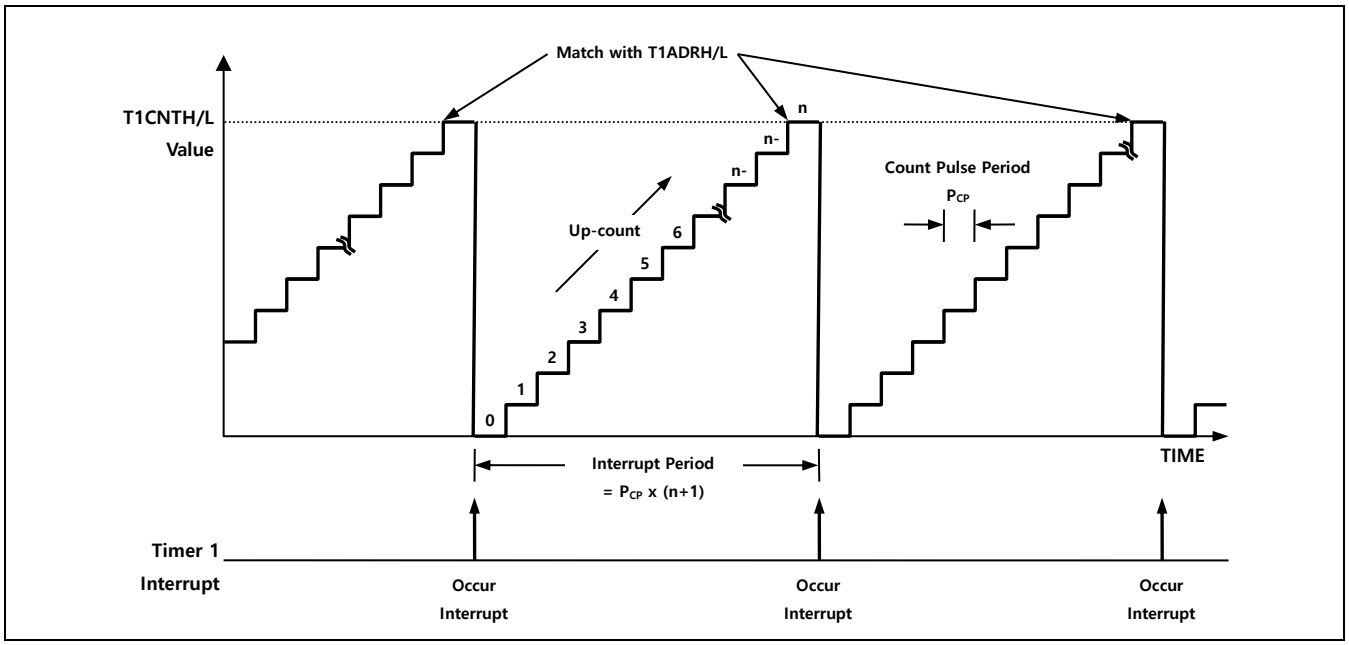


Figure 11.14 16-bit Timer/Counter 1 Example

11.5.3 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 EIPOL0 registers setting, the external interrupt EINT11 function is chosen. Of course, the EINT11 pin must be set as an input port.

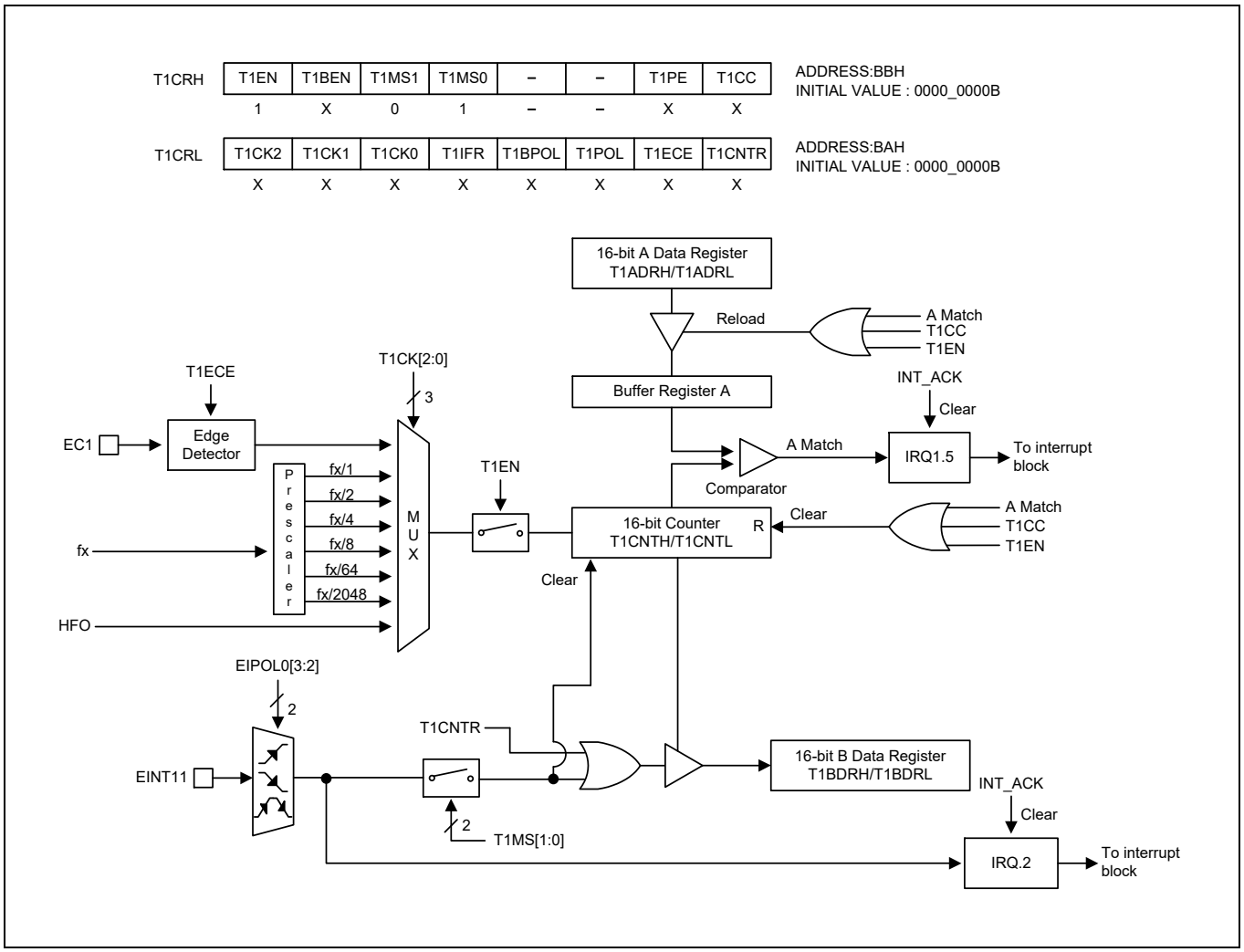


Figure 11.15 16-bit Capture Mode for Timer 1

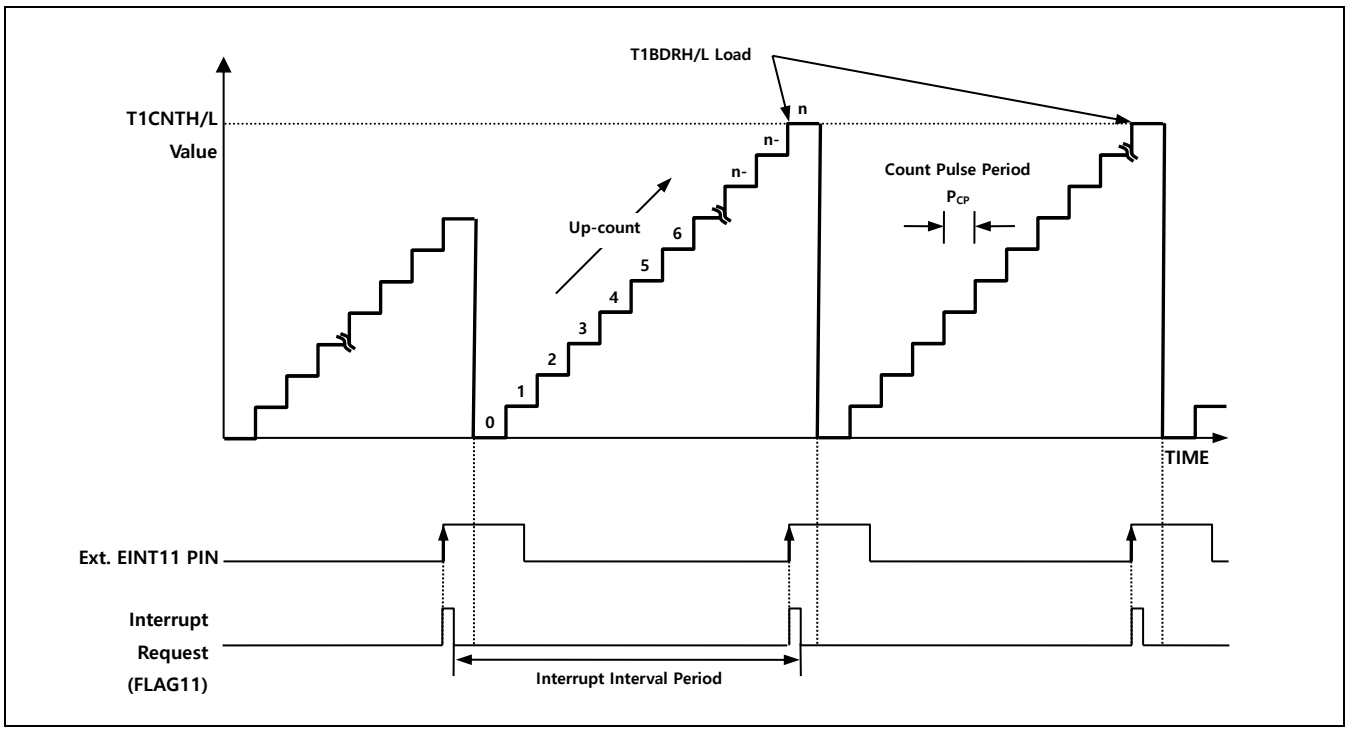


Figure 11.16 Input Capture Mode Operation for Timer 1

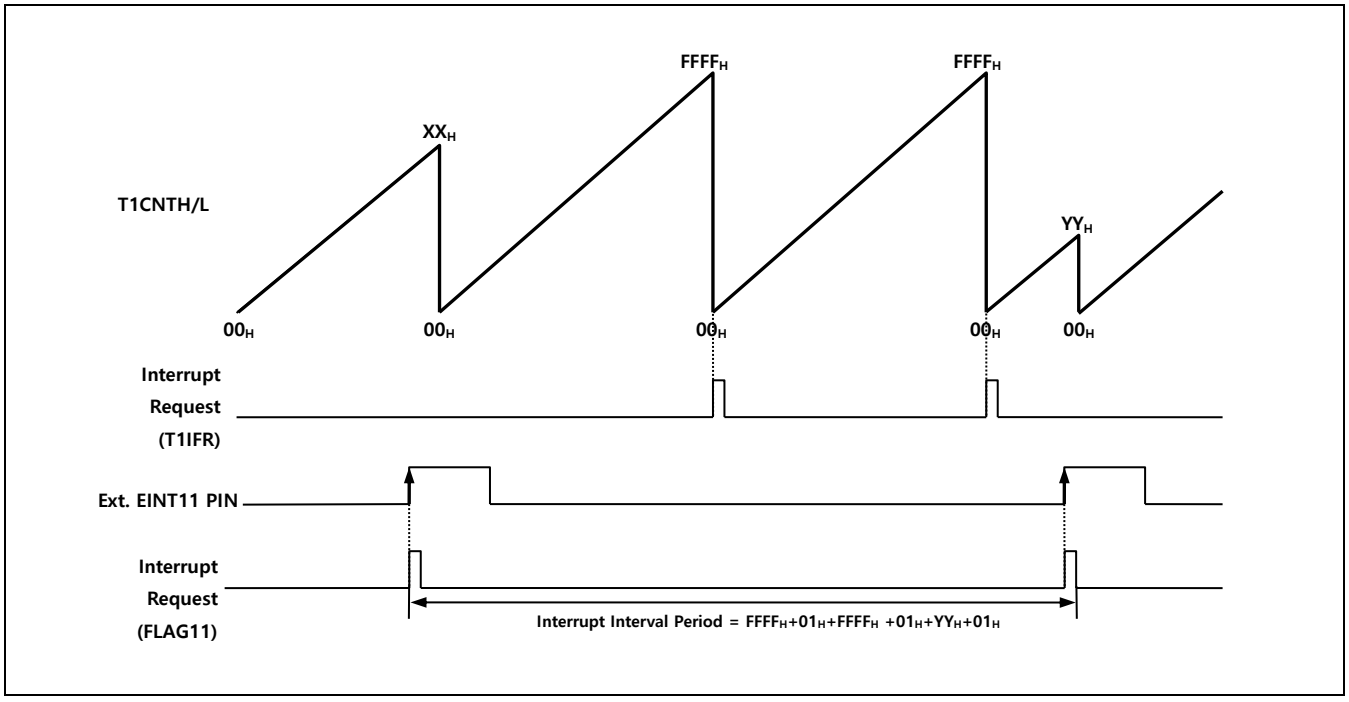
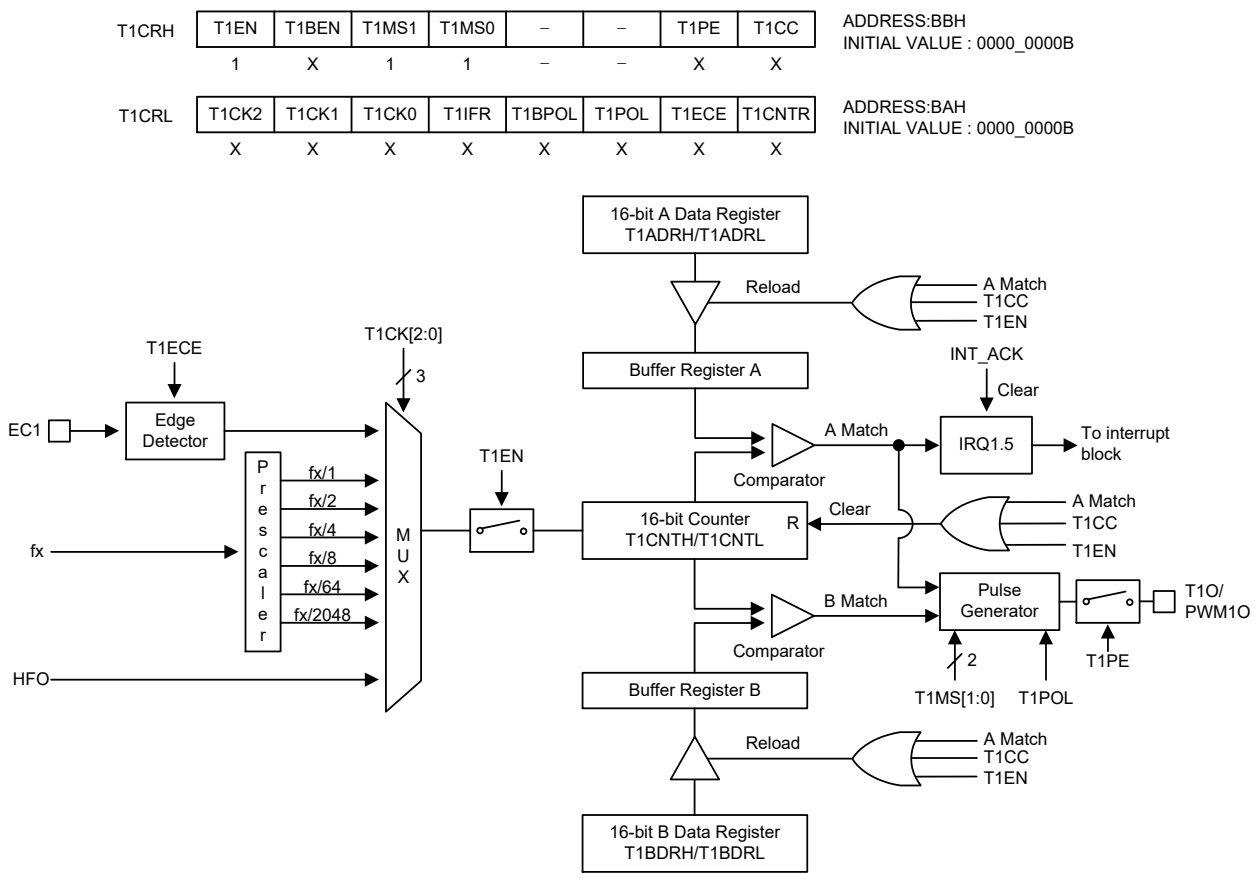


Figure 11.17 Express Timer Overflow in Capture Mode

11.5.4 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_L[3:2] or P0FSRL_L[7:6] to '10'. 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 11.18 16-bit PPG Mode for Timer 1

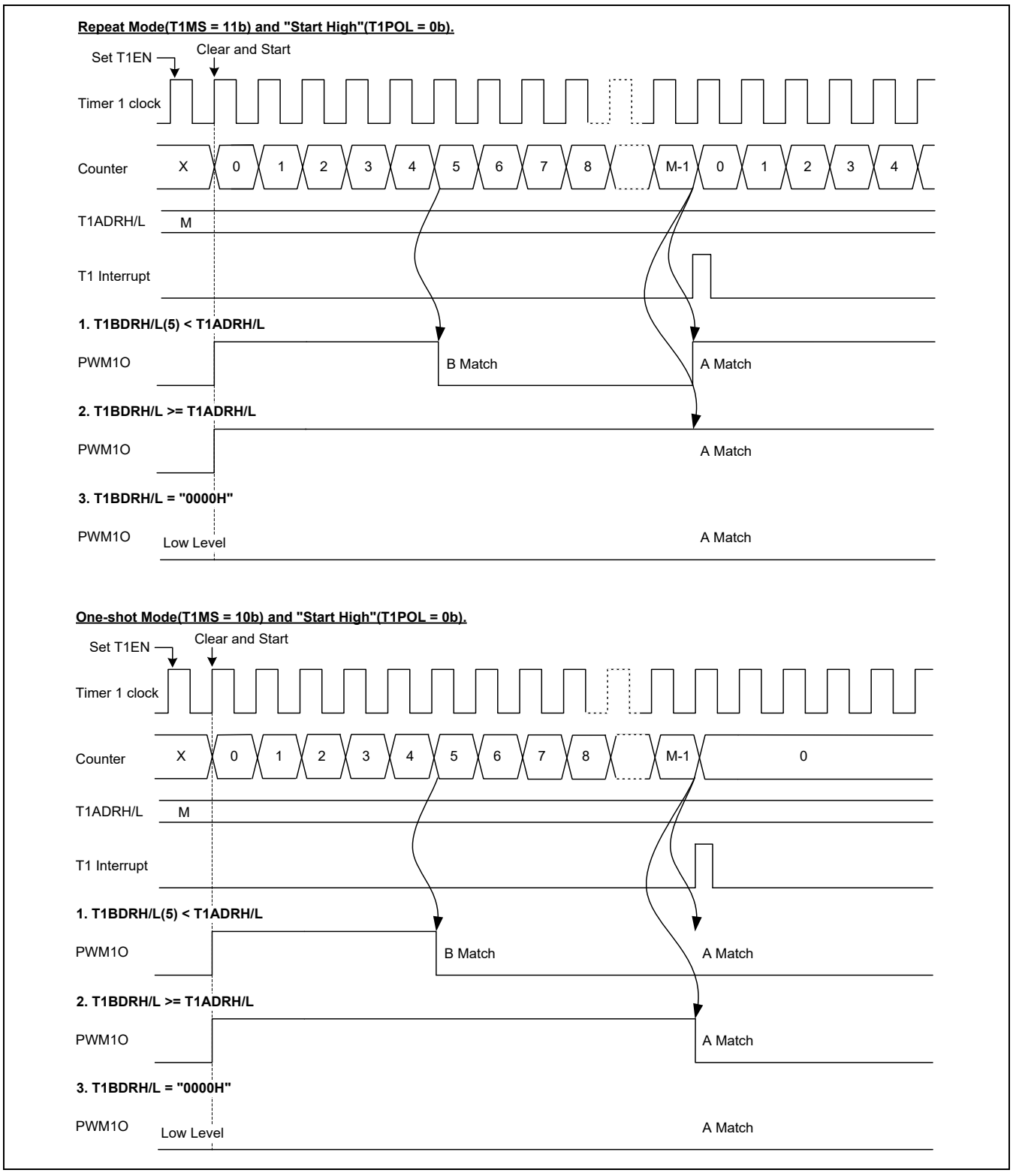
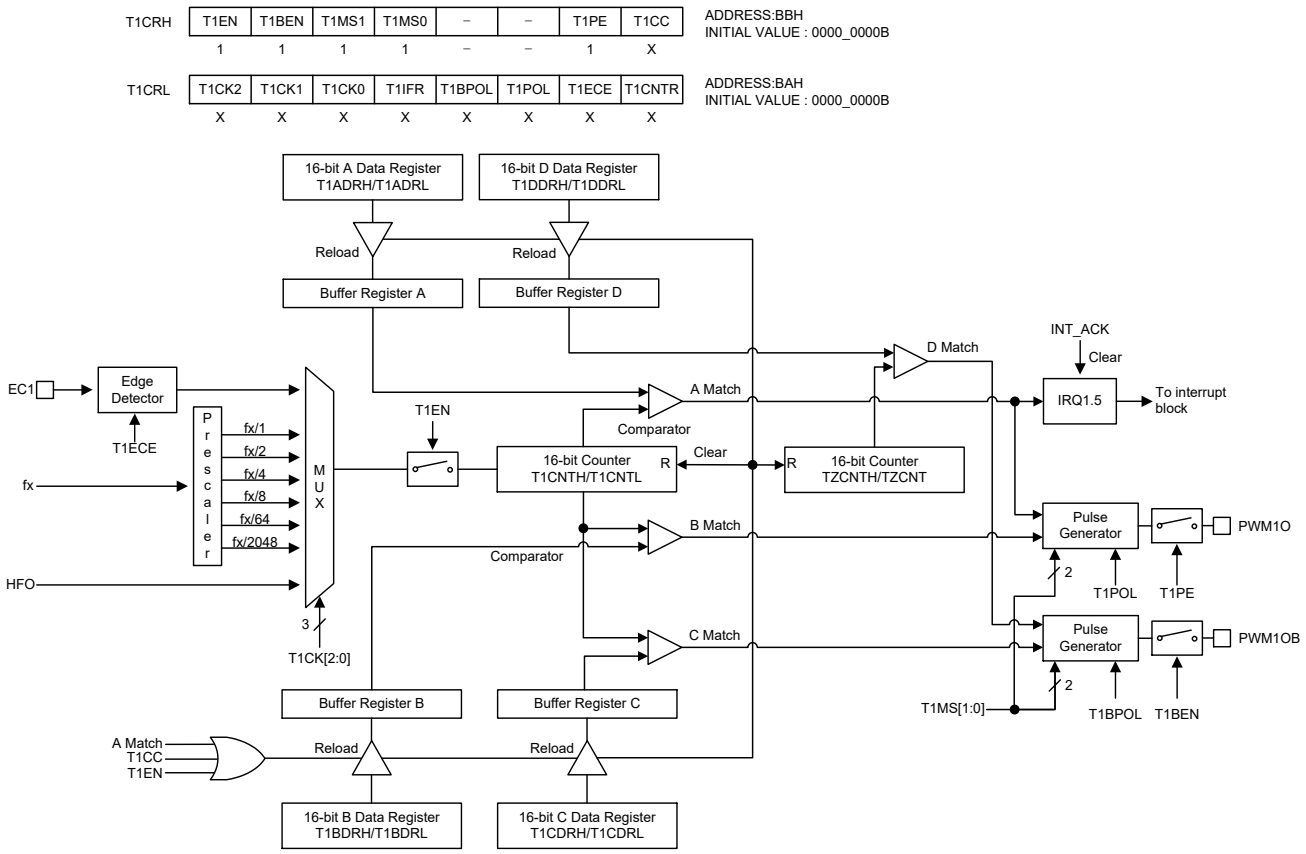


Figure 11.19 16-bit PPG Mode Timming chart for Timer 1

11.5.5 16-bit Complementary PWM mode (Dead Time)

The timer 1 has a Complementary PWM function. The complementary PWM output function operates when T1BEN is set. In PPG mode, PWM1O/PWM1OB pin outputs up to 16-bit resolution complementary PWM output. This pin should be configured as a PWM output by setting P1FSRL_L[3:2] or P0FSRL_L[7:6] to '10'.

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. The delay (dead time) of the complementary pwm output is determined by T1CDRH / T1CDRL. And the duty of the complementary pwm output is determined by T1DDRH / T1DDRL.



NOTE) The T1EN is automatically cleared to logic "0" after one pulse is generated at a PPG one-shot mode.

Figure 11.20 16-bit Complementary PWM Mode for Timer 1

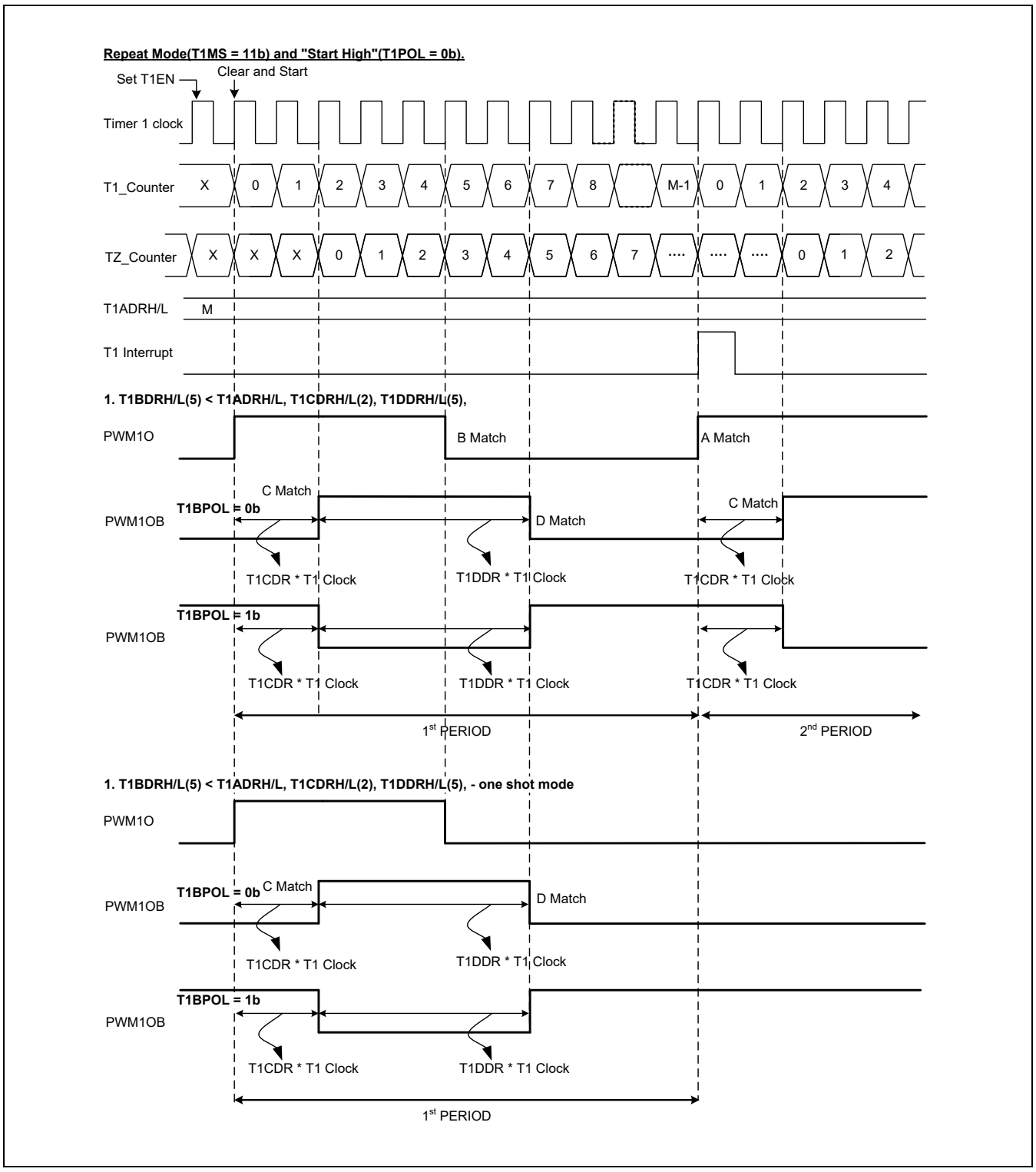


Figure 11.21 16-bit Complementary PWM Mode Timming chart for Timer 1

11.5.6 Block Diagram

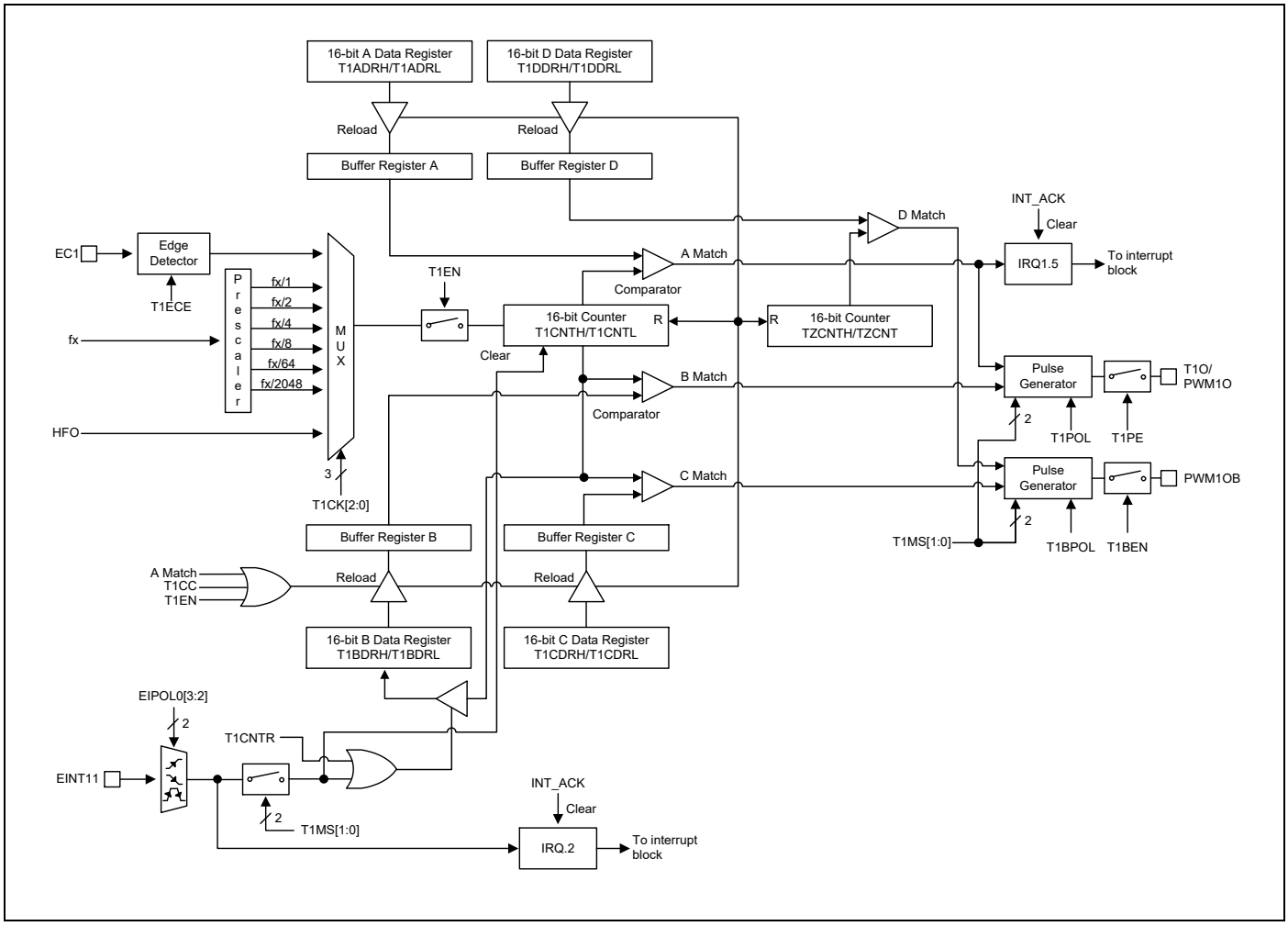


Figure 11.22 16-bit Timer 1 Block Diagram

11.5.7 Register Map

Name	Address	Direction	Default	Description
T1CRL	BAH	R/W	00H	Timer 1 Control Low Register
T1CRH	BBH	R/W	00H	Timer 1 Control High Register
T1ADRL	BCH	R/W	FFH	Timer 1 A Data Low Register
T1ADRH	BDH	R/W	FFH	Timer 1 A Data High Register
T1BDRL	BEH	R/W	FFH	Timer 1 B Data Low Register
T1BDRH	BFH	R/W	FFH	Timer 1 B Data High Register
T1CDRL	CCH	R/W	FFH	Timer 1 C Data Low Register
T1CDRH	CDH	R/W	FFH	Timer 1 C Data High Register
T1DDRL	CEH	R/W	FFH	Timer 1 D Data Low Register
T1DDRH	CFH	R/W	FFH	Timer 1 D Data High Register

Table 11.7 Timer 1 Register Map

11.5.8 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 C data high register (T1CDRH), timer 1 C data low register (T1CDRL), timer 1 D data high register (T1DDRH), timer 1 D data log register (T1DDRL), timer 1 control high register (T1CRH) and timer 1 control low register (T1CRL).

11.5.9 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

T1CDRH (Timer 1 C data High Register) : CDH

7	6	5	4	3	2	1	0
T1CDRH7	T1CDRH6	T1CDRH5	T1CDRH4	T1CDRH3	T1CDRH2	T1CDRH1	T1CDRH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FFH

T1CDRH[7:0] T1 C Data High Byte

T1CDRL (Timer 1 C Data Low Register) : CCH

7	6	5	4	3	2	1	0
T1CDRL7	T1CDRL6	T1CDRL5	T1CDRL4	T1CDRL3	T1CDRL2	T1CDRL1	T1CDRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FFH

T1CDRL[7:0] T1 C Data Low Byte

T1DDRH (Timer 1 D Data High Register) : CFH

7	6	5	4	3	2	1	0
T1DDRH7	T1DDRH6	T1DDRH5	T1DDRH4	T1DDRH3	T1DDRH2	T1DDRH1	T1DDRH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FFH

T1DDRH[7:0] T1 D Data High Byte

T1DDRL (Timer 1 D Data Low Register) : CEH

7	6	5	4	3	2	1	0
T1DDRL7	T1DDRL6	T1DDRL5	T1DDRL4	T1DDRL3	T1DDRL2	T1DDRL1	T1DDRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FFH

T1DDRL[7:0] T1 D Data Low Byte

T1CRH (Timer 1 Control High Register) : BBH

7	6	5	4	3	2	1	0
T1EN	T1BEN	T1MS1	T1MS0	–	–	T1PE	T1CC
R/W	R/W	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)

T1BEN Control Complementary PWM

0 Complementary PWM disable

1 Complementary PWM enable

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)

T1PE Control Timer 1 port output

0 Timer 1 output disable

1 Timer 1 output enable

T1CC Clear Timer 1 Counter

0 No effect

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

***NOTE)** To use T1O/PWM1O as pin3(P11), set Sub-function and set it to Output High.

T1CRL (Timer 1 Control Low Register) : BAH

7	6	5	4	3	2	1	0
T1CK2	T1CK1	T1CK0	T1IFR	T1BPOL	T1POL	T1ECE	T1CNTR
R/W	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 / 64
0	1	0	fx / 8
0	1	1	fx / 4
1	0	0	fx / 2
1	0	1	fx / 1
1	1	0	HFO Direct (32MHz)
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. Writing “1” has no effect.

0 T1 Interrupt no generation

1 T1 Interrupt generation

T1BPOL PWM1OB Polarity Selection

0 Start High (PWM1OB is low level at disable)

1 Start Low (PWM1OB is high level at disable)

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)

11.6 Timer 2

11.6.1 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 clocked by an internal or an external clock source (EC2). 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, 64, 2048, HFO and EC2

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.

T2EN	P1FSRL[6:5]	T2MS[1:0]	T2CK[2:0]	Timer 2
1	01	00	XXX	16 Bit Timer/Counter Mode
1	00	01	XXX	16 Bit Capture Mode
1	01	10	XXX	16 Bit PPG Mode(one-shot mode)
1	01	11	XXX	16 Bit PPG Mode(repeat mode)

Table 11.8 Timer 2 Operating Modes

11.6.2 16-bit Timer/Counter Mode

The 16-bit timer/counter mode is selected by control register as shown in Figure 11.23.

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, 64, 2048 and Internal High Frequency Oscillator (HFO) 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).

The external clock (EC2) counts up the timer at the rising edge. If the EC2 is seledted as a clock source by T1CK[2:0], EC2 port should be set to the input port by P1FSR_L[5:4] to '01'.

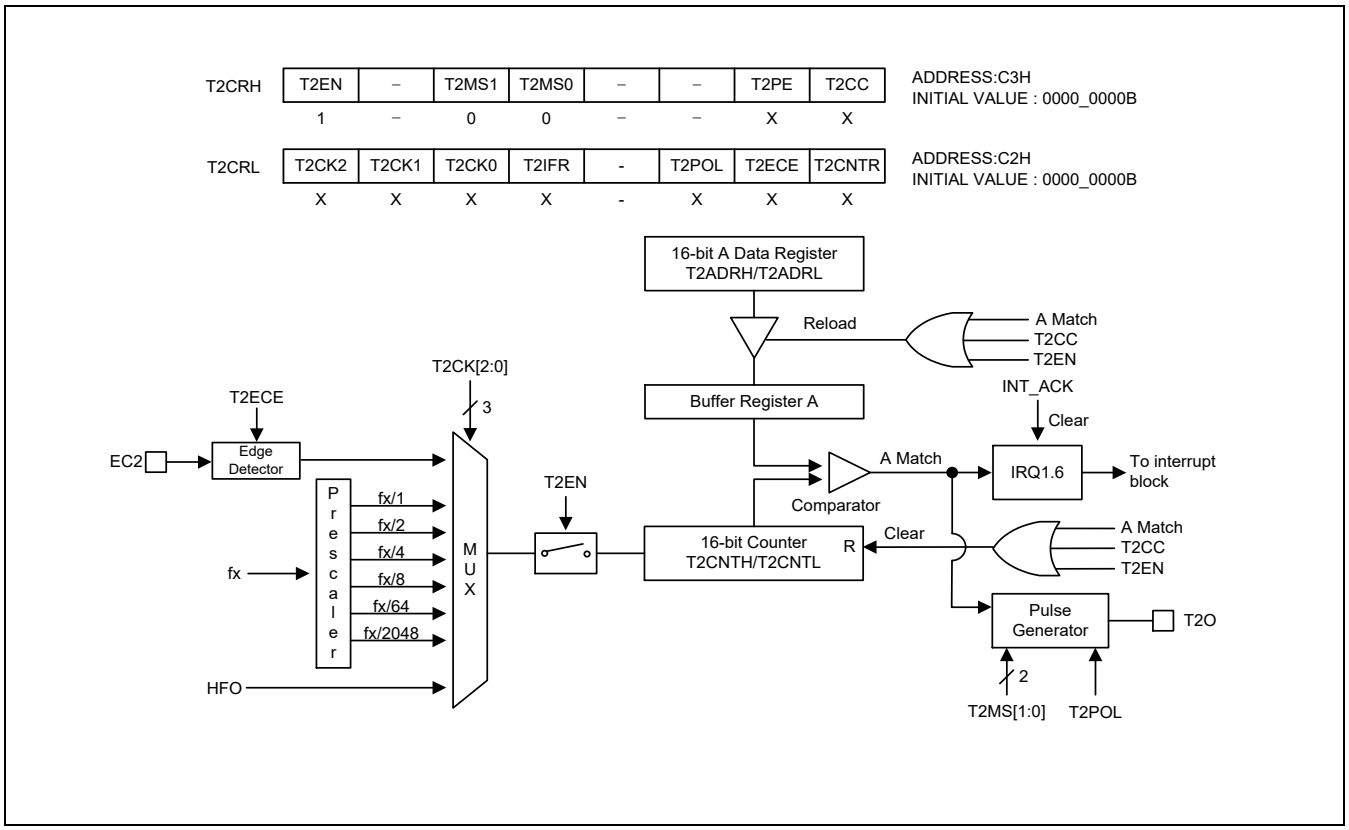


Figure 11.23 16-bit Timer/Counter Mode for Timer 2

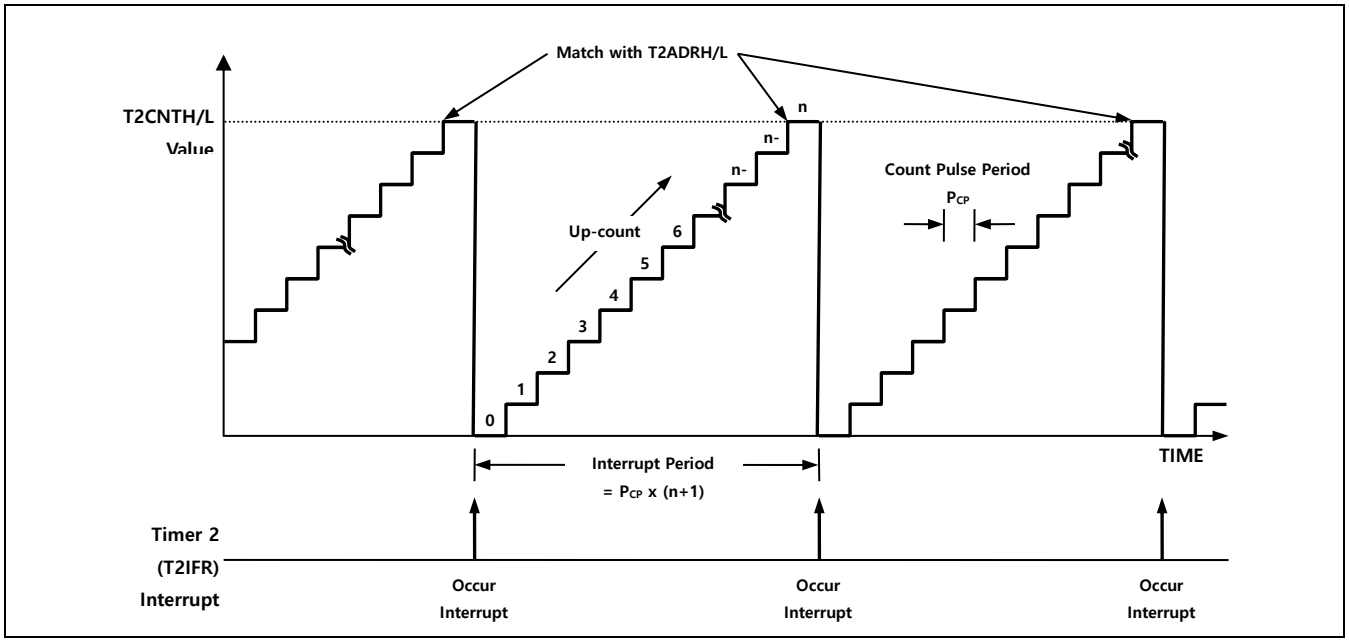


Figure 11.24 16-bit Timer/Counter 2 Example

11.6.3 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 EIPOL0 registers setting, the external interrupt EINT12 function is chosen. Of course, the EINT12 pin must be set to an input port.

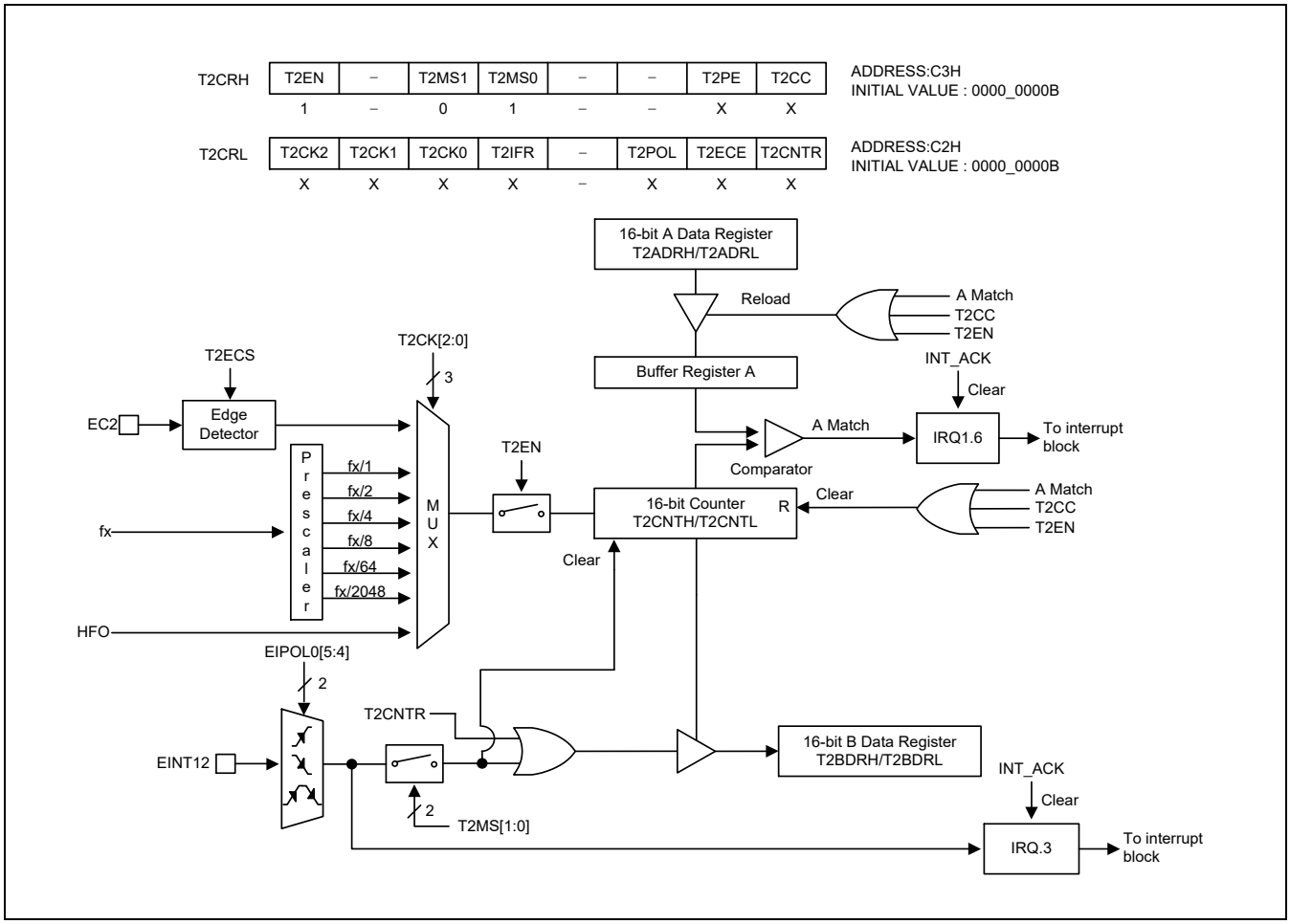


Figure 11.25 16-bit Capture Mode for Timer 2

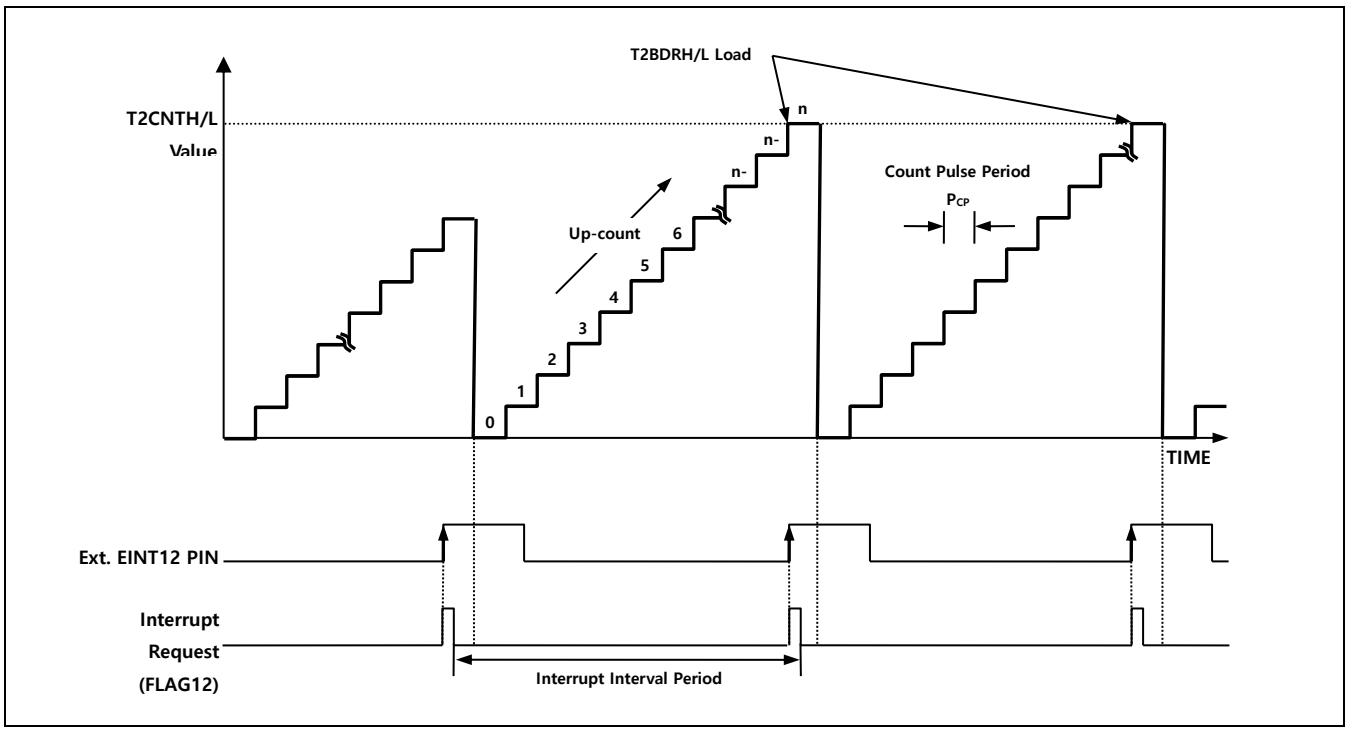


Figure 11.26 Input Capture Mode Operation for Timer 2

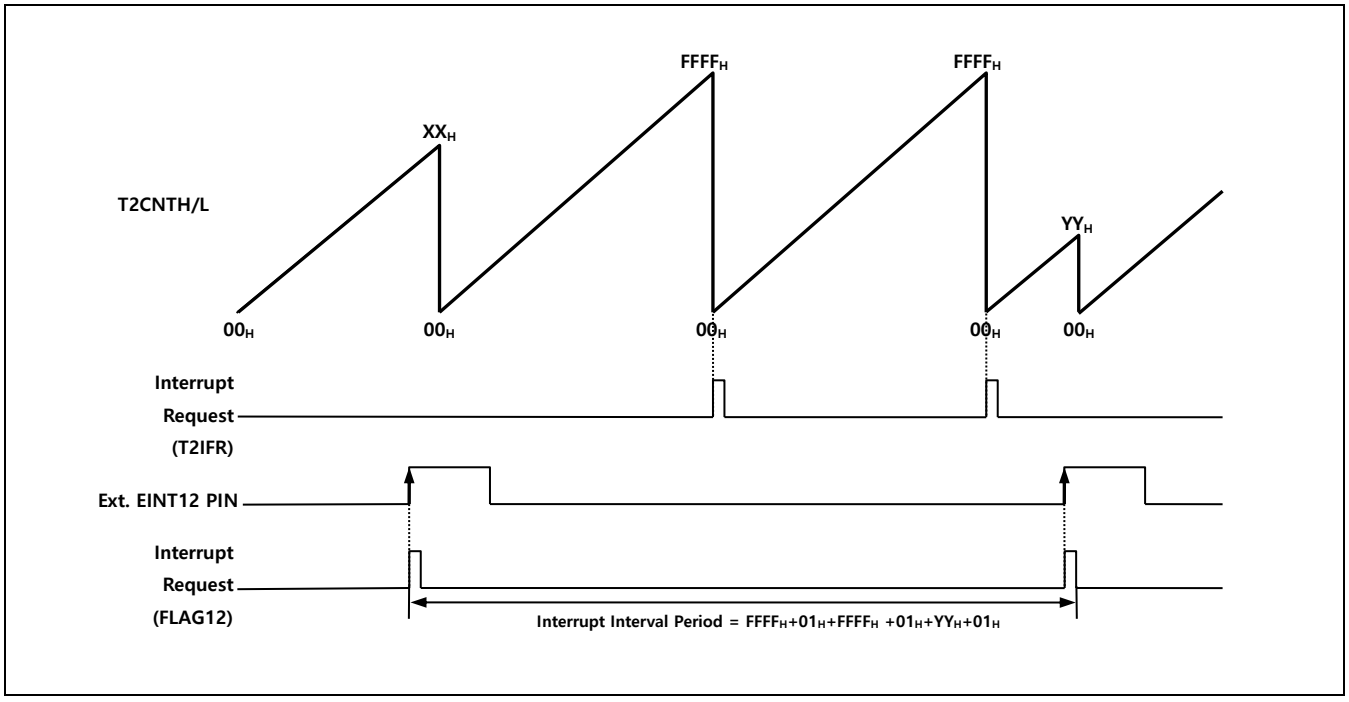
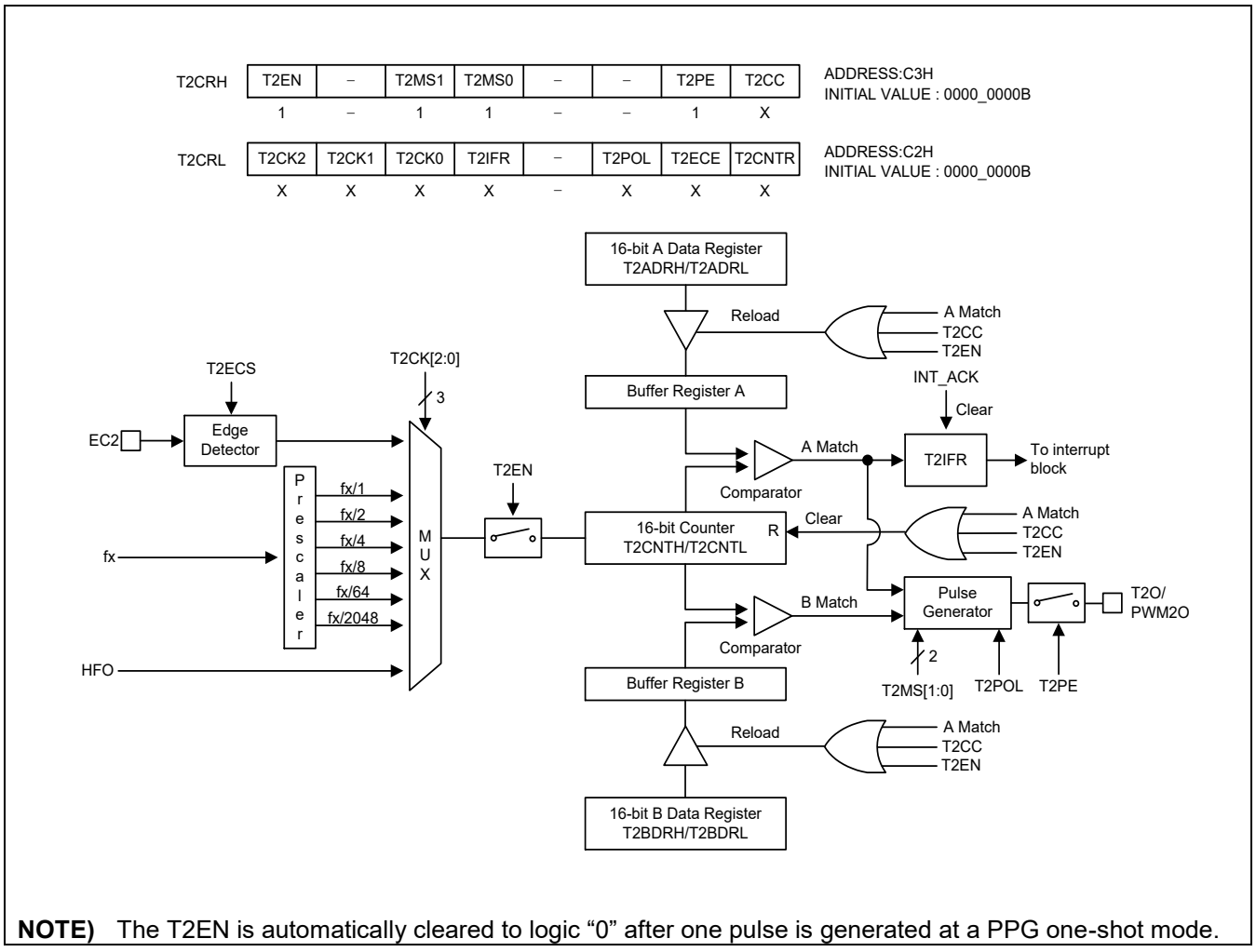


Figure 11.27 Express Timer Overflow in Capture Mode

11.6.4 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 P1FSR_L[5:4] or P1FSR_L[1:0] to '10'. 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 11.28 16-bit PPG Mode for Timer 2

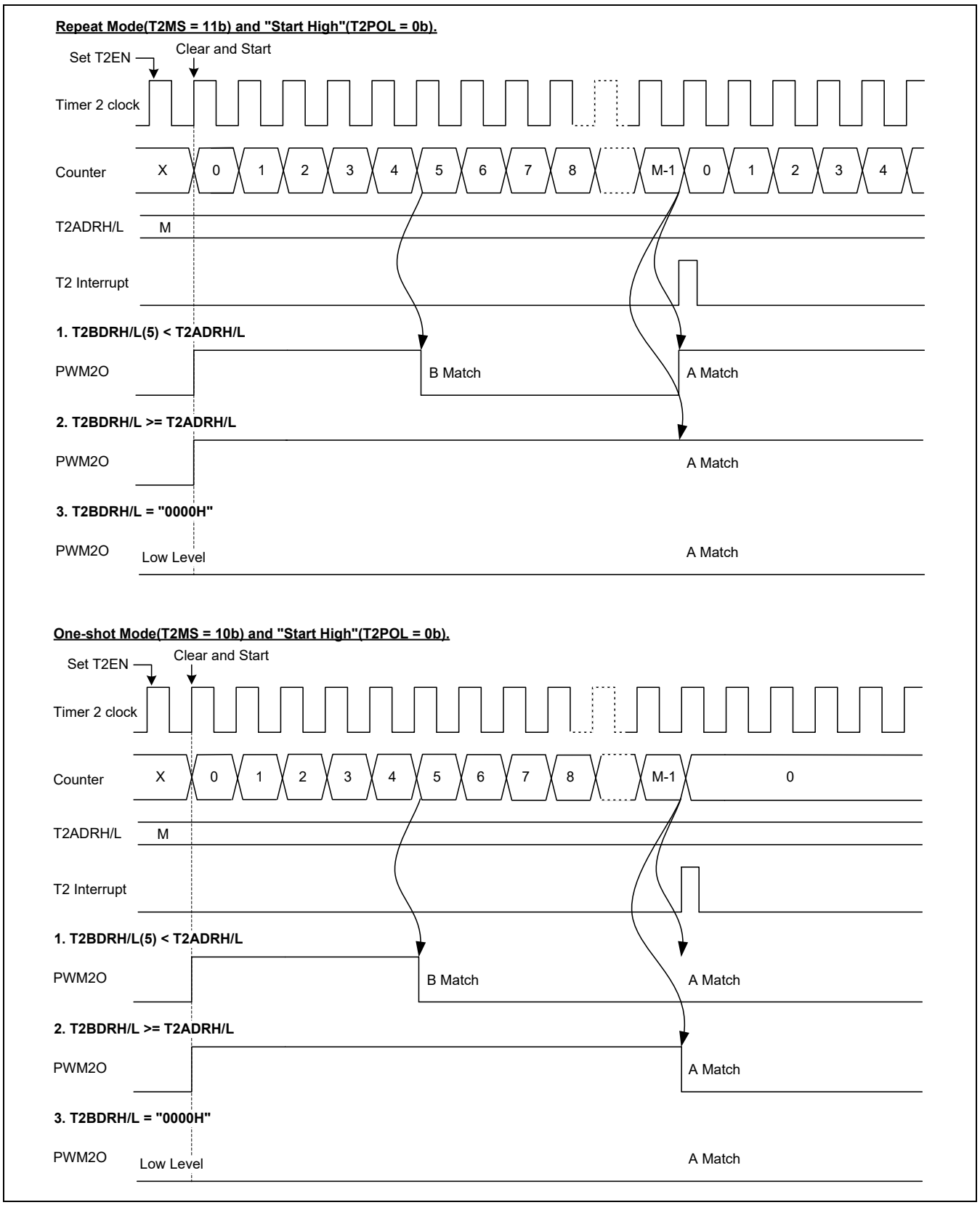


Figure 11.29 16-bit PPG Mode Timming chart for Timer 2

11.6.5 Block Diagram

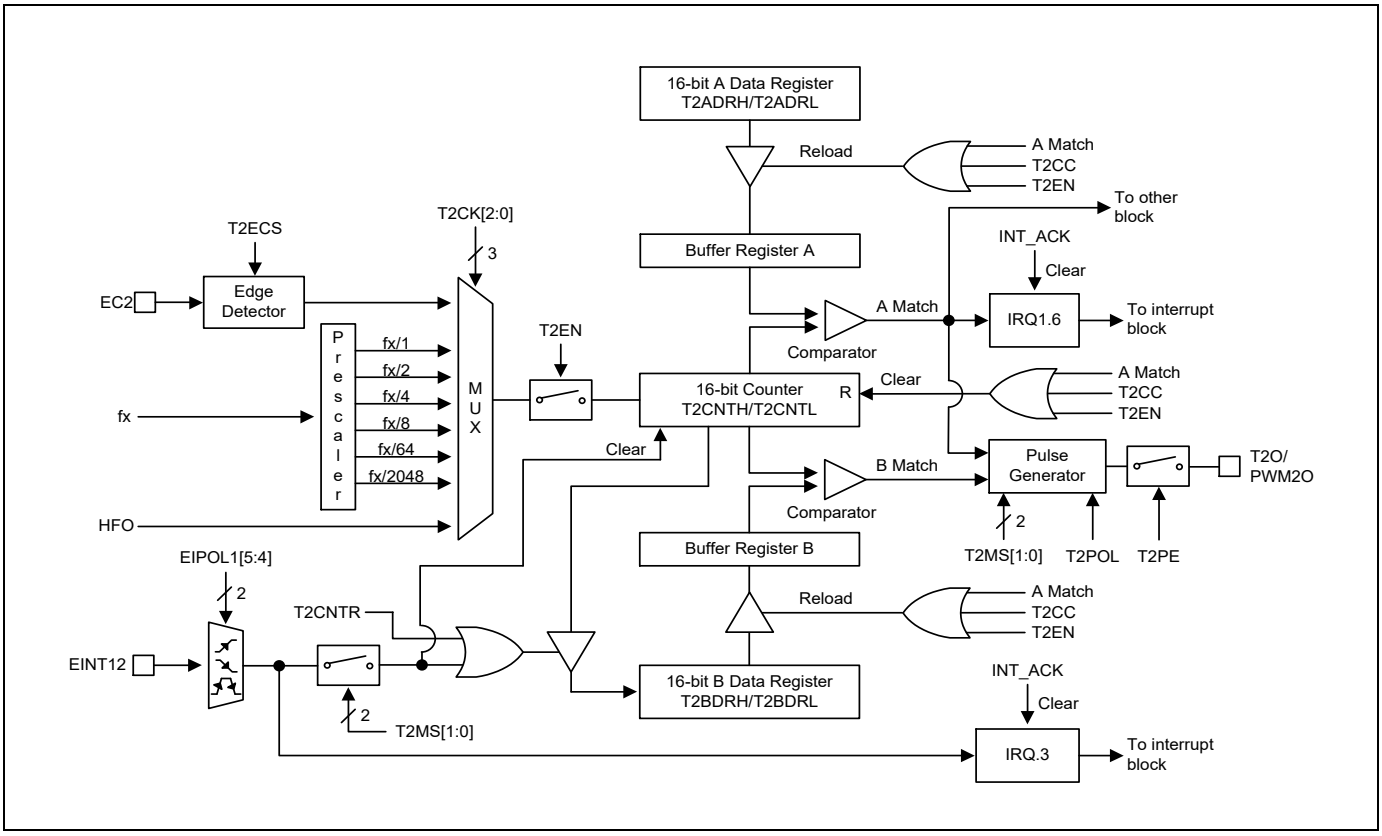


Figure 11.30 16-bit Timer 2 Block Diagram

11.6.6 Register Map

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

Table 11.9 Timer 2 Register Map

11.6.7 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).

11.6.8 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	–	–	T2PE	T2CC
R/W	–	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)

T2PE Control Timer 2 port output

0 Timer 2 output disable

1 Timer 2 output enable

T2CC Clear Timer 2 Counter

0 No effect

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

***NOTE)** To use T2O/PWM2O as pin2(P10), set Sub-function and set it to Output Low.

T2CRL (Timer 2 Control Low Register) : C2H

7	6	5	4	3	2	1	0
T2CK2	T2CK1	T2CK0	T2IFR	–	T2POL	T2ECE	T2CNTR
R/W	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/2048
0	0	1	fx/64
0	1	0	fx/8
0	1	1	fx/4
1	0	0	fx/2
1	0	1	fx/1
1	1	0	HFO Direct (32MHz)
1	1	1	External clock (EC1)

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. Writing “1” has no effect.

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)

T2ECE Timer 2 External Clock Edge Selection

0 External clock falling edge

1 External clock rising edge

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)

11.7 12-bit A/D Converter

11.7.1 Overview

The analog-to-digital converter (A/D) allows conversion of an analog input signal to a corresponding 12-bit digital value. The A/D module has tenth analog inputs. The output of the multiplex is the input into the converter, which generates the result via successive approximation. The A/D module has four registers which are the control register ADCM (A/D Converter Mode Register), ADCM1 (A/D Converter Mode Register 1) and A/D result register ADCHR (A/D Converter Result High Register) and ADCLR (A/D Converter Result Low Register). It is selected for the corresponding channel to be converted by setting ADSEL[3:0]. To executing A/D conversion, ADST bit sets to '1'. The register ADCHR and ADCLR contains the results of the A/D conversion. When the conversion is completed, the result is loaded into the ADCHR and ADCLR, the A/D conversion status bit AFLAG is set to '1', and the A/D interrupt is set. For processing A/D conversion, AFLAG bit is read as '0'. If using STBY (power down) bit, the ADC is disabled. After STBY bit is reset (ADC power enable) and it is restarted, during some cycle, ADC conversion value may have an inaccurate value.

Also internal timer, external generating event, comparator and the timer can start ADC. At this time, interrupt enable of trigger sources is not required. If only the interrupt generation condition of each trigger source is satisfied, start adc.

ADC Conversion Time = ADCLK * 60 cycles

The A/D converter needs at least 20 μs for conversion time. So you must set the conversion time more than 20 μs. If the ADC conversion time is set short, the resolution is degraded.

11.7.2 Block Diagram

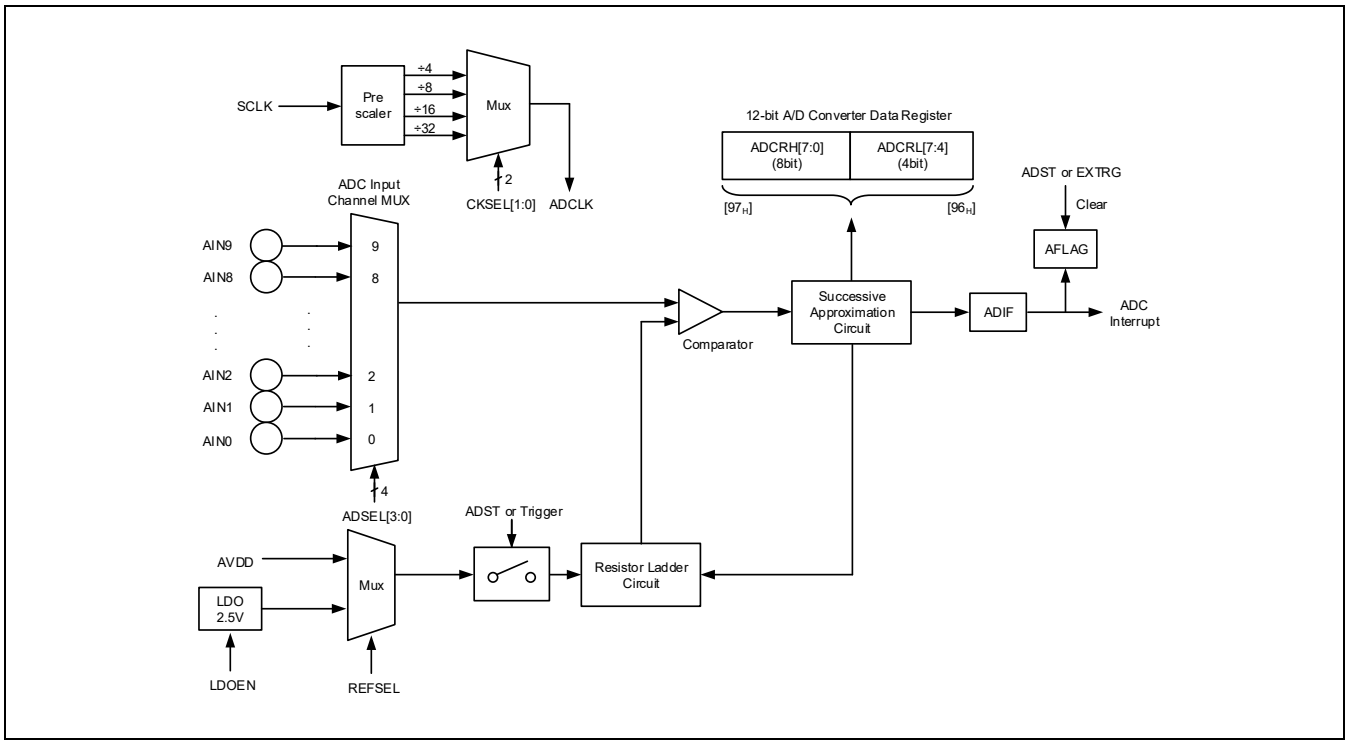


Figure 11.31 ADC Block Diagram

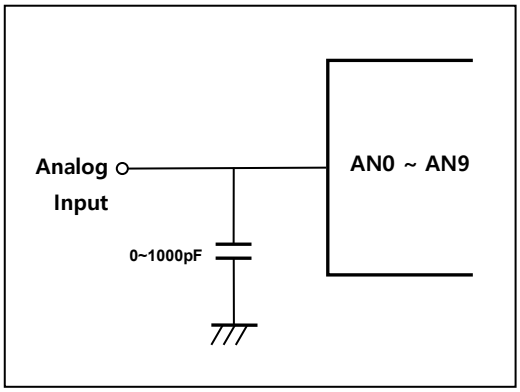


Figure 11.32 A/D Analog Input Pin Connecting Capacitor

11.7.3 ADC Operation

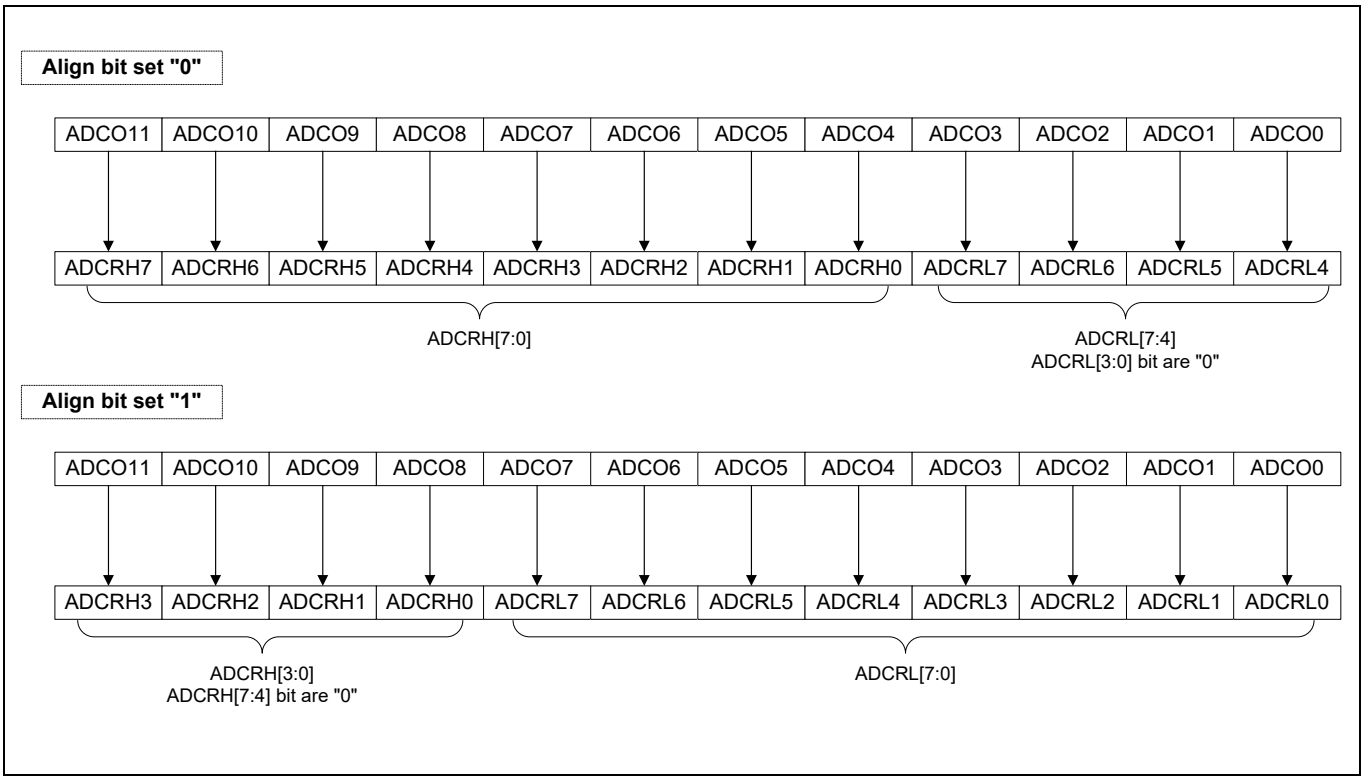


Figure 11.33 ADC Operation for Align bit

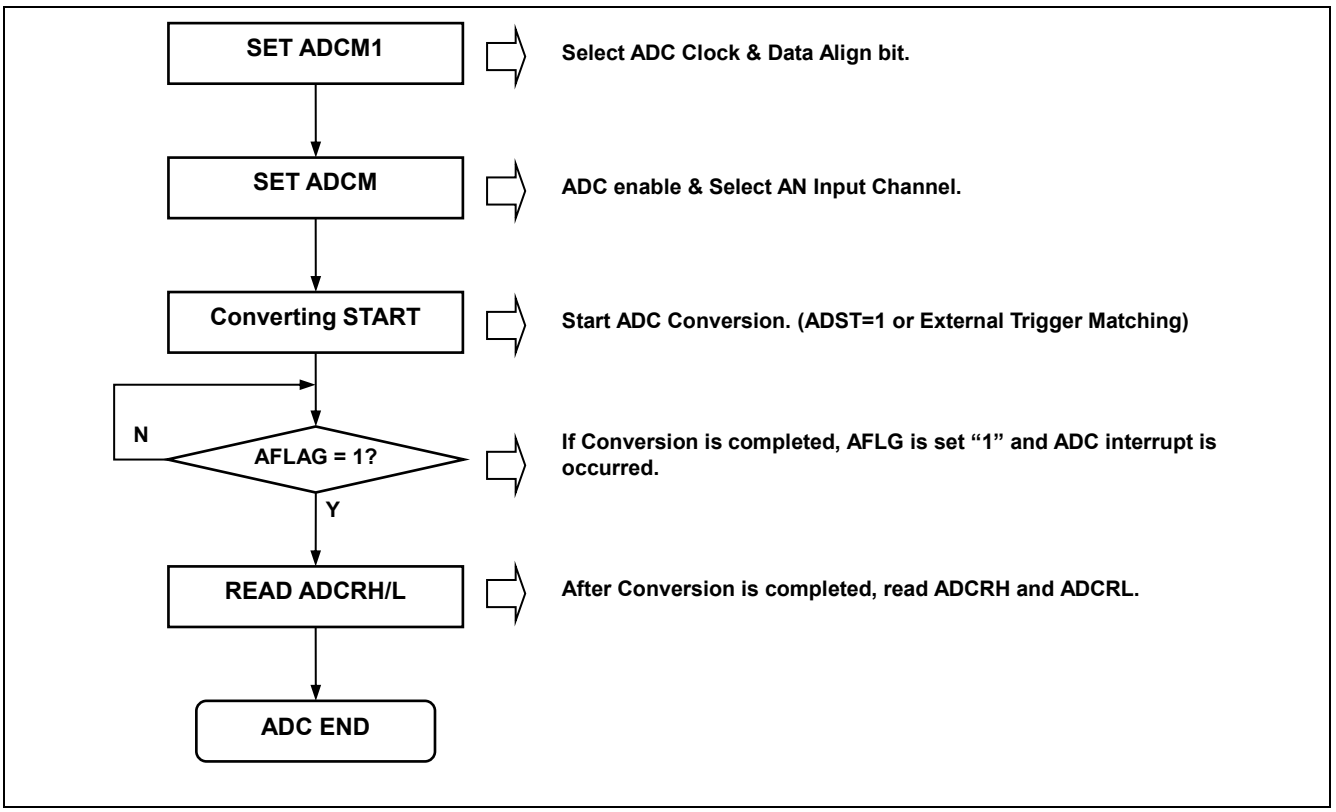


Figure 11.34 Converter Operation Flow

11.7.4 Register Map

Name	Address	Direction	Default	Description
ADCM	95H	R/W	8FH	A/D Converter Mode Register
ADCM1	96H	R/W	01H	A/D Converter Mode 1 Register
ADCRL	96H	R	xxH	A/D Converter Result Low Register
ADCRH	97H	R	xxH	A/D Converter Result High Register

Table 11.10 Register Map

11.7.5 Register Description for ADC

The ADC Register consists of A/D Converter Mode Register (ADCM), A/D Converter Result High Register (ADCRH), A/D Converter Result Low Register (ADCRL), A/D Converter Mode 1 Register (ADCM1)..

NOTE) When STBY bit is set to '1', ADCM1 is read.
If ADC enables, it is possible only to write ADCM1. When reading, ADCRL is read.

ADCM (A/D Converter Mode Register) : 95H

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 : 8FH

STBY Control operation of A/D standby (power down)
- 0 ADC module enable
- 1 ADC module disable (power down)

ADST Control A/D Conversion stop/start.
- 0 ADC Conversion Stop
- 1 ADC Conversion Start

REFSEL A/D Converter reference selection
- 0 Internal Reference (VDD)
- 1 Internal LDO Reference (V_{LDO} = 2.5V)

AFLAG A/D Converter operation state
- 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	Channel0(AN0)
0	0	0	1	Channel1(AN1)
0	0	1	0	Channel2(AN2)
0	0	1	1	Channel3(AN3)
0	1	0	0	Channel4(AN4)
0	1	0	1	Channel5(AN5)
0	1	1	0	Channel6(AN6)
0	1	1	1	Channel7(AN7)
1	0	0	0	Channel8(AN8)
1	0	0	1	Channel9(AN9)
1	0	1	0	reserved
1	0	1	1	reserved
1	1	0	0	reserved
1	1	0	1	reserved
1	1	1	0	reserved
1	1	1	1	reserved

NOTE)

1. When using ports as ADC input port, set corresponding PxFSRH, PxFSRL register to ADC input mode in order to open analog input switch and to prevent digital input.
2. LDO : Low Drop Out

ADCM1 (A/D Converter Mode Register) : 96H

7	6	5	4	3	2	1	0
EXTRG	TSEL2	TSEL1	TSEL0	AMUXEN	ALIGN	CKSEL1	CKSEL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 01H

EXTRG A/D external Trigger

0 External Trigger disable

1 External Trigger enable

TSEL[2:0] A/D Trigger Source selection

TSEL2	TSEL1	TSEL0	Description
0	0	0	Ext. Interrupt 10 (P23)
0	0	1	Ext. Interrupt 11 (P03)
0	1	0	Ext. Interrupt 12 (P12)
0	1	1	-
1	0	0	Timer0 interrupt
1	0	1	Timer1 interrupt
1	1	0	Timer2 interrupt

AMUXEN A/D analog input mux enable

0 AIN[15:0] channel disable(Floating)

1 AIN[15:0] channel enable

ALIGN A/D Converter data align selection.

0 MSB align (ADCRH[7:0], ADCRL[7:4])

1 LSB align (ADCRH[3:0], ADCRL[7:0])

CKSEL[1:0] A/D Converter Clock selection

CKSEL1	CKSEL0	ADC Clock
0	0	fx/4
0	1	fx/8
1	0	fx/16
1	1	fx/32

NOTE) fx : system clock
ADC clock should use below 3MHz

ADCRH (A/D Converter Result High Register) : 97H

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 result (8-bit), default
ADDL[11:8] LSB align, A/D Converter High result (4-bit)

ADCRL (A/D Converter Result Low Register) : 96H

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 result (4-bit), default
ADDL[7:0] LSB align, A/D Converter Low result (8-bit)

11.8 Comparator

11.8.1 Overview

The A9B114 contains analog comparators which can be easily used to compare the external input voltage with internally reference or externally reference voltage. ADC and Comparator have the same input. When the comparator input voltage is larger than the reference voltage comparator output status is '1' and interrupt flag is generated. The Comparator interrupt flag is assigned to one interrupt vector. Comparator output debounce clock and length can be set. The internal reference of the comparator is the resistor distribution method. When AVDD is selected, the reference value is 5V. At this time, 1.6V and 2.2V can be selected as the internal reference voltage of the comparator. If AVDD is below 5V, the internal reference value is reduced at a constant rate. However, when Low Drop Out(LDO) is selected, 1.6V and 2.2V are normally selected as 2.5V reference.

11.8.2 Block Diagram

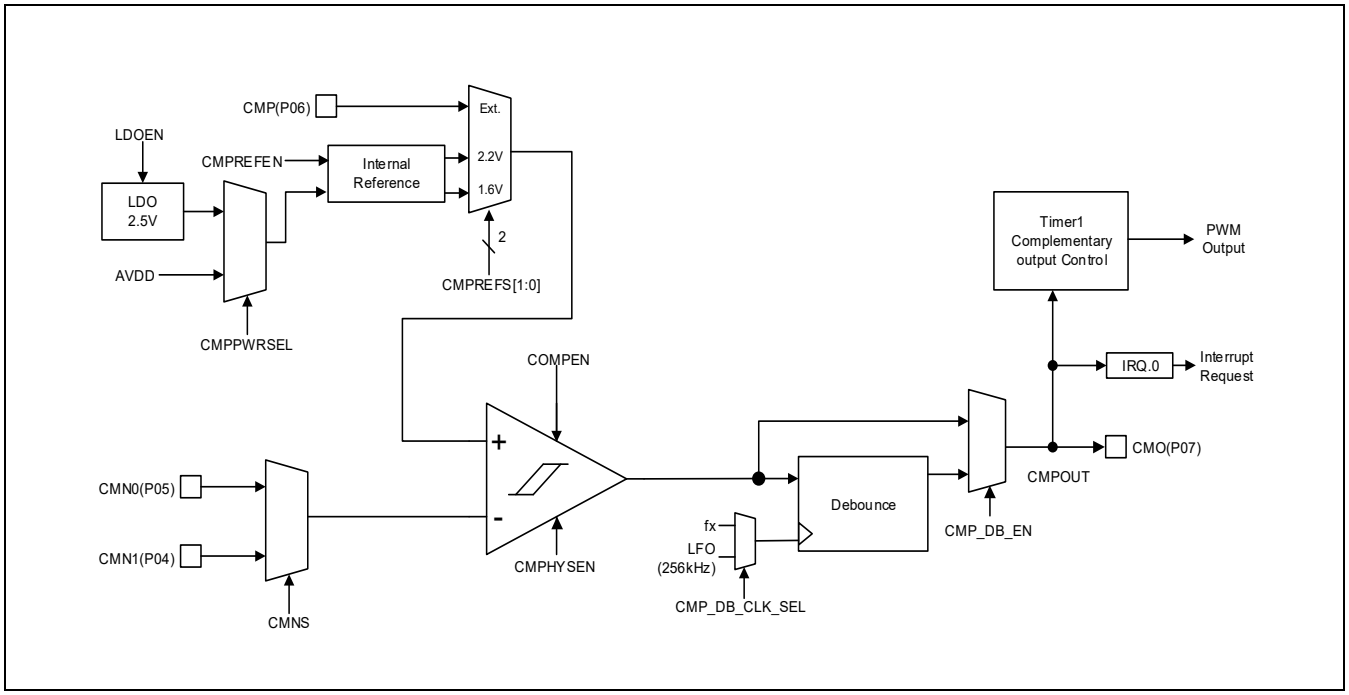


Figure 11.35 Comparator Block Diagram

11.8.3 Register map

Name	Address	Direction	Default	Description
CMPCR	D6H	R/W	01H	Comparator Control Register
CMPTR	D7H	R/W	00H	Comparator Option Register
CMPDBT	DEH	R/W	0FH	Comparator Output de-bounce time Register
LDOCR	AAH	R/W	00H	LDO Control Register

Table 11.11 Register Map

11.8.4 Register Description for Comparator

CMPCR (Comparator Control Register) : D6H

7	6	5	4	3	2	1	0
CMPEN	CMPPWRSEL	CMPREFS1	CMPREFS0	CMPREFEN	CMNS	CMPOF	CMPF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Initial value : 01H

CMPEN Comparator block operation control
- 0 Comparator disable
- 1 Comparator enable

CMPPWRSEL Comparator reference power selection
- 0 LDO 2.5V
- 1 AVDD

CMPREFS[1:0] Comparator reference selection

CMPREFS1	CMPREFS0	Description
0	0	Internal 1.6V
0	1	External PIN (P06)
1	0	Internal 2.2V
1	1	External PIN (P06)

CMPREFEN Comparator internal reference control
- 0 Internal reference disable
- 1 Internal reference enable

CMNS Comparator input pin selection
- 0 CMN1 (P04)
- 1 CMN0 (P05)

CMPOF Comparator output selection
- 0 Output ‘H’ when CMP > CMN and Output ‘L’ when CMP < CMN
- 1 Output ‘H’ when CMP < CMN and Output ‘L’ when CMP > CMN

CMPF Comparator output flag
- 0 Output ‘L’
- 1 Output ‘H’

CMPTR (Comparator Option Register) : D7H

7	6	5	4	3	2	1	0
CMPHYSEN	CMP_DB_EN	–	CMP_DB_CLK_SEL	CMPEDGE1	CMPEDGE0	CMPPWMC1	CMPPWMC0
RW	RW	–	RW	RW	RW	RW	RW

Initial value : 00H

CMPHYSEN Comparator block hysteresis operation control
- 0 Comparator hysteresis disable
- 1 Comparator hysteresis enable

CMP_DB_EN Comparator output de-bounce control
- 0 Output de-bounce disable
- 1 Output de-bounce enable

CMP_DB_CLK_SEL Comparator output de-bounce clock source
- 0 fx (16MHz)
- 1 LFO (256kHz)

CMPEDGE[1:0] Comparator interrupt trigger selection

CMPEDGE1	CMPEDGE0	Description
0	0	Reserve
0	1	Rising edge trigger
1	0	Falling edge trigger
1	1	Both edge trigger

CMPPWMC[1:0] Timer1 PWM output Control by comparator

CMPPWMC1	CMPPWMC0	Description
0	0	Not controlled
0	1	PWM stop when comparator out ‘H’
1	0	PWM stop when CMP > CMN
1	1	PWM stop when CMP < CMN

CMPDBT (Comparator De-bounce Time Register) : DEH

7	6	5	4	3	2	1	0
–	–	–	–	CMPDBT3	CMPDBT2	CMPDBT1	CMPDBT0
–	–	–	–	R/W	R/W	R/W	R/W

Initial value : 0FH

CMPDBT[3:0] Comparator output de-bounce time

Select comparator output de-bounce length

De-bounce length = CMPDBT x 2 x 1/f_{dbclk2}

LDOCR (LDO Control Register) : AAH

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

Initial value : 00H

DSCHGEN LDO by pass mode

0 Disable

1 Enable

LDOEN LDO control

0 LDO disable

1 LDO enable

Note) LDO operating mode

DSCHGEN_I	LDOEN	Description
0	0	LDO disable
0	1	Normal mode LDO output 2.5V
1	0	Bypass mode LDO output is bypass to AVDD

11.9 USART

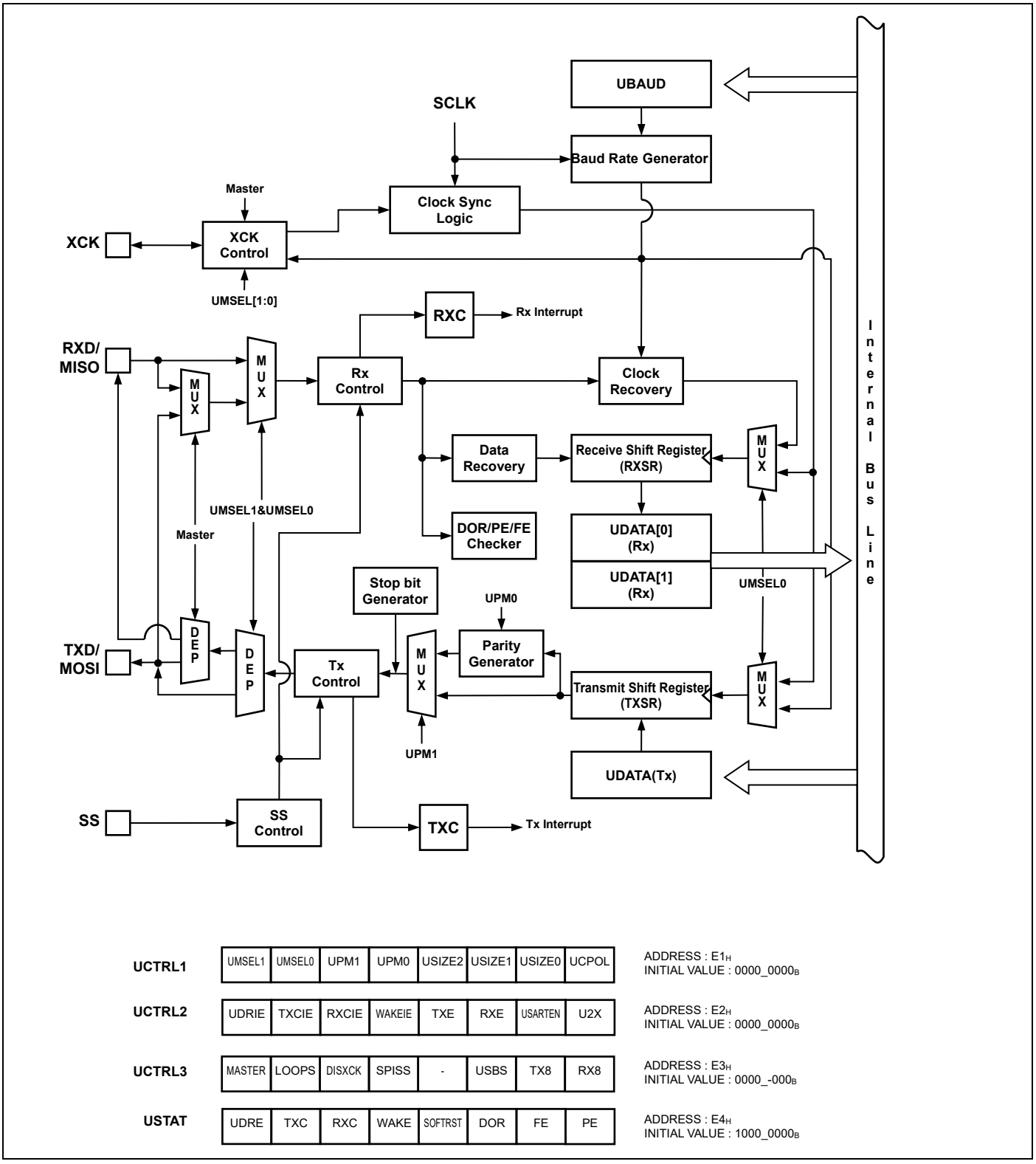
11.9.1 Overview

The Universal Synchronous and Asynchronous serial Receiver and Transmitter (USART) 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
- Master or Slave Clocked Synchronous and SPI Operation
- Supports all four SPI Modes of Operation (Mode 0, 1, 2, 3)
- LSB First or MSB First Data Transfer @SPI mode
- High Resolution 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
- Digital Low Pass Filter
- Three Separate Interrupts on TX Complete, TX Data Register Empty and RX Complete
- Double Speed Asynchronous Communication Mode

USART has three main parts of Clock Generator, Transmitter and Receiver. The Clock Generation logic consists of synchronization logic for external clock input used by synchronous or SPI slave operation, and the baud rate generator for asynchronous or master (synchronous or SPI) operation. The Transmitter consists of a single write buffer, a serial shift register, parity generator and control logic for handling different serial frame formats. The write buffer allows a continuous transfer of data without any delay between frames. The receiver is the most complex part of the USART 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 (UDATA) and control logic. The Receiver supports the same frame formats as the Transmitter and can detect Frame Error, Data OverRun and Parity Errors.

11.9.2 Block Diagram



UCTRL1	UMSEL1	UMSEL0	UPM1	UPM0	USIZE2	USIZE1	USIZE0	UCPOL	ADDRESS : $E1_H$ INITIAL VALUE : 0000_0000_B
UCTRL2	UDRIE	TXCIE	RXCIE	WAKEIE	TXE	RXE	USARTEN	U2X	ADDRESS : $E2_H$ INITIAL VALUE : 0000_0000_B
UCTRL3	MASTER	LOOPS	DISXCK	SPISS	-	USBS	TX8	RX8	ADDRESS : $E3_H$ INITIAL VALUE : 0000_-000_B
USTAT	UDRE	TXC	RXC	WAKE	SOFTRST	DOR	FE	PE	ADDRESS : $E4_H$ INITIAL VALUE : 1000_0000_B

Figure 11.36 USART Block Diagram

11.9.3 Clock Generation

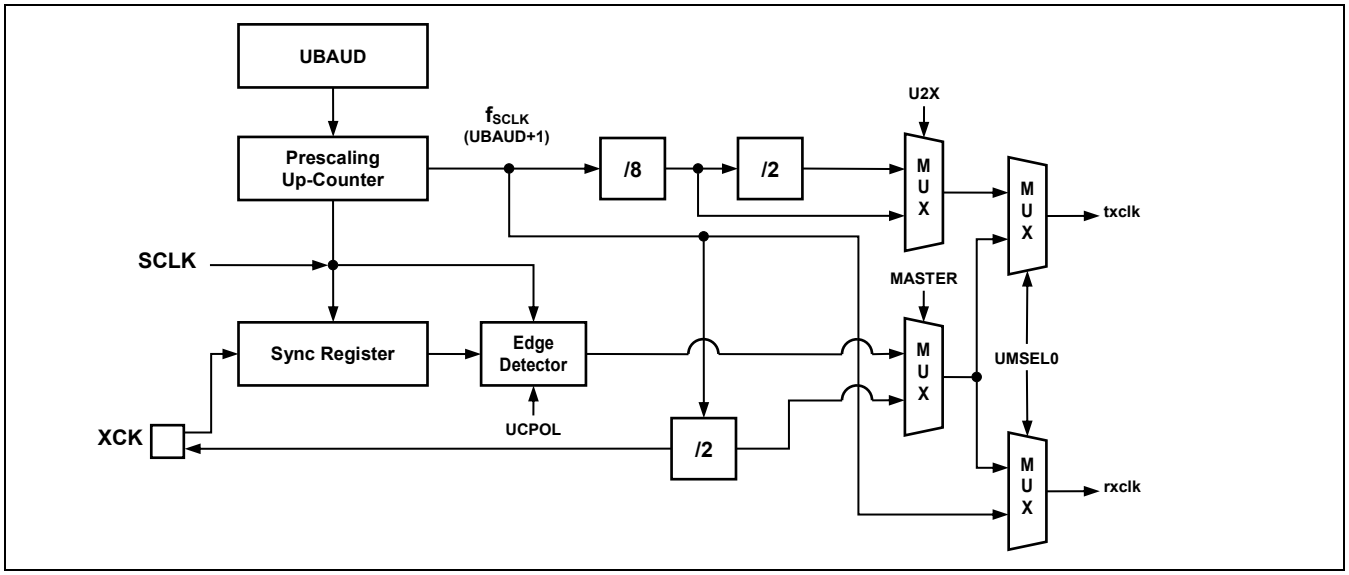


Figure 11.37 Clock Generation Block Diagram

The Clock generation logic generates the base clock for the Transmitter and Receiver. The USART supports four modes of clock operation and those are Normal Asynchronous, Double Speed Asynchronous, Master Synchronous and Slave Synchronous. The clock generation scheme for Master SPI and Slave SPI mode is the same as Master Synchronous and Slave Synchronous operation mode. The UMSELn bit in UCTRL1 register selects between asynchronous and synchronous operation. Asynchronous Double Speed mode is controlled by the U2X bit in the UCTRL2 register. The MASTER bit in UCTRL2 register controls whether the clock source is internal (Master mode, output port) or external (Slave mode, input port). The XCK pin is only active when the USART operates in Synchronous or SPI mode.

Table below contains equations for calculating the baud rate (in bps).

Operating Mode	Equation for Calculating Baud Rate
Asynchronous Normal Mode (U2X=0)	$\text{Baud Rate} = \frac{\text{fSCLK}}{16(\text{UBAUDx} + 1)}$
Asynchronous Double Speed Mode (U2X=1)	$\text{Baud Rate} = \frac{\text{fSCLK}}{8(\text{UBAUDx} + 1)}$
Synchronous or SPI Master Mode	$\text{Baud Rate} = \frac{\text{fSCLK}}{2(\text{UBAUDx} + 1)}$

Table 11.12 Equations for Calculating Baud Rate Register Setting

11.9.4 External Clock (XCK)

External clocking is used by the synchronous or SPI slave modes of operation.

External clock input from the XCK pin is sampled by a synchronization logic to remove meta-stability. The output from the synchronization logic must then pass through an edge detector before it can be used by the Transmitter and Receiver. This process introduces a two CPU clock period delay and the maximum frequency of the external XCK pin is limited by the following equation.

$$fXCK = \frac{fSCLK}{4}$$

Where fXCK is the frequency of XCK and fSCLK is the frequency of main system clock (SCLK).

11.9.5 Synchronous mode Operation

When synchronous or SPI mode is used, the XCK pin will be used as either clock input (slave) or clock output (master). The dependency between the clock edges and data sampling or data change is the same. The basic principle is that data input on RXD (MISO in SPI mode) pin is sampled at the opposite XCK clock edge of the edge in the data output on TXD (MOSI in SPI mode) pin is changed.

The UCPOL bit in UCTRL1 register selects which XCK clock edge is used for data sampling and which is used for data change. As shown in the figure below, when UCPOL is zero the data will be changed at XCK rising edge and sampled at XCK falling edge.

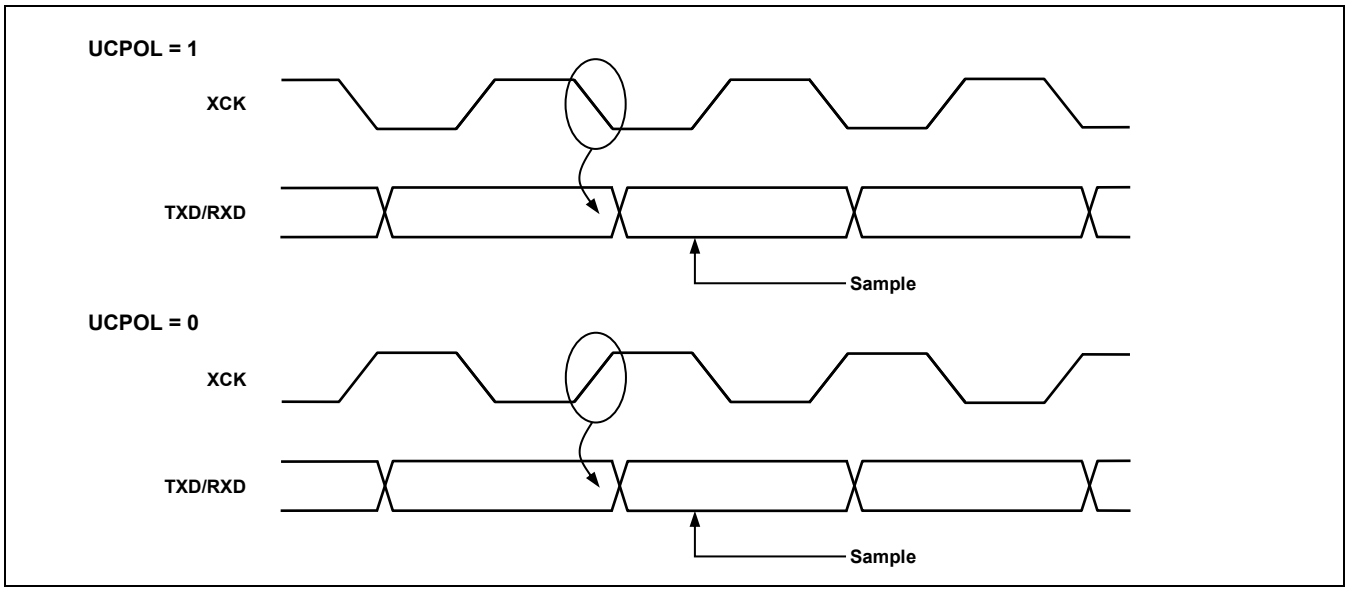


Figure 11.38 Synchronous Mode XCKn Timing

11.9.6 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 checking.

The USART 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 a total of nine, are succeeding, ending with the most significant bit (MSB). If enabled the parity bit is inserted after the data bits, before the stop bits. 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 next figure shows the possible combinations of the frame formats. Bits inside brackets are optional.

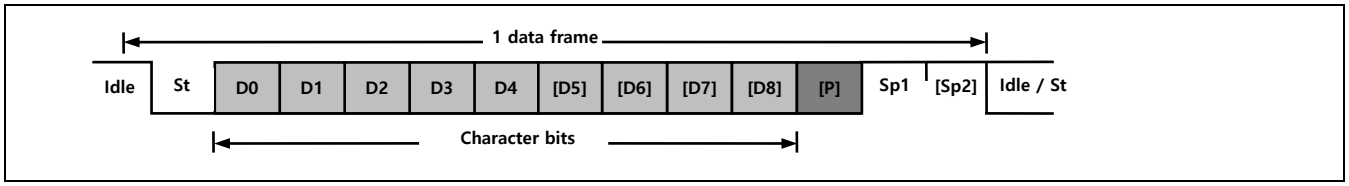


Figure 11.39 frame format

1 data frame consists of the following bits

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

The frame format used by the USART is set by the USIZE[2:0], UPM[1:0] and USBS bits in UCTRL1 register. The Transmitter and Receiver use the same setting.

11.9.7 Parity bit

The parity bit is calculated by doing an exclusive-or of all the data bits. If odd parity is used, the result of the exclusive-or is inverted. The parity bit is located between St + bits 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

11.9.8 USART Transmitter

The USART Transmitter is enabled by setting the TXE bit in UCTRL1 register. When the Transmitter is enabled, the normal port operation of the TXD pin is overridden by the serial output pin of USART. The baud-rate, operation mode and frame format must be setup once before doing any transmissions. If synchronous or SPI operation is used, the clock on the XCK pin will be overridden and used as transmission clock. If USART operates in SPI mode, SS pin is used as SS input pin in slave mode or can be configured as SS output pin in master mode. This can be done by setting SPISS bit in UCTRL3 register.

11.9.8.1 Sending Tx data

A data transmission is initiated by loading the transmit buffer (UDATA 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 at the settings of control registers. If the 9-bit characters are used in asynchronous or synchronous operation mode (USIZE[2:0]=7), the ninth bit must be written to the TX8 bit in UCTRL3 register before loading transmit buffer (UDATA register).

11.9.8.2 Transmitter flag and interrupt

The USART Transmitter has 2 flags which indicate its state. One is USART Data Register Empty (UDRE) and the other is Transmit Complete (TXC). Both flags can be used as interrupt sources.

UDRE flag indicates whether the transmit buffer is ready to be loaded with new data. This bit is set when the transmit buffer is empty and cleared when the transmit buffer contains transmittion data which 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 not valid.

When the Data Register Empty Interrupt Enable (UDRIE) bit in UCTRL2 register is set and the Global Interrupt is enabled, USART Data Register Empty Interrupt is generated while UDRE flag is set.
The Transmit Complete (TXC) flag bit is set when the entire frame in the transmit shift register has been shifted out and there are no more data in the transmit buffer. The TXC flag is automatically cleared when the Transmit Complete Interrupt service routine is executed, or it can be cleared by writing ‘0’ to TXC bit in USTAT register.
When the Transmit Complete Interrupt Enable (TXCIE) bit in UCTRL2 register is set and the Global Interrupt is enabled, USART Transmit Complete Interrupt is generated while TXC flag is set.

11.9.8.3 Parity Generator

The Parity Generator calculates the parity bit for the sending serial frame data. When parity bit is enabled (UPM[1]=1), the transmitter control logic inserts the parity bit between the bits and the first stop bit of the sending frame.

11.9.8.4 Disabling Transmitter

Disabling the Transmitter by clearing the TXE bit will not become effective until ongoing transmission is completed. When the Transmitter is disabled, the TXD pin is used as normal General Purpose I/O (GPIO) or primary function pin.

11.9.9 USART Receiver

The USART Receiver is enabled by setting the RXE bit in the UCTRL1 register. When the Receiver is enabled, the normal pin operation of the RXD pin is overridden by the USART as the serial input pin of the Receiver. The baud-rate, mode of operation and frame format must be set before serial reception. If synchronous or SPI operation is used, the clock on the XCK pin will be used as transfer clock. If USART operates in SPI mode, SS pin is used as SS input pin in slave mode or can be configured as SS output pin in master mode. This can be done by setting SPISS bit in UCTRL3 register.

11.9.9.1 Receiving Rx data

When USART is in synchronous or asynchronous operation mode, the Receiver starts data reception when it detects a valid start bit (LOW) on RXD pin. Each bit after start bit is sampled at pre-defined baud-rate (asynchronous) or sampling edge of XCK (synchronous), and shifted into the receive shift register until the first stop bit of a frame is received. Even if there's 2nd stop bit in the frame, the 2nd 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 UDATA register.

If 9-bit characters are used (USIZE[2:0] = 7), the ninth bit is stored in the RX8 bit position in the UCTRL3 register. The 9th bit must be read from the RX8 bit before reading the low 8 bits from the UDATA register. Likewise, the error flags FE, DOR, PE must be read before reading the data from UDATA register. This is because the error flags are stored in the same FIFO position of the receive buffer.

11.9.9.2 Receiver flag and interrupt

The USART Receiver has one flag that indicates the Receiver state.

The Receive Complete (RXC) flag indicates whether there are unread data present 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 (RXE=0), the receiver buffer is flushed and the RXC flag is cleared.

When the Receive Complete Interrupt Enable (RXCIE) bit in the UCTRL2 register is set and Global Interrupt is enabled, the USART Receiver Complete Interrupt is generated while RXC flag is set.

The USART Receiver has three error flags which are Frame Error (FE), Data OverRun (DOR) and Parity Error (PE). These error flags can be read from the USTAT register. As data received 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 UDATA register, read the USTAT register first which contains error flags.

The Frame Error (FE) flag indicates the state of the first stop bit. The FE flag is set when the stop bit was correctly detected as “1”, and the FE flag is cleared when the stop bit was incorrect, ie detected as “0”. This flag can be used for detecting out-of-sync conditions between data frames.

The Data OverRun (DOR) flag indicates data loss due to a receive buffer full condition. A DOR 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 DOR flag is set, all the incoming data are lost. To prevent data loss or clear this flag, read the receive buffer.

The Parity Error (PE) flag indicates that the frame in the receive buffer had a Parity Error when received. If Parity Check function is not enabled (UPM[1]=0), the PE bit is always read “0”.

NOTE) The error flags related to receive operation are not used when USART is in SPI mode.

11.9.9.3 Parity Checker

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

11.9.9.4 Disabling Receiver

In contrast to Transmitter, disabling the Receiver by clearing RXE bit makes the Receiver inactive immediately. When the Receiver is disabled the Receiver flushes the receive buffer and the remaining data in the buffer is all reset. The RXD pin is not overridden the function of USART, so RXD pin becomes normal GPIO or primary function pin.

11.9.9.5 Asynchronous Data Reception

To receive asynchronous data frame, the USART 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 RXD pin.

The Data recovery logic samples incoming bits and low pass filters them, and this removes the noise of RXD pin.
The next figure illustrates the sampling process of the start bit of an incoming frame. The sampling rate is 16 times the baud-rate for normal mode, and 8 times the baud rate for Double Speed mode (U2X=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.

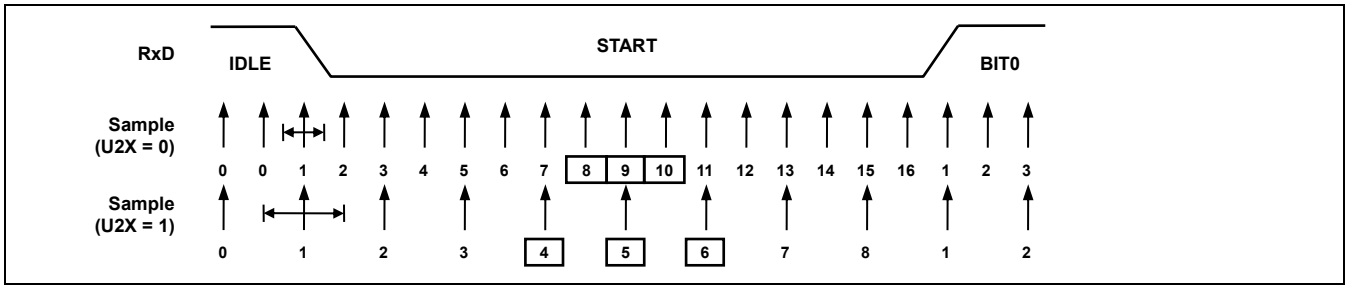


Figure 11.40 Start bit Sampling

When the Receiver is enabled (RXE=1), the clock recovery logic tries to find a high to low transition on the RXD line, the start bit condition. After detecting high to low transition on RXD line, the clock recovery logic uses samples 8,9, and 10 for Normal mode, and samples 4, 5, and 6 for Double Speed 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 for Normal mode, samples 4, 5, and 6 for Double Speed mode. If more than 2 samples have low levels, the received bit is considered to a logic 0 and 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.

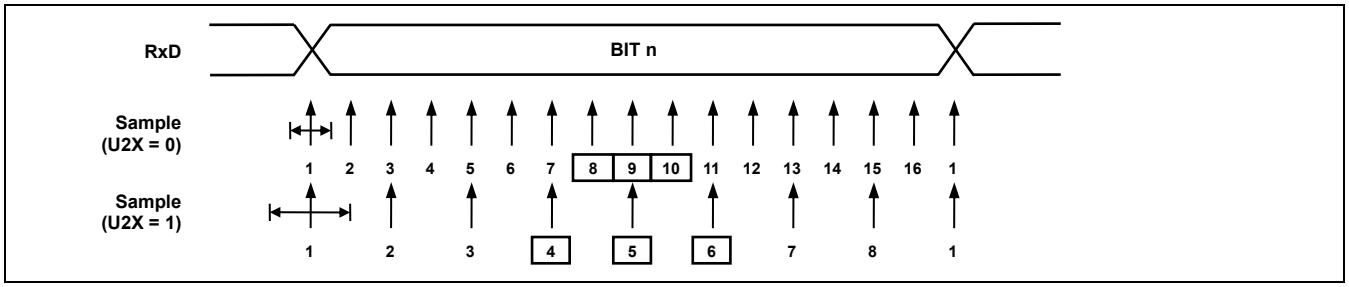


Figure 11.41 Sampling of Data and Parity bit

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

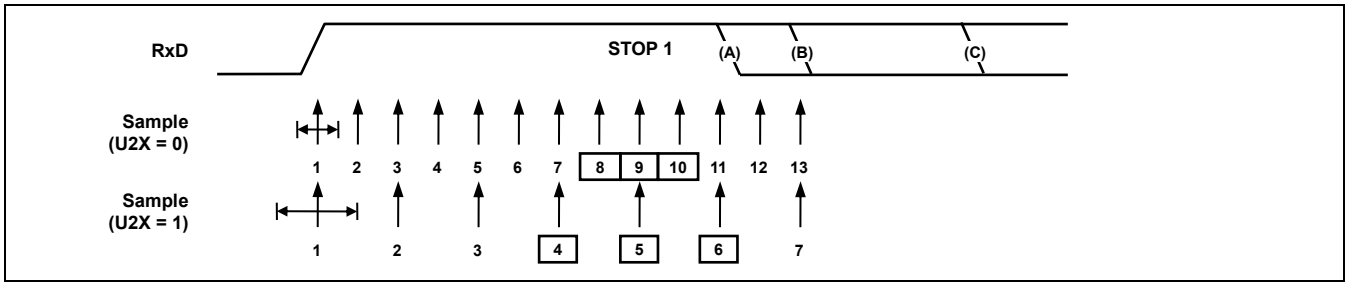


Figure 11.42 Stop bit Sampling and Next Start bit Sampling

11.9.10 SPI Mode

The USART 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
- Master or Slave operation
- Supports all four SPI modes of operation (mode0, 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 (UMSEL[1:0]=3), the Slave Select (SS) pin becomes active low input in slave mode operation, or can be output in master mode operation if SPISS bit is set.

Note that during SPI mode of operation, the pin RXD is renamed as MISO and TXD is renamed as MOSI for compatibility to other SPI devices.

11.9.10.1 SPI Clock formats and timing

To accommodate a wide variety of synchronous serial peripherals from different manufacturers, the USART has a clock polarity bit (UCPOL) and a clock phase control bit (UCPHA) to select one of four clock formats for data transfers. UCPOL selectively insert an inverter in series with the clock. UCPHA selects one of two different clock phase relationships between the clock and data. Note that UCPHA and UCPOL bits in UCTRL1 register have different meanings according to the UMSEL[1:0] bits which decides the operating mode of USART.

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

SPI Mode	UCPOL	UCPHA	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)

Table 11.13 CPOL Functionality

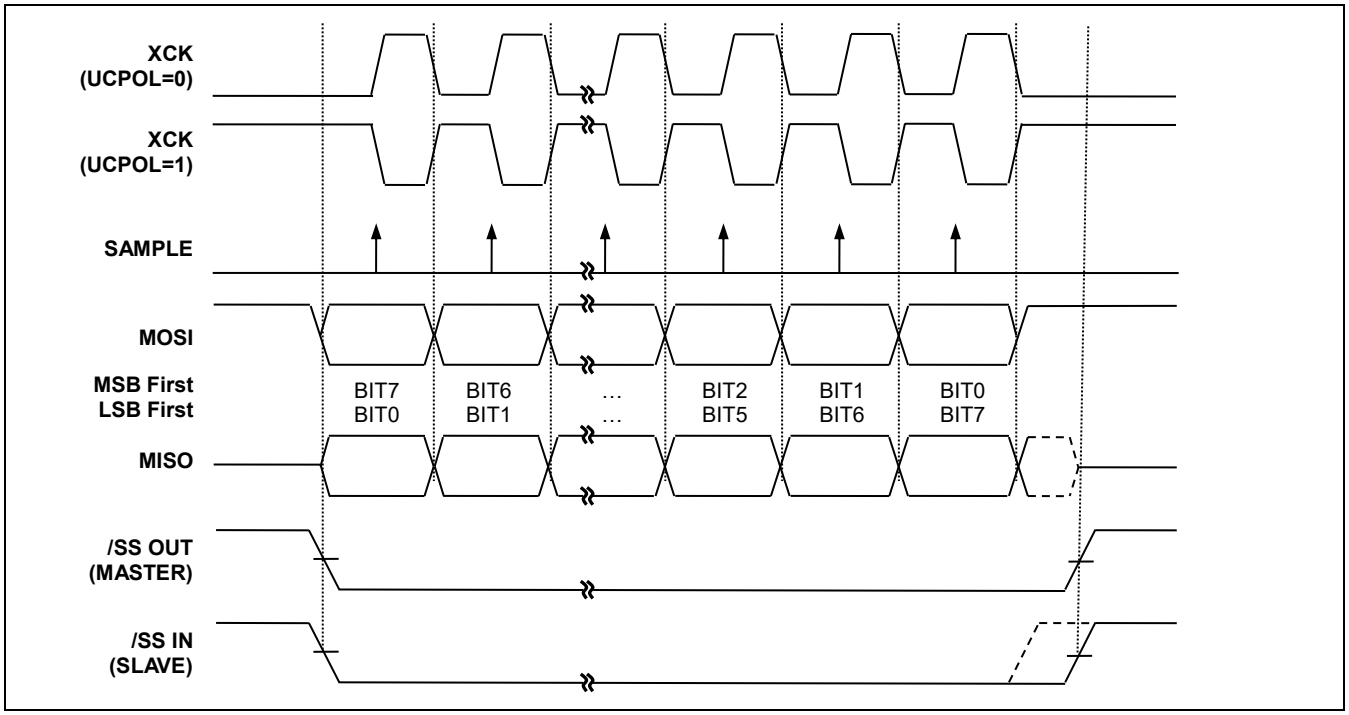


Figure 11.43 SPI Clock Formats when UCPHA=0

When UCPHA=0, the slave begins to drive its MISO output with the first data bit value when SS goes to active low. The first XCK edge causes both the master and the slave to sample the data bit value on their MISO and MOSI inputs, respectively. At the second XCK edge, the USART shifts the second data bit value out to the MOSI and MISO outputs of the master and slave, respectively. Unlike the case of UCPHA=1, when UCPHA=0, the slave's SS 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 SS input.

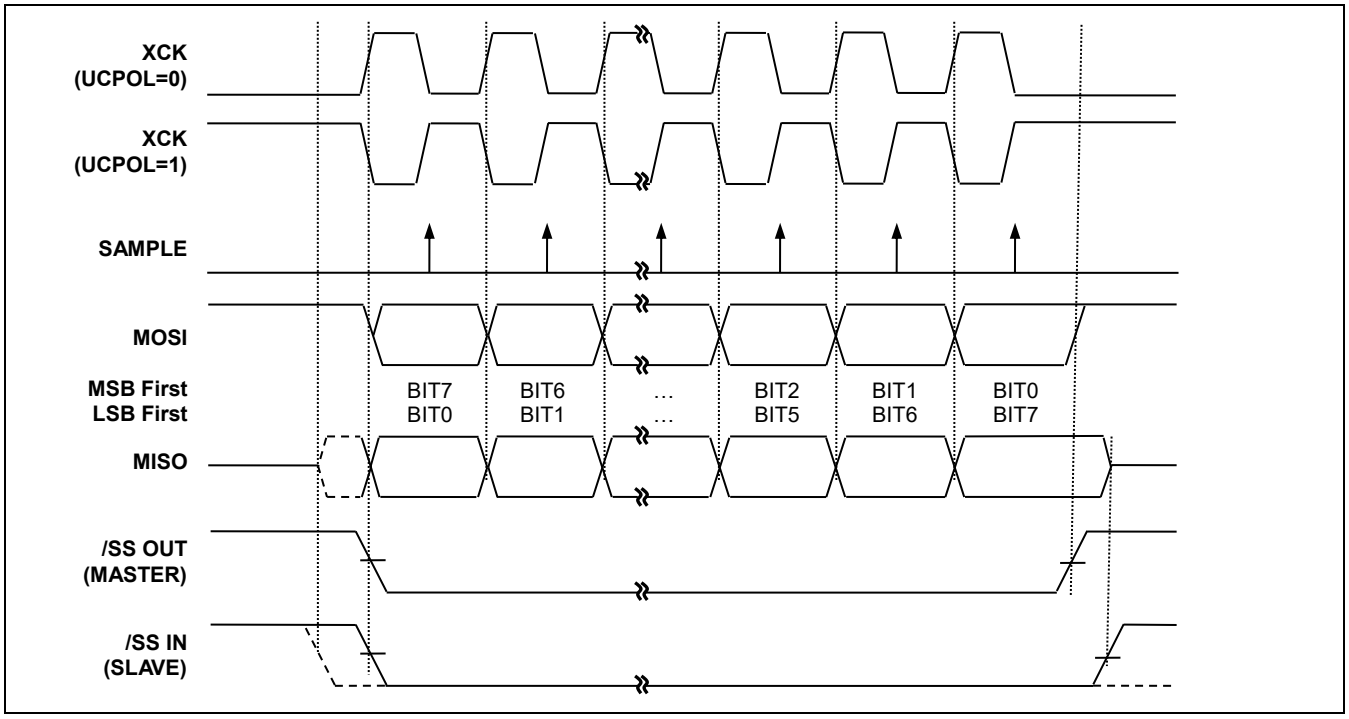


Figure 11.44 SPI Clock Formats when UCPHA=1

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

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

11.9.11 Register Map

Name	Address	Direction	Default	Description
UCTRL1	E1H	R/W	00H	USART Control 1 Register
UCTRL2	E2H	R/W	00H	USART Control 2 Register
UCTRL3	E3H	R/W	00H	USART Control 3 Register
UCTRL4	EBH	R/W	00H	USART Control 4 Register
USTAT	E4H	R	80H	USART Status Register
UBAUD	E5H	R/W	FFH	USART Baud Rate Generation Register
UDATA	E6H	R/W	00H	USART Data Register
FPCR	ECH	R/W	00H	USART Floating Point Counter Register

Table 11.14 USART Register Map

11.9.12 USART Register Description

USART module consists of USART Control 1 Register (UCTRL1), USART Control 2 Register (UCTRL2), USART Control 3 Register (UCTRL3), USART Control 4 Register (UCTRL4), USART Floaint Point Counter (FPCR), USART Status Register (USTAT), USART Data Register (UDATA), and USART Baud Rate Generation Register (UBAUD).

11.9.13 Register Description for USART

UCTRL1 (USART Control 1 Register) E1H

7	6	5	4	3	2	1	0
UMSEL1	UMSEL0	UPM1	UPM0	USIZE2	USIZE1 UDORD	USIZE0 UCPHA	UCPOL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00_H

UMSEL[1:0] Selects operation mode of USART

UMSEL1	UMSEL0	Operating Mode
0	0	Asynchronous Mode (Normal Uart)
0	1	Synchronous Mode (Synchronous Uart)
1	0	Reserved
1	1	SPI Mode

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

UPM1	UPM0	Parity mode
0	0	No Parity
0	1	Reserved
1	0	Even Parity
1	1	Odd Parity

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

USIZE2	USIZE1	USIZE0	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

UDORD This bit is in the same bit position with USIZE1. In SPI mode, when set to one the MSB of the data byte is transmitted first. When set to zero the LSB of the data byte is transmitted first.

0 LSB First

1 MSB First

UCPOL Selects polarity of XCK in synchronous or SPI mode

0 TXD change @Rising Edge, RXD change @Falling Edge

1 TXD change @ Falling Edge, RXD change @ Rising Edge

UCPHA This bit is in the same bit position with USIZE0. In SPI mode, along with UCPOL bit, selects one of two clock formats for different kinds of synchronous serial peripherals. Leading edge means first XCK edge and trailing edge means 2nd or last clock edge of XCK in one XCK pulse. And Sample means detecting of incoming receive bit, Setup means preparing transmit data.

UCPOL	UCPHA	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)

UCTRL2 (USART Control 2 Register) E2H

7	6	5	4	3	2	1	0
UDRIE	TXCIE	RXCIE	WAKEIE	TXE	RXE	USARTEN	U2X
RW	RW	RW	RW	RW	RW	RW	RW

Initial value : 00_H

UDRIE Interrupt enable bit for USART Data Register Empty.
- 0 Interrupt from UDRE is inhibited (use polling)
- 1 When UDRE is set, request an interrupt

TXCIE Interrupt enable bit for Transmit Complete.
- 0 Interrupt from TXC is inhibited (use polling)
- 1 When TXC is set, request an interrupt

RXCIE Interrupt enable bit for Receive Complete
- 0 Interrupt from RXC is inhibited (use polling)
- 1 When RXC is set, request an interrupt

WAKEIE Interrupt enable bit for Asynchronous Wake in STOP mode. When device is in stop mode, if RXD goes to LOW level an interrupt can be requested to wake-up system.
- 0 Interrupt from Wake is inhibited
- 1 When WAKE is set, request an interrupt

TXE Enables the transmitter unit.
- 0 Transmitter is disabled
- 1 Transmitter is enabled

RXE Enables the receiver unit.
- 0 Receiver is disabled
- 1 Receiver is enabled

USARTEN Activate USART module by supplying clock.
- 0 USART is disabled (clock is halted)
- 1 USART is enabled

U2X This bit only has effect for the asynchronous operation and selects receiver sampling rate.
- 0 Normal asynchronous operation
- 1 Double Speed asynchronous operation

UCTRL3 (USART Control 3 Register) E3H

7	6	5	4	3	2	1	0
MASTER	LOOPS	DISXCK	SPISS	-	USBS	TX8	RX8
R/W	R/W	R/W	R/W	-	R/W	R/W	R/W

Initial value : 00_H

MASTER Selects master or slave in SPI or Synchronous mode operation and controls the direction of XCK pin.
- 0 Slave mode operation and XCK is input pin.
- 1 Master mode operation and XCK is output pin

LOOPS Controls the Loop Back mode of USART, for test mode
- 0 Normal operation
- 1 Loop Back mode

DISXCK In Synchronous mode of operation, selects the waveform of XCK output.
- 0 XCK is free-running while USART is enabled in synchronous master mode.
- 1 XCK is active while any frame is on transferring.

SPISS Controls the functionality of SS pin in master SPI mode.
- 0 SS pin is normal GPIO or other primary function
- 1 SS output to other slave device

USBS Selects the length of stop bit in Asynchronous or Synchronous mode of operation.
- 0 1 Stop bit
- 1 2 Stop bit

TX8 The ninth bit of data frame in Asynchronous or Synchronous mode of operation. Write this bit first before loading the UDATA register.
- 0 MSB (9^{th} bit) to be transmitted is '0'
- 1 MSB (9^{th} bit) to be transmitted is '1'

RX8 The ninth bit of data frame in Asynchronous or Synchronous mode of operation. Read this bit first before reading the receive buffer.
- 0 MSB (9^{th} bit) received is '0'
- 1 MSB (9^{th} bit) received is '1'

UCTRL4 (USART Control 4 Register) EBH

7	6	5	4	3	2	1	0
-	-	-	-	-	FPCREN	AOVSSEL	AOVSEN
-	-	-	-	-	R/W	R/W	R/W

Initial value : 00_H

FPCREN Enable baud rate compensation
- 0 Disable
- 1 Enable

AOVSEN Enable additional oversampling rates selection
- 0 Disable
- 1 Enable

AOVSSEL Select additional oversampling rates
- 0 Select X13
- 1 Select X4

USTAT (USART Status Register) E4H

7	6	5	4	3	2	1	0
UDRE	TXC	RXC	WAKE	SOFTRST	DOR	FE	PE
R/W	R/W	R/W	R/W	R/W	R	R	R

Initial value : 80_H

UDRE The UDRE flag indicates if the transmit buffer (UDATA) is ready to be loaded with new data. If UDRE is '1', it means the transmit buffer is empty and can hold one or two new data. This flag can generate an UDRE interrupt. Writing '0' to this bit position will clear UDRE flag.

0 Transmit buffer is not empty.

1 Transmit buffer is empty.

TXC 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 TXC interrupt is executed. It is also cleared by writing '0' to this bit position. This flag can generate a TXC interrupt.

0 Transmission is ongoing.

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

RXC 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 RXC flag can be used to generate a RXC interrupt.

0 There is no data unread in the receive buffer

1 There are more than 1 data in the receive buffer

WAKE This flag is set when the RX pin is detected low while the CPU is in stop mode. This flag can be used to generate a WAKE interrupt. This bit is set only when in asynchronous mode of operation. NOTE

0 No WAKE interrupt is generated.

1 WAKE interrupt is generated.

SOFTRST This is an internal reset and only has effect on USART. Writing '1' to this bit initializes the internal logic of USART and is auto cleared.

0 No operation

1 Reset USART

DOR 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

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

0 No Frame Error

1 Frame Error detected

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

0 No Parity Error

1 Parity Error detected

NOTE) When the WAKE function of USART is used as a release source from STOP mode, it is required to clear this bit in the RX interrupt service routine. Else the device will not wake-up from STOP mode again by the change of RX pin.

UBAUD (USART Baud-Rate Generation Register) E5H

7	6	5	4	3	2	1	0
UBAUD7	UBAUD6	UBAUD5	UBAUD4	UBAUD3	UBAUD2	UBAUD1	UBAUD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FF_H

UBAUD [7:0] The value in this register is used to generate internal baud rate in asynchronous mode or to generate XCK clock in synchronous or SPI mode. To prevent malfunction, do not write '0' in asynchronous mode, and do not write '0' or '1' in synchronous or SPI mode.

UDATA (USART Data Register) E6H

7	6	5	4	3	2	1	0
UDATA7	UDATA6	UDATA5	UDATA4	UDATA3	UDATA2	UDATA1	UDATA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00_H

UDATA [7:0] The USART 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 UDATA register. Reading the UDATA register returns the contents of the Receive Buffer.

Write this register only when the UDRE flag is set. In SPI or synchronous master mode, write this register even if TX is not enabled to generate clock, XCK.

FPCR (USART Floating Point Register) ECH

7	6	5	4	3	2	1	0
FPCR7	FPCR6	FPCR5	FPCR4	FPCR3	FPCR2	FPCR1	FPCR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00_H

FPCR [7:0] USART Floating Point Counter

8-bit floating point counter

NOTE) BAUD RATE compensation can be used in the following ways:

Example1) Condition : sysclk = 16MHz, Baud rate = 9600 bps, Asynchronous Normal Mode (U2X = 0)

Baud rate = sysclk / 16 x (UBAUD + 1)

Calculated UBAUD = (1000000 / Target Baud rate) – 1 = 103.17, Error rate = 0.17 ⇒ **UBAUD = 104**

UCTRL4 = 0x04, Enable baudrate Compensation

Calculated FPCR = (UBAUD - Calculated UBAUD) x 256 = (104 – 103.17) x 256 = 212.48 ⇒ **FPCR = 213**

Example2) Condition : sysclk = 16MHz, Baud rate = 115,200 bps, Asynchronous Normal Mode (U2X = 0)

Baud rate = sysclk / 16 x (UBAUD + 1)

Calculated UBAUD = (1000000 / Target Baud rate) – 1 = 7.68, Error rate = 0.68 ⇒ **UBAUD = 8**

UCTRL4 = 0x04, Enable baudrate Compensation

Calculated FPCR = (UBAUD - Calculated UBAUD) x 256 = (8 – 7.68) x 256 = 81.92 ⇒ **FPCR = 82**

11.9.14 Baud Rate setting (example)

Baud Rate	fOSC=1.00MHz				fOSC=1.8432MHz				fOSC=2.00MHz			
	U2X=0		U2X=1		U2X=0		U2X=1		U2X=0		U2X=1	
	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR
2400	25	0.2%	51	0.2%	47	0.0%	95	0.0%	51	0.2%	103	0.2%
4800	12	0.2%	25	0.2%	23	0.0%	47	0.0%	25	0.2%	51	0.2%
9600	6	-7.0%	12	0.2%	11	0.0%	23	0.0%	12	0.2%	25	0.2%
14.4K	3	8.5%	8	-3.5%	7	0.0%	15	0.0%	8	-3.5%	16	2.1%
19.2K	2	8.5%	6	-7.0%	5	0.0%	11	0.0%	6	-7.0%	12	0.2%
28.8K	1	8.5%	3	8.5%	3	0.0%	7	0.0%	3	8.5%	8	-3.5%
38.4K	1	-18.6%	2	8.5%	2	0.0%	5	0.0%	2	8.5%	6	-7.0%
57.6K	-	-	1	8.5%	1	-25.0%	3	0.0%	1	8.5%	3	8.5%
76.8K	-	-	1	-18.6%	1	0.0%	2	0.0%	1	-18.6%	2	8.5%
115.2K	-	-	-	-	-	-	1	0.0%	-	-	1	8.5%
230.4K	-	-	-	-	-	-	-	-	-	-	-	-

Baud Rate	fOSC=3.6864MHz				fOSC=4.00MHz				fOSC=7.3728MHz			
	U2X=0		U2X=1		U2X=0		U2X=1		U2X=0		U2X=1	
	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR
2400	95	0.0%	191	0.0%	103	0.2%	207	0.2%	191	0.0%	-	-
4800	47	0.0%	95	0.0%	51	0.2%	103	0.2%	95	0.0%	191	0.0%
9600	23	0.0%	47	0.0%	25	0.2%	51	0.2%	47	0.0%	95	0.0%
14.4K	15	0.0%	31	0.0%	16	2.1%	34	-0.8%	31	0.0%	63	0.0%
19.2K	11	0.0%	23	0.0%	12	0.2%	25	0.2%	23	0.0%	47	0.0%
28.8K	7	0.0%	15	0.0%	8	-3.5%	16	2.1%	15	0.0%	31	0.0%
38.4K	5	0.0%	11	0.0%	6	-7.0%	12	0.2%	11	0.0%	23	0.0%
57.6K	3	0.0%	7	0.0%	3	8.5%	8	-3.5%	7	0.0%	15	0.0%
76.8K	2	0.0%	5	0.0%	2	8.5%	6	-7.0%	5	0.0%	11	0.0%
115.2K	1	0.0%	3	0.0%	1	8.5%	3	8.5%	3	0.0%	7	0.0%
230.4K	-	-	1	0.0%	-	-	1	8.5%	1	0.0%	3	0.0%
250K	-	-	1	-7.8%	-	-	1	0.0%	1	-7.8%	3	-7.8%
0.5M	-	-	-	-	-	-	-	-	-	-	1	-7.8%

Baud Rate	fOSC=8.00MHz				fOSC=11.0592MHz				fOSC=14.7456MHz			
	U2X=0		U2X=1		U2X=0		U2X=1		U2X=0		U2X=1	
	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR
2400	207	0.2%	-	-	-	-	-	-	-	-	-	-
4800	103	0.2%	207	0.2%	143	0.0%	-	-	191	0.0%	-	-
9600	51	0.2%	103	0.2%	71	0.0%	143	0.0%	95	0.0%	191	0.0%
14.4K	34	-0.8%	68	0.6%	47	0.0%	95	0.0%	63	0.0%	127	0.0%
19.2K	25	0.2%	51	0.2%	35	0.0%	71	0.0%	47	0.0%	95	0.0%
28.8K	16	2.1%	34	-0.8%	23	0.0%	47	0.0%	31	0.0%	63	0.0%
38.4K	12	0.2%	25	0.2%	17	0.0%	35	0.0%	23	0.0%	47	0.0%
57.6K	8	-3.5%	16	2.1%	11	0.0%	23	0.0%	15	0.0%	31	0.0%
76.8K	6	-7.0%	12	0.2%	8	0.0%	17	0.0%	11	0.0%	23	0.0%
115.2K	3	8.5%	8	-3.5%	5	0.0%	11	0.0%	7	0.0%	15	0.0%
230.4K	1	8.5%	3	8.5%	2	0.0%	5	0.0%	3	0.0%	7	0.0%
250K	1	0.0%	3	0.0%	2	-7.8%	5	-7.8%	3	-7.8%	6	5.3%
0.5M	-	-	1	0.0%	-	-	2	-7.8%	1	-7.8%	3	-7.8%
1M	-	-	-	-	-	-	-	-	-	-	1	-7.8%

Table 11.15 Examples of UBAUD Settings for Commonly Used Oscillator Frequencies

11.9.15 0% Error Baud Rate

This USART system supports the floating point counter logic for the 0% error of baud rate. By using the 8bits floating point counter logic, the cumulative error to below the decimal point can be removed.

The floating point counter value is defined by baud rate error. In the baud rate formula, BAUD is presented the integer count value. For example, If you want to use the 57600 baud rate (fXIN = 16MHz), integer count value must be 16.36 value (BAUD+1 = 16000000/(16×57600) = 17.36). Here, the accurate BAUD value is 16.36. To realize 0% error of baud rate, floating point counter value must be 164 ((17-16.36) x 256 ≒ 164) and BAUD value must be 17. Namely you have to write the 164(decimal number) in USART_FPCR and 17(decimal number) in USART_BAUD.

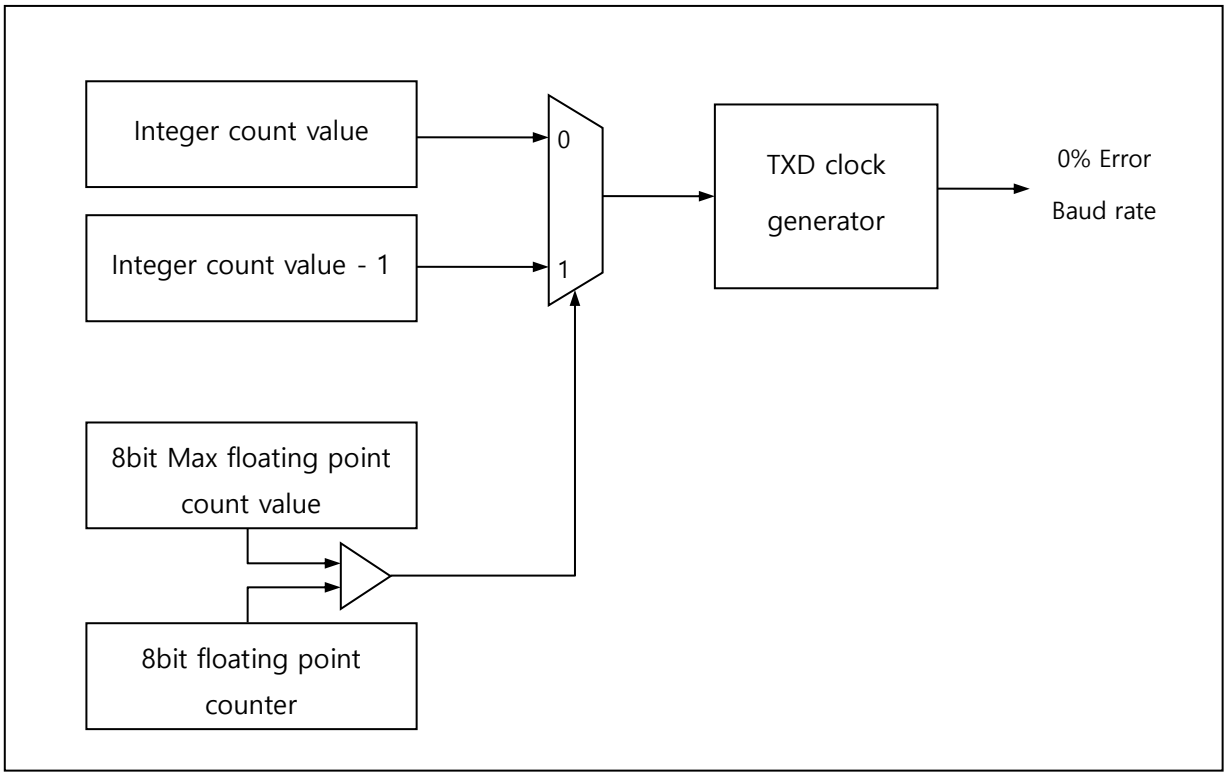


Figure 11.45 0% Error Baud Rate Block Diagram

11.10 I2C

11.10.1 Overview

The I2C is one of industrial standard serial communication protocols, and which uses 2 bus lines Serial Data Line (SDA) and Serial Clock Line (SCL) to exchange data. Because both SDA and SCL 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 speed
- 7-bit address
- Support two slave addresses
- Both master and slave operation
- Bus busy detection

11.10.2 Block Diagram

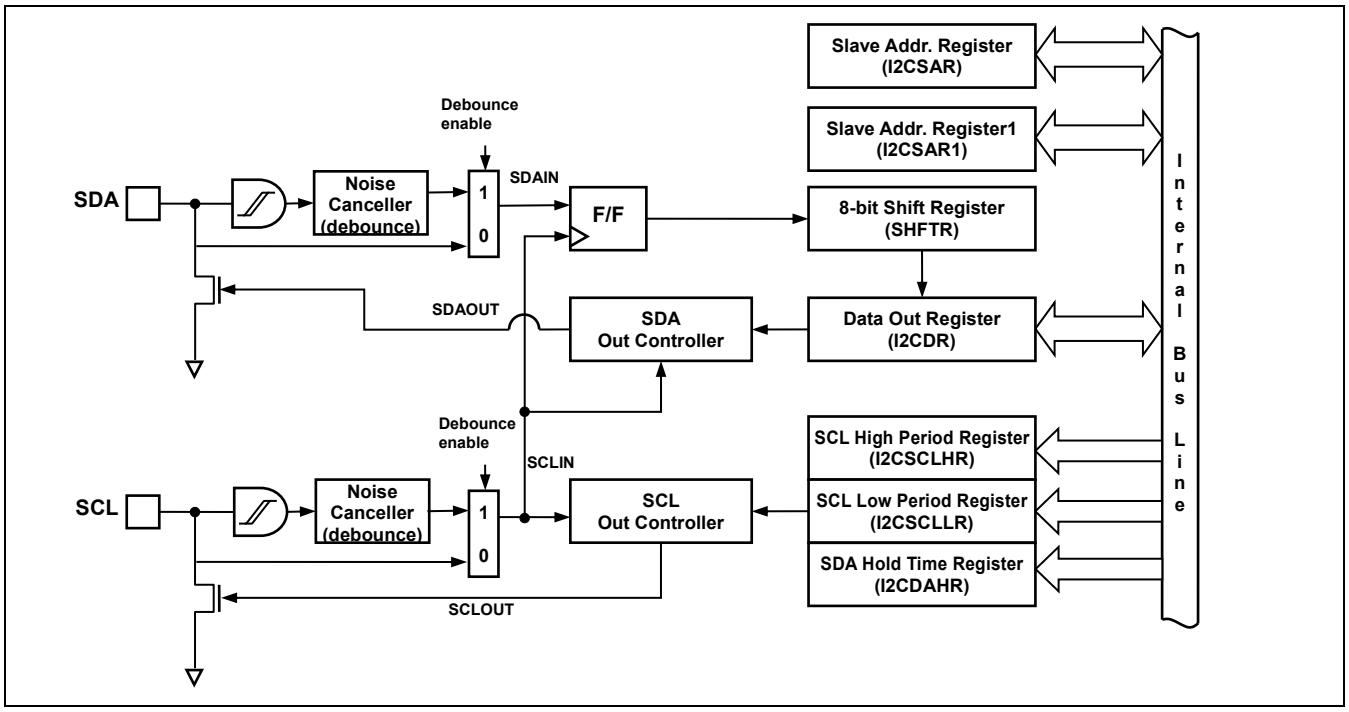


Figure 11.46 I2C Block Diagram

11.10.3 I2C bit Transfer

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

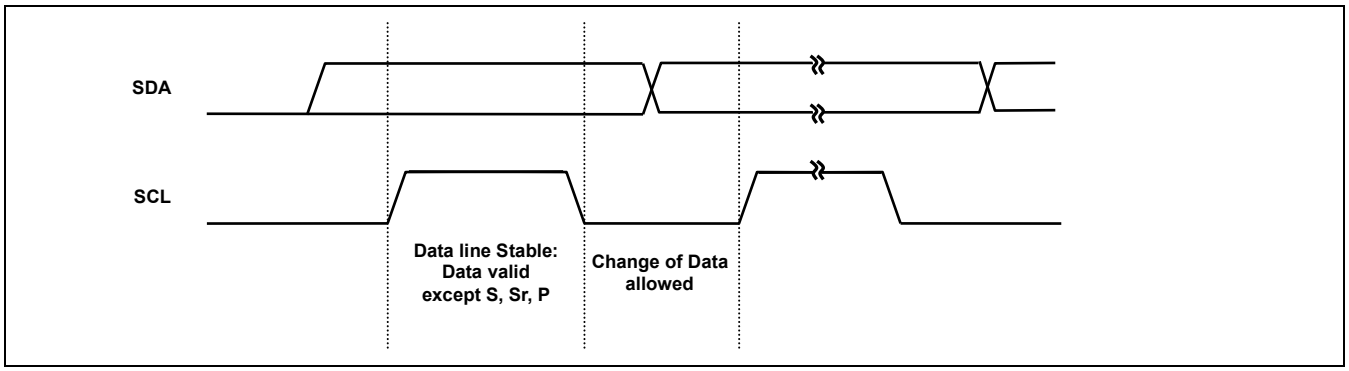


Figure 11.47 Bit Transfer on the I2C-Bus

11.10.4 Start / Repeated Start / Stop

One master can issue a START (S) condition to notice other devices connected to the SCL, SDA 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 SDA line while SCL is high defines a START (S) condition.

A low to high transition on the SDA line while SCL 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.

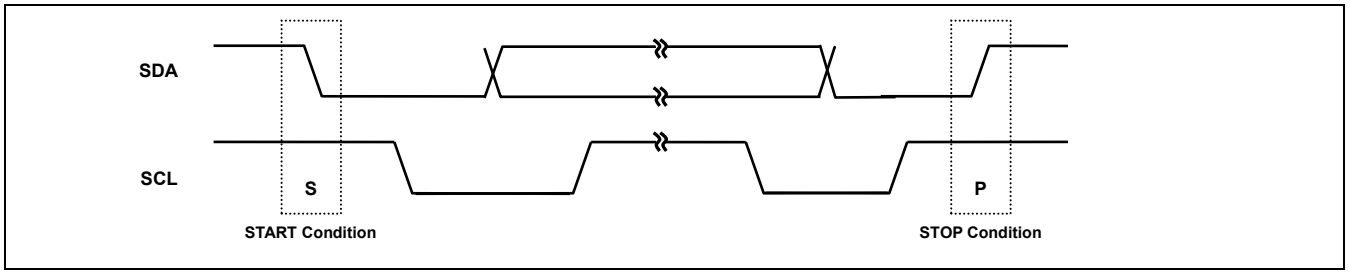


Figure 11.48 START and STOP Condition

11.10.5 Data Transfer

Every byte put on the SDA line must be 8-bit 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 SCL 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 SCL.

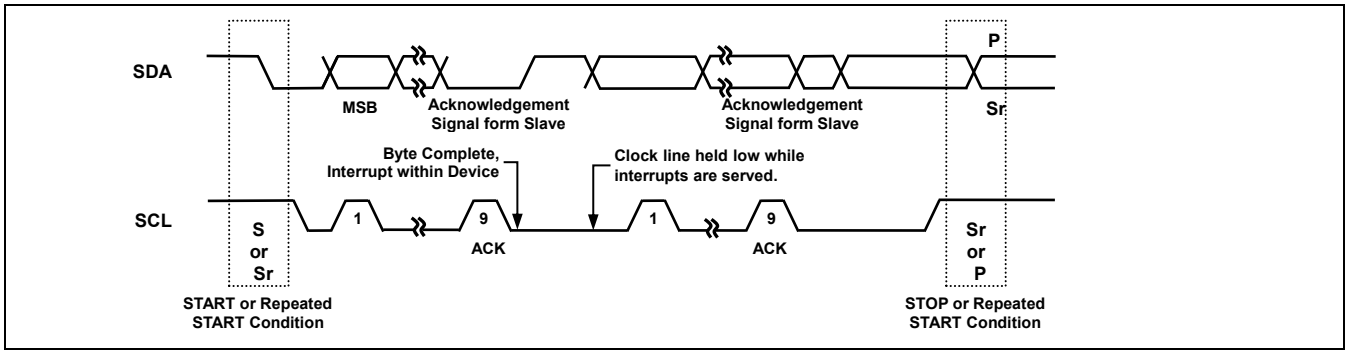


Figure 11.49 STOP or Repeated START Condition

11.10.6 Acknowledge

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

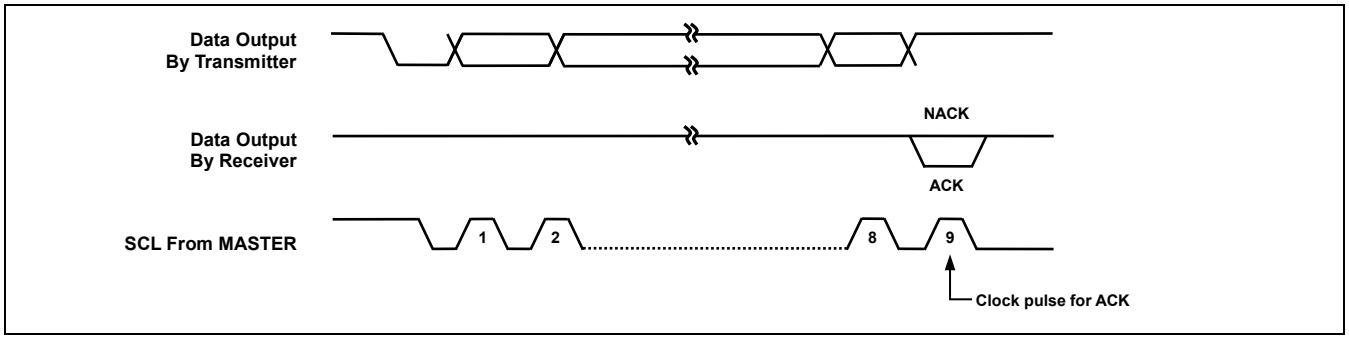


Figure 11.50 Acknowledge on the I2C-Bus

11.10.7 Synchronization / Arbitration

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

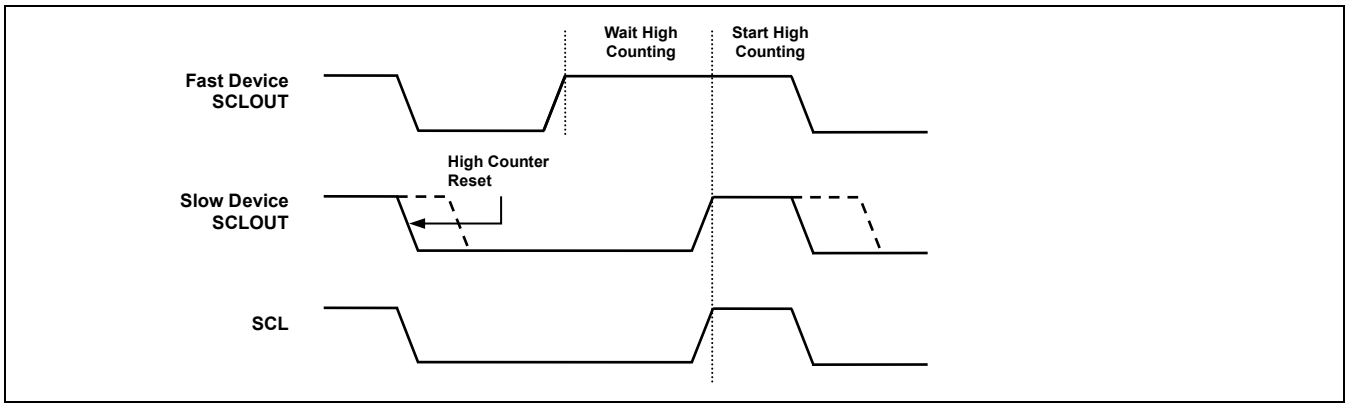


Figure 11.51 Clock Synchronization during Arbitration Procedure

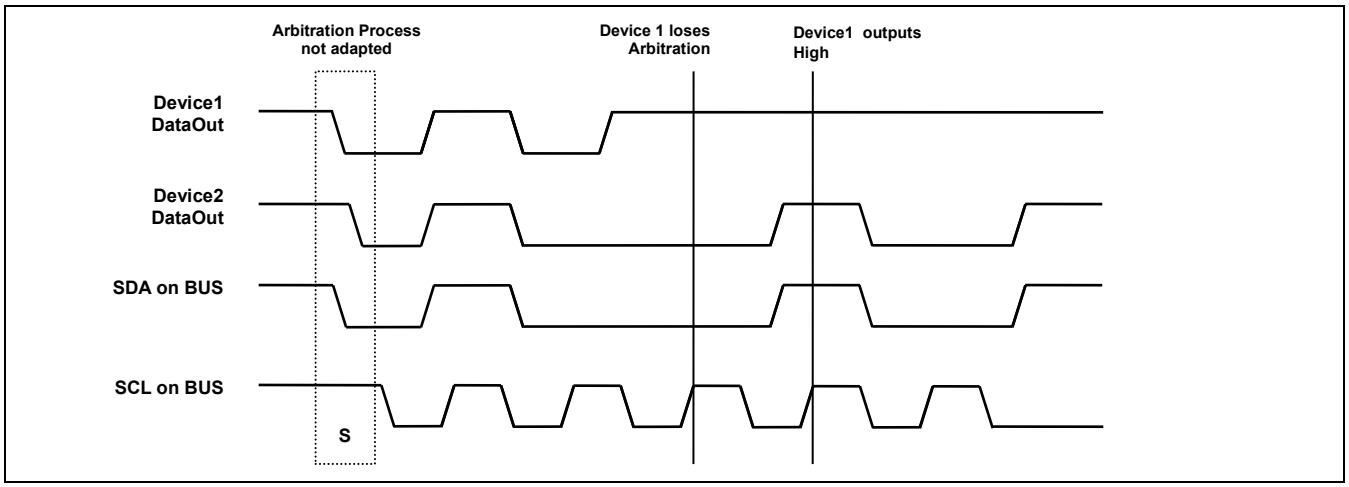


Figure 11.52 Arbitration Procedure of Two Masters

11.10.8 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 execute other operations during a I2C byte transfer.

Note that when a I2C interrupt is generated, IIF flag in I2CMR register is set, it is cleared by writing an arbitrary value to I2CSR. When I2C interrupt occurs, the SCL line is hold LOW until writing any value to I2CSR. When the IIF flag is set, the I2CSR contains a value indicating the current state of the I2C bus. According to the value in I2CSR, 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.

11.10.8.1 Master Transmitter

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

1. Enable I2C by setting IICEN bit in I2CMR. This provides main clock to the peripheral.
2. Load SLA+W into the I2CDR where SLA is address of slave device and W is transfer direction from the viewpoint of the master. For master transmitter, W is '0'. Note that I2CDR is used for both address and data.
3. Configure baud rate by writing desired value to both I2CSCLLR and I2CSCLHR for the Low and High period of SCL line.
4. Configure the I2CSDAHR to decide when SDA changes value from falling edge of SCL. If SDA should change in the middle of SCL LOW period, load half the value of I2CSCLLR to the I2CSDAHR.
5. Set the START bit in I2CMR. This transmits a START condition. And also configure how to handle interrupt and ACK signal. When the START bit is set, 8-bit data in I2CDR 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 SCL. 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 MLOST bit in I2CSR is set, and I2C waits in idle state or can be operate as an addressed slave. To operate as a slave when the MLSOT bit in I2CSR is set, the ACKEN bit in I2CMR must be set and the received 7-bit address must equal to the SLA bits in I2CSAR. In this case I2C operates as a slave transmitter or a slave receiver (go to appropriate section). In this stage, I2C holds the SCL 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 I2CDR.
   2) Master stops data transfer even if it receives ACK signal from slave. In this case, set the STOP bit in I2CMR.
   3) Master transmits repeated START condition with not checking ACK signal. In this case, load SLA+R/W into the I2CDR and set START bit in I2CMR.

After doing one of the actions above, write arbitrary value to I2CSR to release SCL 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 I2CDR 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 SCL as Low. When I2C loses bus mastership while transmitting data arbitrating other masters, the MLOST bit in I2CSR is set. If then, I2C waits in idle state. When the data in I2CDR is transmitted completely, I2C generates TEND interrupt.

   I2C can choose one of the following cases regardless of the reception of ACK signal from slave.

   1) Master continues receiving data from slave. To do this, set ACKEN bit in I2CMR to ACKnowledge and 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 ACKEN bit in I2CMR.
   3) Because no ACK signal is detected, master terminates data transfer. In this case, set the STOP bit in I2CMR.
   4) No ACK signal is detected, and master transmits repeated START condition. In this case, load SLA+R/W into the I2CDR and set the START bit in I2CMR.

   After doing one of the actions above, write arbitrary value to I2CSR to release SCL 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 I2CDR, 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 I2CSR, write arbitrary value to I2CSR. After this, I2C enters idle state.

11.10.8.2 Master Receiver

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

1. Enable I2C by setting IICEN bit in I2CMR. This provides main clock to the peripheral.
2. Load SLA+R into the I2CDR 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 I2CDR is used for both address and data.
3. Configure baud rate by writing desired value to both I2CSCLLR and I2CSCLHR for the Low and High period of SCL line.
4. Configure the I2CSDAHR to decide when SDA changes value from falling edge of SCL. If SDA should change in the middle of SCL LOW period, load half the value of I2CSCLLR to the I2CSDAHR.
5. Set the START bit in I2CMR. This transmits a START condition. And also configure how to handle interrupt and ACK signal. When the START bit is set, 8-bit data in I2CDR 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 SCL. 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 MLOST bit in I2CSR is set, and I2C waits in idle state or can be operate as an addressed slave. To operate as a slave when the MLSOT bit in I2CSR is set, the ACKEN bit in I2CMR must be set and the received 7-bit address must equal to the SLA bits in I2CSAR. In this case I2C operates as a slave transmitter or a slave receiver (go to

appropriate section). In this stage, I2C holds the SCL 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 ACKEN bit in I2CMR 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 STOP bit in I2CMR.
3) Master transmits repeated START condition due to no ACK signal from slave. In this case, load SLA+R/W into the I2CDR and set START bit in I2CMR.

After doing one of the actions above, write arbitrary value to I2CSR to release SCL 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 I2CDR 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 SCL LOW. When 1-byte of data is received completely, I2C generates TEND interrupt.

   I2C can choose one of the following cases according to the RXACK flag in I2CSR.

   4) Master continues receiving data from slave. To do this, set ACKEN bit in I2CMR to ACKnowledge and the next data to be received.
   5) Master wants to terminate data transfer when it receives next data by not generating ACK signal. This can be done by clearing ACKEN bit in I2CMR.
   6) Because no ACK signal is detected, master terminates data transfer. In this case, set the STOP bit in I2CMR.
   7) No ACK signal is detected, and master transmits repeated START condition. In this case, load SLA+R/W into the I2CDR and set the START bit in I2CMR.

   After doing one of the actions above, write arbitrary value to I2CSR to release SCL 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 I2CDR, 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 I2CSR, write arbitrary value to I2CSR. After this, I2C enters idle state.

11.10.8.3 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 SCL, load value 0x00 into I2CSDAHR to make SDA change within one system clock period from the falling edge of SCL. Note that the hold time of SDA is calculated by SDAH x period of SCLK where SDAH is multiple of number of SCLK coming from I2CSDAHR. When the hold time of SDA is longer than the period of SCLK, I2C (slave) cannot transmit serial data properly.
2. Enable I2C by setting IICEN bit and INTEN bit in I2CMR. This provides main clock to the peripheral.
3. When a START condition is detected, I2C receives one byte of data and compares it with SLA bits in I2CSAR. If the GCALLEN bit in I2CSAR is enabled, I2C compares the received data with value 0x00, the general call address.
4. If the received address does not equal to SLA bits in I2CSAR, I2C enters idle state ie, waits for another START condition. Else if the address equals to SLA bits and the ACKEN bit is enabled, I2C generates SSEL interrupt and the SCL line is held LOW. Note that even if the address equals to SLA bits, when the ACKEN bit is disabled, I2C enters idle state. When SSEL interrupt occurs, load transmit data to I2CDR and write arbitrary value to I2CSR to release SCL line.
5. 1-byte of data is being transmitted.
6. In this step, I2C generates TEND interrupt and holds the SCL line as 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 I2CDR.

   After doing one of the actions above, write arbitrary value to I2CSR to release SCL 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 STOP bit indicates that data transfer between master and slave is over. To clear I2CSR, write arbitrary value to I2CSR. After this, I2C enters idle state.

11.10.8.4 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 SCL, load value 0x00 into I2CSDAHR to make SDA change within one system clock period from the falling edge of SCL. Note that the hold time of SDA is calculated by SDAH x period of SCLK where SDAH is multiple of number of SCLK coming from I2CSDAHR. When the hold time of SDA is longer than the period of SCLK, I2C (slave) cannot transmit serial data properly.
2. Enable I2C by setting IICEN bit and INTEN bit in I2CMR. This provides main clock to the peripheral.
3. When a START condition is detected, I2C receives one byte of data and compares it with SLA bits in I2CSAR. If the GCALLEN bit in I2CSAR is enabled, I2C compares the received data with value 0x00, the general call address.
4. If the received address does not equal to SLA bits in I2CSAR, I2C enters idle state ie, waits for another START condition. Else if the address equals to SLA bits and the ACKEN bit is enabled, I2C generates SSEL interrupt and

the SCL line is held LOW. Note that even if the address equals to SLA bits, when the ACKEN bit is disabled, I2C enters idle state. When SSEL interrupt occurs and I2C is ready to receive data, write arbitrary value to I2CSR to release SCL line.

5. 1-byte of data is being received.
6. In this step, I2C generates TEND interrupt and holds the SCL 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 (ACKEN=0) and I2C waits STOP or repeated START condition.
   2) ACK signal is detected (ACKEN=1) and I2C can continue to receive data from master.

   After doing one of the actions above, write arbitrary value to I2CSR to release SCL 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 STOP bit indicates that data transfer between master and slave is over. To clear I2CSR, write arbitrary value to I2CSR. After this, I2C enters idle state.

11.10.9 Register Map

Name	Address	Dir	Default	Description
I2CSR	F8H	R	00H	I2C Status Register 0
I2CMR	F9H	R/W	00H	I2C Mode Control Register
I2CMR1	EEH	R/W	00H	I2C Mode Control Register 1
I2CSCLLR	FAH	R/W	3FH	SCL Low Period Register
I2CSCLHR	FBH	R/W	3FH	SCL High Period Register
I2CSDAHR	FCH	R/W	01H	SDA Hold Time Register
I2CDR0	FDH	R/W	FFH	I2C Data Register
I2CSAR	FEH	R/W	00H	I2C Slave Address 0 Register
I2CSAR1	FFH	R/W	00H	I2C Slave Address 1 Register

Table 11.16 I2C Register Map

11.10.10 I2C Register Description

I2C Registers are composed of I2C Mode Control Register (I2CMR), I2C Mode Control Register 1 (ICMR1), I2C Status Register (I2CSR), SCL Low Period Register (I2CSCLLR), SCL High Period Register (I2CSCLHR), SDA Hold Time Register (I2CSDAHR), I2C Data Register (I2CDR), I2C Slave Address 0 Register (I2CSAR1) and I2C Slave Address 1 Register (I2CSAR1).

11.10.11 Register Description for I2C

I2CMR (I²C Mode Control Register) : F9H

7	6	5	4	3	2	1	0
IIF	IICEN	RESET	INTEN	ACKEN	MASTER	STOP	START
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W

Initial value : 00H

IIF This is interrupt flag bit.
- 0 No interrupt is generated or interrupt is cleared
- 1 An interrupt is generated

IICEN Enable I²C Function Block (by providing clock)
- 0 I²C is inactive
- 1 I²C is active

RESET Initialize internal registers of I²C.
- 0 No operation
- 1 Initialize I²C, auto cleared

INTEN Enable interrupt generation of I²C.
- 0 Disable interrupt, operates in polling mode
- 1 Enable interrupt

ACKEN Controls ACK signal generation at ninth SCL period.
NOTE) ACK signal is output (SDA=0) for the following 3 cases.
When received address packet equals to SLA bits in I2CSAR
When received address packet equals to value 0x00 with GCALL enabled
When I²C operates as a receiver (master or slave)
- 0 No ACK signal is generated (SDA=1)
- 1 ACK signal is generated (SDA=0)

MASTER Represent operating mode of I²C
- 0 I²C is in slave mode
- 1 I²C is in master mode

STOP When I²C is master, generates STOP condition.
- 0 No operation
- 1 STOP condition is to be generated

START When I²C is master, generates START condition.
- 0 No operation
- 1 START or repeated START condition is to be generated

I2CMR1 (I2C Mode Control Register 1) : EEH

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	DIS_SDAH
-	-	-	-	-	-	-	R/W

Initial value : 00H

DIS_SDAH Disable SDA hold time
- 0 Enable SDA hold time
- 1 Disable SDA hold time

I2CSR (I²C Status Register) : F8H

7	6	5	4	3	2	1	0
GCALL	TEND	STOP	SSEL	MLOST	BUSY	TMODE	RXACK
R	R	R	R	R	R	R	R

Initial value : 00H

GCALL This bit has different meaning depending on whether I²C is master or slave. NOTE 1)
When I²C is a master, this bit represents whether it received AACK (Address ACK) from slave.
When I2C is a slave, this bit is used to indicate general call.
0 No AACK is received (Master mode)
1 AACK is received (Master mode)
0 Received address is not general call address (Slave mode)
1 General call address is detected (Slave mode)

TEND This bit is set when 1-Byte of data is transferred completely. NOTE 1)
0 1 byte of data is not completely transferred
1 1 byte of data is completely transferred

STOP This bit is set when STOP condition is detected. NOTE 1)
0 No STOP condition is detected
1 STOP condition is detected

SSEL This bit is set when I²C is addressed by other master. NOTE 1)
0 I²C is not selected as slave
1 I²C is addressed by other master and acts as a slave

MLOST This bit represents the result of bus arbitration in master mode. NOTE 1)
0 I²C maintains bus mastership
1 I²C has lost bus mastership during arbitration process

BUSY This bit reflects bus status.
0 I²C bus is idle, so any master can issue a START condition
1 I²C bus is busy

TMODE This bit is used to indicate whether I2C is transmitter or receiver.
0 I²C is a receiver
1 I²C is a transmitter

RXACK This bit shows the state of ACK signal.
0 No ACK is received
1 ACK is generated at ninth SCL period

NOTE 1) These bits can be source of interrupt.
When an I²C interrupt occurs except for STOP interrupt, the SCL line is hold LOW. To release SCL, write arbitrary value to I2CSR. When I2CSR is written, the TEND, STOP, SSEL, LOST, RXACK bits are cleared.

I2CSCLLR (SCL Low Period Register) : FAH

7	6	5	4	3	2	1	0
SCLL7	SCLL6	SCLL5	SCLL4	SCLL3	SCLL2	SCLL1	SCLL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 3FH

SCLL[7:0] This register defines the LOW period of SCL when I²C operates in master mode. The base clock is SCLK, the system clock, and the period is calculated by the formula : $t_{SCLK} \times (4 \times SCLL + 1)$ where t_{SCLK} is the period of SCLK.

I2CSCLHR (SCL High Period Register) : FBH

7	6	5	4	3	2	1	0
SCLH7	SCLH6	SCLH5	SCLH4	SCLH3	SCLH2	SCLH1	SCLH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 3FH

SCLH[7:0] This register defines the HIGH period of SCL when I²C operates in master mode. The base clock is SCLK, the system clock, and the period is calculated by the formula : $t_{SCLK} \times (4 \times SCLH + 3)$ where t_{SCLK} is the period of SCLK.

So, the operating frequency of I²C in master mode (fI2C) is calculated by the following equation.

$$fI2C = \frac{1}{tSCLK \times (4\,(SCLL + SCLH) + 4)}$$

I2CSDAHR (SDA Hold Time Register) : FCH

7	6	5	4	3	2	1	0
SDAH7	SDAH6	SDAH5	SDAH4	SDAH3	SDAH2	SDAH1	SDAH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 01H

SDAH[7:0] This register is used to control SDA output timing from the falling edge of SCL. Note that SDA is changed after $t_{SCLK} \times$ SDAH. In master mode, load half the value of SCLL to this register to make SDA change in the middle of SCL. In slave mode, configure this register regarding the frequency of SCL from master. The SDA is changed after $t_{SCLK} \times$ (SDAH + 1). So, to insure normal operation in slave mode, the value $t_{SCLK} \times$ (SDAH + 1) must be smaller than the period of SCL.

I2CDR (I²C Data Register) : FDH

7	6	5	4	3	2	1	0
ICD7	ICD6	ICD5	ICD4	ICD3	ICD2	ICD1	ICD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FFH

ICD[7:0] When I²C is configured as a transmitter, load this register with data to be transmitted. When I²C is a receiver, the received data is stored into this register.

I2CSAR0 (I²C Slave Address Register 0) : FEH

7	6	5	4	3	2	1	0
SLA7	SLA6	SLA5	SLA4	SLA3	SLA2	SLA1	GCALLEN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

SLA[7:1] These bits configure the slave address of this I²C module when I2C operates in slave mode.

GCALLEN This bit decides whether I²C allows general call address or not when I²C operates in slave mode.

0 Ignore general call address

1 Allow general call address

I2CSAR1 (I2C Slave Address Register 1) : FFH

7	6	5	4	3	2	1	0
SLA7	SLA6	SLA5	SLA4	SLA3	SLA2	SLA1	GCALLEN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

SLA[7:1] These bits configure the slave address of this I2C module when I2C operates in slave mode.

GCALLEN This bit decides whether I2C allows general call address or not when I2C operates in slave mode.

0 Ignore general call address

1 Allow general call address

11.11 CRC

11.11.1 Overview

Using the CRC, it can be monitor the memory of the specified area. This is a one-time operation, and reset is required for continuous operation. In CRC MNT mode, when the CRC read is finished, CRC_FLAG occurs. In CRC validate mode, if the CRC validate fail after the CRC reading is finished, CRC_FLAG occurs. CRC_FAIL indicates the status of validate results when the CRC read is finished. If the CRC_FLAG is generated and the interrupt is enabled, interrupt service routine is served. CRC-FLAG is not cleared by hardware. CRC-TYPE 0~3 are not supported. Validate is done by comparing the CRC_MNT register and the CRC register value. CRC are not automatically initialized, you need to calculate a new CRC after CRC_H,CRC_L Clear.

CRC TYPE	CRC mode	Input	Condition of CRC_FLAG	Condition of CRC reset
CRC_TYPE = 4	MNT	FLASH	After CRC reading	Validate fail
CRC_TYPE = 5	MNT	IXRAM	After CRC reading	Validate fail
CRC_TYPE = 6	Validate	FLASH	After CRC reading & Validate fail	Validate fail
CRC_TYPE = 7	validate	IXRAM	After CRC reading & Validate fail	Validate fail

Table 11.17 CRC mode

11.11.2 Block Diagram

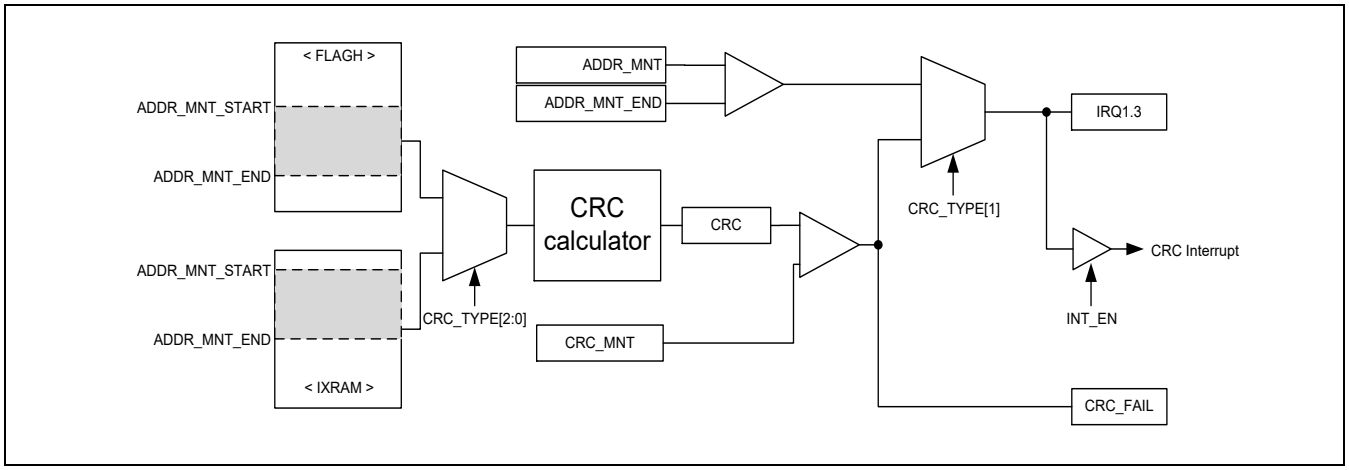


Figure 11.53 CRC block diagram

11.11.3 Register Map

Name	Address	Direction	Default	Description
CRC_CON	20E0H	R/W	00H	CRC Control Register
CRC_H	20E3H	R/W	00H	CRC High Register
CRC_L	20E4H	R/W	00H	CRC Low Register
CRC_MNT_H	20E5H	R/W	00H	CRC Monitor High Register
CRC_MNT_L	20E6H	R/W	00H	CRC Monitor Low Register
CRC_ADDR_START_M	20EBH	R/W	00H	CRC Start Address Middle Register
CRC_ADDR_START_L	20ECH	R/W	00H	CRC Start Address Low Register
CRC_ADDR_END_M	20EDH	R/W	00H	CRC End Address Middle Register
CRC_ADDR_END_L	20EEH	R/W	00H	CRC End Address Low Register

Table 11.18 CRC Register Map

11.11.4 CRC Register description

CRC_CON (CRC Control Register) : 20E0H

7	6	5	4	3	2	1	0
CRC_FLAG	CRC_INTEN	–	CRC_EN	CRC_FAIL	CRC_TYPE[2]	CRC_TYPE[1]	CRC_TYPE[0]
R/W	R/W	–	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

CRC_FLAG CRC flag. The flag is cleared only by writing a '0' to the bit. So, the flag should be cleared by software. Writing "1" has no effect.
* When go to interrupt service routine, CRC_FLAG is not cleared.
0 CRC flag not occur
1 CRC flag occur

CRC_INTEN Enable CRC interrupt
0 CRC interrupt disable
1 CRC interrupt enable

CRC_EN Enable CRC operation, it is cleared automatically after the CRC reading is finished
0 CRC disable
1 CRC enable

CRC_FAIL Status of CRC validate.
0 Validate pass
1 Validate fail

CRC_TYPE[2:0] Select the CRC input data type.
0xx Not used
100 Specified flash data
101 Specified IXRAM data
For XRAM data CRC, add 256 to the corresponding address.
(Physical Address : IRAM (0~0xFF), XRAM (0x100~0x1FF)
110 Specified flash data, validate CRC value
111 Specified IXRAM data, validate CRC value
For XRAM data CRC, add 256 to the corresponding address.
(Physical Address : IRAM (0~0xFF), XRAM (0x100~0x1FF)

CRC_H (CRC High Register) : 20E3H

7	6	5	4	3	2	1	0
CRC[15]	CRC[14]	CRC[13]	CRC[12]	CRC[11]	CRC[10]	CRC[9]	CRC[8]
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

CRC_L (CRC Low Register) : 20E4H

7	6	5	4	3	2	1	0
CRC[7]	CRC[6]	CRC[5]	CRC[4]	CRC[3]	CRC[2]	CRC[1]	CRC[0]
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

CRC[15:0] CRC result

CRC_MNT_H (CRC Monitor High Register) : 20E5H

7	6	5	4	3	2	1	0

CRC_MNT[15]	CRC_MNT[14]	CRC_MNT[13]	CRC_MNT[12]	CRC_MNT[11]	CRC_MNT[10]	CRC_MNT[9]	CRC_MNT[8]
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

CRC_MNT_L (CRC Monitor Low Register) : 20E6H

7	6	5	4	3	2	1	0
CRC_MNT[7]	CRC_MNT[6]	CRC_MNT[5]	CRC_MNT[4]	CRC_MNT[3]	CRC_MNT[2]	CRC_MNT[1]	CRC_MNT[0]
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

CRC_MNT[15:0] CRC compare register, when performing a validate

CRC_ADDR_START_M (CRC Start Address Middle Register) : 20EBH

7	6	5	4	3	2	1	0
CRC_ADDR_START[15]	CRC_ADDR_START[14]	CRC_ADDR_START[13]	CRC_ADDR_START[12]	CRC_ADDR_START[11]	CRC_ADDR_START[10]	CRC_ADDR_START[9]	CRC_ADDR_START[8]
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

CRC_ADDR_START_L (CRC Start Address Low Register) : 20ECH

7	6	5	4	3	2	1	0
CRC_ADDR_START[7]	CRC_ADDR_START[6]	CRC_ADDR_START[5]	CRC_ADDR_START[4]	CRC_ADDR_START[3]	CRC_ADDR_START[2]	CRC_ADDR_START[1]	CRC_ADDR_START[0]
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

CRC_ADDR_START[16:0] CRC start address

CRC_ADDR_END_M (CRC End Address Middle Register) : 20EDH

7	6	5	4	3	2	1	0
CRC_ADDR_END[15]	CRC_ADDR_END[14]	CRC_ADDR_END[13]	CRC_ADDR_END[12]	CRC_ADDR_END[11]	CRC_ADDR_END[10]	CRC_ADDR_END[9]	CRC_ADDR_END[8]
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

CRC_ADDR_END_L (CRC End Address Low Register) : 20EEH

7	6	5	4	3	2	1	0
CRC_ADDR_END[7]	CRC_ADDR_END[6]	CRC_ADDR_END[5]	CRC_ADDR_END[4]	CRC_ADDR_END[3]	CRC_ADDR_END[2]	CRC_ADDR_END[1]	CRC_ADDR_END[0]
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

CRC_ADDR_END[16:0] CRC end address

NOTE) CRC Polynomial
f(x) = 1+x^7+x^10+x^11+x^15+x^16
CRC16, Polynomial representations Normal : 0x8C81

12 Power Down Operation

12.1 Overview

The A94B114 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 two kinds of power saving functions, IDLE and STOP mode. In two modes, program is stopped.

12.2 Peripheral Operation in IDLE/STOP 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 Ring OSC)
Timer0~2	Operates Continuously	Halted (Only when the Event Counter Mode is Enabled, Timer operates Normally)
ADC	Operates Continuously	Stop
Comparator	Operates Continuously	Stop (Can be operated with Ring OSC)
USART	Operates Continuously	Stop
I2C	Operates Continuously	Stop
Internal RC OSC (16MHz)	Oscillation	Stop when the system clock (fx) is fIRC
Internal Ring OSC (256kHz)	Can be operated with setting value	Can be operated with setting value
Main OSC (0.4~12MHz)	Oscillation	Stop when fx = fXIN
I/O Port	Retain	Retain
Control Register	Retain	Retain
Address Data Bus	Retain	Retain
Release Method	By RESET, all Interrupts	By RESET, Timer Interrupt(EC), External Interrupt, I2C Interrupt, Comparator Interrupt, USART by RX, WDT, LVI

Table 12.1 Peripheral Operation during Power Down Mode

12.3 IDLE Mode

The power control register is set to '01h' to enter the IDLE Mode. In this mode, the internal oscillation circuits remain active. Oscillation continues and peripherals are operated normally but CPU stops. It is released by reset or interrupt. To be released by interrupt, interrupt should be enabled before IDLE mode. If using reset, because the device becomes initialized state, the registers have reset value.

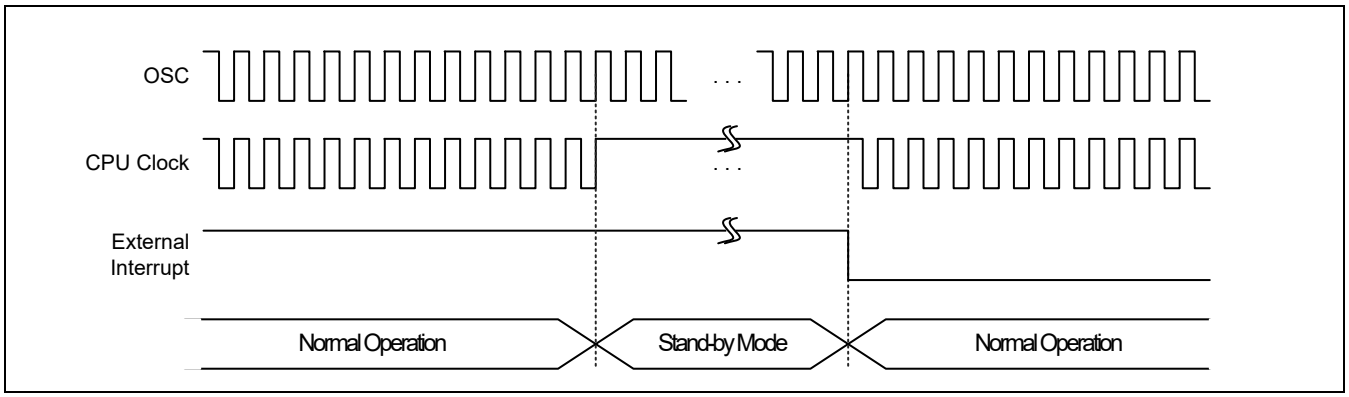


Figure 12.1 IDLE Mode Release Timing by External Interrupt

12.4 STOP Mode

The power control register is set to '03H' to enter the STOP Mode. In the stop mode, the selected oscillator, system clock and peripheral clock is stopped, but watch timer can be continued to operate with sub clock. With the clock frozen, all functions are stopped, but the on-chip RAM and control registers are held. For example, If the internal RC oscillator (f_{IRC}) is selected for the system clock and the sub clock (f_{SUB}) is oscillated, the internal RC oscillator stops oscillation and the sub clock is continuously oscillated in stop mode. At that time, the watch timer can be operated with the sub clock.

The source for exit from STOP mode is hardware reset and interrupts. The reset re-defines all the control registers.

When exit from STOP mode, enough oscillation stabilization time is required to normal operation. Figure 12.2 shows the timing diagram. When released from STOP mode, the Basic interval timer is activated on wake-up. Therefore, before STOP instruction, user must be set its relevant prescale divide ratio to have long enough time. This guarantees that oscillator has started and stabilized.

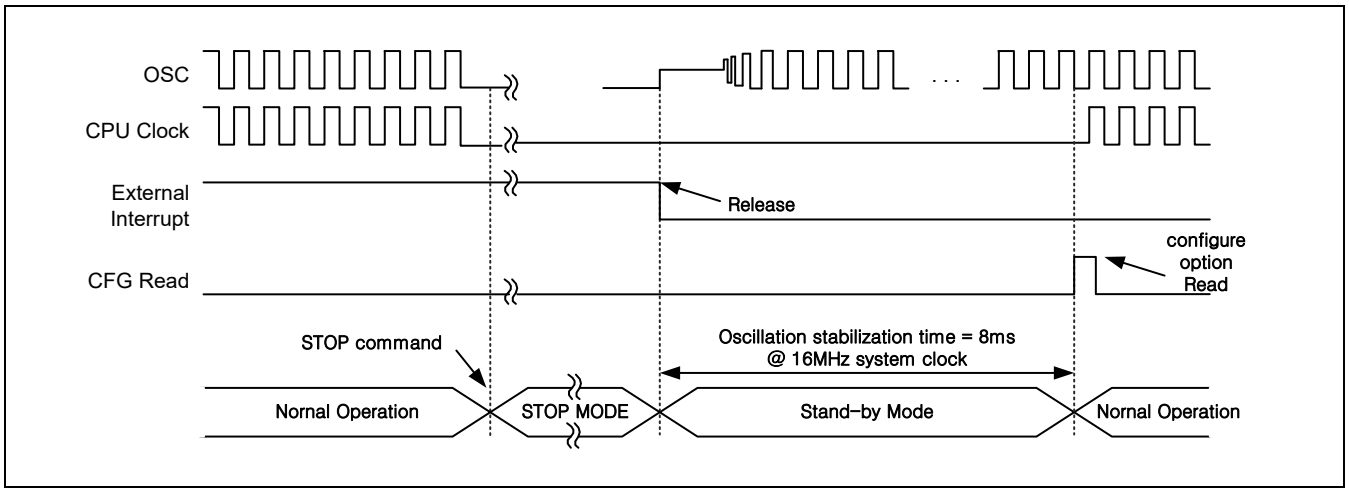


Figure 12.2 STOP Mode Release Timing by External Interrupt

12.5 Release Operation of STOP Mode

After STOP mode is released, the operation begins according to content of related interrupt register just before STOP mode start (Figure 12.3). If the global interrupt Enable Flag (IE.EA) is set to `1`, the STOP mode is released by the interrupt which each interrupt enable flag = `1` and the CPU jumps to the relevant interrupt service routine. Even if the IE.EA bit is cleared to '0', the STOP mode is released by the interrupt of which the interrupt enable flag is set to '1'.

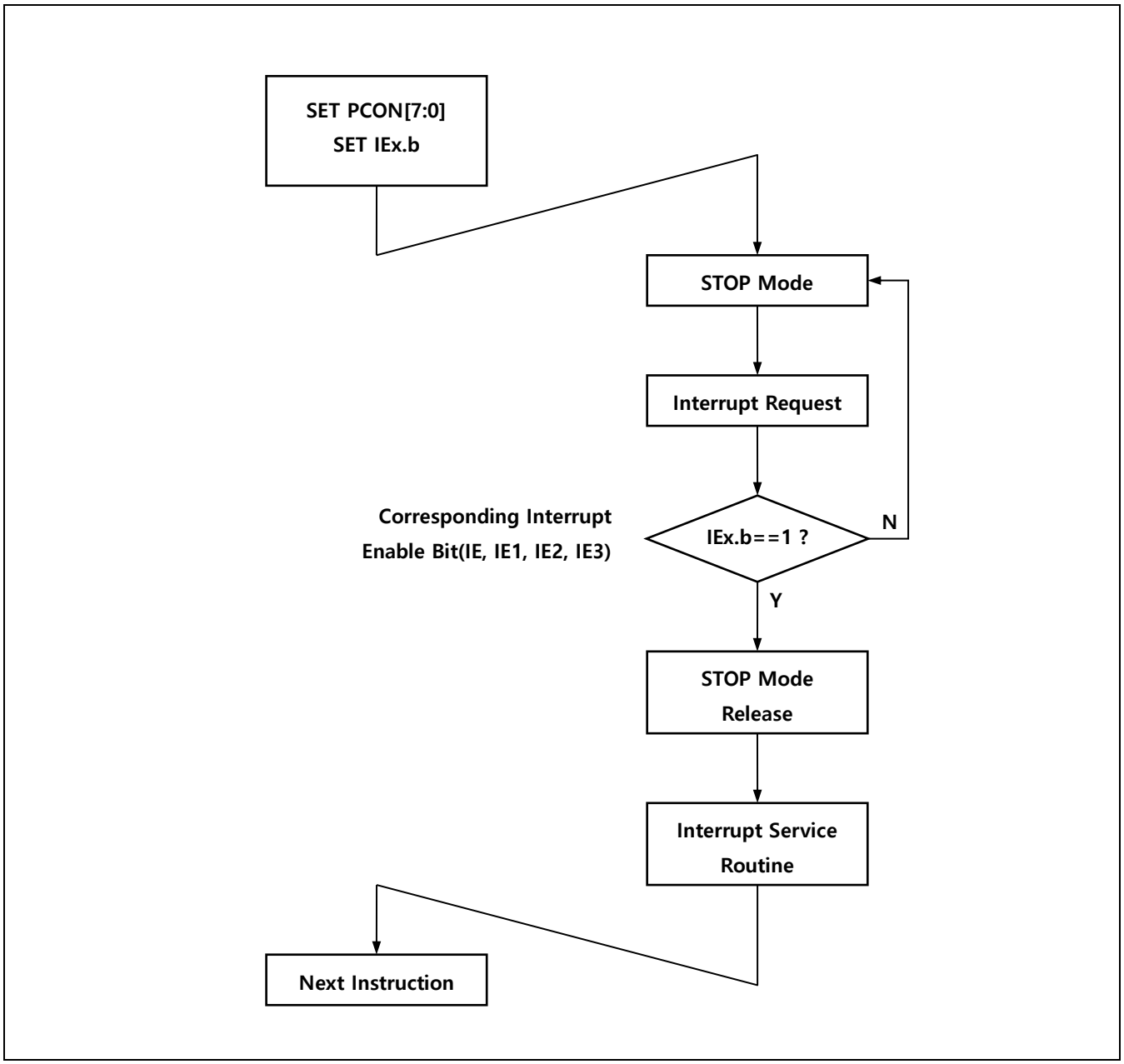


Figure 12.3 STOP Mode Release Flow

12.6 Register Map

Name	Address	Direction	Default	Description
PCON	87H	R/W	00H	Power Control Register

Table 12.2 Power Down Operation Register Map

12.7 Power Down Operation Register Description

The power down operation register consists of the power control register (PCON).

12.8 Register Description for Power Down Operation

PCON (Power Control Register): 87H

7	6	5	4	3	2	1	0
–	–	–	–	PCON3	PCON2	PCON1	PCON0
–	–	–	–	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

NOTE)

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.

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

Table 13.1 Reset State

13.2 Reset Source

The A94B114 has four 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 `)

13.3 Block Diagram

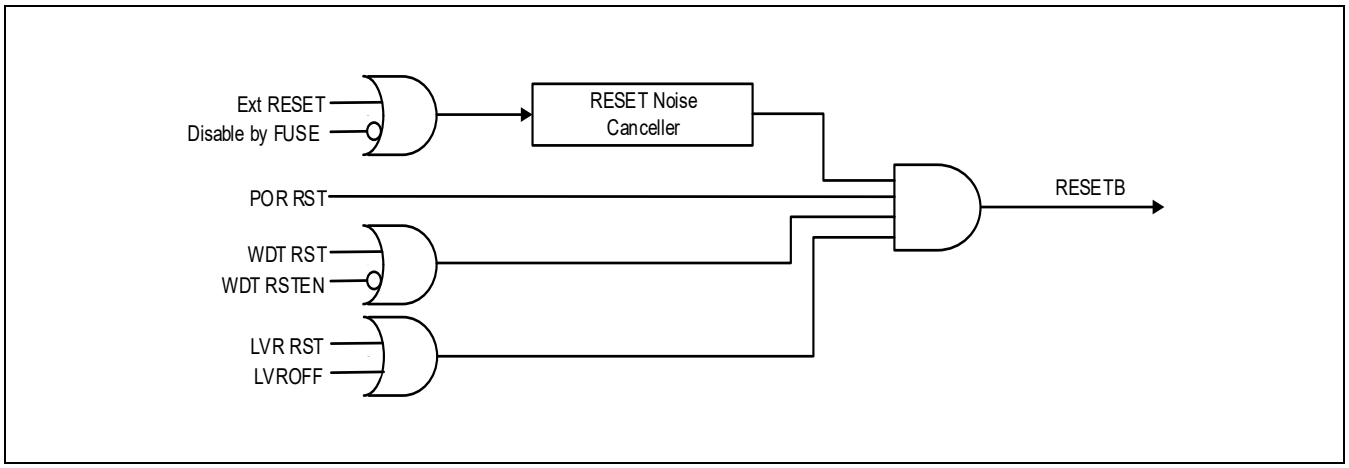


Figure 13.1 RESET Block Diagram

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.

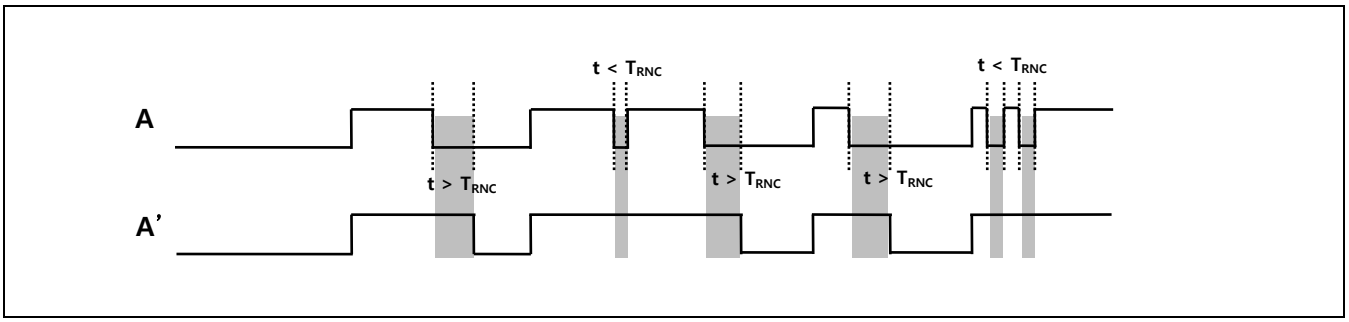


Figure 13.2 Reset noise canceller timer diagram

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. And the external RESET PIN is able to use as normal I/O pin.

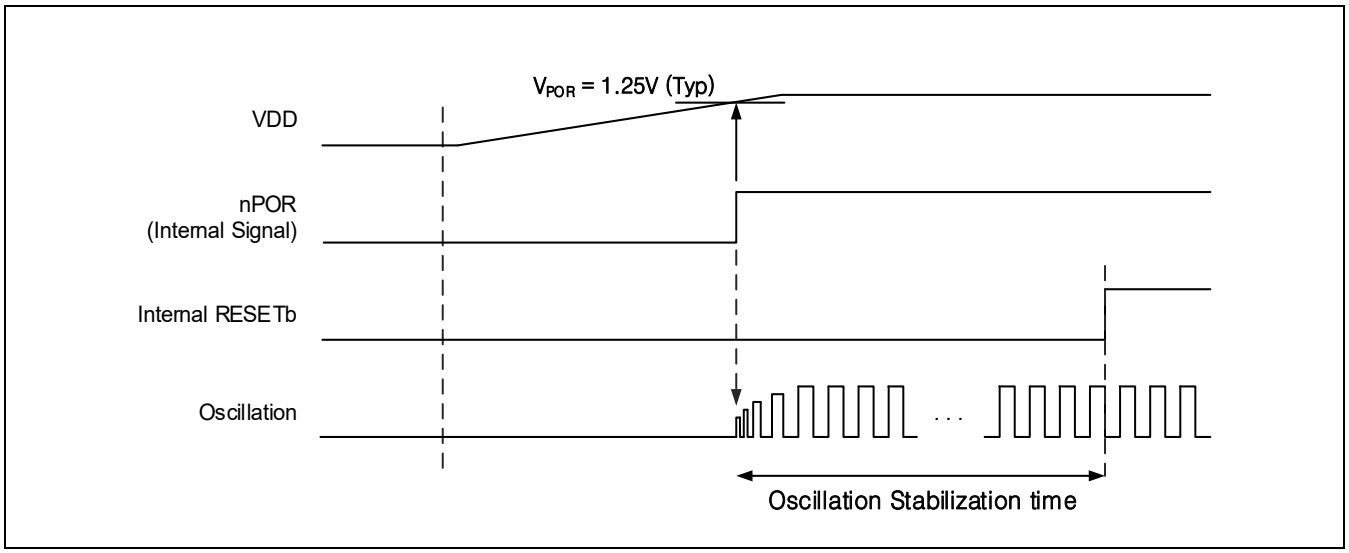


Figure 13.3 Internal RESET Release Timing On Power-Up

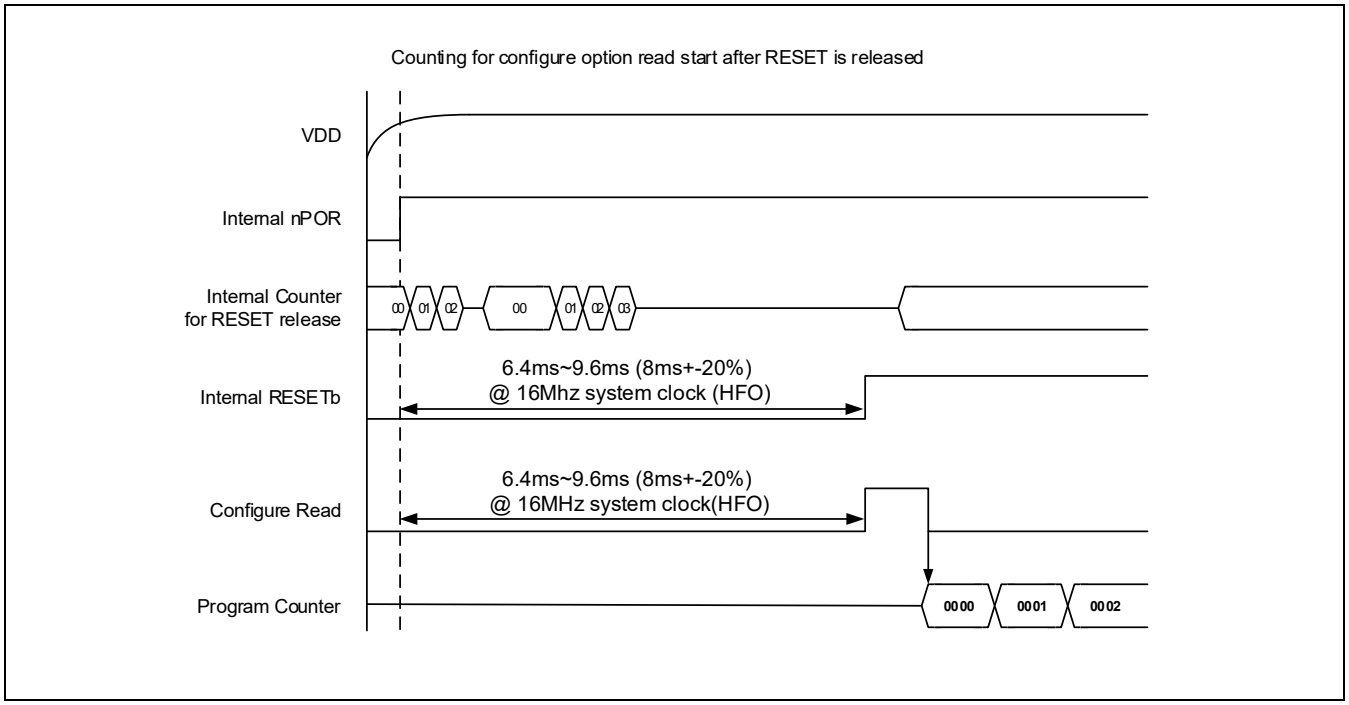


Figure 13.4 Configuration Timing when Power-on

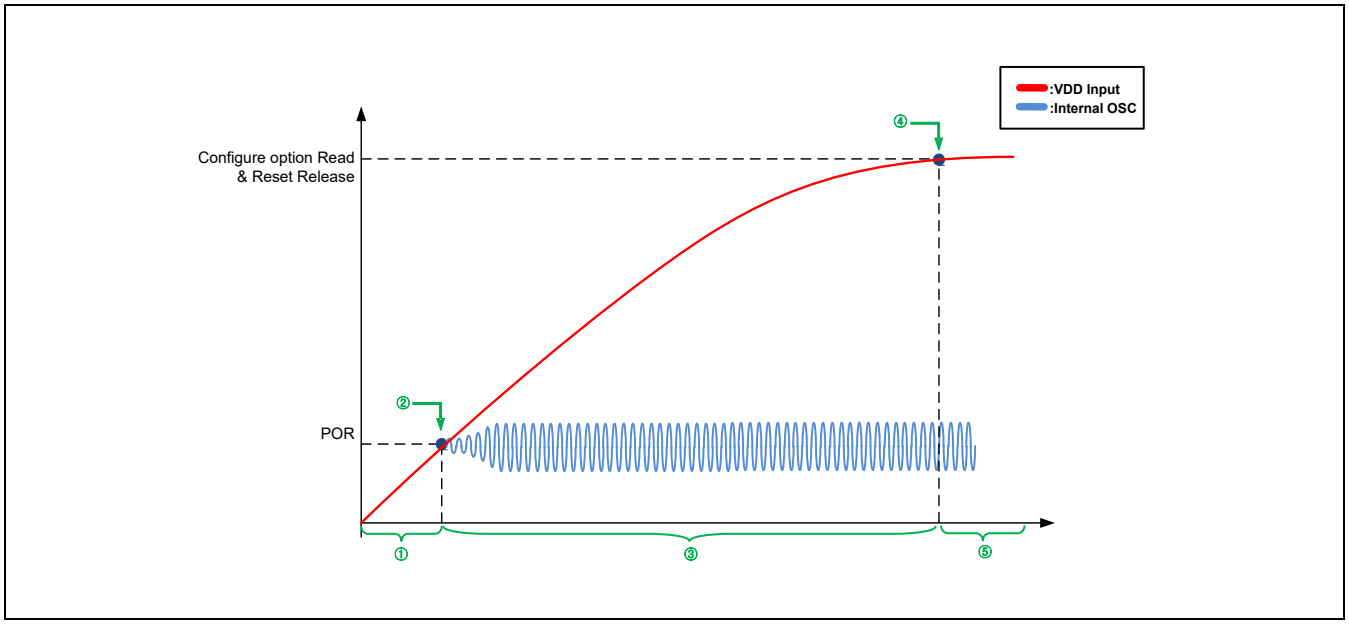


Figure 13.5 Boot Process WaveForm

Process	Description	Remarks
①	-No Operation	
②	- 1st POR level Detection - Internal OSC (32MHz) ON	-about 1.07V ~ 1.55V
③	- Delay section (=8ms) - VDD input voltage must rise over than flash operating voltage for Configure option read	-Slew Rate >= 0.05V/ms
④	- Configure option read point - Reset Release section	-Configure Value is determined by Writing Option
⑤	-Normal operation	

Table 13.2 Boot Process Description

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. Its debounce clock is only Internal High Frequency Oscillator (HFO) IRC. So do not turn off the HFO IRC to use it. 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.

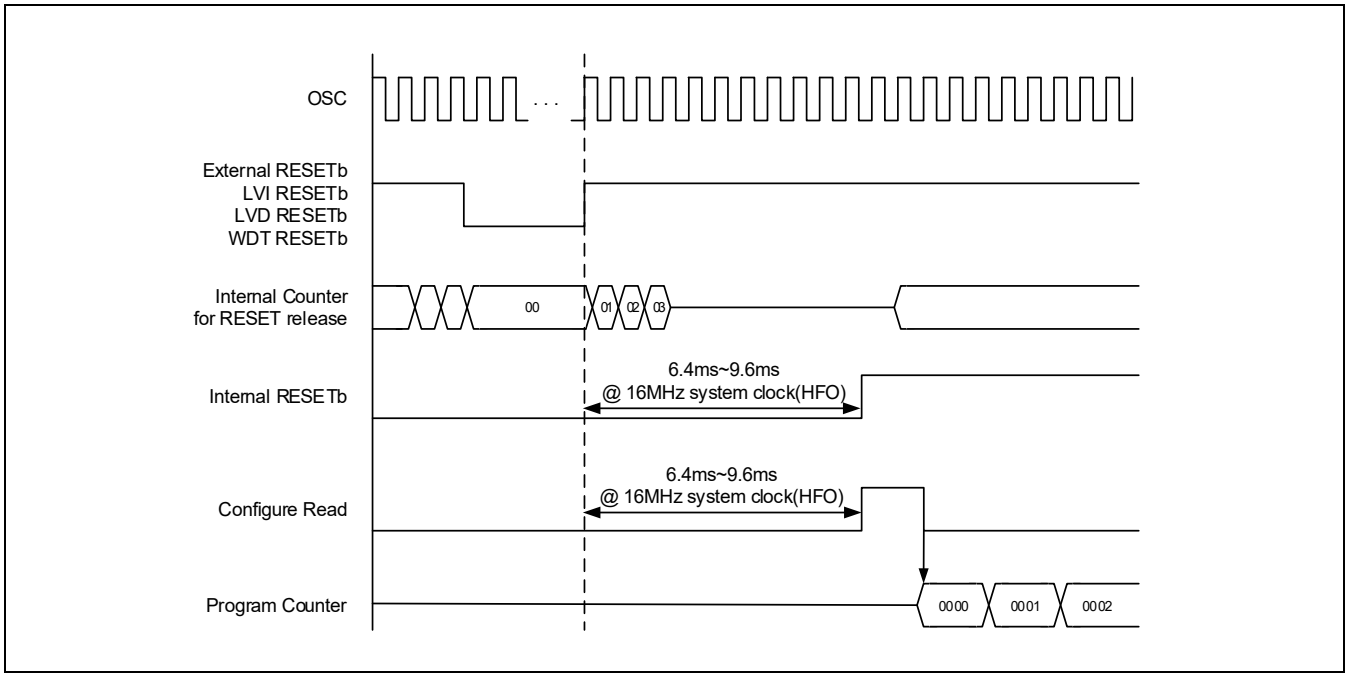


Figure 13.6 Timing Diagram after RESET

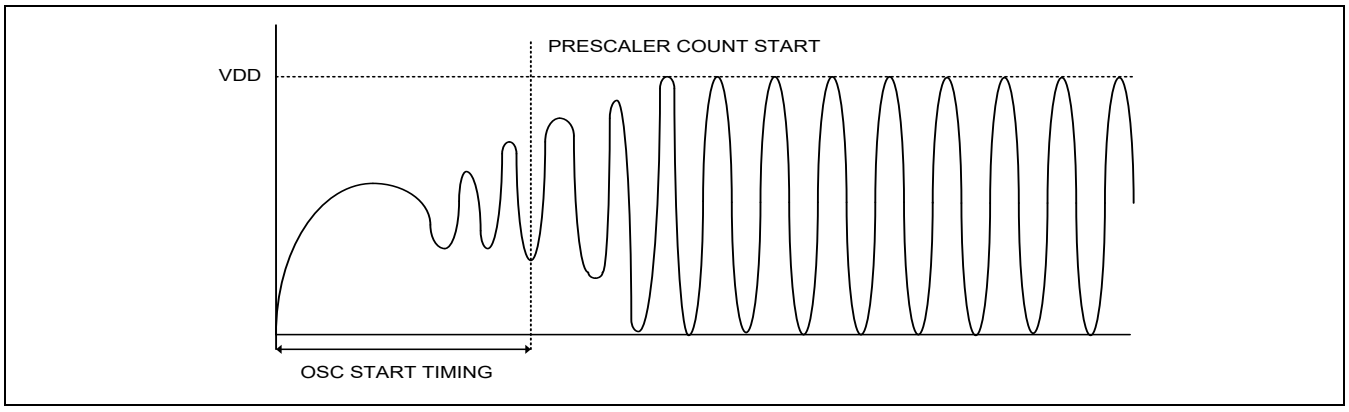


Figure 13.7 Oscillator generating waveform example

NOTE)

1. As shown Figure 13.7, the stable generating time is not included in the start-up time.
2. The RESETB pin has a Pull-up register by hardware

13.7 Low Voltage Reset and Low Voltage Indicator Processor

The A94B114 has an On-chip Low Voltage Rest (LVR) and On-chip Low Voltage Indicator circuit (LVI) for monitoring the VDD level during operation by comparing it to a fixed trigger level. The LVR terriger level is 1.75V and it is enabled by default. The LVR can be disable by register setting. The trigger level for the LVI can be selected by LVIS[2:0] bit to be 2.4V, 2.9V, 3.9V, 4.2V. In the STOP mode, this will contribute significantly to the total current consumption. So to minimize the current consumption, LVR is set to off by software. LVI is not operation in STOP mode.

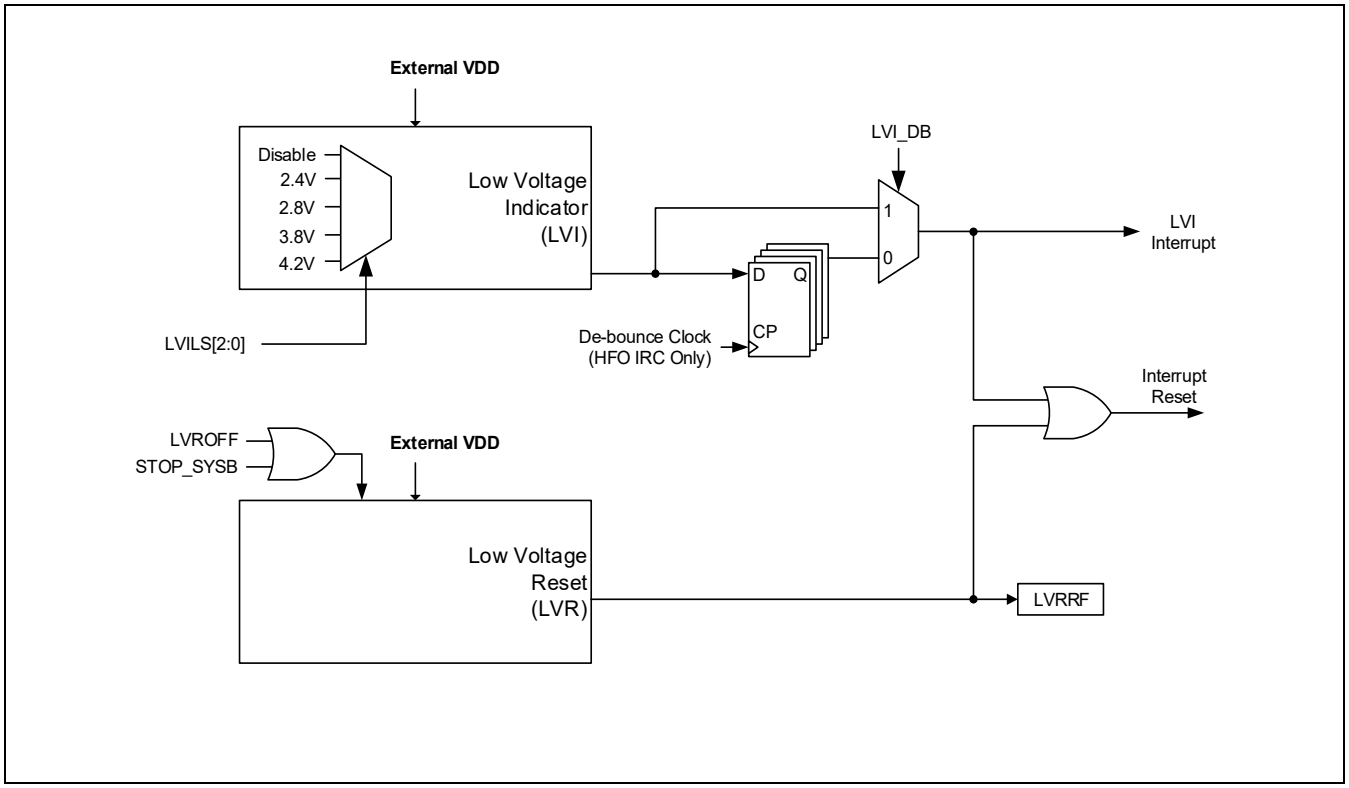


Figure 13.8 Block Diagram of LVR and LVI

13.8 Register Map

Name	Address	Direction	Default	Description
RSFR	98H	R/W	84H	Reset Flag Register
LVIR	89H	R/W	00H	Low Voltage Indicator Control Register
DBTSR	84H	R/W	00H	De-bounce Time Selection Register

Table 13.3 Reset Operation Register Map

13.9 Reset Operation Register Description

The reset control register consists of the reset flag register (RSFR), low voltage indicator control register (LVIR) and de-bounce time selection register (DBTSR).

13.10 Register Description for Reset Operation

RSFR (Reset Flag Register) : 98H

7	6	5	4	3	2	1	0
PORF	EXTRF	WDTRF	OCDRF	-	LVRF	–	–
R/W	R/W	R/W	R/W	R/W	R/W	–	–

Initial value : 84H

PORF Power-On Reset flag bit. The bit 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 reset by writing '0' to this bit or by Power-On Reset.
- 0 No detection
- 1 Detection

OCDRF On-chip debugger reset flag bit. The bit 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 reset by writing '0' to this bit.
- 0 No detection
- 1 Detection

NOTE)

1. When the Power-On Reset occurs, the PORF bit is only set to "1", EXTRF, WDTRF, OCDRF bits are all cleared to "0".
2. 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.

LVIR (Low Voltage Indicator Control Register) : 89H

7	6	5	4	3	2	1	0
LVROFF	LVI_INT_ON	LVI_DB	–	–	LVILS2	LVILS1	LVILS0
R/W	R/W	R/W	–	–	R/W	R/W	R/W

Initial value : 00H

LVROFF LVR ON/OFF selection
- 0 LVR ON
- 1 LVR OFF

LVI_INT_ON LVI Interrupt selection
- 0 Interrupt
- 1 Do not select.

LVI_DB LVI Reset de-bounce control
- 0 Disable
- 1 Enable (4us De-bounce, HFO IRC only)

LVILS[2:0] LVI level Voltage

LVILS2	LVILS1	LVILS0	Description
0	0	0	LVI disable
0	0	1	2.4V
0	1	0	2.9V
0	1	1	3.9V
1	0	0	4.2V
1	0	1	Not used
1	1	0	Not used
1	1	1	Not used

DBTSR (De-bounce Time Selection Register) : 84H

7	6	5	4	3	2	1	0
–	–	–	–	RDB1	RDB0	PDB1	PDB0
–	–	–	–	R/W	R/W	R/W	R/W

Initial value : 00_H

RDB[1:0] External Reset De-bounce selection register

RDB1	RDB0	Description
0	0	10us
0	1	20us
1	0	40us
1	1	80us

PDB[1:0] Port De-bounce selection register

PDB1	PDB0	Description
0	0	1.2us
0	1	2.5us
1	0	5us
1	1	10us

NOTE) 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. Its de-bounce clock is only HFO IRC. So do not turn off the HFO IRC to use it.

14 On-chip Debug System

14.1 Overview

14.1.1 Description

A94B114 can use On-chip debug(OCD). On-chip debug system (OCD) of A94B114 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
- Operating frequency
  - Supports the maximum frequency of the target MCU

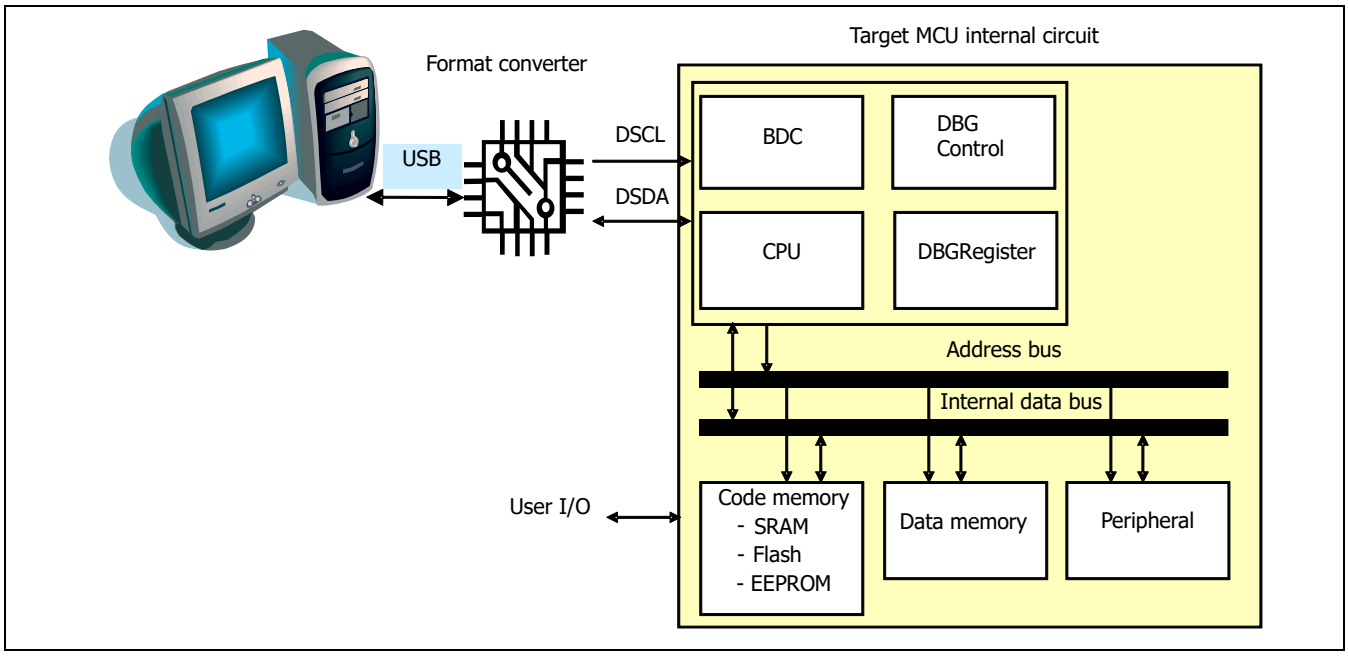


Figure 14.1 Block Diagram of On-Chip Debug System

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.

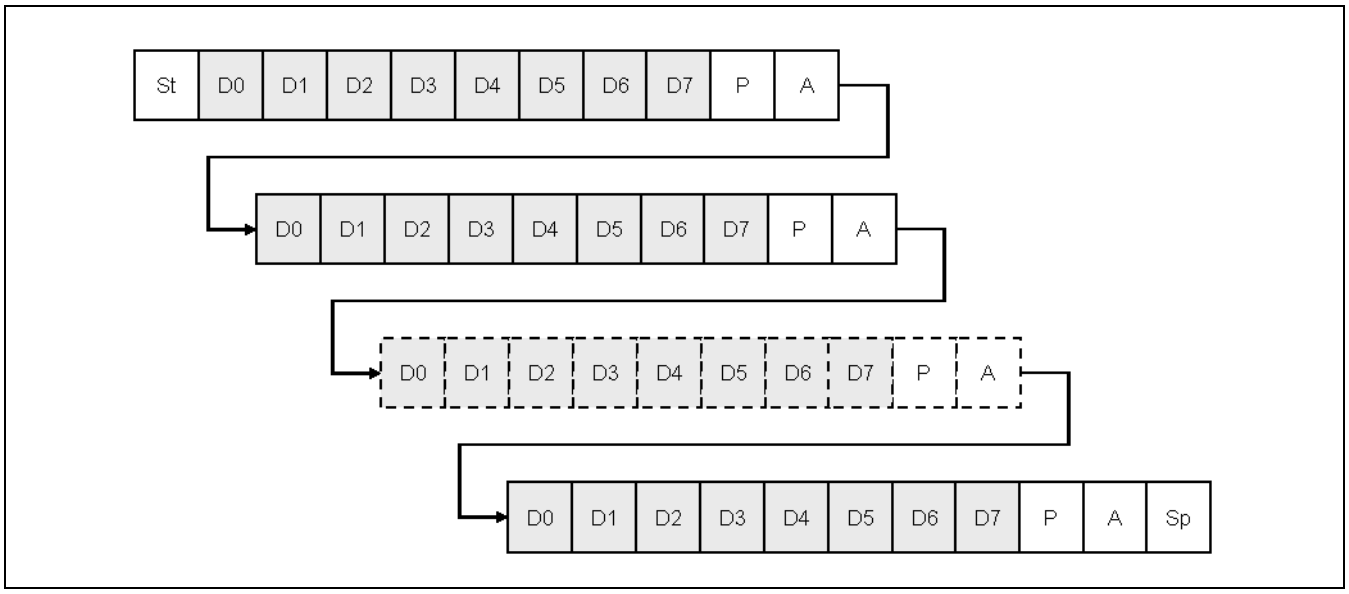


Figure 14.2 10-bit Transmission Packet

14.2.2 Packet Transmission Timing

14.2.2.1 Data Transfer

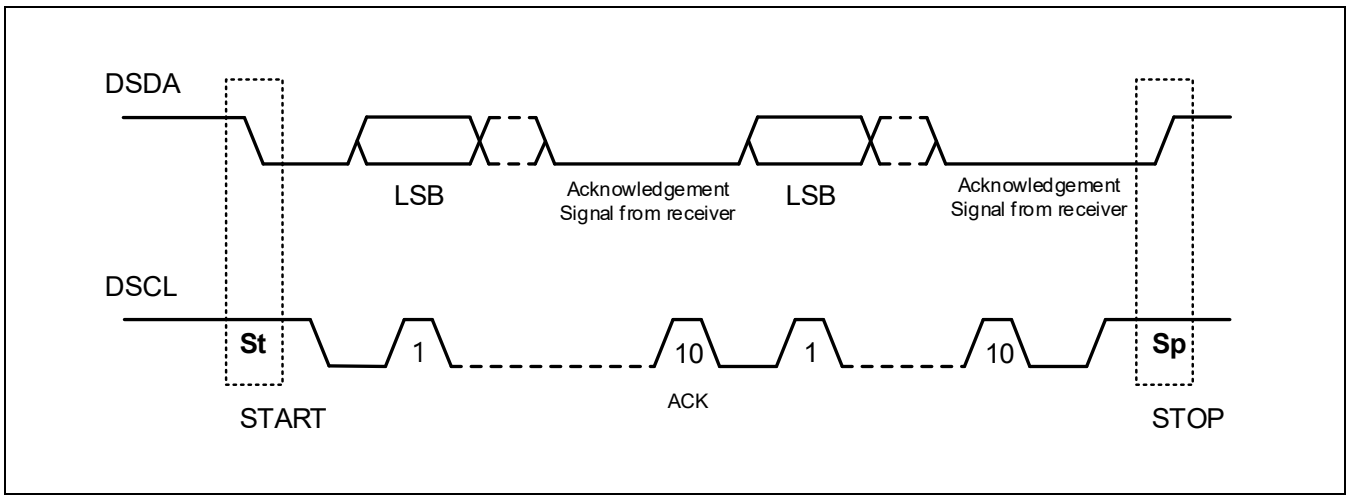


Figure 14.3 Data Transfer on the Twin Bus

14.2.2.2 Bit Transfer

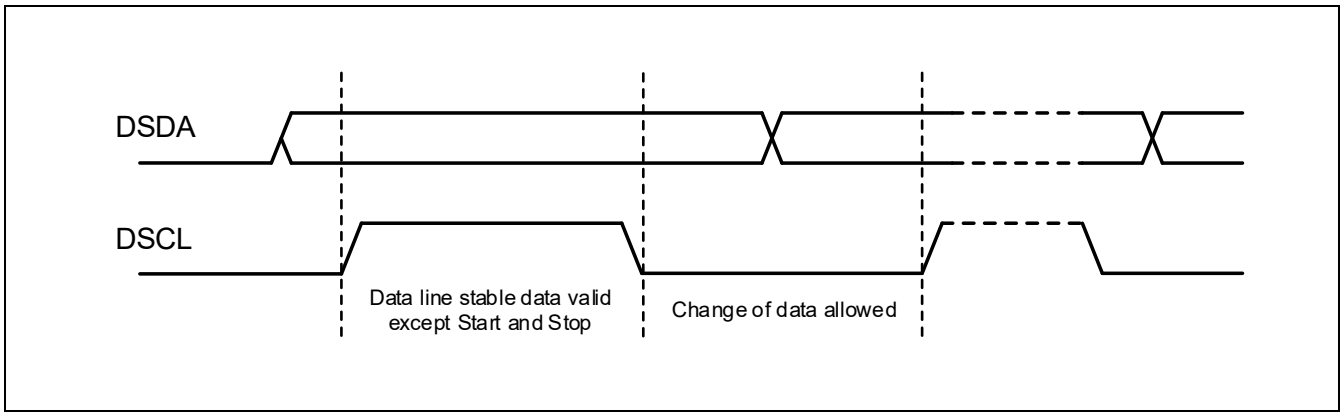


Figure 14.4 Bit Transfer on the Serial Bus

14.2.2.3 Start and Stop Condition

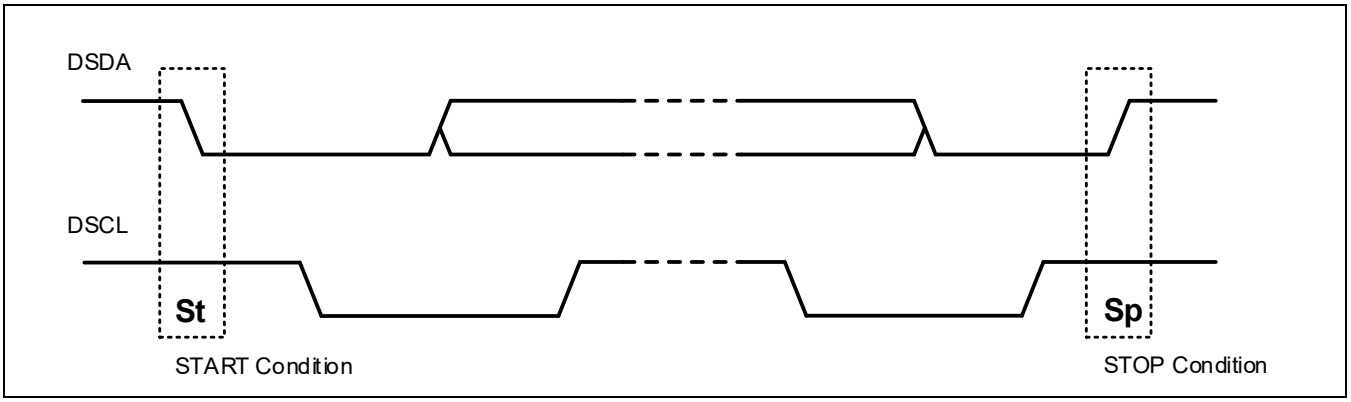


Figure 14.5 Start and Stop Condition

14.2.2.4 Acknowledge Bit

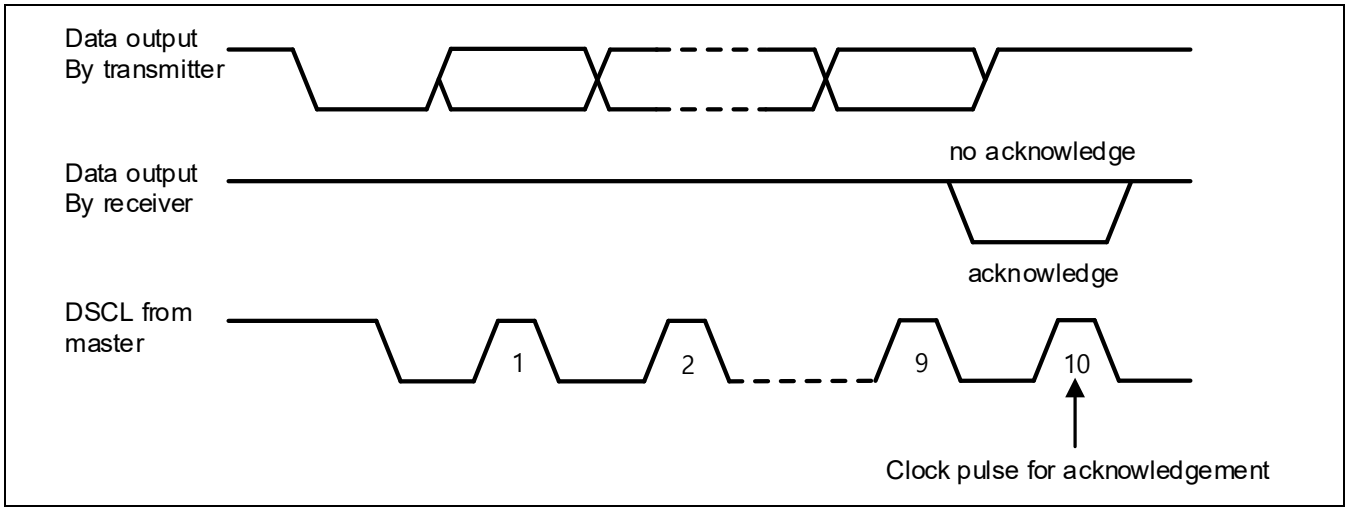


Figure 14.6 Acknowledge on the Serial Bus

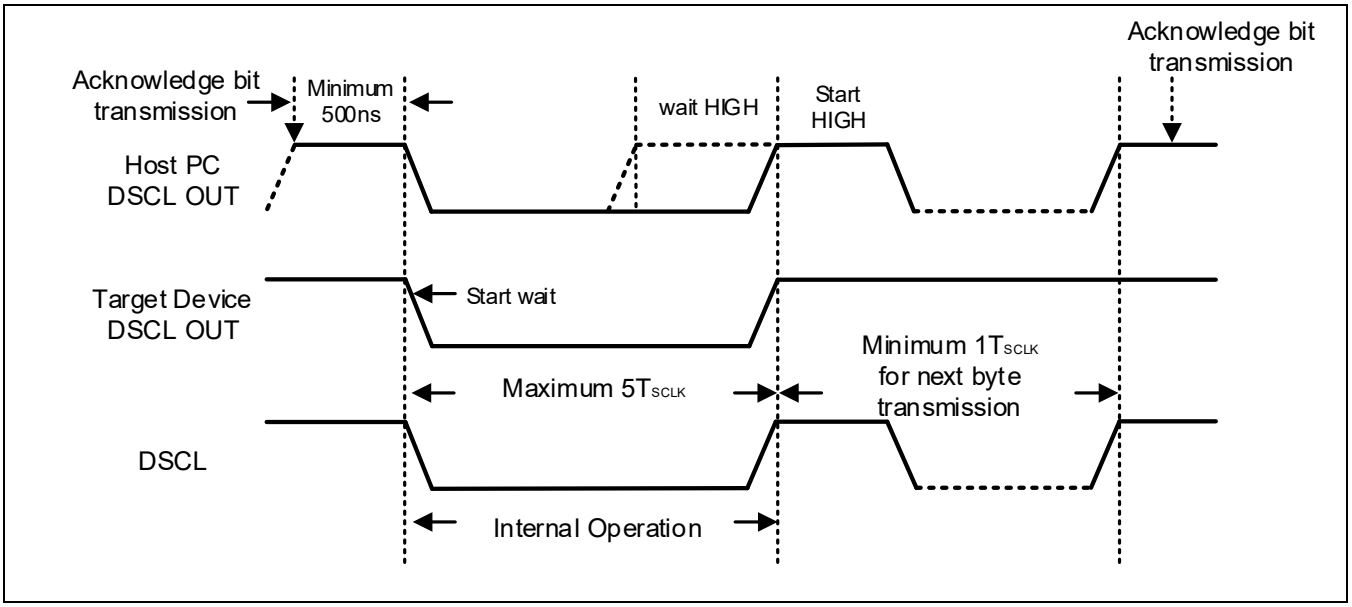


Figure 14.7 Clock Synchronization during Wait Procedure

14.2.3 Connection of Transmission

Two-pin interface connection uses open-drain(wire-AND bidirectional I/O).

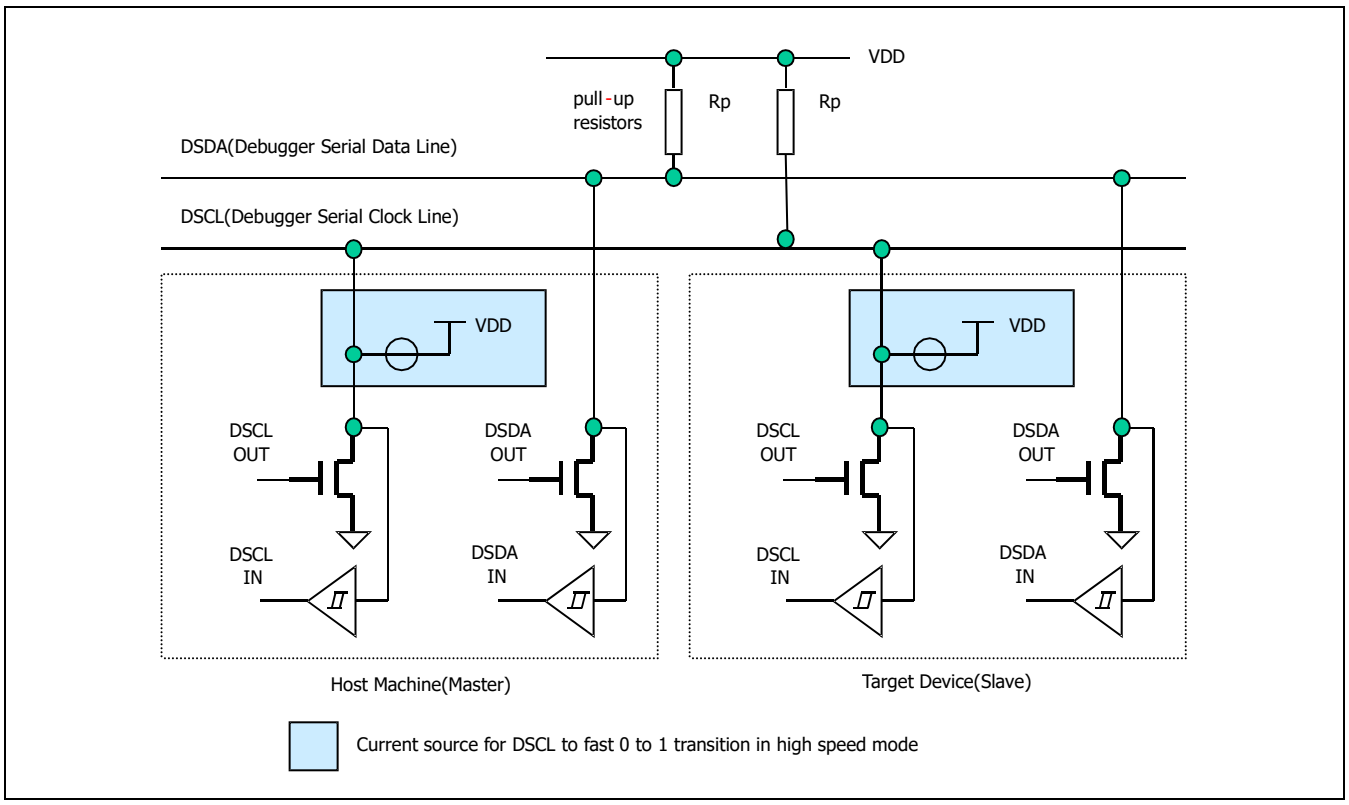


Figure 14.8 Connection of Transmission

15 Memory Programming

15.1 Overview

15.1.1 Description

A94B114 has flash memory to which a program can be written, erased, and overwritten while mounted on the board.
This device support the self program/erase mode in user soft mode in sram-jump.
Serial ISP mode is supported.

15.1.2 Features

- Flash Size : 8 Kbytes
- Single power supply program and erase
- Command interface for fast program and erase operation
- Up to 10,000 program/erase cycles at typical voltage and temperature for flash memory
- Security feature

15.2 Flash Control and status register

Registers to control Flash are Mode Register (FEMR), Control Register (FECR), Status Register (FESR), Time Control Register (FETCR), Address Low Register (FEARL), Address Middle Register (FEARM) and address High Register (FEARH). They are mapped to SFR area and can be accessed only in programming mode.

15.2.1 Register Map

Name	Address	Dir	Default	Description
FEMR	F1H	R/W	00H	Flash Mode Register
FECR	F2H	R/W	03H	Flash Control Register
FESR	F3H	R/W	80H	Flash Status Register
FETCR	F4H	R/W	00H	Flash Time Control Register
FEARL	F5H	R/W	00H	Flash Address Low Register
FEARM	F6H	R/W	00H	Flash Address Middle Register
FEARH	F7H	R/W	00H	Flash Address High Register
ENTRY_0	10D8H (XRAM)	R/W	00H	0xAA
ENTRY_1	10DAH (XRAM)	R/W	00H	0x55
ENTRY_2	10DDH (XRAM)	R/W	00H	0xA5
PAGE_BUF	10E0H ~ 10FFH	R/W	00H	Flash Data Buffer

Table 15.1 Register MAP

15.2.2 Register Description for Flash

FEMR (Flash Mode Register) : F1H

7	6	5	4	3	2	1	0
FSEL	-	PGM	ERASE	PBUFF	OTPE	VFY	FEEN
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

FSEL Select flash memory.
- 0 Deselect flash memory
- 1 Select flash memory

PGM Enable program or program verify mode with VFY
- 0 Disable program or program verify mode
- 1 Enable program or program verify mode

ERASE Enable erase or erase verify mode with VFY
- 0 Disable erase or erase verify mode
- 1 Enable erase or erase verify mode

PBUFF Select page buffer
- 0 Deselect page buffer
- 1 Select page buffer

OTPE Select OTP area instead of program memory
- 0 Deselect OTP area
- 1 Select OTP area

VFY Set program or erase verify mode with PGM or ERASE

Program Verify: PGM=1, VFY=1

Erase Verify: ERASE=1, VFY=1

FEEN Enable program and erase of Flash. When inactive, it is possible to read as normal mode
- 0 Disable program and erase
- 1 Enable program and erase

FECR (Flash Control Register) : F2H

7	6	5	4	3	2	1	0
AEF	-	EXIT1	EXIT0	WRITE	READ	nFERST	nPBRST
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 03H

AEF — Enable flash bulk erase mode
- 0 Disable bulk erase mode of Flash memory
- 1 Enable bulk erase mode of Flash memory

EXIT[1:0] — Exit from program mode. It is cleared automatically after 1 clock

EXIT1	EXIT0	Description
0	0	Don't exit from program mode
0	1	Don't exit from program mode
1	0	Don't exit from program mode
1	1	Exit from program mode

WRITE — Start to program or erase of Flash. It is cleared automatically after 1 clock
- 0 No operation
- 1 Start to program or erase of Flash

READ — Start auto-verify of Flash. It is cleared automatically after 1 clock
- 0 No operation
- 1 Start auto-verify of Flash

nFERST — Reset Flash control logic. It is cleared automatically after 1 clock
- 0 No operation
- 1 Reset Flash control logic.

nPBRST — Reset page buffer with PBUFF. It is cleared automatically after 1 clock

PBUFF	nPBRST	Description
0	0	Page buffer reset
1	0	Write checksum reset

WRITE and READ bits can be used in program, erase and verify mode with FEAR registers. Read or writes for memory cell or page buffer uses read and write enable signals from memory controller. Indirect address mode with FEAR is only allowed to program, erase and verify

FESR (Flash Status Register) : F3H

7	6	5	4	3	2	1	0
PEVBSY	VFYGOOD	-	-	ROMINT	WMODE	EMODE	VMODE
R	R/W	R	R	R/W	R	R	R

Initial value : 80H

PEVBSY Operation status flag. It is clear automatically when operation starts. Operations are program, erase or verification
- 0 Busy (Operation processing)
- 1 Complete Operation

VFYGOOD Auto-verification result flag.
- 0 Auto-verification fails
- 1 Auto-verification successes

ROMINT Flash interrupt request flag. Auto-cleared when program/erase/verify starts. Active in program/erase/verify completion
- 0 Not interrupt request.
- 1 Interrupt request.

WMODE Write mode flag

EMODE Erase mode flag

VMODE Verify mode flag

FEARL (Flash address low Register) : F5H

7	6	5	4	3	2	1	0
ARL7	ARL6	ARL5	ARL4	ARL3	ARL2	ARL1	ARL0
W	W	W	W	W	W	W	W

Initial value : 00H

ARL[7:0] Flash address low

FEARM (Flash address middle Register) : F6H

7	6	5	4	3	2	1	0
ARM7	ARM6	ARM5	ARM4	ARM3	ARM2	ARM1	ARM0
W	W	W	W	W	W	W	W

Initial value : 00H

ARM[7:0] Flash address middle

FEARH (Flash address high Register) : F7H

7	6	5	4	3	2	1	0
ARH7	ARH6	ARH5	ARH4	ARH3	ARH2	ARH1	ARH0
W	W	W	W	W	W	W	W

Initial value : 00H

ARH[7:0] Flash address high

FEAR registers are used for program, erase and auto-verify. In program and erase mode, it is page address and ignored the same least significant bits as the number of bits of page address. In auto-verify mode, address increases automatically by one.

FEARs are write-only register. Reading these registers returns 24-bit checksum result

FETCR (Flash Time control Register) : F4H

7	6	5	4	3	2	1	0
TCR7	TCR6	TCR5	TCR4	TCR3	TCR2	TCR1	TCR0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

TCR[7:0] Flash Time control

Program and erase time is controlled by setting FETCR register. Program and erase timer uses 10-bit counter. It increases by internal RC oscillator clock frequency (HFO IRC). It is cleared when program or erase starts. Timer stops when 10-bit counter is same to FETCR. PEVBSY is cleared when program, erase or verify starts and set when program, erase or verify stops.

Max program/erase time : (255+1) * 2 * (31.25ns * 256) = 4.096ms

In the case of 10% of error rate of counter source clock, program or erase time is 3.6~4.5ms

* * Program/erase time calculation

for page write or erase, Tpe = (TCON+1) * 2 * (31.25ns * 256)

for bulk erase, Tbe = (TCON+1) * 4 * (31.25ns * 256)

	Min	Typ	Max	Unit
program/erase Time	2.4	2.5	2.6	ms
Bulk erase Time	-	5.0	-	ms

Table 15.2 Program/erase Time

※ Recommended program/erase time (FETCR = A0h)

15.3 Memory map

15.3.1 Flash Memory Map

Program memory uses 8-Kbyte of Flash memory. It is read by byte and written by page. One page is 32-byte

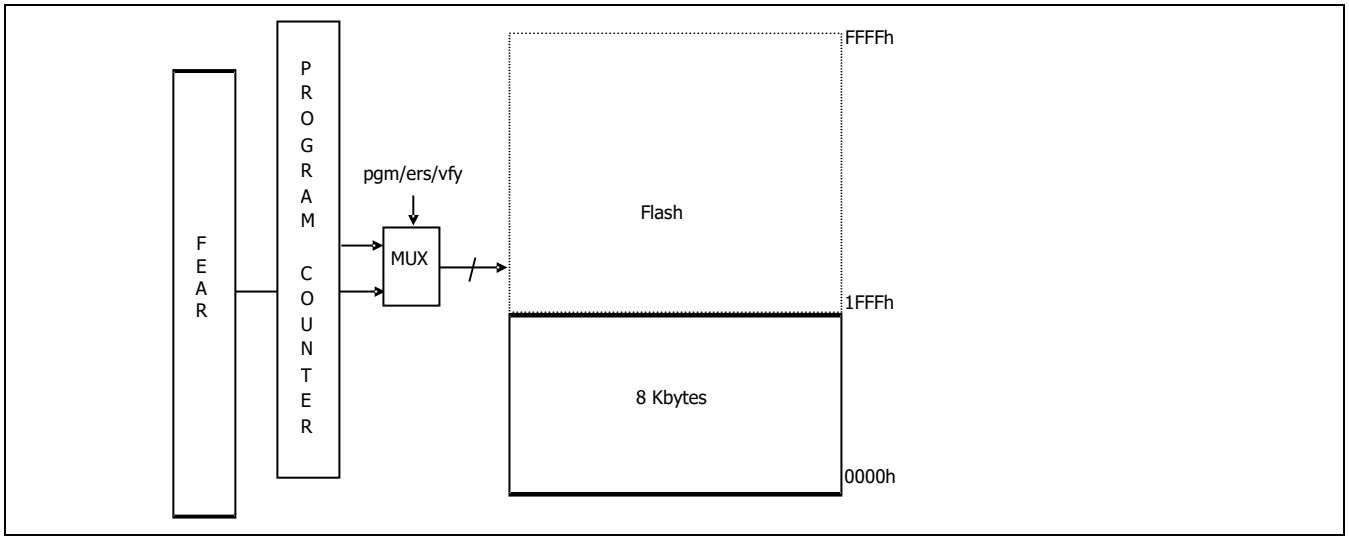


Figure 15.1 Flash Memory Map

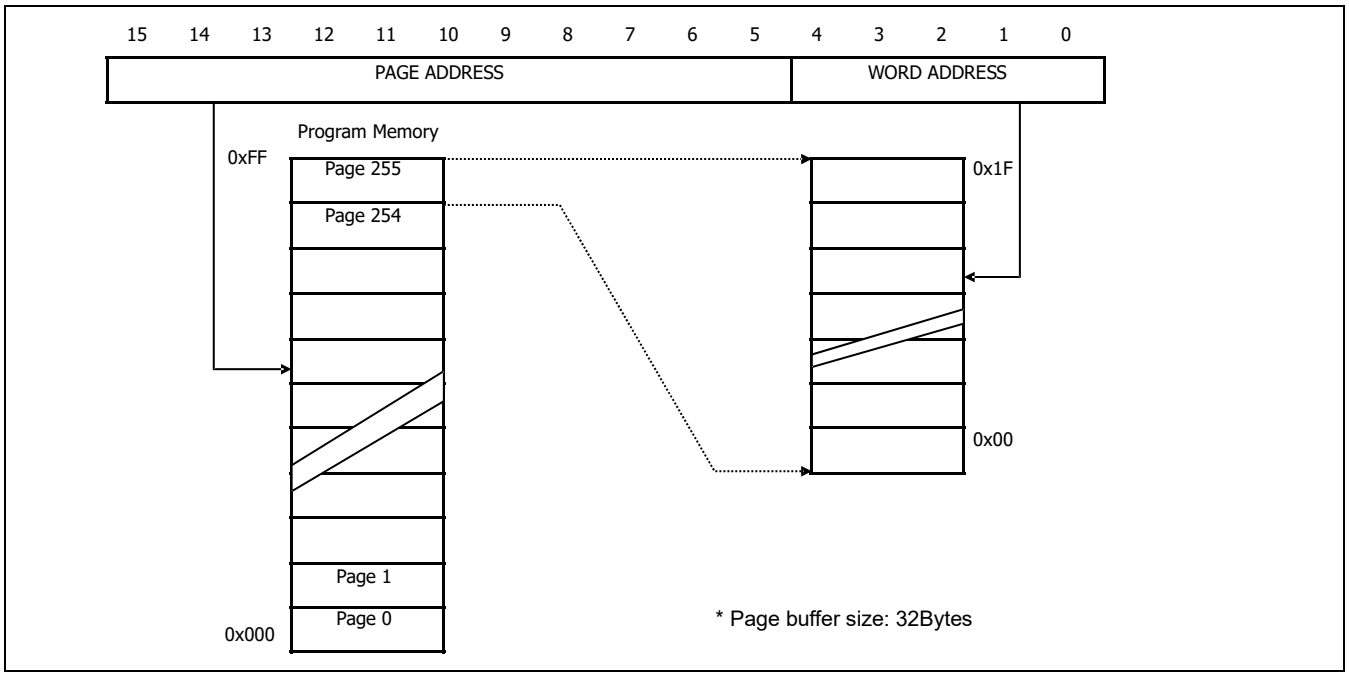


Figure 15.2 Address configuration of Flash memory

15.4 Serial In-System Program Mode

Serial in-system program uses the interface of debugger which uses two wires. Refer to chapter 14 in details about debugger.

15.4.1 Flash operation

Configuration (This Configuration is just used for follow description)

7	6	5	4	3	2	1	0
-	FEMR[4] &[1]	FEMR[5] &[1]	-	-	FEMR[2]	FECR[6]	FECR[7]
-	ERASE&VFY	PGM&VFY	-	-	OTPE	AEE	AEF

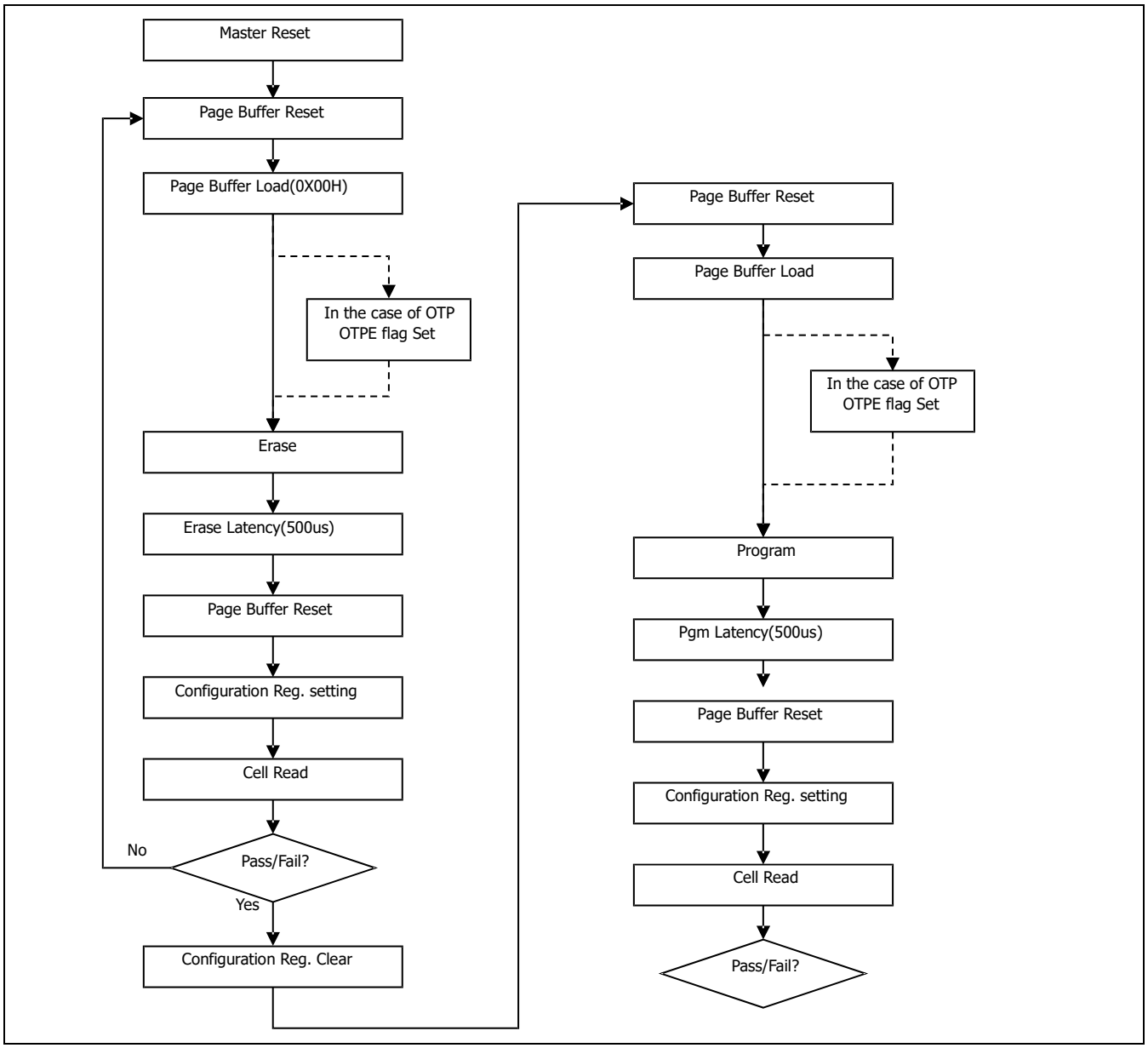


Figure 15.3 The sequence of page program and erase of Flash memory

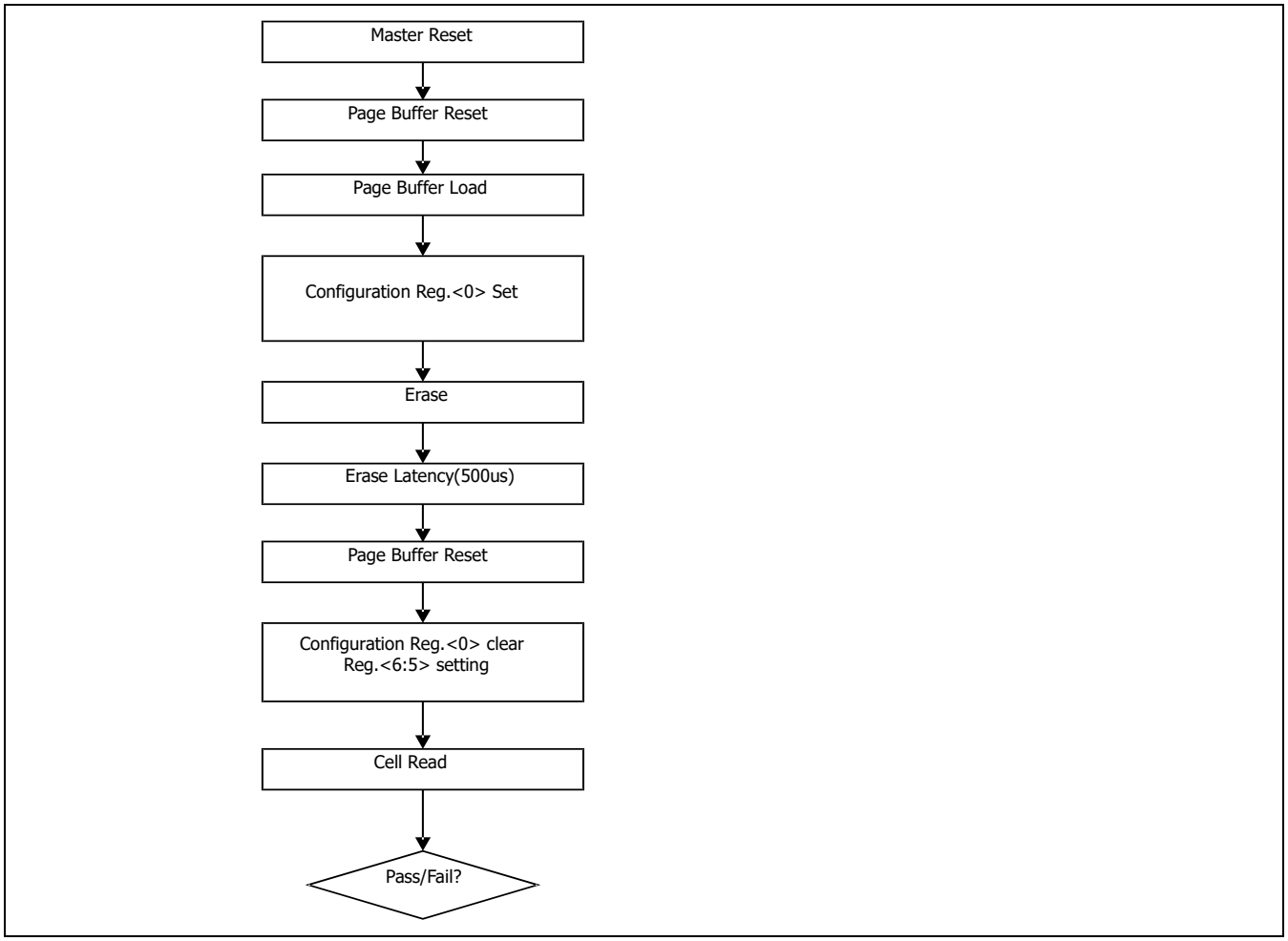


Figure 15.4 The sequence of bulk erase of Flash memory

15.4.1.1 Flash Read

Step 1. Enter OCD(=ISP) mode.

Step 2. Set ENBDM bit of BCR.

Step 3. Enable debug and Request debug mode.

Step 4. Read data from Flash.

15.4.1.2 Enable program mode

Step 1. Enter OCD(=ISP) mode.[1]

Step 2. Set ENBDM bit of BCR.

Step 3. Enable debug and Request debug mode.

Step 4. Enter program/erase mode entry sequence.[2]

(1) Write 0xAA to 0x10D8 (XRAM).

(2) Write 0x55 to 0x10DA (XRAM).

(3) Write 0xA5 to 0x10DD (XRAM).

[1] Refer to how to enter ISP mode..

[2] Command sequence to activate Flash write/erase mode. It is composed of sequentially writing data of Flash memory.

15.4.1.3 Flash write mode

Step 1. Enable program mode.

Step 2. Reset page buffer. FEMR: 1000_0001 FECR:0000_0010

Step 3. Select page buffer. FEMR:1000_1001

Step 4. Write data to page buffer.(Address automatically increases by twin.)

Step 5. Set write mode. FEMR:1010_0001

Step 6. Set page address. FEARH:FEARM:FEARL=20'hx_xxxx

Step 7. Set FETCR.

Step 8. Start program. FECR:0000_1011

Step 9. Insert one NOP operation

Step 10. Read FESR until PEVBSY is 1.

Step 11. Repeat step2 to step 8 until all pages are written.

15.4.1.4 Flash page erase mode

Step 1. Enable program mode.

Step 2. Reset page buffer. FEMR: 1000_0001 FECR:0000_0010

Step 3. Select page buffer. FEMR:1000_1001

Step 4. Write 'h00 to page buffer. (Data value is not important.)

Step 5. Set erase mode. FEMR:1001_0001

Step 6. Set page address. FEARH:FEARM:FEARL=20'hx_xxxx

Step 7. Set FETCR.

Step 8. Start erase. FECR:0000_1011

Step 9. Insert one NOP operation

Step 10. Read FESR until PEVBSY is 1.

Step 11. Repeat step2 to step 8 until all pages are erased.

15.4.1.5 Flash bulk erase mode

Step 1. Enable program mode.

Step 2. Reset page buffer. FEMR: 1000_0001 FECR:0000_0010

Step 3. Select page buffer. FEMR:1000_1001

Step 4. Write 'h00 to page buffer. (Data value is not important.)

Step 5. Set erase mode. FEMR:1001_0001.

(Only main cell area is erased. For bulk erase including OTP area, select OTP area.(set FEMR to 1000_1101.)

Step 6. Set FETCR

Step 7. Start bulk erase. FECR:1000_1011

Step 8. Insert one NOP operation

Step 9. Read FESR until PEVBSY is 1.

15.4.1.6 Flash OTP area read mode

Step 1. Enter OCD(=ISP) mode.

Step 2. Set ENBDM bit of BCR.

Step 3. Enable debug and Request debug mode.

Step 4. Select OTP area. FEMR:1000_0101

Step 5. Read data from Flash.

15.4.1.7 Flash OTP area write mode

Step 1. Enable program mode.

Step 2. Reset page buffer. FEMR: 1000_0001 FECR:0000_0010

Step 3. Select page buffer. FEMR:1000_1001

Step 4. Write data to page buffer.(Address automatically increases by twin.)

Step 5. Set write mode and select OTP area. FEMR:1010_0101

Step 6. Set page address. FEARH:FEARM:FEARL=20'hx_xxxx

Step 7. Set FETCR.

Step 8. Start program. FECR:0000_1011

Step 9. Insert one NOP operation

Step 10. Read FESR until PEVBSY is 1.

15.4.1.8 Flash OTP area erase mode

Step 1. Enable program mode.

Step 2. Reset page buffer. FEMR: 1000_0001 FECR:0000_0010

Step 3. Select page buffer. FEMR:1000_1001

Step 4. Write 'h00 to page buffer. (Data value is not important.)

Step 5. Set erase mode and select OTP area. FEMR:1001_0101

Step 6. Set page address. FEARH:FEARM:FEARL=20'hx_xxxx

Step 7. Set FETCR.

Step 8. Start erase. FECR:0000_1011

Step 9. Insert one NOP operation

Step 10. Read FESR until PEVBSY is 1.

15.4.1.9 Flash program verify mode

Step 1. Enable program mode.

Step 2. Set program verify mode. FEMR:1010_0011

Step 3. Read data from Flash.

15.4.1.10 OTP program verify mode

Step 1. Enable program mode.

Step 2. Set program verify mode. FEMR:1010_0111

Step 3. Read data from Flash.

15.4.1.11 Flash erase verify mode

Step 1. Enable program mode.

Step 2. Set erase verify mode. FEMR:1001_0011

Step 3. Read data from Flash.

15.4.1.12 Flash page buffer read

Step 1. Enable program mode.

Step 2. Select page buffer. FEMR:1000_1001

Step 3. Read data from Flash.

15.4.2 Summary of Flash Program/Erase Mode

Operation mode		Description
FLASH	Flash read	Read cell by byte.
	Flash write	Write cell by page.
	Flash page erase	Erase cell by page.
	Flash bulk erase	Erase the whole cells.
	Flash program verify	Read cell in verify mode after programming.
	Flash erase verify	Read cell in verify mode after erase.
	Flash page buffer load	Load data to page buffer.

Table 15.3 Operation Mode

15.5 Mode entrance method of ISP mode

TARGET MODE	DSDA	DSCL	DSDA
OCD(ISP)	‘hC	‘hC	‘hC

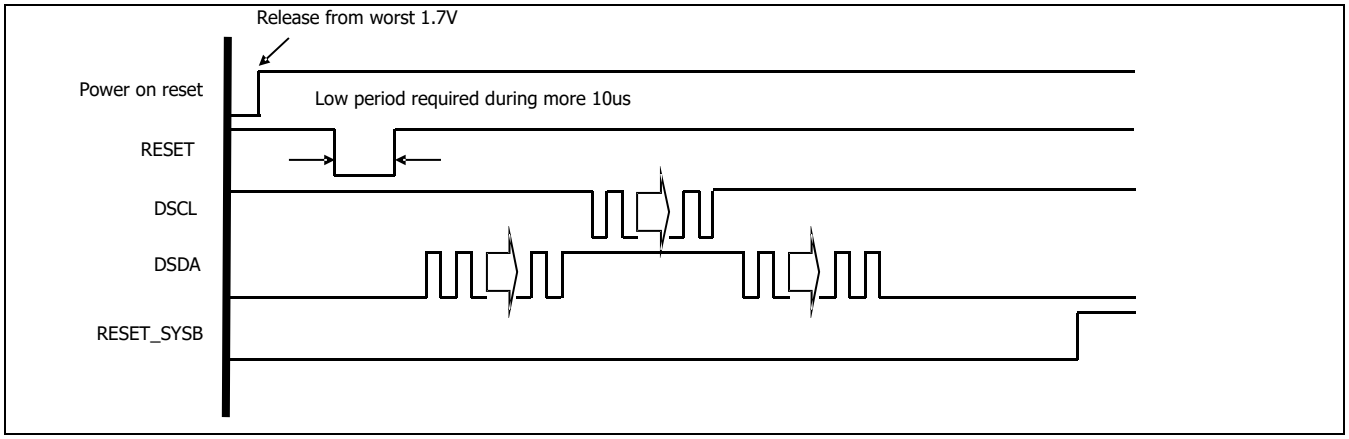


Figure 15.5 ISP mode

15.6 SRAM-Jump Program mode

In this device SRAM-jump program is used to self programing in user mode. When writing the flash data memory, cpu should operate in sram not flash. SRAM address 0x00 is assigned to program address 0x8000. Refer to the following example.

```
main()
{
....

// 1. Flash mode entry with movx instruction
   *(unsigned char xdata *)0x10D8 = 0xAA;
   *(unsigned char xdata *)0x10DA = 0x55;
   *(unsigned char xdata *)0x10DD = 0xA5;

   FETCR = 0x9D;   // 2.5ms PGM time

   FEMR = 0x81;     // 3. Reset page buffer
   FECR = 0x02;

   write_page_buffer(); // 4. Write page buffer

   do_flash_at_sram(); // 5. Write flash

   FECR = 0x33; // 6. Flash mode exit
....
   Return ;
}


void do_flash_at_sram_src()
{
   FEMR = 0xA1;  //
   FECR = 0x0B;  // Enable program

   while( !(FESR & 0x80) );

   FEMR = 0;
}


void do_flash_at_sram()
{
   #pragma asm
      mov dptr,#do_flash_at_sram_src
      mov r0,#0x30
      mov r1,#0x13
      do_flash_at_sram_loop:
         clr a
         movc a,@a+dptr
         mov @r0,a
         inc dptr
         inc r0
         djnz r1,do_flash_at_sram_loop
      ljmp 0x8030
   #pragma endasm
}
```

```
void write_page_buffer()
{
   #pragma asm
      mov dptr,#write_page_buffer_src
      mov r0,#0x30
      mov r2,#0x12       ;sram
      write_page_buffer_loop:
         clr a
         movc a,@a+dptr
         mov @r0,a
         inc dptr
         inc r0
         djnz r2,write_page_buffer_loop
      ljmp 0x8030                    ; jump sram region
   #pragma endasm
}



void write_page_buffer_src()
{
   FEMR = 0x81;

  #pragma asm
     mov r0,#32
     mov dptr,#0x10E0  ; page buffer address
     write_page_buffer_src_loop:
      mov a, @r1          // write data is written in the SRAM
                          // previosly and r1 has the address
      movx @dptr,a
      inc r1
      inc dptr
      djnz r0,write_page_buffer_src_loop
   #pragma endasm

   FEMR = 0;
}
```

Figure 15.6 Code example of flash write by SRAM-jump mode

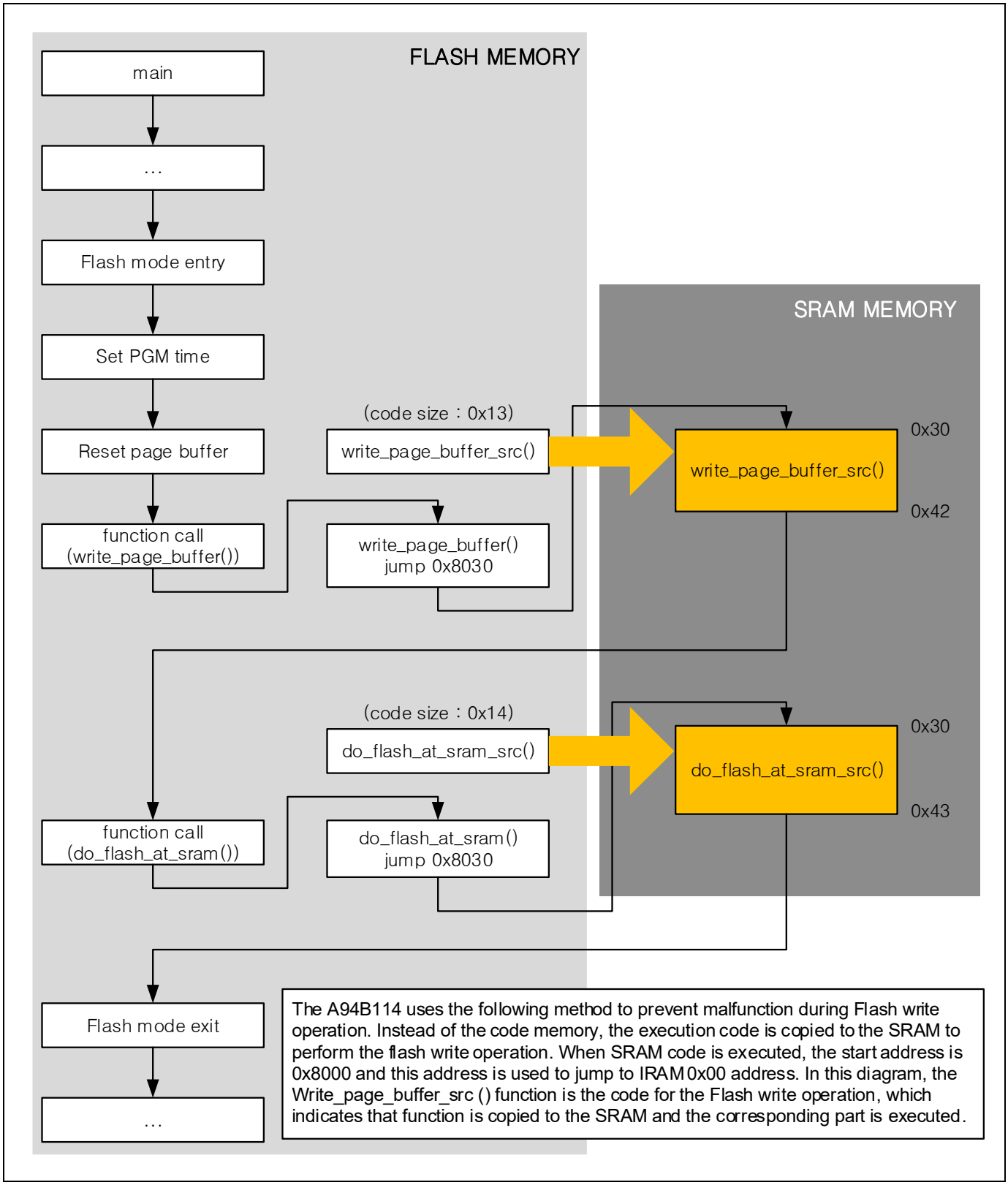


Figure 15.7 Memory diagram and flow of flash write by SRAM-jump mode

16 Configure Option

16.1 Configure Option Control

FUSE_CFG0 (Pseudo-Configure Data)

7	6	5	4	3	2	1	0
RSTEN	–	XOENA	XIENA	–	LOCKVA	LOCKHL	LOCKF
R	–	R	R	–	R	R	R

Initial value : 00H

RSTEN Select RESETB pin
- 0 Disable RESETB pin (default)
- 1 Enable RESETB pin

XOENA Enable Main OSC output
- 0 Disable
- 1 Enable

XIENA Enable Main OSC input
- 0 Disable
- 1 Enable

LCKVA Vector Area(00~FF) Protection selection
- 0 Enable protection (Not erasable by instruction)
- 1 Disable protection (Erasable by instruction)

LOCKHL Code Write Protection
- 0 Enable protection
- 1 Disable protection

LOCKF Code Read Protection
- 0 Disable protection
- 1 Enable protection

FUSE_CFG1 (Pseudo-Configure Data)

7	6	5	4	3	2	1	0
–	–	–	–	PASEL	PASS2	PASS1	PASS0
–	–	–	–	R	R	R	R

Initial value : 00H

PASEL Specific Area Write Protection Selection
- 0 Enable protection (Not erasable by instruction)
- 1 Disable protection (Erasable by instruction)

PASS[2:0] When PASEL is 0 (Enable protection), Code write protection is possible in the set address area.
- 0 0 0 0100H ~ 1BFFH
- 0 0 1 0100H ~ 1DFFH
- 0 1 0 0100H ~ 1EFFH
- 0 1 1 0100H ~ 1F7FH
- -
- 1 0 1 0100H ~ 17FFH
- 1 1 0 0100H ~ 0FFFH
- 1 1 1 0100H ~ 07FFH

17 APPENDIX

17.1 Instruction Table

Instructions are either 1, 2 or 3 bytes long as listed in the 'Bytes' column below.

Each instruction takes either 1, 2, 3, 4 or 5 machine cycles to execute as listed in the following table. 1 machine cycle comprises 1 system clock cycles.

ARITHMETIC				
Mnemonic	**Description**	**Bytes**	**Clocks**	**Hex code**
ADD A,Rn	Add register to A	1	2	28-2F
ADD A,dir	Add direct byte to A	2	3	25
ADD A,@Ri	Add indirect memory to A	1	3	26-27
ADD A,#data	Add immediate to A	2	2	24
ADDC A,Rn	Add register to A with carry	1	2	38-3F
ADDC A,dir	Add direct byte to A with carry	2	3	35
ADDC A,@Ri	Add indirect memory to A with carry	1	3	36-37
ADDC A,#data	Add immediate to A with carry	2	2	34
SUBB A,Rn	Subtract register from A with borrow	1	2	98-9F
SUBB A,dir	Subtract direct byte from A with borrow	2	3	95
SUBB A,@Ri	Subtract indirect memory from A with borrow	1	3	96-97
SUBB A,#data	Subtract immediate from A with borrow	2	2	94
INC A	Increment A	1	1	04
INC Rn	Increment register	1	2	08-0F
INC dir	Increment direct byte	2	3	05
INC @Ri	Increment indirect memory	1	3	06-07
DEC A	Decrement A	1	1	14
DEC Rn	Decrement register	1	2	18-1F
DEC dir	Decrement direct byte	2	3	15
DEC @Ri	Decrement indirect memory	1	3	16-17
INC DPTR	Increment data pointer	1	1	A3
MUL AB	Multiply A by B	1	8	A4
DIV AB	Divide A by B	1	8	84
~~DA A~~ [NOTE1]	~~Decimal Adjust A~~	~~1~~	~~1~~	~~D4~~

LOGICAL				
Mnemonic	**Description**	**Bytes**	**Clocks**	**Hex code**
ANL A,Rn	AND register to A	1	2	58-5F
ANL A,dir	AND direct byte to A	2	3	55
ANL A,@Ri	AND indirect memory to A	1	3	56-57
ANL A,#data	AND immediate to A	2	2	54
ANL dir,A	AND A to direct byte	2	3	52
ANL dir,#data	AND immediate to direct byte	3	3	53
ORL A,Rn	OR register to A	1	2	48-4F
ORL A,dir	OR direct byte to A	2	3	45
ORL A,@Ri	OR indirect memory to A	1	3	46-47
ORL A,#data	OR immediate to A	2	2	44
ORL dir,A	OR A to direct byte	2	3	42
ORL dir,#data	OR immediate to direct byte	3	3	43
XRL A,Rn	Exclusive-OR register to A	1	2	68-6F
XRL A,dir	Exclusive-OR direct byte to A	2	3	65
XRL A, @Ri	Exclusive-OR indirect memory to A	1	3	66-67
XRL A,#data	Exclusive-OR immediate to A	2	2	64
XRL dir,A	Exclusive-OR A to direct byte	2	3	62
XRL dir,#data	Exclusive-OR immediate to direct byte	3	3	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**	**Clocks**	**Hex code**
MOV A,Rn	Move register to A	1	2	E8-EF
MOV A,dir	Move direct byte to A	2	3	E5
MOV A,@Ri	Move indirect memory to A	1	3	E6-E7
MOV A,#data	Move immediate to A	2	2	74
MOV Rn,A	Move A to register	1	2	F8-FF
MOV Rn,dir	Move direct byte to register	2	3	A8-AF
MOV Rn,#data	Move immediate to register	2	2	78-7F
MOV dir,A	Move A to direct byte	2	2	F5
MOV dir,Rn	Move register to direct byte	2	2	88-8F
MOV dir,dir	Move direct byte to direct byte	3	3	85
MOV dir,@Ri	Move indirect memory to direct byte	2	3	86-87
MOV dir,#data	Move immediate to direct byte	3	3	75
MOV @Ri,A	Move A to indirect memory	1	2	F6-F7
MOV @Ri,dir	Move direct byte to indirect memory	2	3	A6-A7
MOV @Ri,#data	Move immediate to indirect memory	2	3	76-77
MOV DPTR,#data	Move immediate to data pointer	3	3	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	1	F2-F3
MOVX @DPTR,A	Move A to external data(A16)	1	1	F0
PUSH dir	Push direct byte onto stack (except PUSH SP)[NOTE1]	2	3	C0
POP dir	Pop direct byte from stack (except POP SP)[NOTE1]	2	3	D0
XCH A,Rn	Exchange A and register	1	2	C8-CF
XCH A,dir	Exchange A and direct byte	2	4	C5
XCH A,@Ri	Exchange A and indirect memory	1	3	C6-C7
XCHD A,@Ri	Exchange A and indirect memory nibble	1	3	D6-D7

BOOLEAN				
Mnemonic	**Description**	**Bytes**	**Clocks**	**Hex code**
CLR C	Clear carry	1	1	C3
CLR bit	Clear direct bit	2	3	C2
SETB C	Set carry	1	1	D3
SETB bit	Set direct bit	2	3	D2
CPL C	Complement carry	1	1	B3
CPL bit	Complement direct bit	2	3	B2
ANL C,bit	AND direct bit to carry	2	3	82
ANL C,/bit	AND direct bit inverse to carry	2	3	B0
ORL C,bit	OR direct bit to carry	2	3	72
ORL C,/bit	OR direct bit inverse to carry	2	3	A0
MOV C,bit	Move direct bit to carry	2	3	A2
MOV bit,C	Move carry to direct bit	2	3	92

BRANCHING				
Mnemonic	**Description**	**Bytes**	**Clocks**	**Hex code**
ACALL addr 11	Absolute jump to subroutine	2	4	11→F1
LCALL addr 16	Long jump to subroutine	3	2	12
RET	Return from subroutine	1	4	22
RETI	Return from interrupt	1	4	32
AJMP addr 11	Absolute jump unconditional	2	3	01→E1
LJMP addr 16	Long jump unconditional	3	4	02
SJMP rel	Short jump (relative address)	2	3	80
JC rel	Jump on carry = 1	2	3	40
JNC rel	Jump on carry = 0	2	3	50
JB bit,rel	Jump on direct bit = 1	3	5	20
JNB bit,rel	Jump on direct bit = 0	3	5	30
JBC bit,rel	Jump on direct bit = 1 and clear	3	5	10
JMP @A+DPTR	Jump indirect relative DPTR	1	2	73
JZ rel	Jump on accumulator = 0	2	3	60
JNZ rel	Jump on accumulator ≠0	2	3	70
CJNE A,dir,rel	Compare A,direct jne relative	3	5	B5
CJNE A,#d,rel	Compare A,immediate jne relative	3	4	B4
CJNE Rn,#d,rel	Compare register, immediate jne relative	3	4	B8-BF
CJNE @Ri,#d,rel	Compare indirect, immediate jne relative	3	5	B6-B7
DJNZ Rn,rel	Decrement register, jnz relative	2	4	D8-DF
DJNZ dir,rel	Decrement direct byte, jnz relative	3	5	D5

MISCELLANEOUS				
Mnemonic	**Description**	**Bytes**	**Clocks**	**Hex code**
NOP	No operation	1	1	00

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.

NOTE1) PUSH SP, POP SP, and DA instruction behave differently from 8051 operations.
The use of this instruction only occurs when using the assembly language, and its use is very limited. Refer to "ERRATA_ABOV_94 CPU(CM8051) Incompatible Instruction" for more details.

17.2 Package relation

	A94B114FR (20-Pin)	**A94B114FD (20-Pin)**	**A94B114AE (13-Pin)**
Pin count	20		16
Max I/O	18		14
Difference (removed functions on standard A94B114)	-		AN8, AN9, EINT1, EINT10 EC0, T0O, PWM1OB

Table of contents