CMOS single-chip 8-bit MCU with 12-bit A/D converter and CEC Controller

Main features

- **8-bit Microcontroller With High Speed 8051 CPU**
- **Basic MCU Function**
  - 64KB Flash Code Memory
  - 4,352 bytes SRAM
  - 2KB Data EEPROM
- **Built-in Analog Function**
  - Power-On Reset and Low Voltage Detect Reset
  - Internal 16MHz RC Oscillator (±1.5%, T_A = 0 ~ +50°C)
  - Watchdog Timer RC Oscillator (256kHz)
- **Peripheral Features**
  - 12-bit Analog to Digital Converter (10 inputs)
  - USART 8-bit x 2-ch
  - SPI 8-bit x 2-ch
  - I2C 8-bit x 2-ch
  - Hardware CEC Controller
  - Hardware IR Receiver
- **I/O and Packages**
  - Up to 46 programmable I/O lines with 48-LQFP
  - 48-LQFP
- **Operating Conditions**
  - 2.7V to 5.5V Wide Voltage Range
  - -40°C to 85°C Temperature Range
- **Application**
  - Home Appliance

A97C450

User's manual

V 1.02

Revised 31 Jan, 2018

Revision History

Version	Date	Revision list
1.00	2017.08.11	The First Edition.
1.01	2017.10.26	Maximum value of IDD6 (STOP2) is added in “Table 7.9 DC Characteristics”. The content of flowchart is changed in “Figure 11.64 Procedure for Reading RTC”. Note about the writing to I2CSR is added in “11.12.11”
1.02	2018.01.31	Add Device Nomenclature. Awake-up interrupt from power down mode is removed in 11.14. Fix 11.17.9 RTC Register description (YEAR Register bit (Address 0x8FBA)).

Version 1.02

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.

For more information, please visit ABOV’s website (http://www.abov.co.kr).

1 Overview

1.1. Description

The A97C450 is advanced CMOS 8-bit microcontroller with FLASH (64KB) and EEPROM (2KB). This is powerful microcontroller which provides a highly flexible and cost effective solution to many embedded control applications. This provides the following features : 64KB of FLASH, 2KB of Data EEPROM, 256 bytes of SRAM, 4,096 bytes of XRAM, general purpose I/O, 8/16-bit timer/counter, watchdog timer, watch timer, RTC with calendar, SPI, USART, I2C, on-chip POR and LVI, 12-bit A/D converter, analog comparator, buzzer driving port, 8/16-bit PWM output, on-chip oscillator, CEC, IR receiver and clock circuitry. The A97C450 also supports power saving modes to reduce power consumption.

1.2 Ordering information

Device Name	Flash	XRAM	IRAM	EEPROM	ADC	I/O PORT	Package
A97C450CLN	64KB	4,096 bytes	256 bytes	2KB	10 inputs	46	48-LQFP

Table 1.1 Ordering Information of A97C450

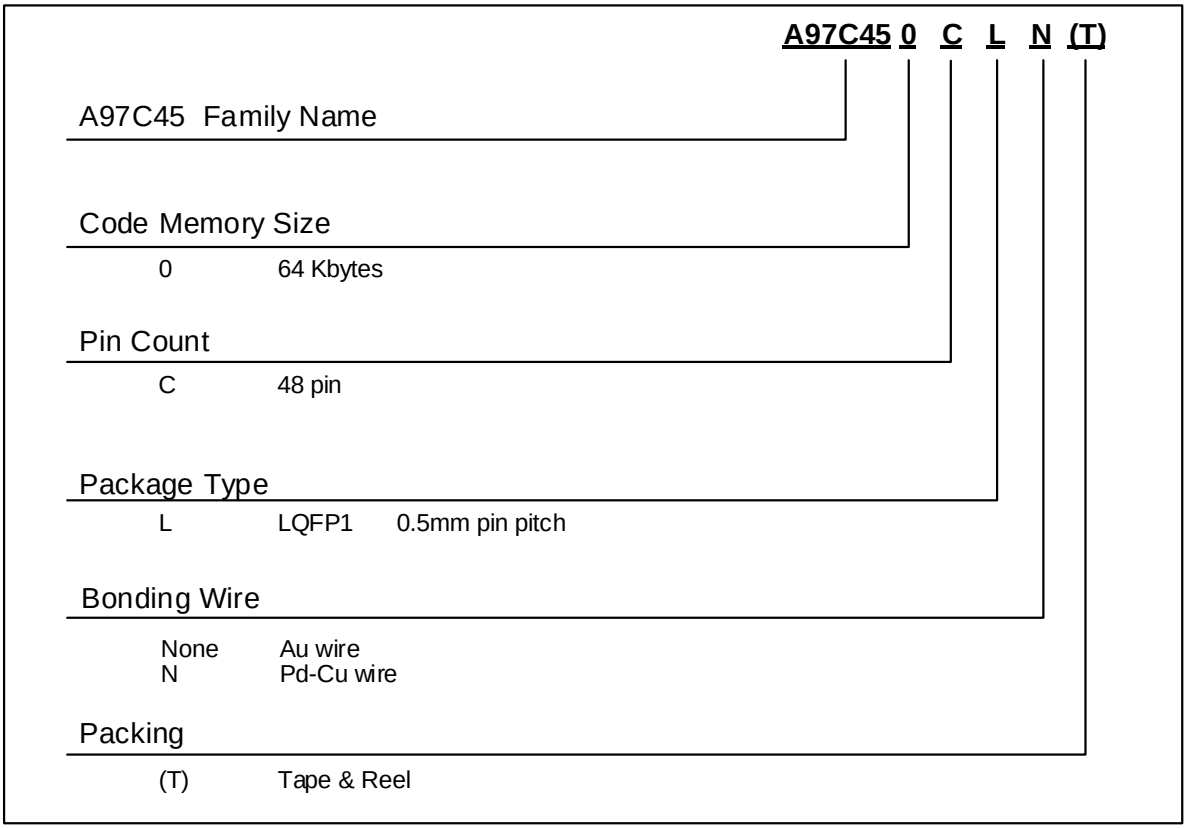


Figure 1.1 Device Nomenclature

1.3 Features

- **CPU**
  - 8-bit CISC core (M8051, 2 clocks per cycle)
- **ROM (FLASH) Capacity**
  - 64KB Flash with self-read and write capability
  - In-System Programming(ISP)
  - Endurance : 10,000 times
  - Retention : 10 years
- **2KB On-chip EEPROM**
  - Endurance : 300,000 times
  - Retention : 10 years
- **256 bytes IRAM / 4,096 bytes XRAM**
- **General Purpose I/O (GPIO)**
  - Normal I/O : 46 Ports (P0, P1, P2, P3, P4, P5[5:0])
- **Boot Loader Protection with Boot Swap**
- **External stack with stack overflow interrupt**
- **Hardware CEC**
- **Hardware IR Receiver with Noise Filter**
- **Timer/Counter**
  - Basic Interval Timer (BIT) 8-bit × 1-ch
  - Watch Dog Timer (WDT) 8-bit × 1-ch
    256kHz internal RC oscillator for WDT
  - 8-bit × 2-ch (T0/T1)
  - 16-bit × 5-ch (T2/T3/T4/T5/T6)
- **Programmable Pulse Generation**
  - Pulse generation (by T2/T3/T4/T5/T6)
  - 8-bit PWM (by T0/T1)
- **Watch Timer (WT)**
  - 3.91ms/0.25s/0.5s/1s /1min interval at 32.768kHz
- **RTC with 100 Year Calendar**
- **Buzzer**
  - 8-bit × 1-ch
- **SPI**
  - 8-bit × 2-ch
- **USART**
  - 8-bit USART × 2-ch
- **I2C**
  - I2C × 2-ch
- **12-bit A/D Converter**
  - 10 Input channels
- **Analog Comparator**
  - 2 Internal Comparator
- **Power On Reset**
  - Reset release level (1.4V)
- **Low Voltage Reset**
  - 1.6V
- **Low Voltage Indicator**
  - 3 levels detect (2.4V / 2.8V / 3.5V)
- **Interrupt Sources**
  - External Interrupts
    (EINT0, EINT1, EINT2~4, EINT5~7) (4)
  - Timer(0/1/2/3/4~6) (5)
  - CEC (1)
  - IR (1)
  - USART (4), SPI (2), I2C (2)
  - WDT (1)
  - WT (1)
  - RTC (1)
  - BIT (1)
  - ADC (1)
  - EEPROM (1)
  - LVI (1)
  - Analog comparator (1)
  - CRC (1)
  - Stack OVF (1)
- **Internal RC Oscillator**
  - Internal RC frequency: 16MHz ±1.5% (T_A= 0 ~ +50°C)
- **Power Down Mode**
  - STOP1/2, SLEEP1/2, IDLE mode
- **Operating Voltage and Frequency**
  - 2.7V~ 5.5V (@ 32 ~ 38kHz with Sub Crystal)
  - 2.7V~ 5.5V (@ 0.4 ~ 12.0MHz with Crystal)
  - 2.7V~ 5.5V (@ 1.0 ~ 16.0MHz with Internal RC)
  - Voltage dropout converter included for core
- **Minimum Instruction Execution Time**
  - 125ns (@16MHz clock)
- **Operating Temperature**
  - –40 ~ +85℃
- **Oscillator Type**
  - 0.4 - 12MHz Crystal or Ceramic for main clock
  - 32.768kHz Crystal for sub clock
- **Package Type**
  - 48-LQFP

1.4 Development tools

1.4.1 Compiler

ABOV Semiconductor does not provide compiler. It is recommended that you consult a compiler provider.

The A97C450 core is Mentor 8051 and the ROM size is smaller than 64KB. Therefore, developer can use the standard 8051 compiler from other providers.

1.4.2 OCD(On-chip debugger) emulator and debugger

The OCD (On Chip Debug) emulator supports ABOV Semiconductor's 8051 series MCU emulation. The OCD interface uses two-wire connection between PC and MCU which is attached to user's system. The OCD can read or change the value of MCU internal memory and I/O peripherals. And the OCD also controls MCU internal debugging logic, it means OCD controls emulation, step run, monitoring, etc.

The OCD debugger program works on Microsoft-Windows NT, 2000, XP, Vista (32-bit), 7, 8, 10 operating system. If you want to see more details, please refer to OCD debugger manual. You can download debugger S/W and manual from our web-site (http://www.abov.co.kr).

Connection:

- DSCL (A97C450 P36 port)
- DSDA (A97C450 P37 port)

OCD connector diagram: Connect OCD with user system

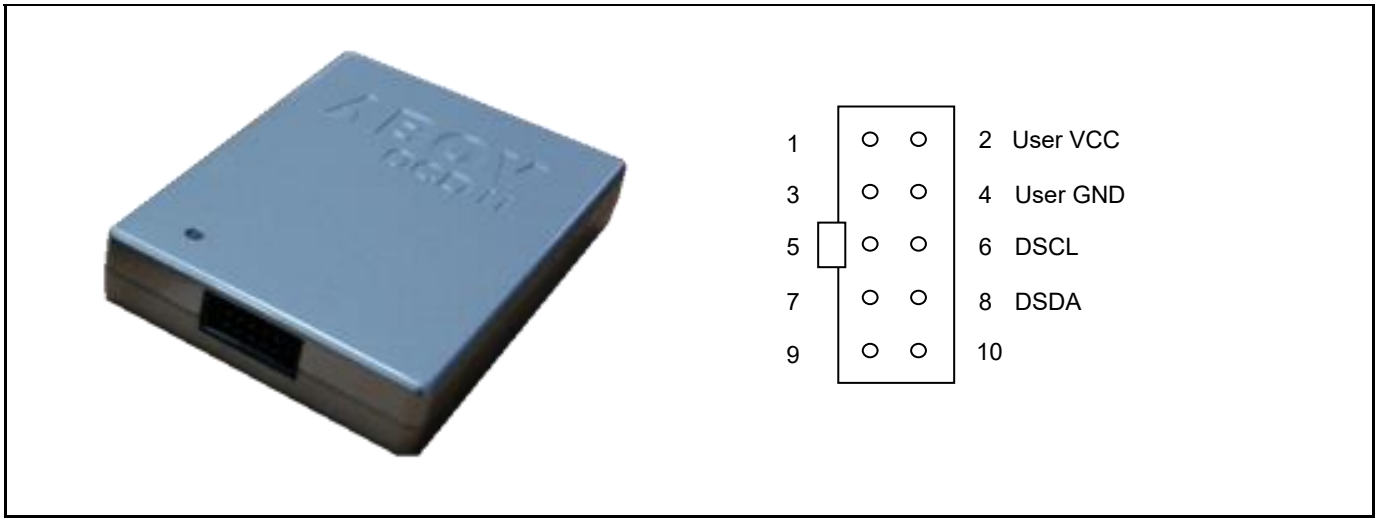


Figure 1.2 debugger and pin description

OCD2 (On Chip Debug) Emulator

- MCU emulation control via 2pin or 3pin OCD interface.
- 2pin interface : OCD2 clock & data.
- 1pin option interface
  - Support device OCD2 mode entry during user S/W is running.
  - Support exact emulation time measurement.
- 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 OCD and 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.

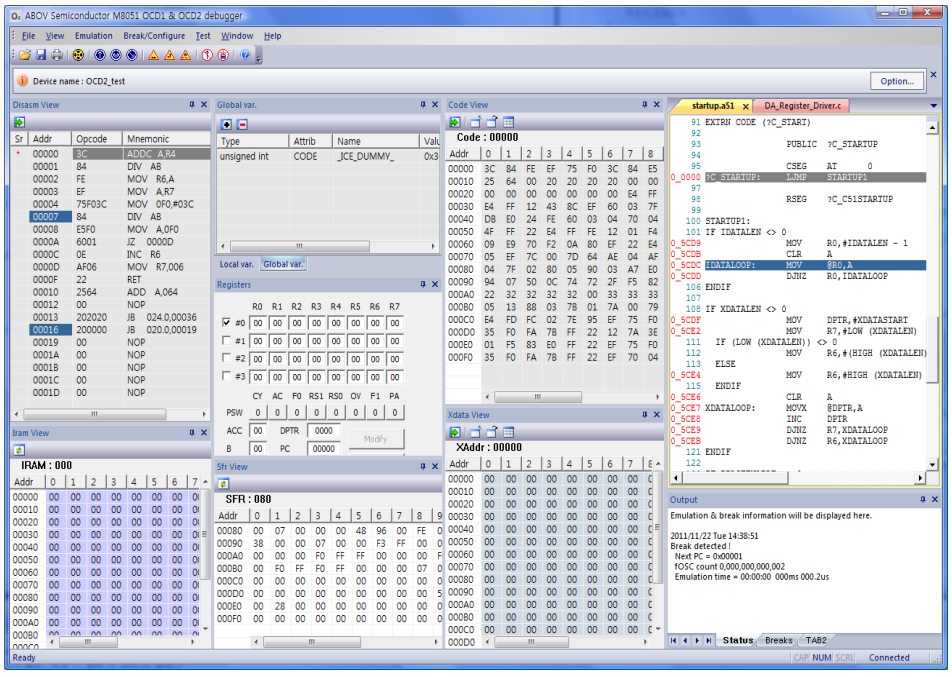


Figure 1.3 OCD Debugger

Support Devices

- MC94xxxx
- MC95xxxx
- MC96xxxx
- MC97xxxx

E-PGM +

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

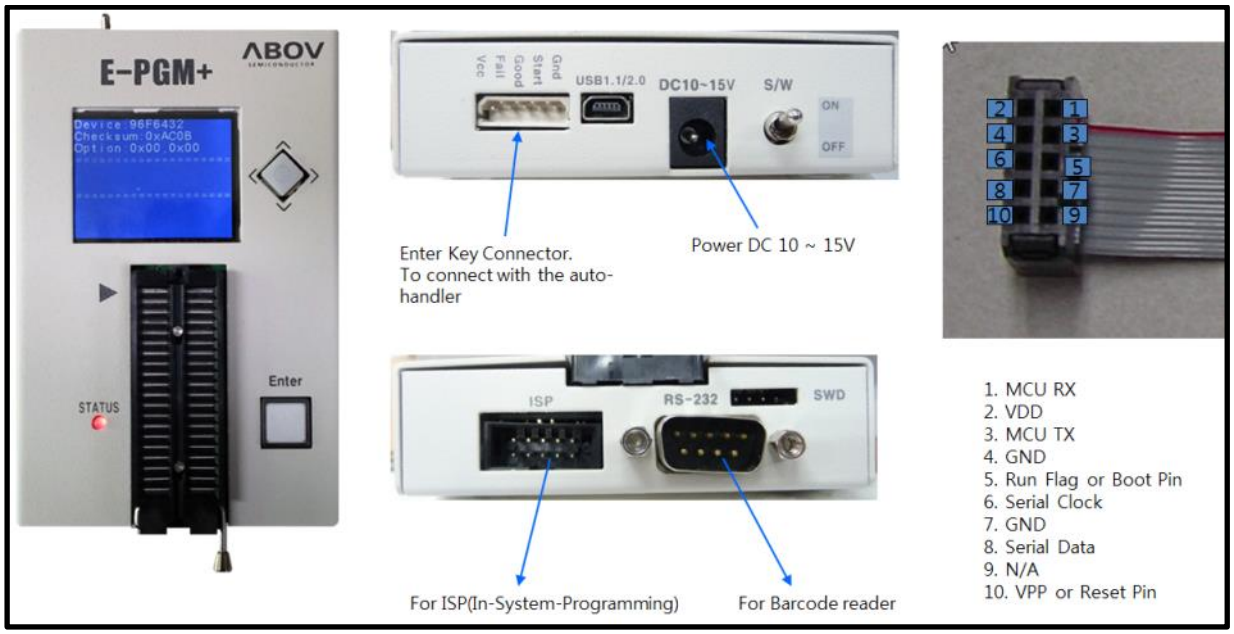


Figure 1.4 component and connector

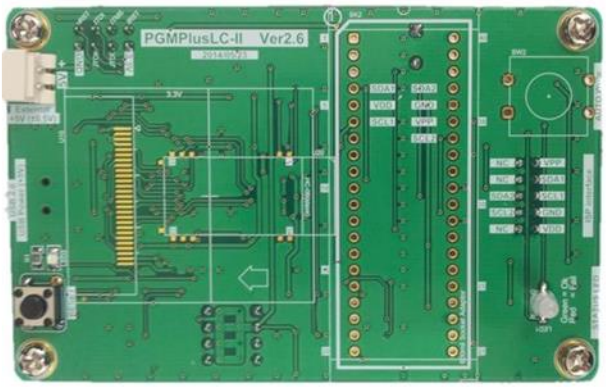
PGMPlusLC 2

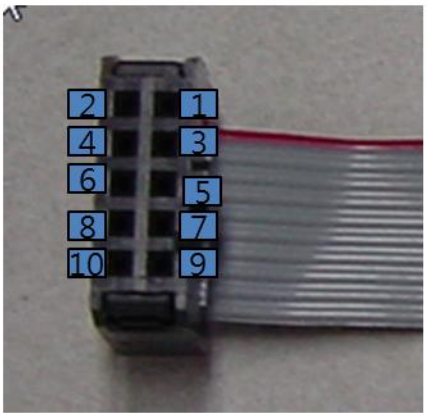
Description

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).
- Fast 32-bit Cortex-M3 MCU is used.
- Supported high voltage Max 18V.
- PGMplusLC2 is based on PC environment.
- PGMplusLC2 is faster than PGMplusLC.
- Transmission speed is 64kbyte/s





2. Vdd
4. GND
6. Serial Clock
8. Serial Data

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

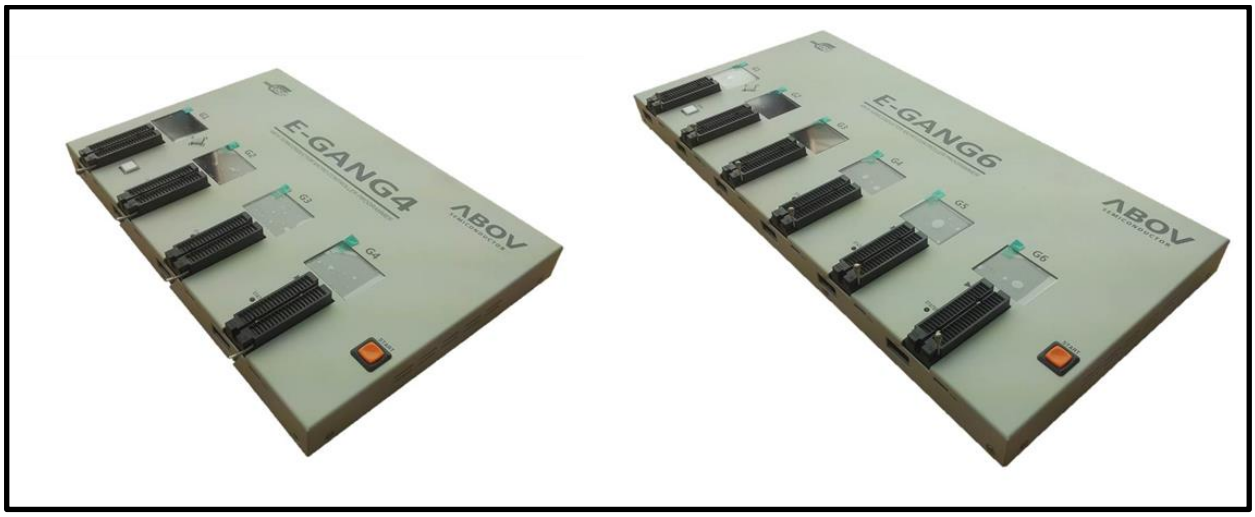


Figure 1.6 E-PGM+ Gang4/6 Programmer

1.5 MTP programming

1.5.1 Overview

The program memory of A97C450 is MTP Type. This flash is accessed by serial data format. There are four pins(DSCL, DSDA, VDD and VSS) for programming/reading the flash.

Pin name	Main chip pin name	During programming	
		I/O	Description
DSCL	P36	I	Serial clock pin. Input only pin.
DSDA	P37	I/O	Serial data pin. Output port when reading and input port when programming. Can be assigned as input/push-pull output port.
VDD, VSS	VDD, VSS	-	Logic power supply pin.

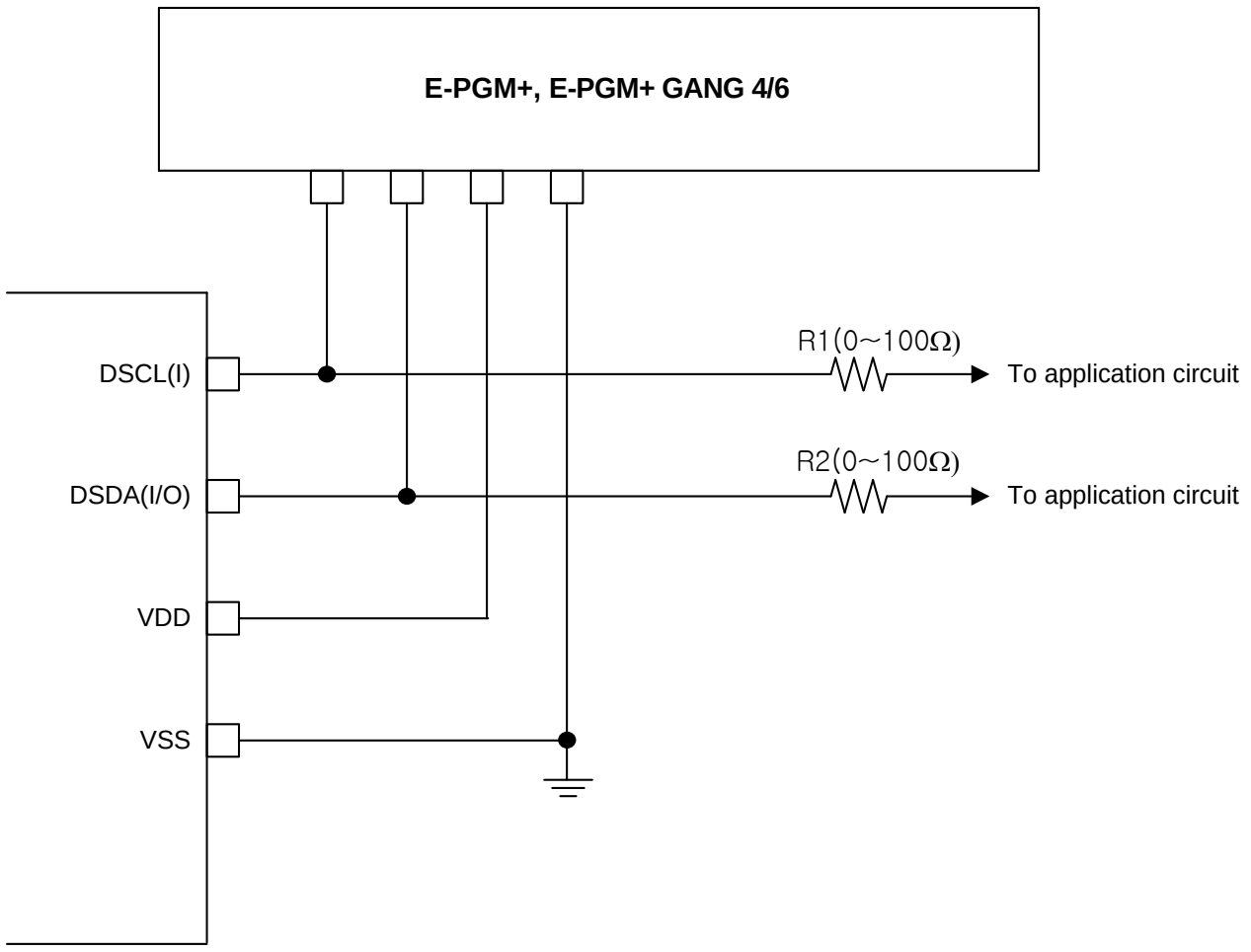
Table 1.2 Descriptions of pins which are used to programming/reading the Flash

1.5.2 On-Board programming

The A97C450 needs only four signal lines including VDD and VSS pins for programming FLASH with serial communication protocol. Therefore the on-board programming is possible if the programming signal lines are ready at the PCB of application board is designed.

1.5.2.1 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.

The value of R1 and R2 is recommended value and it varies with circuit of system.

Figure 1.7 PCB design guide for on board programming

2 Block diagram

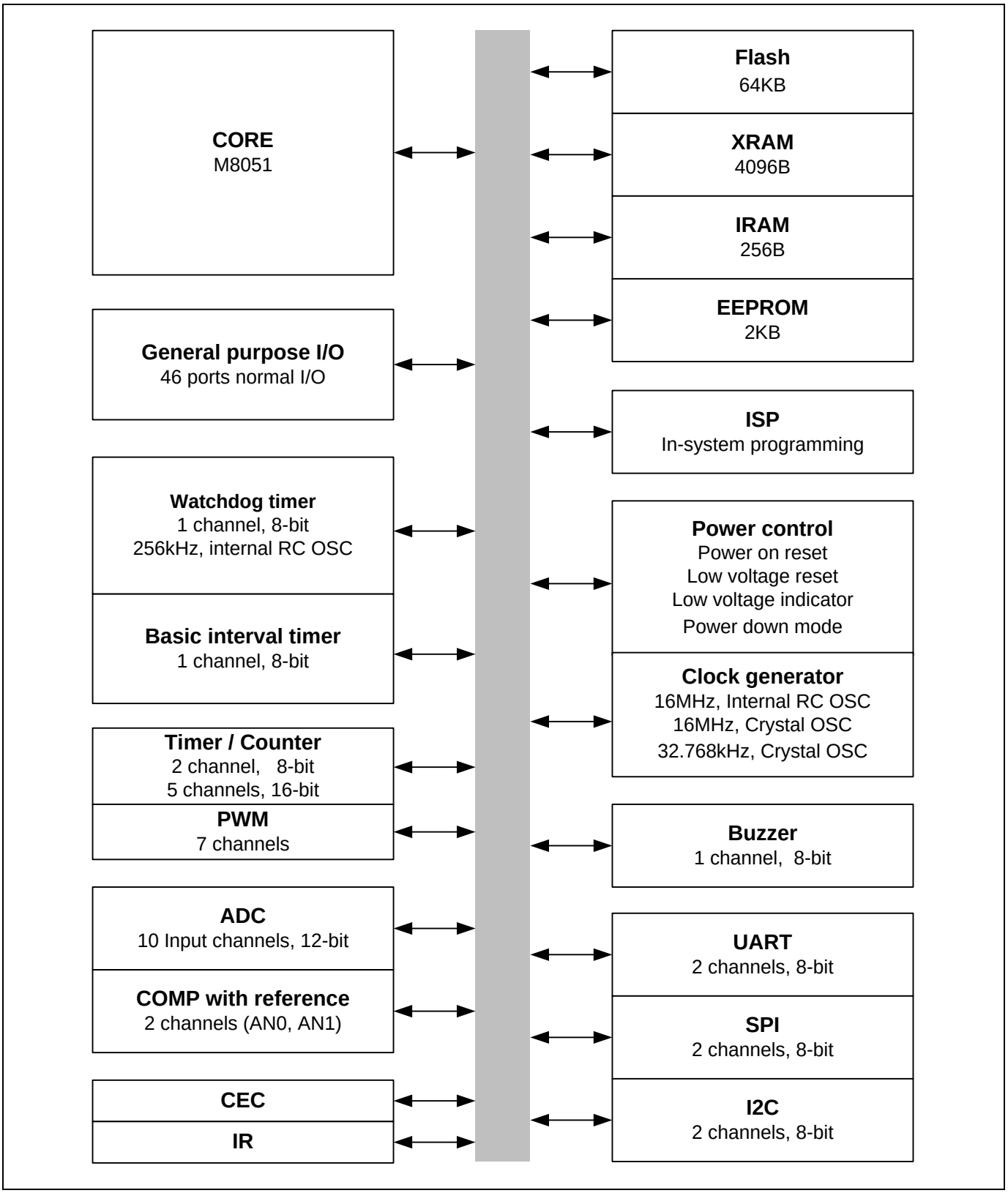
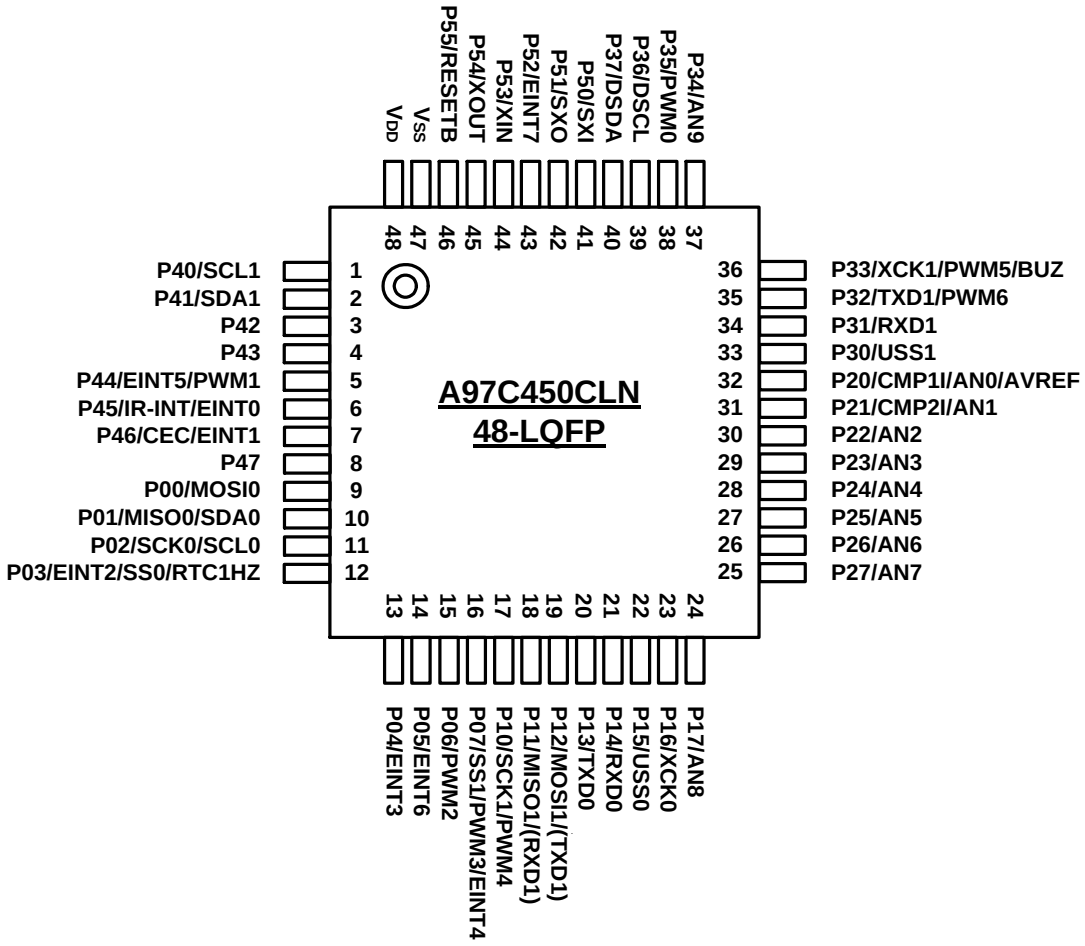


Figure 2.1 Block diagram of A97C450

3 Pin assignment



NOTE)

1. The programmer (E-PGM+, E-PGM+ GANG 4/6) uses P36, P37 pin as DSCL, DSDA.

Figure 3.1 A97C450CLN 48-LQFP pin assignment

4 Package Diagram

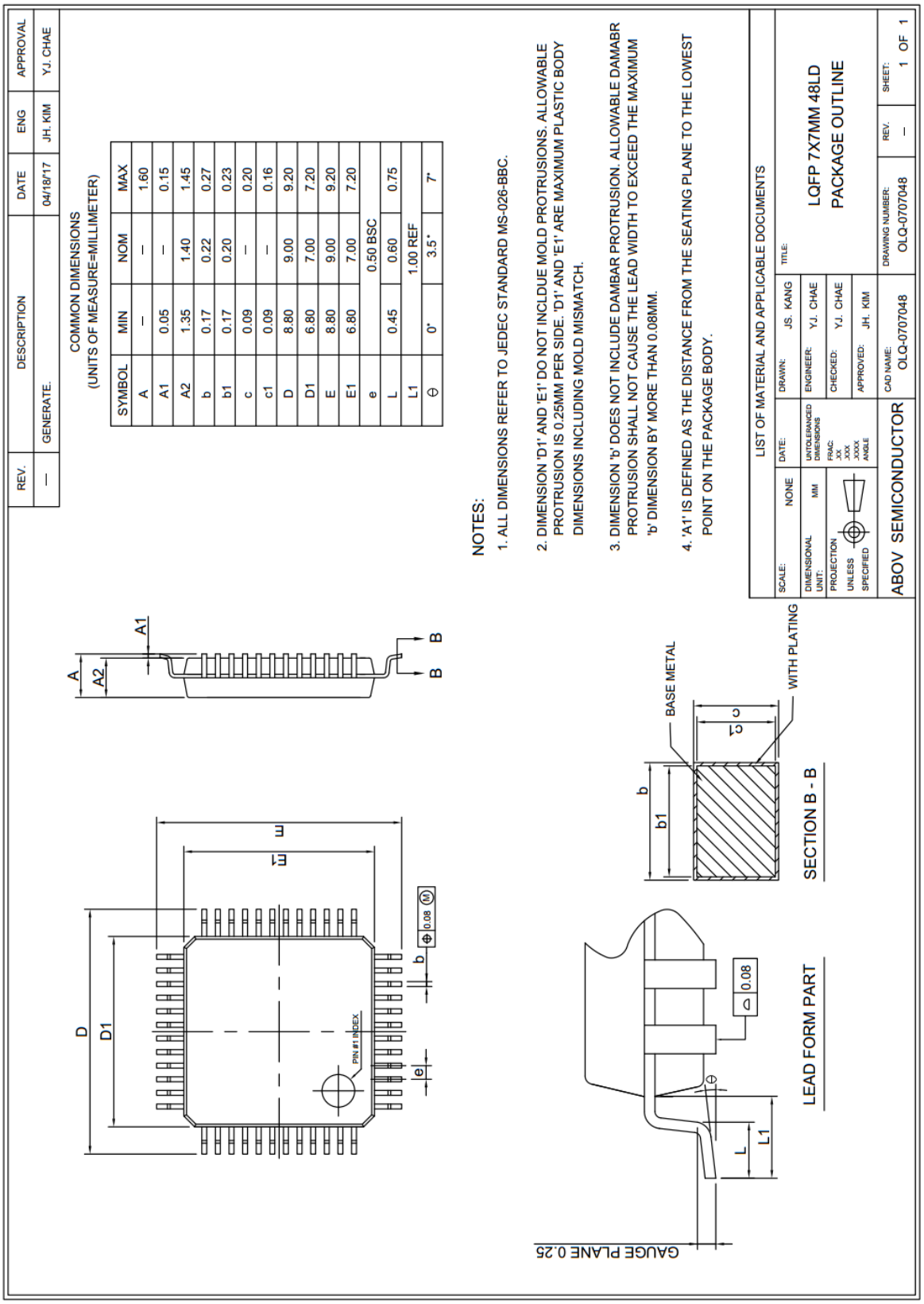


Figure 4.1 48 pin LQFP package

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 an input, a push-pull output, or an open-drain output. A pull-up resistor can be specified in 1-bit unit.	Input	MOSI0
P01				SDA0/MISO0
P02				SCL0/SCK0
P03				EINT2/SS0/RTC1HZ
P04				EINT3
P05				EINT6
P06				PWM2
P07				SS1/PWM3/EINT4
P10	I/O	Port 1 is a bit-programmable I/O port which can be configured as an input, a push-pull output, or an open-drain output. A pull-up resistor can be specified in 1-bit unit.	Input	SCK1/PWM4
P11				MISO1/(RXD1)
P12				MOSI1/(TXD1)
P13				TXD0
P14				RXD0
P15				USS0
P16				XCK0
P17				AN8
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.	Input	CMP1I/AN0/AVref
P21				CMP2I/AN1
P22				AN2
P23				AN3
P24				AN4
P25				AN5
P26				AN6
P27				AN7
P30	I/O	Port 3 is a bit-programmable I/O port which can be configured as a schmitt-trigger input, a push-pull output, or an open-drain output. A pull-up resistor can be specified in 1-bit unit.	Input	USS1
P31				RXD1
P32				TXD1/PWM6
P33				XCK1/PWM5/BUZ
P34				AN9
P35				PWM0
P36				DSCL
P37				DSDA
P40	I/O	Port 4 is a bit-programmable I/O port which can be configured as an input, a push-pull output, or an open-drain output. A pull-up resistor can be specified in 1-bit unit.	Input	SCL1
P41				SDA1
P42				P42
P43				P43
P44				EINT5/PWM1
P45				IR-INT/EINT0
P46				CEC/EINT1
P47				

Table 5.1 Normal Pin description

PIN Name	I/O	Function	@RESET	Shared with
P50	I/O	Port 5 is a bit-programmable I/O port which can be configured as a schmitt-trigger input, a push-pull output, or an open-drain output. A pull-up resistor can be specified in 1-bit unit.	Input	SXI
P51				SXO
P52				EINT7
P53				XIN
P54				XOUT
P55				RESETB
VDD	–	Power input pins	–	–
VSS	–	Power input pins	–	–

Table 5.2 Normal Pin Description (Continued)

Port IO Direction Register(PxIO)	Pull_Up Register (PxPU)	Open Drain Register (PxOD)	**Port output attribute**
ON (1)	OFF (0)	OFF (0)	Push-pull
		ON (1)	Open drain
	ON (1)	OFF (0)	Push-pull
		ON (1)	Internal Pull-Up

Port IO Direction Register	Pull_Up Register	Open Drain Register	**Port Input attribute**
OFF (0)	OFF (0)	X	External condition
	ON (1)	X	Internal Pull-Up

Table 5.3 Port attribute setting (PxIO, PxPU, PxOD, x= Port number (0, 1, 2, 3, 4, 5)

6 Port Structures

6.1 General Purpose I/O Port with Analog Input

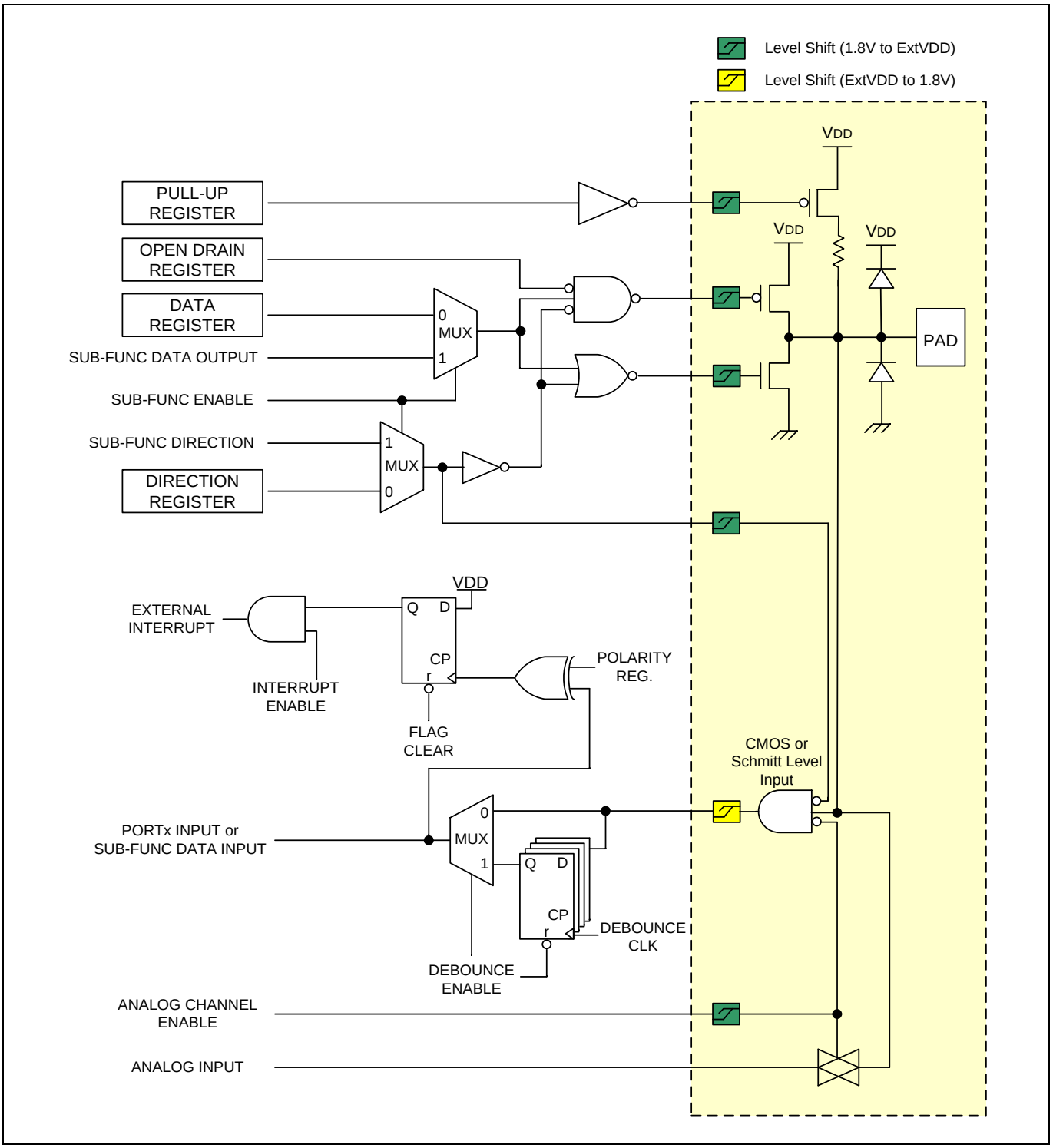


Figure 6.1 General Purpose I/O Port

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}	25	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)

1. 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= 32 ~ 38kHz	Sub Crystal	2.7	–	5.5	V
		f_X= 0.4 ~ 12MHz	Main Crystal	2.7	–	5.5	
		f_X= 1, 2, 4, 8, 16MHz	Internal RC	2.7	–	5.5	
Operating Temperature	T_{OPR}	VDD=2.7~5.5V		-40	–	85	°C

Table 7.2 Recommended Operating Conditions

7.3 A/D Converter Characteristics

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

Parameter	Symbol	Conditions		MIN	TYP	MAX	Unit
Resolution	–	–		–-	12	–	bit
Integral Non-Linear	INL	AVREF= 2.7V – 5.5V ADCCLK= 2MHz		–	–	±6	LSB
Differential Non-Linearity	DNL			–	–	±3	
Zero Offset Error	ZOE			–	–	±5	
Full Scale Error	FSE			–	–	±5	
Conversion Time	t_{CONV}	12-bit resolution, 2MHz		–	30	–	us
Analog Input Voltage	V_{AIN}	–		VSS	–	AVREF	V
Analog Reference Voltage	AVREF	*NOTE 3		2.2	–	VDD	
Analog Input Leakage Current	I_{AIN}	AVREF=5.12V		–	–	2	uA
ADC Operating Current	I_{ADC}	Enable	VDD=5.12V	–	1	2	mA
		Disable		–	–	0.1	uA

Table 7.3 A/D Converter Characteristics

NOTE)

1. Zero offset error is the difference between 0x000 and the converted output for zero input voltage (VSS).
2. Full scale error is the difference between 0xFFF and the converted output for full-scale input voltage (AVREF).
3. When AVREF is lower than 2.7V, the ADC resolution is worse.

7.4 Analog Comparator Characteristics

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
Operating Voltage	AVDD	-	2.2	–	5.5	V
Operating Temperature	T_J	-	-40	25	125	℃
Operating Current	$IDD_{(RMS)}$	-	-	30	50	uA
STOP Current	$ISTOP_{(RMS)}$	0.7uA@105℃, 1.5uA@125℃	-	0.05	1.5	uA
Input Offset Voltage	V_{OS}	-	-50	-	+50	mV
Internal VREF stabilization time	t_{WKUP}	-	50	-	-	us
Propagation Delay(t_{PD})	t_{PHL}	@ Overdirve=50mV, AVDD=3.45V	-	0.5	1.0	us
	t_{PLH}		-	0.5	1.0	

Table 7.4 Analog Comparator Characteristics

7.5 Power-On Reset Characteristics

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

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
RESET Release Level	V_{POR}	–	–	1.4	–	V
VDD Voltage Rising Time	t_R	0.5V to 2.0V	0.05	–	30.0	V/ms
POR Current	I_{POR}	–	–	0.2	–	uA

Table 7.5 Power-on Reset Characteristics

7.6 Low Voltage Reset and Low Voltage Indicator Characteristics

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

Parameter	Symbol	Conditions		MIN	TYP	MAX	Unit
Detection Level	V_{LVR} V_{LVI}	The LVR level is only fixed to Typ. 1.6V but LVI level can be selected to Typ. 2.4V, 2.8V or 3.5V.		–	1.60	1.80	V
				2.20	2.40	2.60	
				2.60	2.80	3.0	
				3.30	3.50	3.70	
Hysteresis	△V	–		–	50	150	mV
Minimum Pulse Width	t_{LW}	–		100	–	–	us
LVR and LVI Current	I_{BL}	Enable	VDD= 3V, RUN Mode	–	30.0	–	uA
		Disable	VDD= 3V	–	–	0.1	

Table 7.6 LVR and LVI Characteristics

7.7 Internal RC Oscillator Characteristics

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

Parameter	Symbol	Conditions		MIN	TYP	MAX	Unit
Frequency	f_{IRC}	V_{DD} = 2.7 – 5.5V		15.76	16	16.24	MHz
Tolerance	–	T_A = 0°C to +50°C	With 0.1uF Bypass capacitor	–	–	±1.5	%
		T_A = -20°C to +85°C				±2.5	
		T_A = -40°C to +85°C				±3.5	
Clock Duty Ratio	T_{OD}	–		40	50	60	%
Stabilization Time	T_{HFS}	–		–	–	100	us
IRC Current	I_{IRC}	Enable		–	0.2	–	mA
		Disable		–	–	0.1	uA

Table 7.7 High Internal RC Oscillator Characteristics

NOTE)

1. A 0.1uF bypass capacitor should be connected to VDD and VSS.

7.8 Internal Watch-Dog Timer RC Oscillator Characteristics

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

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
Frequency	f_{WDTRC}	–	230	256	282	kHz
Stabilization Time	t_{WDTS}	–	–	–	1	ms
WDTRC Current	I_{WDTRC}	Enable	–	5	–	uA
		Disable	–	–	0.1	

Table 7.8 Internal WDTRC Oscillator Characteristics

7.9 DC Characteristics

(T_A = -40°C ~ +85°C, VDD = 2.7V ~ 5.5V, VSS = 0V, f_{XIN} = 8MHz)

Parameter	Symbol	Conditions		MIN	TYP	MAX	Unit
Input High Voltage	V_{IH1}	P0, P1, P2, P3, P4, P5		0.7VDD	–	VDD	V
Input Low Voltage	V_{IL1}	P0, P1, P2, P3, P4, P5		–	–	0.3VDD	V
Output High Voltage	V_{OH1}	VDD= 3.3V, I_{OH}= -4mA, All output ports;		VDD-1.0	–	–	V
	V_{OH2}	VDD= 5.0V, I_{OH}= -10mA, All output ports;		VDD-1.0	–	–	V
Output Low Voltage	V_{OL1}	VDD=3.3V, I_{OL}= 10mA, All output ports;		–	–	1.0	V
	V_{OL2}	VDD=5.0V, I_{OL}= 20mA, 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}	VI=0V, T_A= 25°C All Input ports	VDD=5.0V	25	50	100	kΩ
			VDD=3.0V	50	100	200	
OSC feedback resistor	R_{X1}	XIN= VDD, XOUT= VSS T_A= 25°C, VDD= 5V		700	1000	2500	kΩ
	R_{X2}	SXIN=VDD, SXOUT=VSS T_A= 25°C, VDD=5V		6700	13000	29000	
Supply Current	I_{DD1} (RUN)	f_{XIN}= 12MHz	VDD= 5V±10%	–	4.5	9.0	mA
		f_{XIN}= 12MHz	VDD= 3V±10%	–	3.6	6.0	
		f_{IRC}= 16MHz	VDD= 5V±10%	–	4.4	5.5	
	I_{DD2} (IDLE)	f_{XIN}= 12MHz	VDD= 5V±10%	–	2.0	4.0	mA
		f_{XIN}= 12MHz	VDD= 3V±10%	–	1.2	2.0	
		f_{IRC}= 16MHz	VDD= 5V±10%	–	1.2	3.0	
	I_{DD3} (SLP1)	SLEEP1 MODE WDTRC OSC Enable		–	-	1.0	mA
	I_{DD4} (SLP2)	SLEEP2 MODE WDTRC OSC Enable		–	-	300	uA
	I_{DD5} (STOP1)	STOP1 MODE WDTRC OSC Enable, LVR Disable		–	-	60	uA
	I_{DD6} (STOP2)	STOP2 MODE, T_A=25°C WDTRC OSC Disable, LVR Disable		–	5	10	uA

Table 7.9 DC Characteristics

NOTE)

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

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

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

7.10 AC Characteristics

(T_A = -40°C ~ +85°C, VDD = 2.7V ~ 5.5V)

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
RESETB Input Low Width	t_{RST}	Input, VDD= 5V	10	–	–	us
Interrupt Input High, Low Width	t_{IWH}, t_{IWL}	All interrupt, VDD= 5V	200	–	–	ns
External Counter Input High, Low Pulse Width	t_{ECWH}, t_{ECWL}	ECn, VDD = 5V (n= 0~7)	200	–	–	
External Counter Transition Time	t_{REC}, t_{FEC}	ECn, VDD = 5V (n= 0~7)	20	–	–	

Table 7.10 AC Characteristics

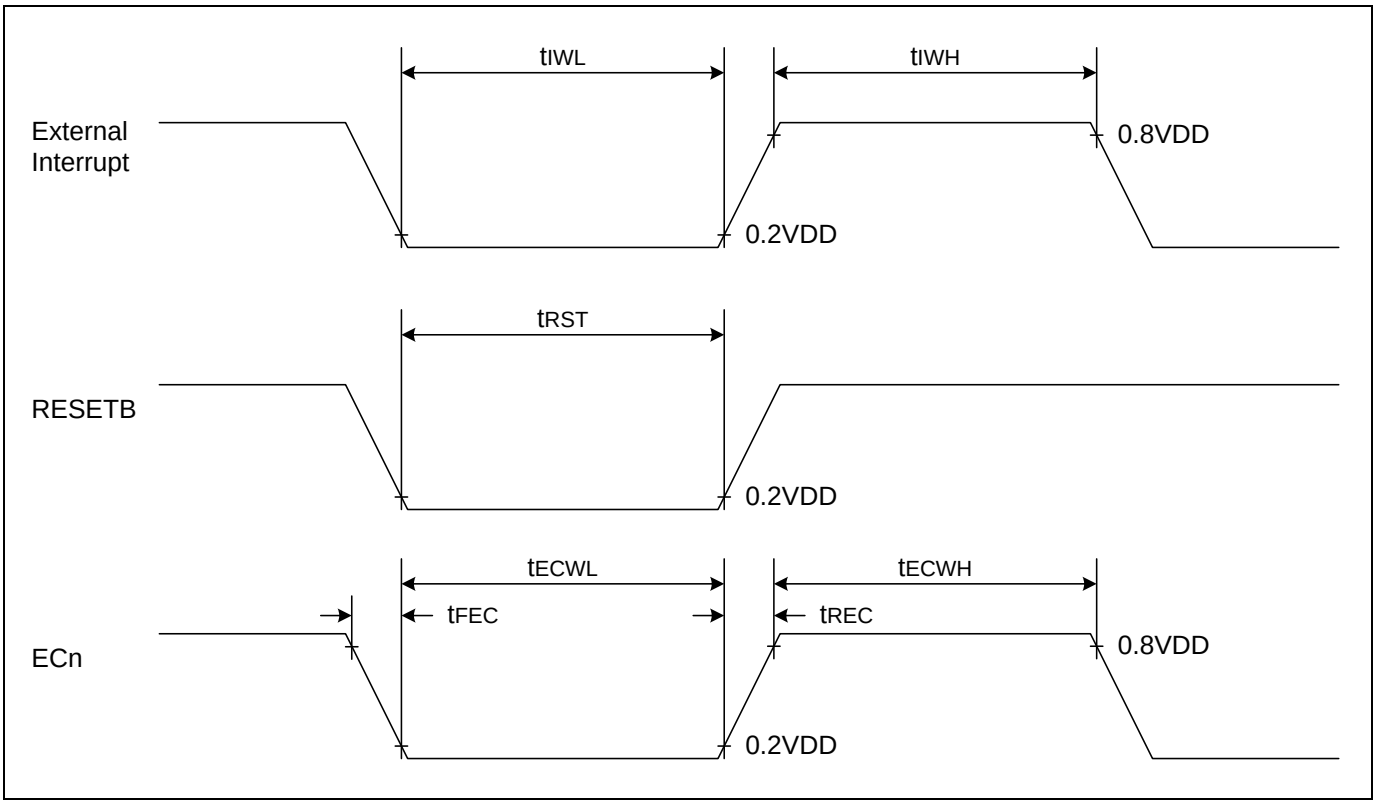


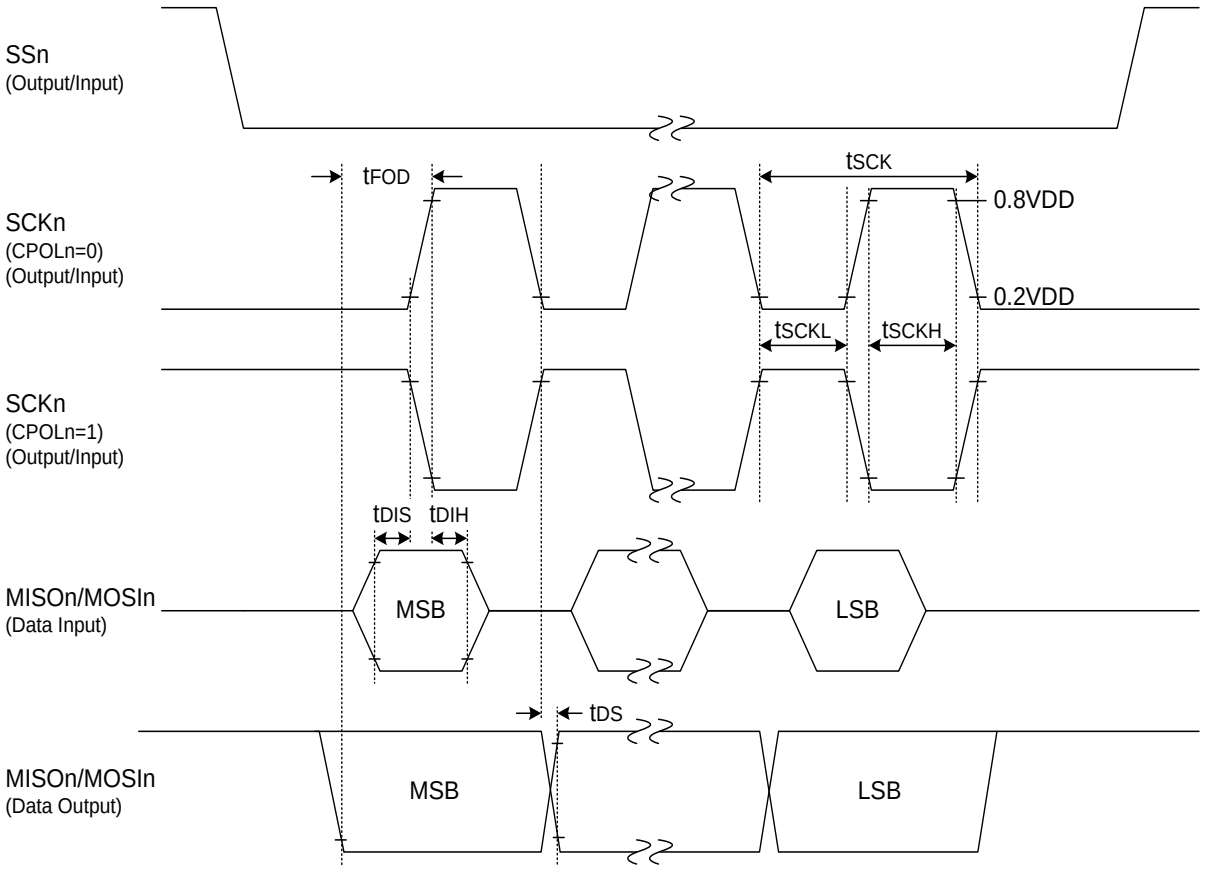
Figure 7.1 AC Timing

7.11 SPI Characteristics

(T_A = -40°C ~ +85°C, VDD = 2.7V ~ 5.5V)

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

Table 7.11 SPI Characteristics



NOTE)

1. n = 0 and 1

Figure 7.2 SPI Timing

7.12 USART Characteristics

(T_A = -40°C ~ +85°C, VDD = 2.7V ~ 5.5V, f_{XIN} = 8MHz)

Parameter	Symbol	MIN	TYP	MAX	Unit
Serial port clock cycle time	t_{SCK}	1800	$t_{CPU} \times 16$	2200	ns
Output data setup to clock rising edge	t_{S1}	810	$t_{CPU} \times 13$	–	
Clock rising edge to input data valid	t_{S2}	–	–	590	
Output data hold after clock rising edge	t_{H1}	t_{CPU} – 50	t_{CPU}	–	
Input data hold after clock rising edge	t_{H2}	0	–	–	
Serial port clock High, Low level width	t_{HIGH}, t_{LOW}	720	$t_{CPU} \times 8$	1280	

Table 7.12 USART Characteristics

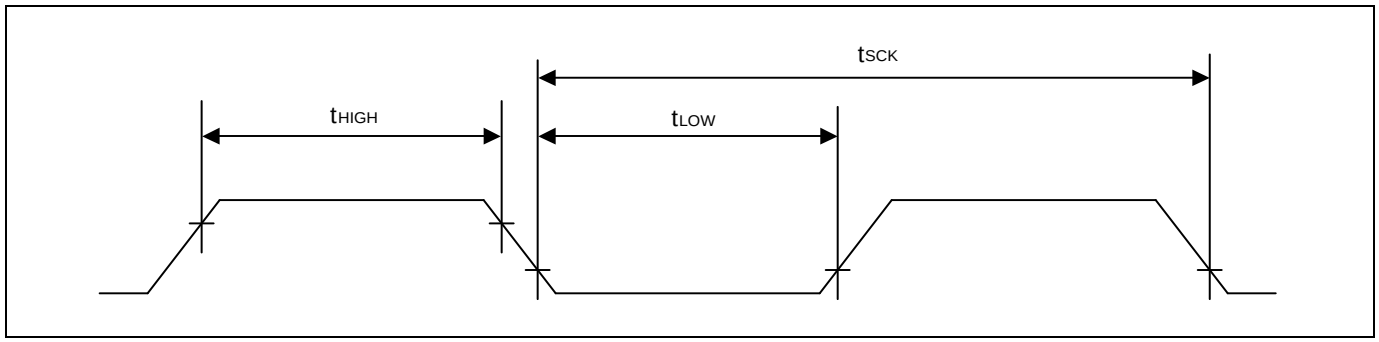


Figure 7.3 Waveform for USART Timing Characteristics

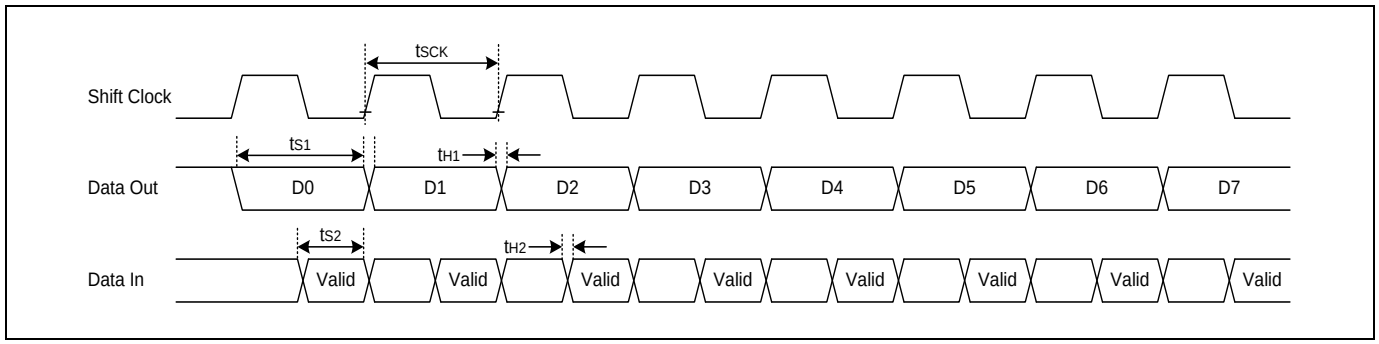


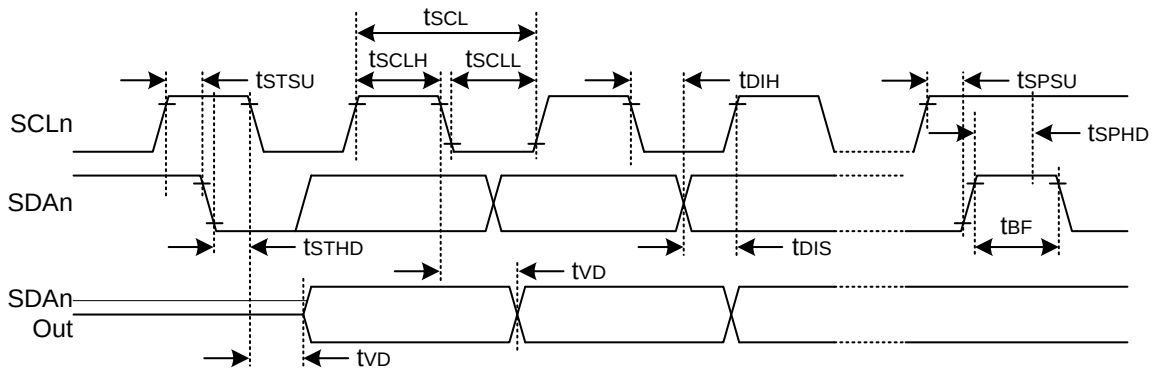
Figure 7.4 Timing Waveform for the USART Module

7.13 I2C Characteristics

(T_A = -40°C ~ +85°C, VDD = 2.7V ~ 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



NOTE)

1. n= 0 and 1

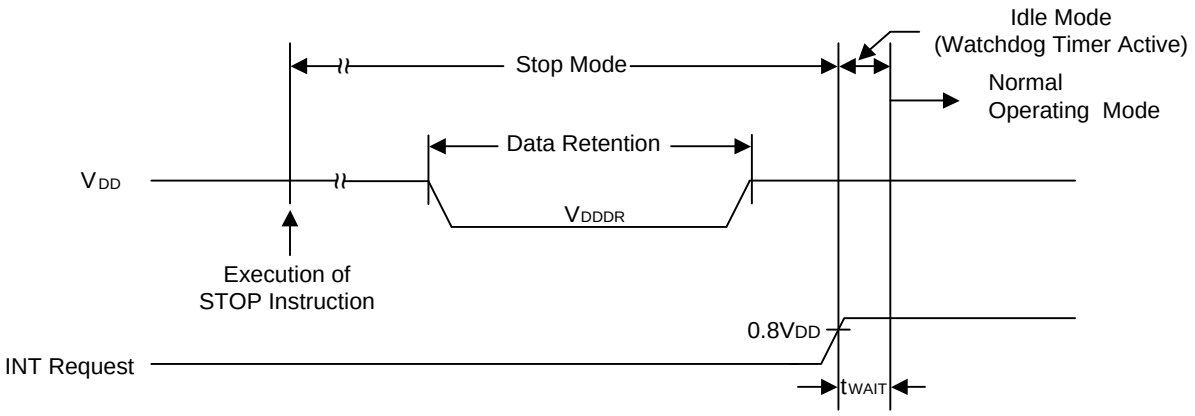
Figure 7.5 I2C Timing(DIS_SDAH=0)

7.14 Data Retention Voltage in Stop Mode

(T_A = -40°C ~ +85°C, VDD = 2.7V ~ 5.5V)

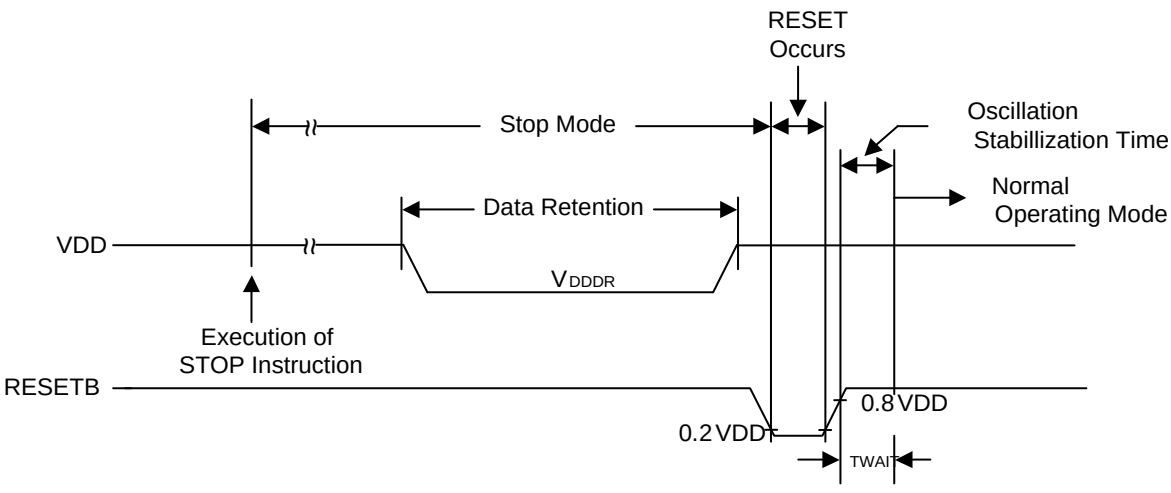
Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
Data retention supply voltage	V_{DDDR}	–	2.7	–	5.5	V
Data retention supply current	I_{DDDR}	V_{DDR} = 2.7V(T_A = 25°C), Stop mode	–	–	1	uA

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 7.6 Stop Mode Release Timing when Initiated by an Interrupt



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

Figure 7.7 Stop Mode Release Timing when Initiated by RESETB

7.15 Internal Flash ROM Characteristics

(T_A = -40°C ~ +85°C, VDD = 2.7V ~ 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
Flash Data Retention Time	t_{RT}	–	10			years

Table 7.15 Internal Flash Rom Characteristics

7.16 Internal EEPROM Characteristics

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

Parameter	Symbol	Condition	MIN	TYP	MAX	Unit
Sector/Byte Write Time	t_{FSW}	–	–	2.5	2.7	ms
Sector/Byte Erase Time	t_{FSE}	–	–	2.5	2.7	
Data Write Protection Time	t_{FHL}	–	–	2.5	2.7	
Page Buffer Reset Time	t_{FBR}	–	–	–	5	us
EEPROM Programming Frequency	f_{PGM}	–	0.4	–	–	MHz
Endurance of Write/Erase	NF_{WE}	–	–	–	300,000	cycles
EEPROM Data Retention Time	t_{RT}	–	10			years

Table 7.16 Internal Flash Rom Characteristics

7.17 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

7.18 Main Clock Oscillator Characteristics

(T_A = -40°C ~ +85°C, VDD = 2.7V ~ 5.5V)

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

Table 7.18 Main Clock Oscillator Characteristics

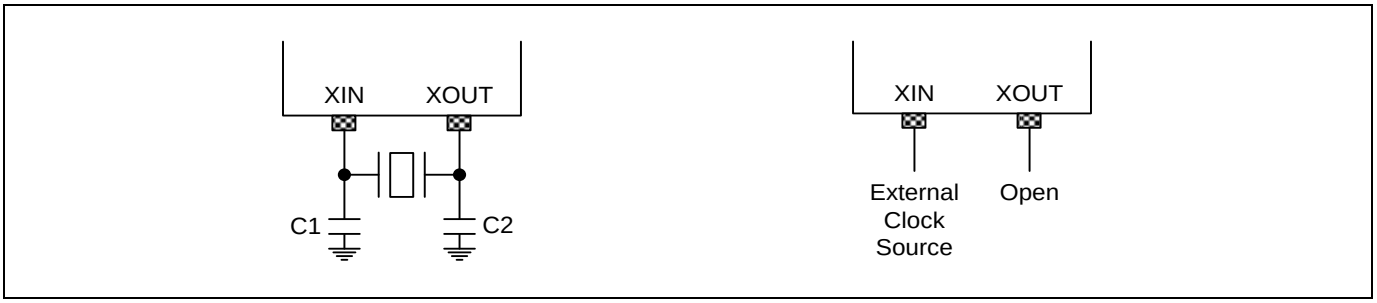


Figure 7.8 Crystal/Ceramic Oscillator & External Clock Circuit

(T_A = -40°C ~ +85°C, VDD = 2.2V ~ 5.5V)

Oscillator	Parameter	MIN	TYP	MAX	Unit
Crystal	fx > 4MHz, VDD = 2.7V ~ 5.5V,	–	–	15	ms
	fx > 1MHz, VDD = 2.7V, T_A=-40°C			60	
Ceramic	-	–	–	10	ms
External Clock	f_{XIN} = 0.4 to 8MHz XIN input high and low width (t_{XH}, t_{XL})	42	–	1250	ns

Table 7.19 Main Oscillation Stabilization Characteristics

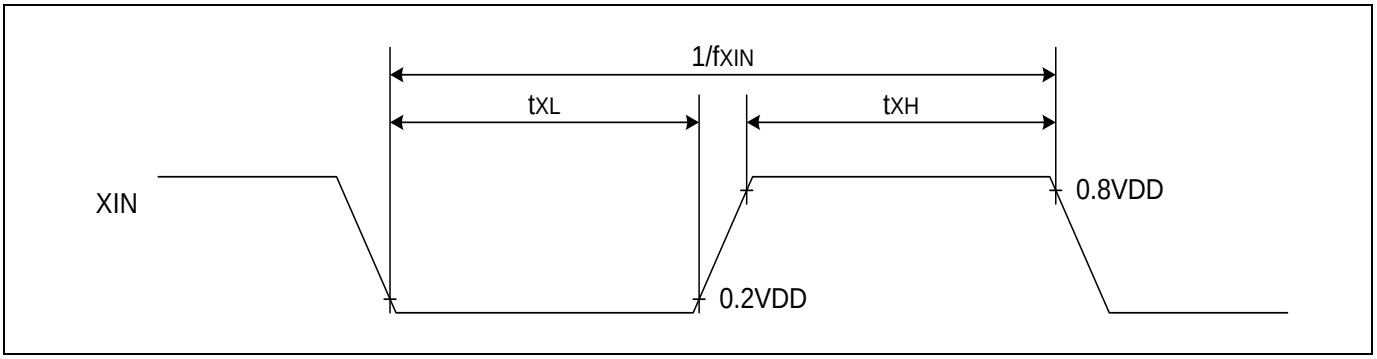


Figure 7.9 Clock Timing Measurement at XIN

7.19 Sub Clock Oscillator Characteristics

(T_A = -40°C ~ +85°C, VDD = 2.7V ~ 5.5V)

Parameter	Symbol	MIN	TYP	MAX	Unit	Condition
Sub oscillation frequency	f_{SUB}	32	32.768	38	kHz	Crystal
SXIN input frequency	f_{SUB}	32	–	100	kHz	External Clock
Operating Current	I_{DD}		3	5	uA	-
External Load Capacitor	C_L	5	15	35	pF	-
Feedback resistance	R_{FB}	6.7	13.7	29	MΩ	
Crystal Input (Low)	V_{IL}			0.2*V_{DDINT}	V	@Max. Current Mode @V_{DDINT} = Typ. 1.85V (on-chip LDO regulator Output)
Crystal Input (High)	V_{IH}	0.8*V_{DDINT}			V	
Crystal Output (Low)	V_{OL}			0.2*V_{DDINT}	V	
Crystal Output (High)	V_{OH}	0.8*V_{DDINT}			V	
Output Clock Duty	D	45	50	55	%	-
Input Swing Level (Peak to Peak)	V_{SW}	1			V	-

Table 7.20 Sub Clock Oscillator Characteristics

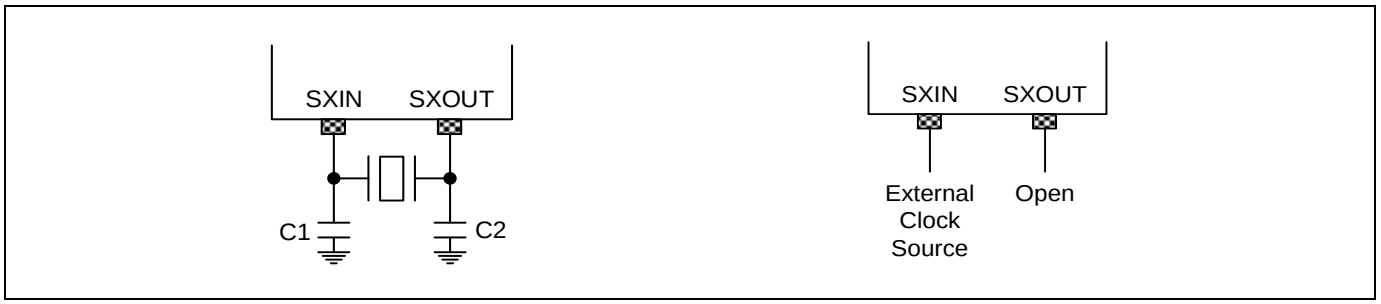


Figure 7.10 Sub Crystal Oscillator & External Clock Cuircit

(T_A = -40°C ~ +85°C, VDD = 2.2V ~ 5.5V)

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

Table 7.1 Sub Oscillation Stabilization Characteristics

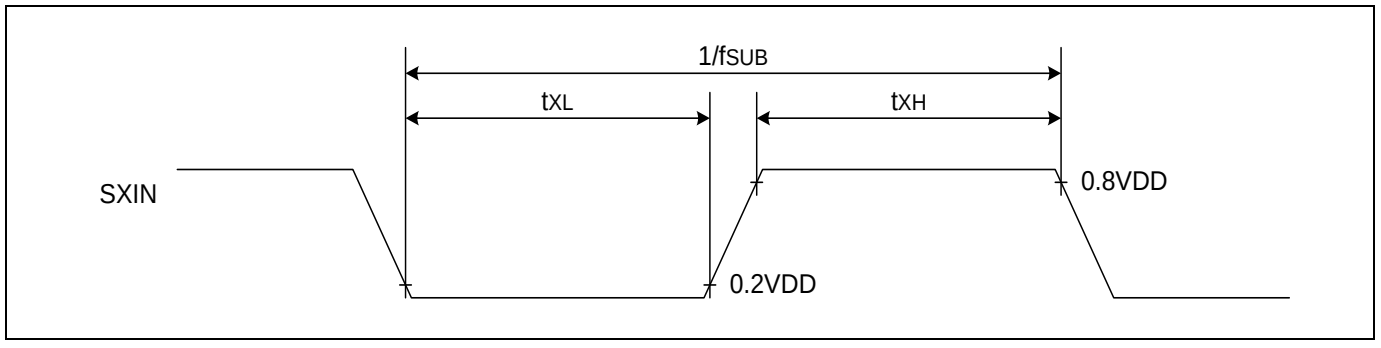


Figure 7.11 Clock Timing Measurement at SXIN

7.20 Operating Voltage Range

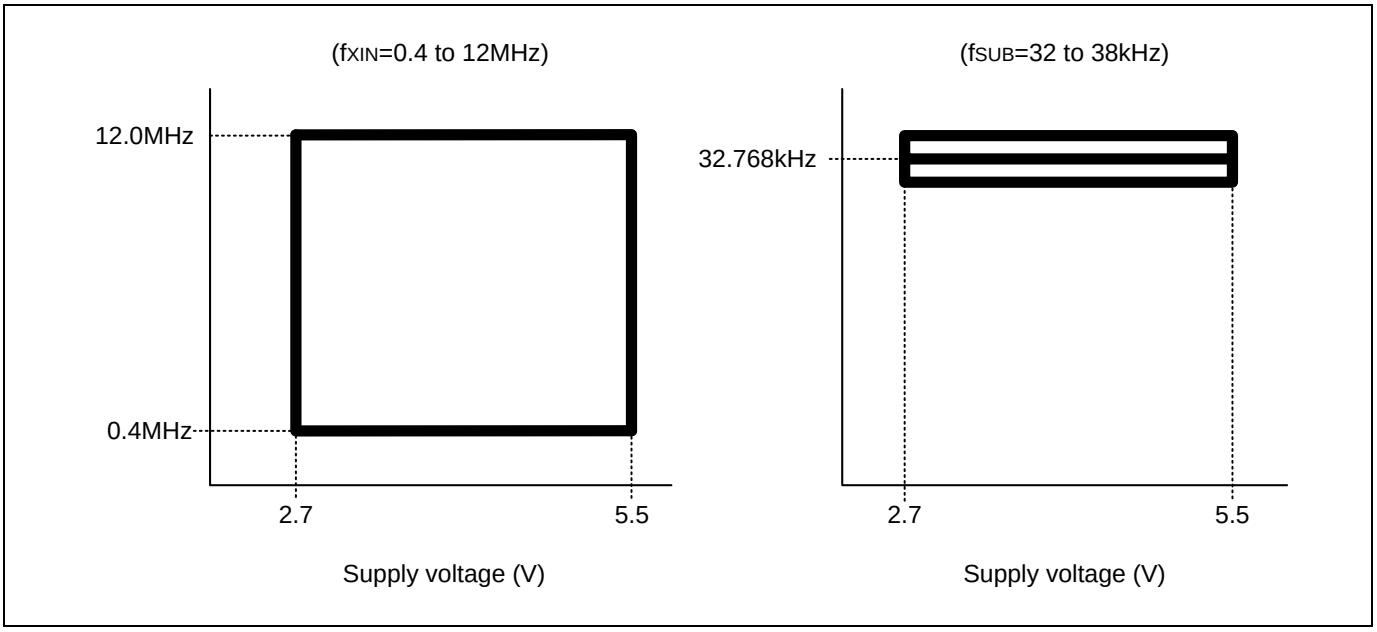


Figure 7.12 Operating Voltage Range

7.21 Recommended Circuit and Layout

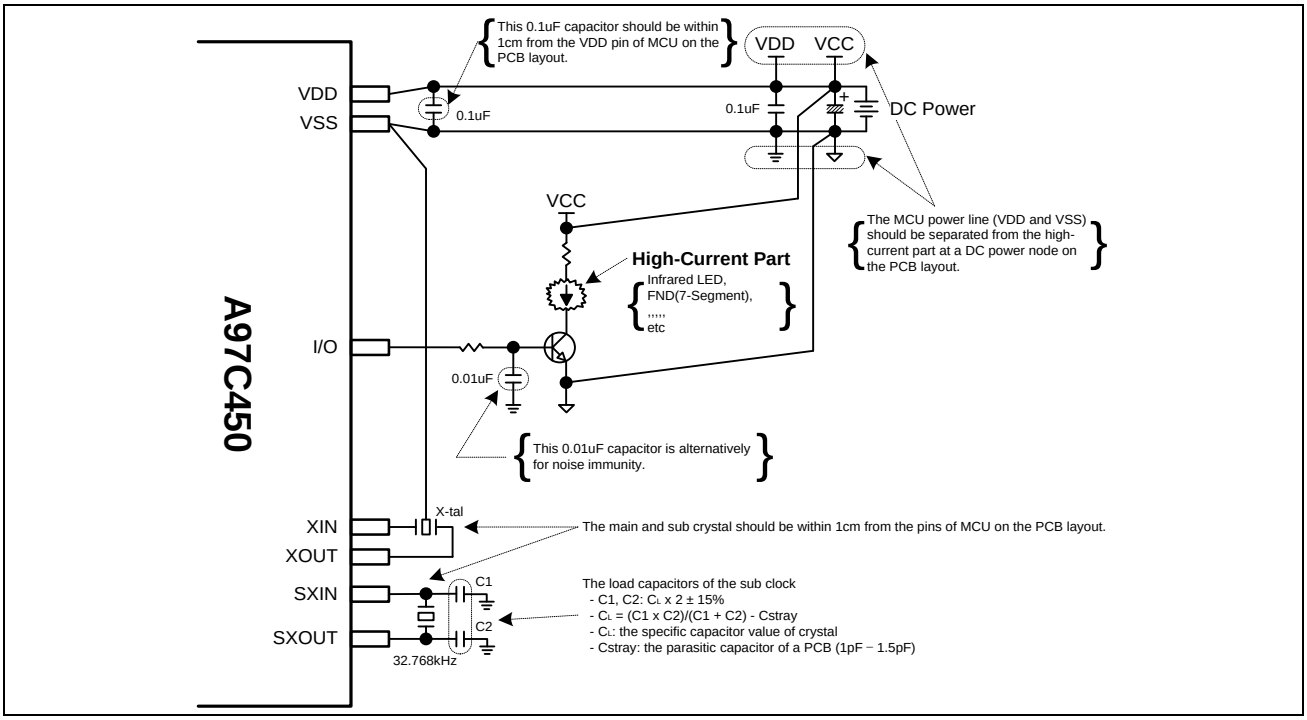
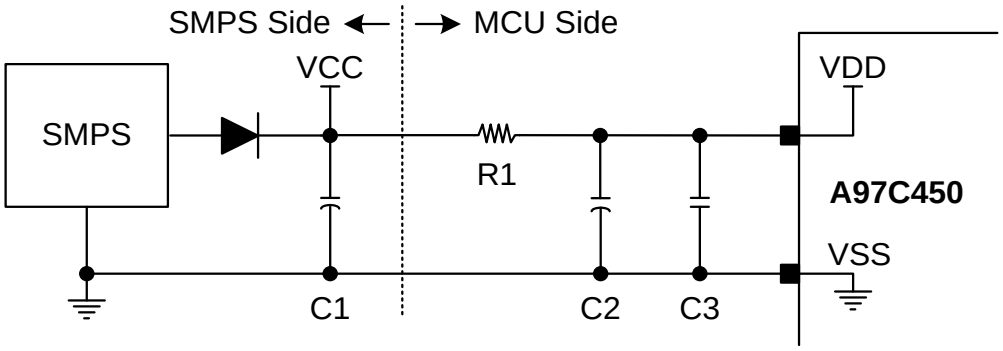


Figure 7.13 Recommended Circuit and Layout

7.22 Recommended Circuit and Layout with SMPS Power



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

Figure 7.14 Recommended Circuit and Layout with SMPS Power

7.23 Typical Characteristics

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

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

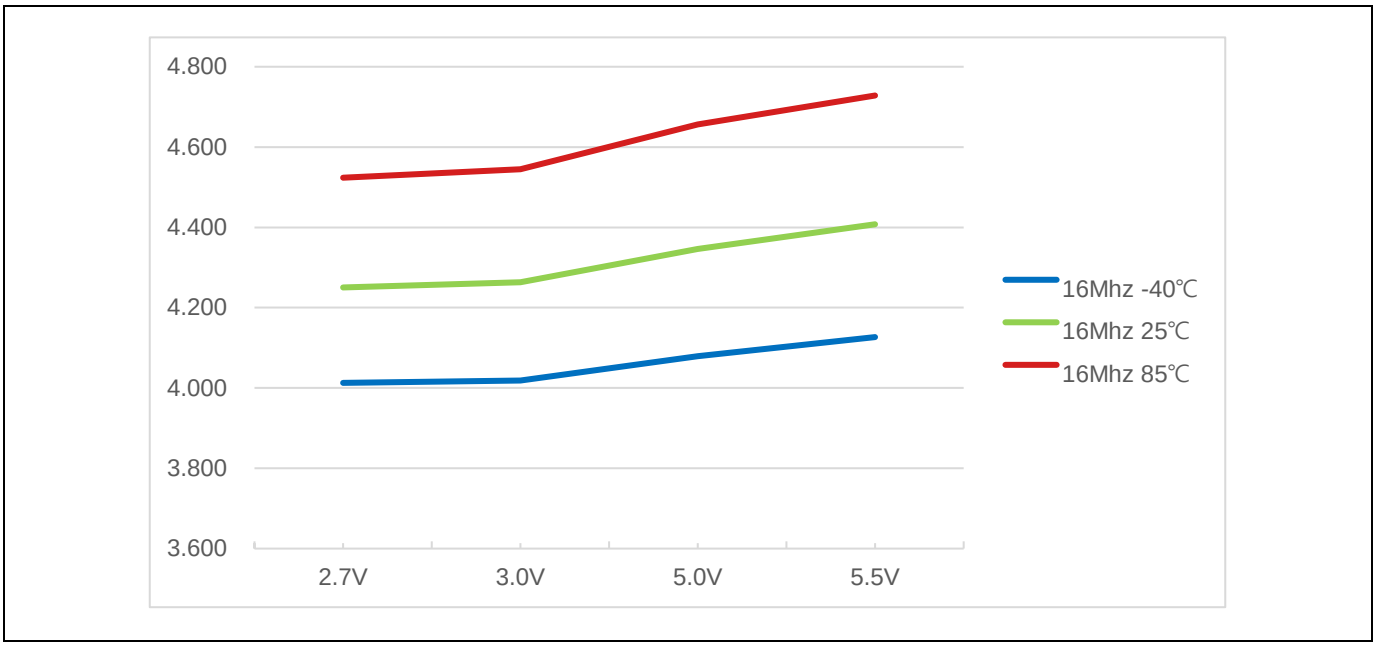


Figure 7.15 RUN (IDD1) Current

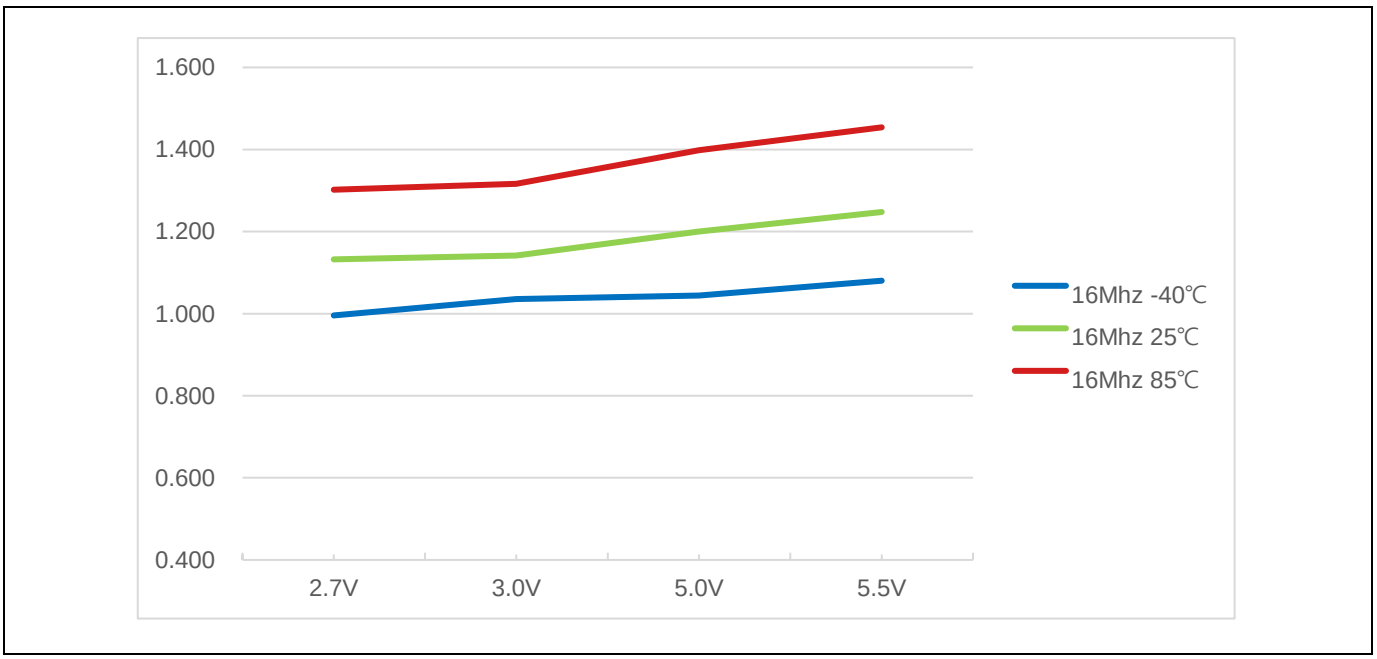


Figure 7.16 IDLE (IDD2) Current

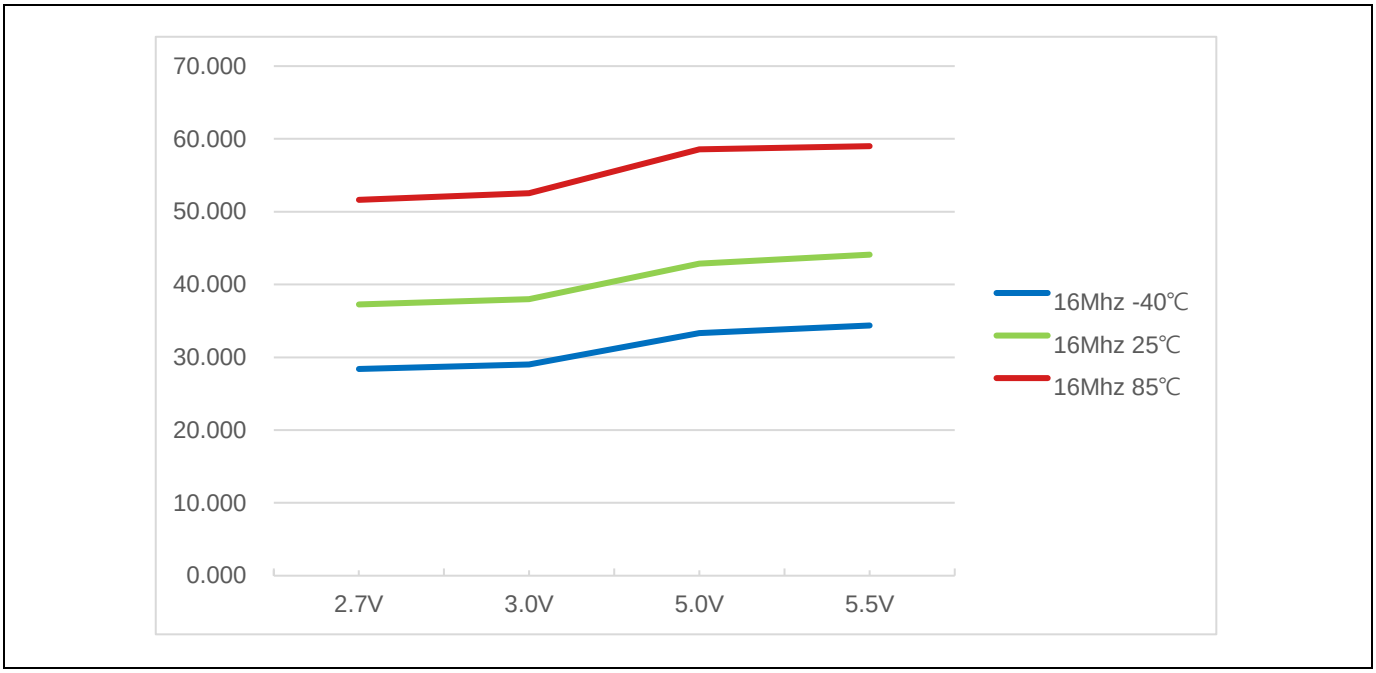


Figure 7.17 STOP1 (IDD5) Current

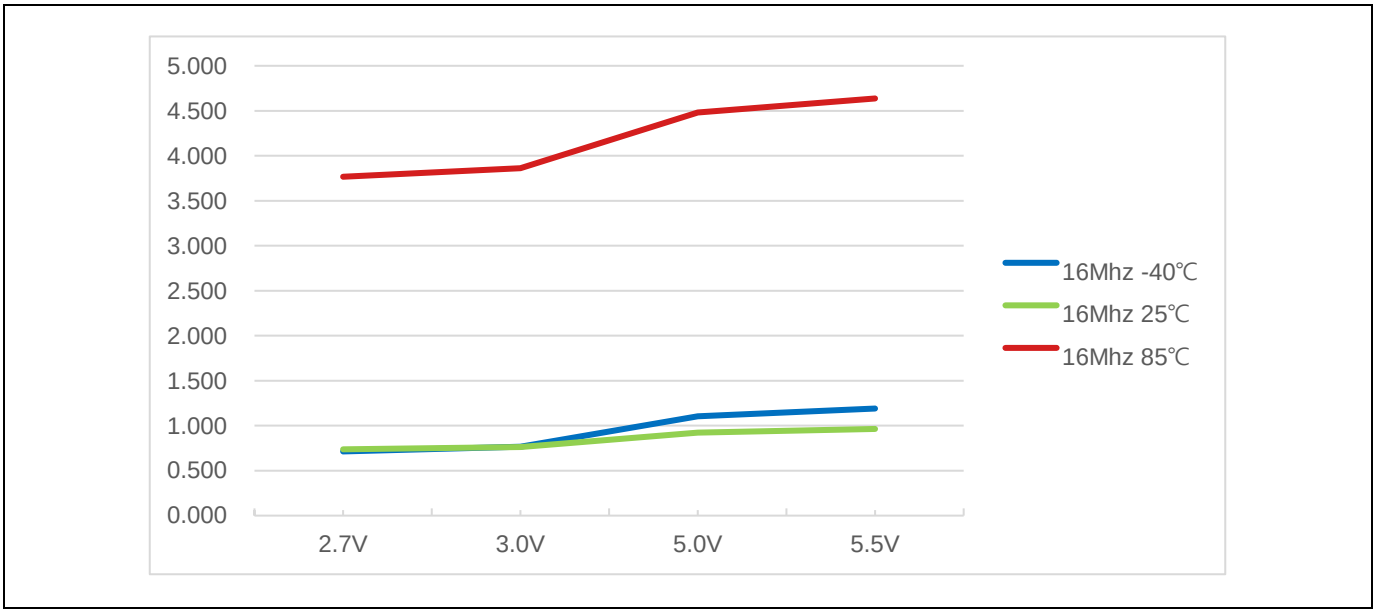


Figure 7.18 STOP2 (IDD6) Current

8 Memory

The A97C450 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.

A97C450 provides on-chip 64KB of the ISP type flash program memory, which can be read and written to. Internal data memory (IRAM) is 256 bytes and it includes the stack area. External data memory (XRAM) is 4,096 bytes and it can be used as the extended stack area.

8.1 Program Memory

A 16-bit program counter is capable of addressing up to 64KB, and this device has 64KB as program memory space.

Figure 8.1 shows the map of the program memory. After reset, the CPU begins execution from address 0000H. Each interrupt is assigned to a fixed address in program memory. The interrupt causes the CPU to jump to that address, where it commences execution of the service routine. External interrupt 0, for example, is assigned to address 000BH. If external interrupt 0 would be requested, its service routine must begin at location 000BH. When the interrupt is not used, the Rom address is used as general purpose program memory. If an interrupt service routine is short enough (as is often case in S/W), it can reside entirely within that 8 bytes interval. Longer service routines can use a jump instruction to skip over subsequent interrupt locations, if other interrupts are in use.

Internal and external data memory is used as program memory which is assigned to address of code bank1. Memory extension registers (MEX1, MEX2, MEX3, MEXSP) are used to access this memory address.

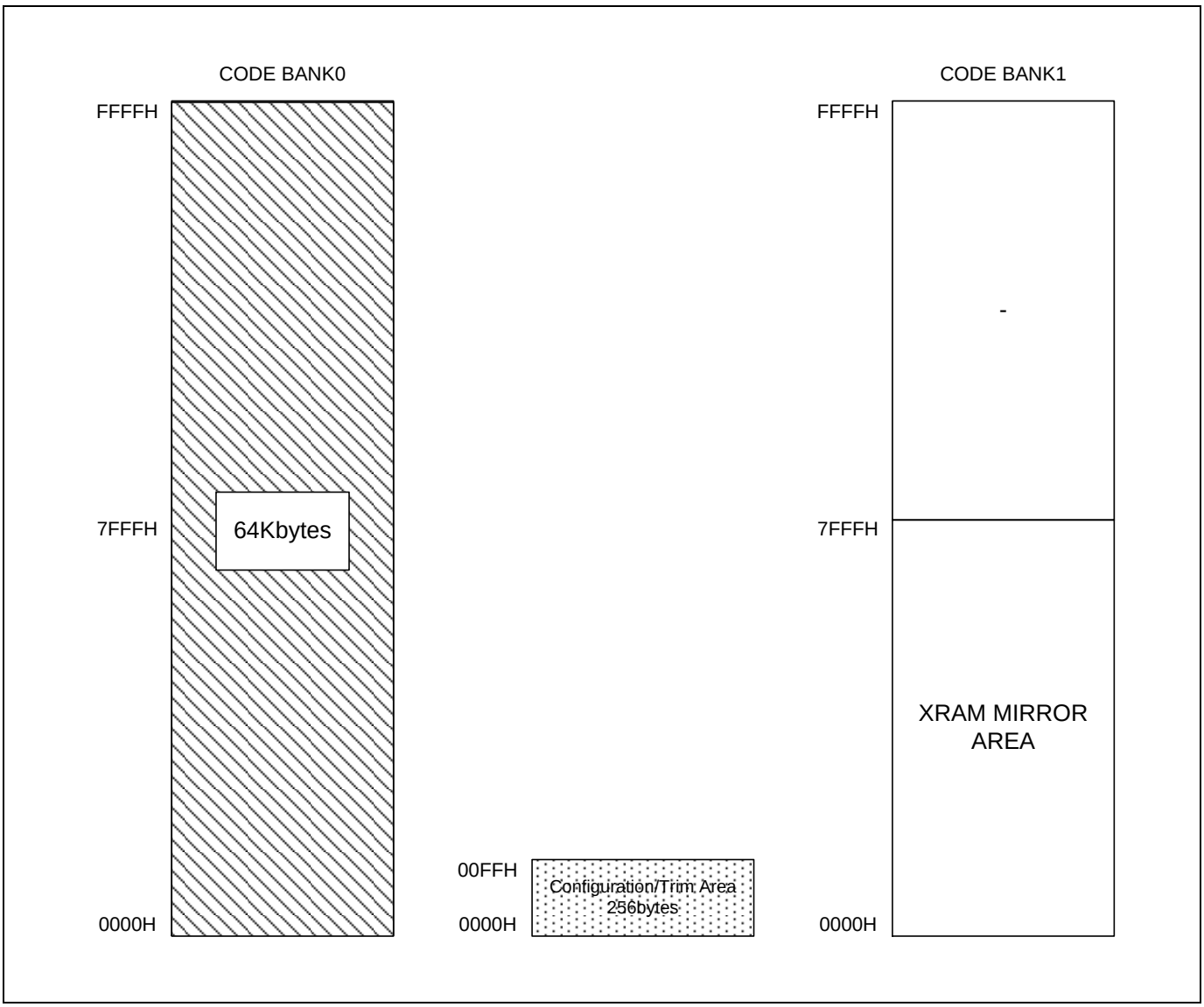


Figure 8.1 Program Memory

NOTE)

1. 64KB Including Interrupt Vector address

8.2 Data Memory

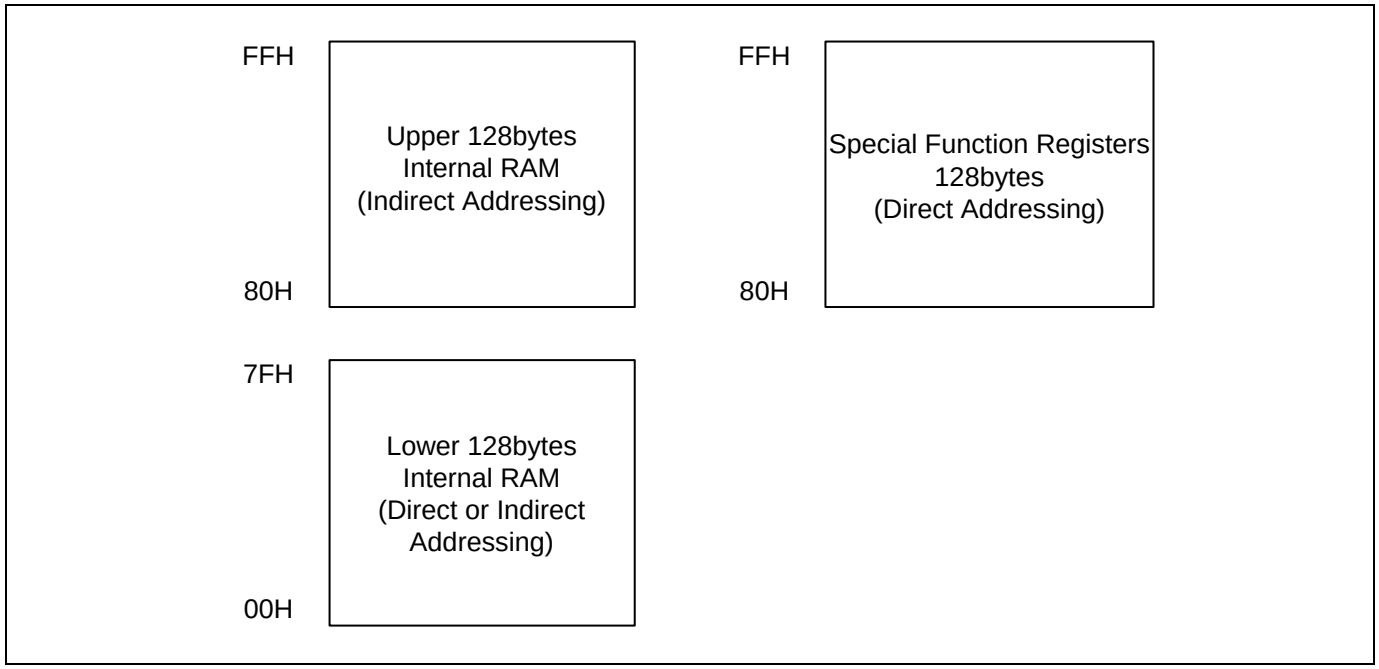


Figure 8.2 Data Memory Map

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 address is one byte wide, which implies an address space of only 256bytes. 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 usage of code space, since register instructions needs less bytes than direct addressing instructions.

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.

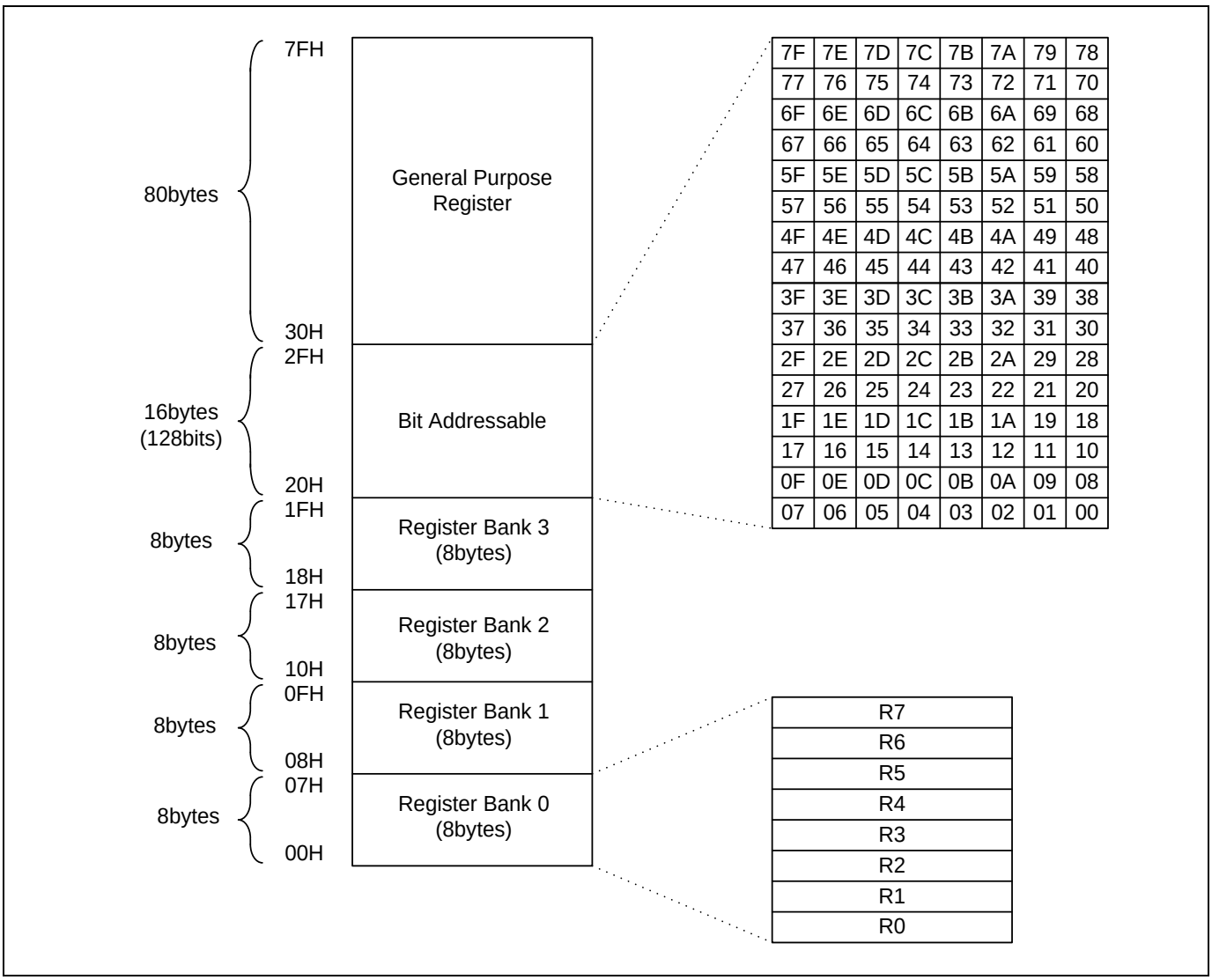


Figure 8.3 Lower 128bytes RAM

8.3 External Data Memory

A97C450 has 2Kbytes EEPROM data memory and 4,096 bytes XRAM and XSFR. This area has no relation with RAM/FLASH. It can be read and written to through SFR with 8-bit unit.

The 2Kbytes data EEPROM is assigned to address from A000H to A7FFH of external memory.

For more information about EEPROM data memory, see chapter15.

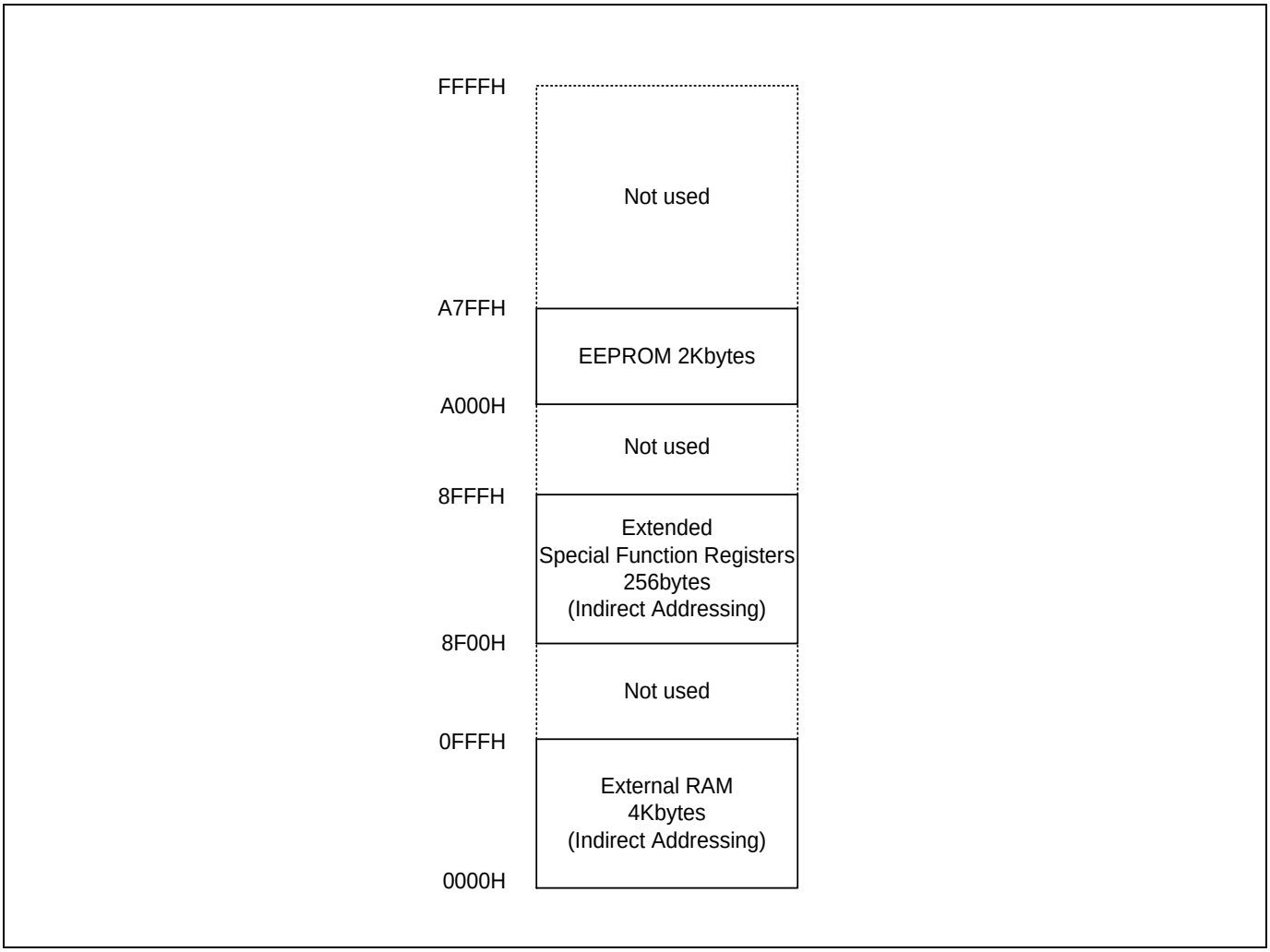


Figure 8.4 XDATA Memory Area

8.4 Extended Stack Pointer(XSP)

The XSP register is a high of stack pointer when XSPEN bit of the XSPCR register is set. In this case stack is located in XDATA memory and the maximum size of stack is equal to 64KB. If the XSPEN bit is set to '0', the XSP register is ignored and stack is located in internal memory.

8.5 Memory Extension Stack

The memory extension stack (MXSTAKCK) store code bank address. When executing the LCALL instruction it stores the code bank address, when executing the RET and RETI instructions exports the code bank address. The MXSTACK is capable of stack 128bit of program address.

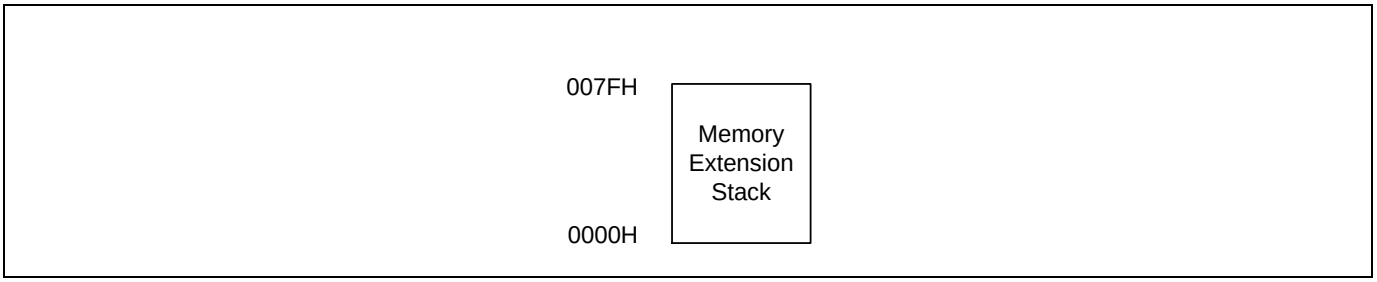


Figure 8.5 Memory extension stack

The memory extension stack (MXSTACK) store code bank address. When executing the LCALL instruction it stores the code bank address, when executing the RET and RETI instructions exports the code bank address. The MXSTACK is capable of stack 128bit of program address.
According to the following example, stack pointer memory diagram is shown below.

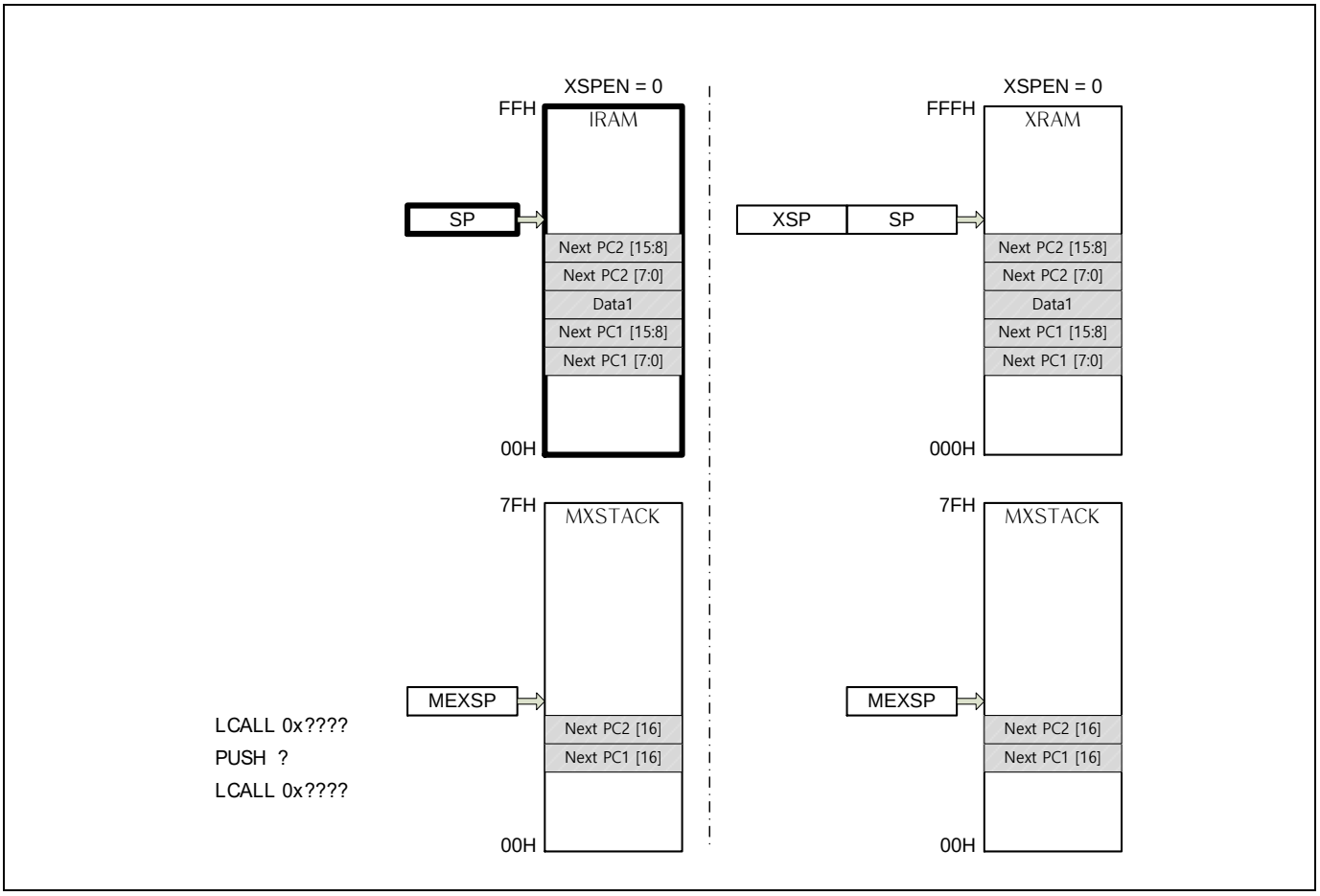


Figure 9.1 Stack pointer memory map

8.6 Bootloader Swap

The A97C450 support Bootloader swap operation for the area selected by write protection selection register (BSIZE). Setting config option enables bootloader swapping. After downloading the bootloader to be changed to the desired boot page and executing boot swap command and resetting, the changed bootloader is seen from address 0000H. Refer to the following example.

Step 1: Jump Boot page 0
Step 2: Enter program mode
Step 3: Update boot page 1
Step 4: Set the OTP boot page selection bit
Step 5: Reset
Step 6: During reset boot page 1 is changed to boot page 0
Step 7: Execute code at the updated boot page 0

FUSE_SWAPEN = 0x5A : Enable bootloader swap.
FUSE_BTSEL = 0xA5 : Select Boot page 1 (default select boot page 0)

Name	Boot page 0 area	Boot page 1 area	Size
BSIZE = 0x0	0x0000~0x01FF	0x0200~0x03FF	512B
BSIZE = 0x1	0x0000~0x03FF	0x0400~0x07FF	1KB
BSIZE = 0x2	0x0000~0x07FF	0x0800~0x0FFF	2KB
BSIZE = 0x3	0x0000~0x0FFF	0x1000~0x1FFF	4KB

Table 8.1 Bootloader area

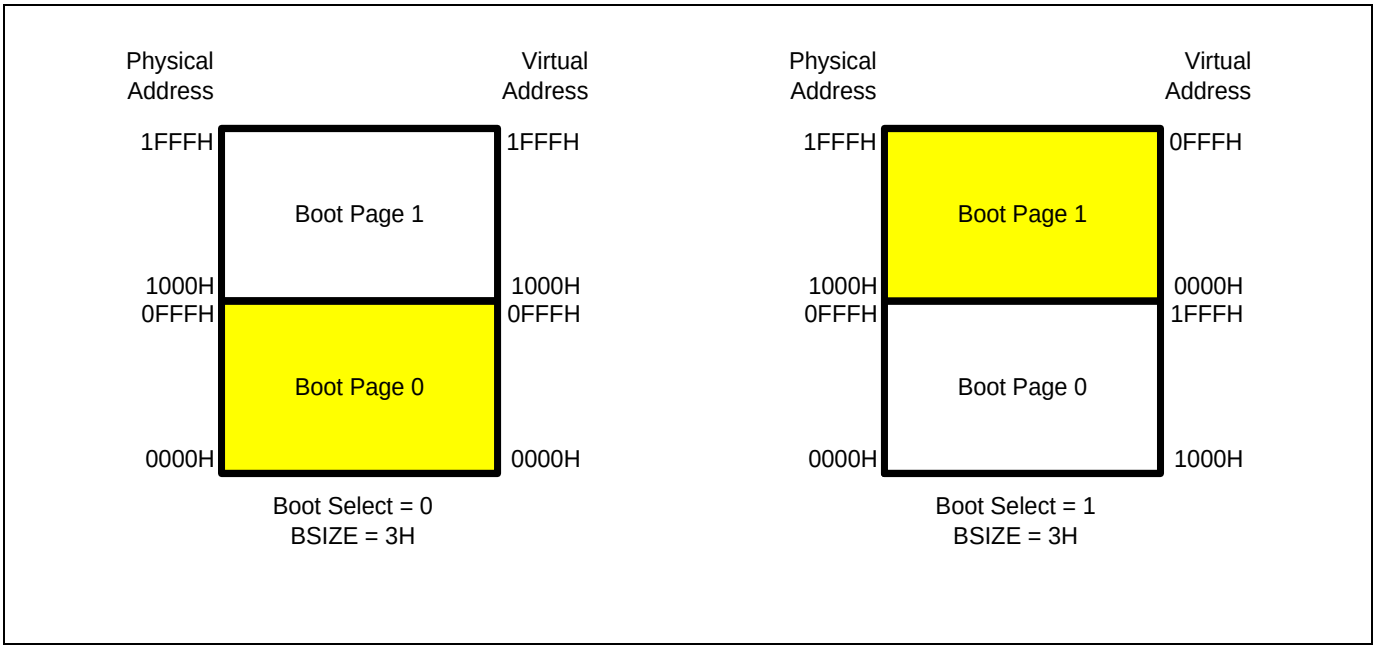


Figure 9.2 Bootloader selection

8.7 SFR Map

8.7.1 SFR Map Summary

- Reserved
- M8051 compatible

	0H/8H	1H/9H	2H/AH	3H/BH	4H/CH	5H/DH	6H/EH	7H/FH
F8H	T6BDRL	T6BDRH	UCTRL11	UCTRL12	UCTRL13	USTAT1	UBAUD1	UDATA1
F0H	B	T6CRL	FEARL	FEARM	FEARH	FEDR	-	T6CRH
E8H	SPICR1	SPIDR1	FEMR	FECR	FESR	FETCR	T6ADRL	T6ADRH
E0H	ACC	SPISR1	UCTRL01	UCTRL02	UCTRL03	USTAT0	UBAUD0	UDATA0
D8H	SPICR0	SPIDR0	I2CMR00	I2CSR0	I2CSCLLR0	I2CSCLHR0	I2CSDAHR0	I2CDR0
D0H	PSW	SPISR0	T5ADRL	T5ADRH	T5BDRL	T5BDRH	I2CSAR01	I2CSAR00
C8H	T4CRL	T4CRH	T4ADRL	T4ADRH	T4BDRL	T4BDRH	T5CRL	T5CRH
C0H	T2BDRL	T2BDRH	T3CRL	T3CRH	T3ADRL	T3ADRH	T3BDRL	T3BDRH
B8H	VDCMOD	P5IO	T1DR	T1CR	T2CRL	T2CRH	T2ADRL	T2ADRH
B0H	P5	P4IO	TINTCR	T0CNT	T0DR	T0CR	TIFLAG0	T1CNT
A8H	IE	IE1	IE2	IE3	IE4	XSP	XSPCR	AUTHORITY
A0H	P4	P3IO	EO	EIENAB	EIFLAG	EIEDGE	EIPOLA	EIBOTH
98H	P3	P2IO	ADCM	ADCRL	ADCRH	WTMR	WTR	BUZCR
90H	P2	P1IO	ADCM2	OTPADD	MEX1	MEX2	MEX3	MEXSP
88H	P1	P0IO	SCCR	BCCR	BITR	WDTMR	WDTR	BUZDR
80H	P0	SP	DPL	DPH	XBANK	RSFR	LVIR	PCON

Table 9.1 SFR Map Summary

NOTE)

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

	0H/8H	1H/9H	2H/AH	3H/BH	4H/CH	5H/DH	6H/EH	7H/FH
8FF8H	-	-	-	-	-	-	-	-
8FF0H	-	-	-	-	-	-	-	-
8FE8H	-	CRC_ADDR_START_H	CRC_ADDR_START_M	CRC_ADDR_START_L	CRC_ADDR_END_H	CRC_ADDR_END_M	CRC_ADDR_END_L	-
8FE0H	CRC_CON	-	CRC_H	CRC_L	CRC_MNT_H	CRC_MNT_L	-	-
8FD8H	SPMUSR	PADDRH	PADDRM	PADDRL	STPTRH	STPTRL	XSTPTR	
8FD0H	SPMUCR	PADDR_OVFH	PADDR_OVFM	PADDR_OVFL	STPTR_OVFH	STPTR_OVFL	STPTR_UDFH	STPTR_UDFL
8FC8H	UCTRL14	FPCR1	-	-	-	-	-	-
8FC0H	UCTRL04	FPCR0	-	-	-	-	-	-
8FB8H	WEEK	MONTH	YEAR	ALARMWM	ALARMWH	ALARMWW	-	-
8FB0H	-	RTCC0	RTCC1	ERRCOR	SEC	MIN	HOUR	DAY
8FA8H	IP4L	IP4H	INT_OFFSET	-	-	-	-	-
8FA0H	IP0L	IP0H	IP1L	IP1H	IP2L	IP2H	IP3L	IP3H
8F98H	-	-	-	-	-	-	-	-
8F90H	CEC_RXD_17	CEC_RXD_18	CEC_RXD_19	-	-	-	-	-
8F88H	CEC_RXD_9	CEC_RXD_10	CEC_RXD_11	CEC_RXD_12	CEC_RXD_13	CEC_RXD_14	CEC_RXD_15	CEC_RXD_16
8F80H	CEC_RXD_1	CEC_RXD_2	CEC_RXD_3	CEC_RXD_4	CEC_RXD_5	CEC_RXD_6	CEC_RXD_7	CEC_RXD_8
8F78H	CEC_TXD_13	CEC_TXD_14	CEC_TXD_15	CEC_TXD_16	CEC_TXD_17	CEC_TXD_18	CEC_TXD_19	CEC_RXH
8F70H	CEC_TXD_5	CEC_TXD_6	CEC_TXD_7	CEC_TXD_8	CEC_TXD_9	CEC_TXD_10	CEC_TXD_11	CEC_TXD_12
8F68H	CEC_FSTAT	CEC_ICLR	CEC_FCLR	CEC_TXH	CEC_TXD_1	CEC_TXD_2	CEC_TXD_3	CEC_TXD_4
8F60H	CEC_PRES1	CEC_PRES0	CEC_CONF1	CEC_CONF0	CEC_GCTRL	CEC_ICTRL	CEC_FCTRL	CEC_ISTAT
8F58H	-	-	-	-	-	-	-	-
8F50H	-	-	-	-	-	-	-	-
8F48H	IRC_PRES1	IRC_PRES0	IRC_FRMP1	IRC_FRMP0	IRC_CONF	IRC_CTRL	IRC_EDGE1	IRC_EDGE0
8F40H	PSR1	PSR2	-	-	-	-	-	-
8F38H	-	-	-	-	-	-	-	-
8F30H	COMPCON	COMPVREFSEL	COMPST	COMPDBCNT	-	-	-	-
8F28H	BISCCON	INTCNTH	INTCNTL	XTLCNTH	XTLCNTL		-	-
8F20H	I2CMR10	I2CSR1	I2CSCLLR1	I2CSCLHR1	I2CSDAHR1	I2CDR1	I2CSAR10	I2CSAR11
8F18H	P0DB	P1DB	P2DB	P3DB	P4DB	P5DB	I2CMR01	I2CMR11
8F10H	P4OD	P5OD	-	-	-	-	-	-
8F08H	IODBCNT	-	-	-	P0OD	P1OD	P2OD	P3OD
8F00H	P0PU	P1PU	P2PU	P3PU	P4PU	P5PU	-	-

Table 9.2 XSFR Map Summary

8.7.2 SFR Map

ACC (Accumulator Register) : E0H

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

Initial value : 00H

ACC Accumulator

B (B Register) : F0H

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

Initial value : 00H

B B Register

SP (Stack Pointer) : 81H

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

Initial value : 07H

SP Stack Pointer

XSP (Extend Stack Pointer) : ADH

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

Initial value : 00H

XSP Extended Stack Pointer

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

XSPCR (Extended Stack Pointer Control Register) : AEH

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

Initial value : 00H

SP Extended Stack Pointer Enable/Disable

0 Disable

1 Enable

DPL (Data Pointer Register Low) : 82H

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

Initial value : 00H

DPL Data Pointer Register Low Byte

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 Register High Byte

PSW (Program Status Word) : D0H

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

Initial value : 00H

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

EO (Extended Operation Register) : A2H

7	6	5	4	3	2	1	0
-	-	-	TRAP_EN	-	DPSEL.2	DPSEL.1	DPSEL.0
R	R	R	R/W	R	R/W	R/W	R/W

Initial value : 00H

TRAP_EN Select the instruction

0 Select MOVC @(DPTR++), A

1 Select Software TRAP instruction

DPSEL[2:0] Select Banked Data Point Register

DPSEL2	DPSEL1	DPSEL0	
0	0	0	DPTR0
0	0	1	DPTR1
Reserved			-

XBANK (XRAM Bank Pointer) : 8CH

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

Initial value : 00H

XBANK XRAM Bank Pointer

NOTE)

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

AUTHORITY (AUTHORITY register) : AFH

7	6	5	4	3	2	1	0
AUTHORITY							
R	R	R	R	R	R	R	R

Initial value : 00H

AUTHORITY The following registers can be written only when authorized.
The registers are SCCR, LVIR, BCCR, RSFR, VDCMOD.

0 Unauthorized

1 Authorized

MEX1 (Memory Extension register 1) : 94H

7	6	5	4	3	2	1	0
CB19	CB18	CB17	CB16	NB19	NB18	NB17	NB16
R	R	R	R	R/W	R/W	R/W	R/W

Initial value : 00H

This register records the 'current' and 'next' memory bank numbers for Program code

CB19-CB16 Current Bank

NB19-NB16 Next Bank

MEX2 (Memory Extension register 2) : 95H

7	6	5	4	3	2	1	0
MCM	MCB18	MCB17	MCB16	IB19	IB18	IB17	IB16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

This register controls the current memory bank numbers for interrupt service routine code and for memory constants

MCM Memory Constant Mode. Set to '1' when Memory Bank used.

MCB18-MCB16 Memory Constant Bank (with MEX3.7)

IB19-IB16 Interrupt Bank

MEX3 (Memory Extension register 3) : 95H

7	6	5	4	3	2	1	0
MCB19	UB1	UB0	MXB19	MXM	MX18	MX17	MX16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

This register chiefly controls the current memory bank number for external data memory.

MCB19	Memory Constant Bank MSB. See MEX2.
UB3, UB4	Bits available to the user.
MXM	XRAM Bank selector. When set to '1', the MOVX Bank bits MX19–MX16 are used as XRAMA19–16 instead of the Current Bank (CB)
MX19-MX16	XRAM Bank

MEXSP (Memory Extension Stack Pointer) : 97H

7	6	5	4	3	2	1	0
0	Memory Extension Stack Pointer						
-	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 7FH

This register is the Memory Extension Stack Pointer. It provides for a stack depth of up to 128 bytes (Bit 7 is always 0). It is pre-incremented by call instructions and post-decremented by return instructions.

9 I/O Ports

9.1 I/O Ports

The A97C450 has six groups of I/O ports (P0 ~ P5). Each port can be easily configured by software as I/O pin, internal pull up and open-drain pin to meet various system configurations and design requirements. Also P0 includes function that can generate interrupt according to change of state of the pin.

9.2 Port Register

9.2.1 Data Register (Px)

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

9.2.2 Direction Register (PxIO)

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

9.2.3 Pull-up Resistor Selection Register (PxPU)

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

9.2.4 Open-drain Selection Register (PxOD)

There are internally open-drain selection registers (PxOD) for P0 ~ P4 and a bit for P5. The open-drain selection register controls the open-drain enable/disable of each port. Almost ports become push-pull by a system reset, but some ports become open-drain by a system reset.

9.2.5 De-bounce Enable Register (PxDB)

P0 ~ P5 support debounce function. Debounce time of each ports has about 4us.

9.2.6 Port Selection Register 1 (PSR1)

PSR1 register select ports P2 attribute to analog-to-digital converter (A/D)s or general I/O port.

PSR1 registers prevent the input leakage current when ports are connected to analog inputs. If the bit of PSR1 is ‘1’, the dynamic current path of the Schmitt OR gate of the port is cut off and the digital input of the corresponding port is always ‘0’.

9.2.7 Port Selection Register 2 (PSR2)

PSR2[1:0] register select ports P17, P34 attribute to analog-to-digital converter (A/D)s or general I/O port. PSR2[1:0] registers prevent the input leakage current when ports are connected to analog inputs. If the bit of PSR2[1:0] is '1', the dynamic current path of the Schmitt OR gate of the port is cut off and the digital input of the corresponding port is always '0'.

PSR[2] is ADC input mux enable register. PSR2[4] register selects ports to be used as a USART1. PSR2[6:5] register selects ports debounce clock.

9.2.8 Port Debounce Length Selection Register (IODBCNT)

IODBCNT register selects debounce length of port input. Clock selected by PSR[6:5] register counts up to IODBCNT.

9.2.9 Register Map

Name	Address	Direction	Default	Description
P0	80H	R/W	00H	P0 Data Register
P0IO	89H	R/W	00H	P0 Direction Register
P0PU	8F00H	R/W	00H	P0 Pull-up Resistor Selection Register
P0OD	8F0CH	R/W	00H	P0 Open-drain Selection Register
P0DB	8F18H	R/W	00H	P0 Debounce Enable Register
P1	88H	R/W	00H	P1 Data Register
P1IO	92H	R/W	00H	P1 Direction Register
P1PU	8F01H	R/W	00H	P1 Pull-up Resistor Selection Register
P1OD	8F0DH	R/W	00H	P1 Open-drain Selection Register
P1DB	8F19H	R/W	00H	P1 Debounce Enable Register
P2	90H	R/W	00H	P2 Data Register
P2IO	99H	R/W	00H	P2 Direction Register
P2PU	8F02H	R/W	00H	P2 Pull-up Resistor Selection Register
P2OD	8F0EH	R/W	00H	P2 Open-drain Selection Register
P2DB	8F1AH	R/W	00H	P2 Debounce Enable Register
P3	98H	R/W	00H	P3 Data Register
P3IO	A1H	R/W	00H	P3 Direction Register
P3PU	8F03H	R/W	00H	P3 Pull-up Resistor Selection Register
P3OD	8F0FH	R/W	00H	P3 Open-drain Selection Register
P3DB	8F1BH	R/W	00H	P3 Debounce Enable Register
P4	A0H	R/W	00H	P4 Data Register
P4IO	B1H	R/W	00H	P4 Direction Register
P4PU	8F04H	R/W	00H	P4 Pull-up Resistor Selection Register
P4OD	8F10H	R/W	00H	P4 Open-drain Selection Register
P4DB	8F1CH	R/W	00H	P4 Debounce Enable Register
P5	B0H	R/W	00H	P5 Data Register
P5IO	B9H	R/W	00H	P5 Direction Register
P5PU	8F05H	R/W	00H	P5 Pull-up Resistor Selection Register
P5OD	8F11H	R/W	00H	P5 Open-drain Selection Register
P5DB	8F1DH	R/W	00H	P5 Debounce Enable Register
PSR1	8F40H	R/W	00H	Port Selection Register 1
PSR2	8F41H	R/W	00H	Port Selection Register 2
IODBCNT	8F08H	R/W	00H	Port Debounce Length Selection register

Table 10.1 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): 89H

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

Initial value: 00H

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

P0PU (P0 Pull-up Resistor Selection Register): 8F00H

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): 8F0CH

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

Initial value: 00H

P0OD[7:0] Configure Open-drain of P0 Port
0 Push-pull output
1 Open-drain output

P0DB (P0 De-bounce Enable Register): 8F18H

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

Initial value: 00H

P0DB[7:0]	Configure debounce 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.

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

The port de-bounce is not automatically disabled at stop mode.

9.4 P1 Port

9.4.1 P1 Port Description

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

9.4.2 Register description for P1

P1 (P1 Data Register): 88H

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

Initial value: 00H

P1[7:0] I/O Data

P1IO (P1 Direction Register): 91H

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

Initial value: 00H

P1IO[7:0] P1 Data I/O Direction.
0 Input
1 Output

P1PU (P1 Pull-up Resistor Selection Register): 8F01H

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

Initial value: 00H

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

P1OD (P1 Open-drain Selection Register): 8F0DH

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

Initial value: 00H

P1OD[7:0] Configure Open-drain of P1 Port
0 Push-pull output
1 Open-drain output

P1DB (P1 De-bounce Enable Register): 8F19H

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

Initial value: 00H

P1DB[7:0]	Configure debounce of Px 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.

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

The port de-bounce is not automatically disabled at stop mode.

9.5 P2 Port

9.5.1 P2 Port Description

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

9.5.2 Register description for P2

P2 (P2 Data Register): 90H

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

Initial value: 00H

P2[7:0] I/O Data

P2IO (P2 Direction Register): 99H

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

Initial value: 00H

P2IO[7:0] P2 Data I/O Direction.
0 Input
1 Output

P2PU (P2 Pull-up Resistor Selection Register): 8F02H

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

Initial value: 00H

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

P2OD (P2 Open-drain Selection Register): 8F0EH

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

Initial value: 00H

P2OD[7:0] Configure Open-drain of P2 Port
0 Push-pull output
1 Open-drain output

P2DB (P2 De-bounce Enable Register): 8F1AH

7	6	5	4	3	2	1	0
P27DB	P26DB	P25DB	P24DB	P23DB	P22DB	P21DB	P20DB
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

P2DB[7:0]	Configure debounce 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.

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

The port de-bounce is not automatically disabled at stop mode.

9.6 P3 Port

9.6.1 P3 Port Description

P3 is 8-bit I/O port. P3 control registers consist of P3 data register (P3), P3 direction register (P3IO), debounce enable register (P3DB), P3 pull-up resistor selection register (P3PU) and P3 open-drain selection register (P3OD). Refer to the port function selection registers for the P3 function selection.

9.6.2 Register description for P3

P3 (P3 Data Register): 98H

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

Initial value: 00H

P3[7:0]	I/O Data

P3IO (P3 Direction Register): A1H

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

Initial value: 00H

P3IO[7:0]	P3 Data I/O Direction. 0 Input 1 Output

P3PU (P3 Pull-up Resistor Selection Register): 8F03H

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

Initial value: 00H

P3PU[7:0]	Configure Pull-up Resistor of P3 Port 0 Disable 1 Enable

P3OD (P3 Open-drain Selection Register): 8F0FH

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

Initial value: 00H

P3OD[7:0] Configure Open-drain of P3 Port
0 Push-pull output
1 Open-drain output

P3DB (P3 De-bounce Enable Register): 8F1BH

7	6	5	4	3	2	1	0
P37DB	P36DB	P35DB	P34DB	P33DB	P32DB	P31DB	P30DB
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

P3DB[7:0] Configure debounce of P3 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.

A pulse level should be input for the duration of 3 clock or more to be actually detected as a valid edge.
The port de-bounce is not automatically disabled at stop mode.

9.7 P4 Port

9.7.1 P4 Port Description

P4 is 8-bit I/O port. P4 control registers consist of P4 data register (P4), P4 direction register (P4IO), debounce enable register (P4DB), P4 pull-up resistor selection register (P4PU) and P4 open-drain selection register (P4OD). Refer to the port function selection registers for the P4 function selection.

9.7.2 Register description for P4

P4 (P4 Data Register): A0H

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

Initial value: 00H

P4[7:0]	I/O Data

P4IO (P4 Direction Register): B1H

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

Initial value: 00H

P4IO[7:0]	P4 Data I/O Direction. 0 Input 1 Output

P4PU (P4 Pull-up Resistor Selection Register): 8F04H

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

Initial value: 00H

P4PU[7:0]	Configure Pull-up Resistor of P4 Port 0 Disable 1 Enable

P4OD (P4 Open-drain Selection Register): 8F10H

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

Initial value: 00H

P4OD[7:0] Configure Open-drain of P4 Port
0 Push-pull output
1 Open-drain output

P4DB (P4 De-bounce Enable Register): 8F1CH

7	6	5	4	3	2	1	0
P47DB	P46DB	P45DB	P44DB	P43DB	P42DB	P41DB	P40DB
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

P4DB[7:0] Configure debounce of P4 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.

A pulse level should be input for the duration of 3 clock or more to be actually detected as a valid edge.
The port de-bounce is not automatically disabled at stop mode.

9.8 P5 Port

9.8.1 P5 Port Description

P5 is 8-bit I/O port. P5 control registers consist of P5 data register (P5), P5 direction register (P5IO), debounce enable register (P5DB), P5 pull-up resistor selection register (P5PU) and P5 open-drain selection register (P5OD). Refer to the port function selection registers for the P5 function selection.

9.8.2 Register description for P5

P5 (P5 Data Register): B0H

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

Initial value: 00H

P5[7:0]	I/O Data

P5IO (P5 Direction Register): B9H

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

Initial value: 00H

P5IO[7:0]	P5 Data I/O Direction. 0 Input 1 Output

P5PU (P5 Pull-up Resistor Selection Register): 8F05H

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

Initial value: 00H

P5PU[7:0]	Configure Pull-up Resistor of P5 Port 0 Disable 1 Enable

P5OD (P5 Open-drain Selection Register): 8F11H

7	6	5	4	3	2	1	0
-	-	P55OD	P54OD	P53OD	P52OD	P51OD	P50OD
-	-	RW	RW	RW	RW	RW	RW

Initial value: 00H

P5OD[7:0] Configure Open-drain of P5 Port
0 Push-pull output
1 Open-drain output

P5DB (P5 De-bounce Enable Register): 8F1DH

7	6	5	4	3	2	1	0
-	-	P55DB	P54DB	P53DB	P52DB	P51DB	P50DB
-	-	RW	RW	RW	RW	RW	RW

Initial value: 00H

P5DB[7:0] Configure debounce of P5 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.

A pulse level should be input for the duration of 3 clock or more to be actually detected as a valid edge.
The port de-bounce is not automatically disabled at stop mode.
Refer to the configure option for the P55/RESETB.

PSR1 (Port Selection Register 1) : 8F40H

7	6	5	4	3	2	1	0
PSR17	PSR16	PSR15	PSR14	PSR13	PSR12	PSR11	PSR10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

PSR1[7:0] P20~P27port selection register
- 0 Disable analog channel AN[7:0].
- 1 Enable analog channel AN[7:0].

PSR2 (Port Selection Register 2) : 8F41H

7	6	5	4	3	2	1	0
PSR27	PSR26	PSR25	PSR24	PSR23	AMUXEN	PSR21	PSR20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

PSR2[7] Not used

PSR2[6:5] Select port debounce clock
- 00 Internal 256KHz RCOSC
- 01 F_{SCLK} (system clock)
- 10 F_{SCLK} / 2
- 11 F_{SCLK} / 4

PSR2[4] Select ports to be used as USART1
- 0 P31, P32 port is used as a USART1
- 1 P11, P12 port is used as a USART1

PSR2[3] Not used

AMUXEN ADC input mux enable or disable
- 0 disable
- 1 enable

PSR2[1] P34 port selection register
- 0 Disable analog channel AN[9].
- 1 Enable analog channel AN[9].

PSR2[0] P17 port selection register
- 0 Disable analog channel AN[8].
- 1 Enable analog channel AN[8].

IODBCNT (Port Debounce Length Selection Register) : 8F08H

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

Initial value : 00H

IODBCNT[2:0] Port debounce length selection register
Deboounce length = IODBCNT x 1/f_{dbclk}
f_{dbclk} is the clock selected by PSR2[6:5]

10 Interrupt Controller

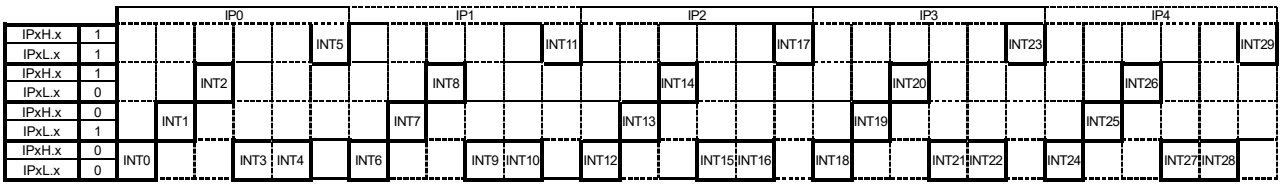
10.1 Overview

The A97C450 supports up to 30 interrupt sources. The interrupts have separate enable register bits associated with them, allowing software control. They can also have four levels of priority assigned to them. The non-maskable interrupt source is always enabled with a higher priority than any other interrupt source, and is not controllable by software. The interrupt controller has following features:

- Receive the request from 30 interrupt source
- Individual priority
- 4 priority levels
- Multi Interrupt possibility
- If the requests of different priority levels are received simultaneously, the request of higher priority level is served first.
- Each interrupt source can be controlled by EA bit and each IEx bit
- Interrupt latency: 3~9 machine cycles in single interrupt system

The non-maskable interrupt is always enabled. The maskable interrupts are enabled through four pair of interrupt enable registers (IE, IE1, IE2, IE3, IE4). Each bit of IE, IE1, IE2, IE3, IE4 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 A97C450 supports a four-level priority scheme. Each maskable interrupt is individually assigned to one of four priority levels according to IP0H/L, IP1H/L, IP2H/L, IP3H/L, and IP4H/L.

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



10.2 External Interrupt

The external interrupt on EINT0 ~ EINT7 pins receive various interrupt request depending on the edge selection register EIEDGE (External Interrupt Edge register) and EIPOLA (External Interrupt Priority register) as shown in Figure 10-1. Also each external interrupt source has control setting bits. The EIFLAG (External interrupt flag register) register provides the status of external interrupts.

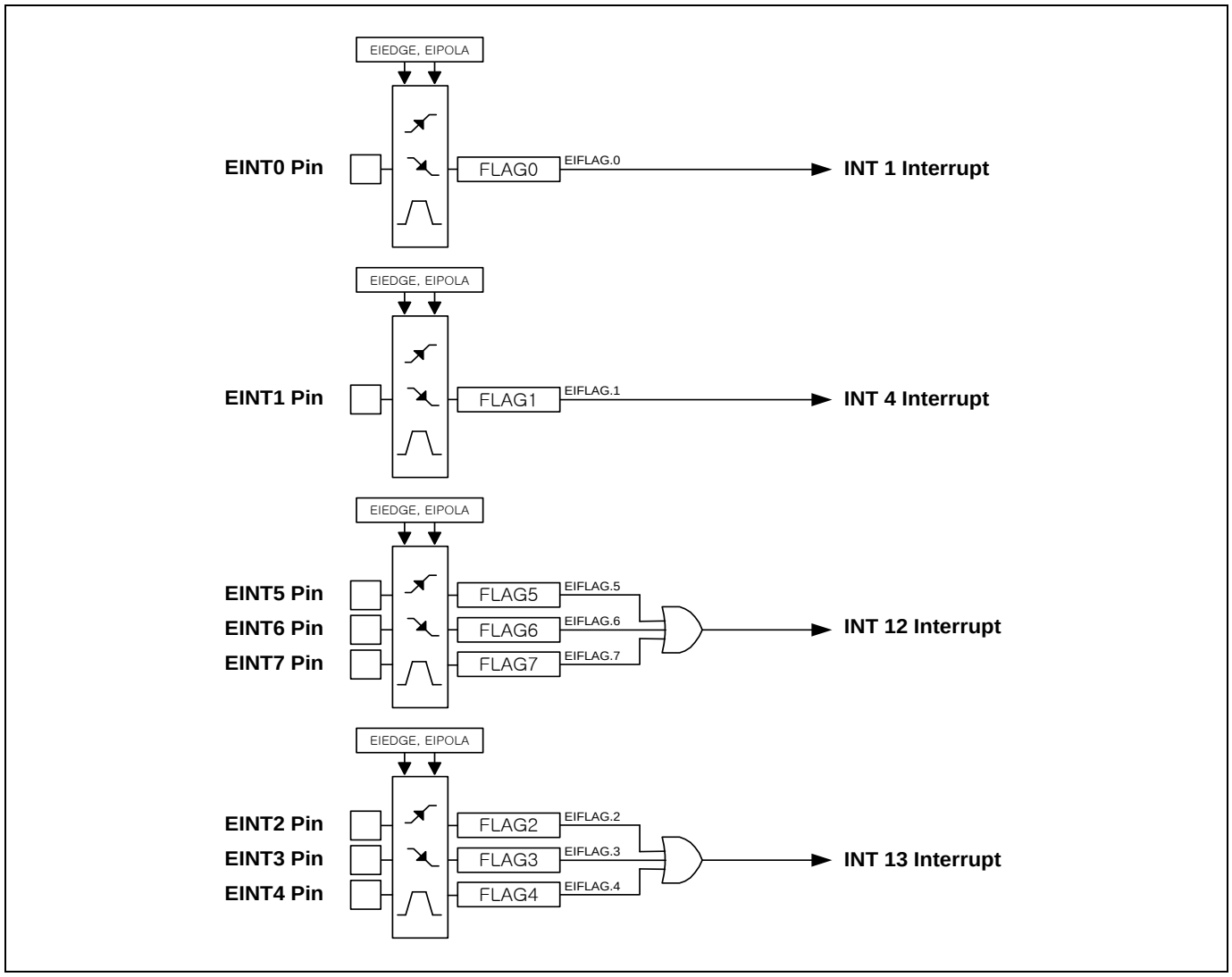


Figure 10.1 External Interrupt Description

10.3 Block Diagram

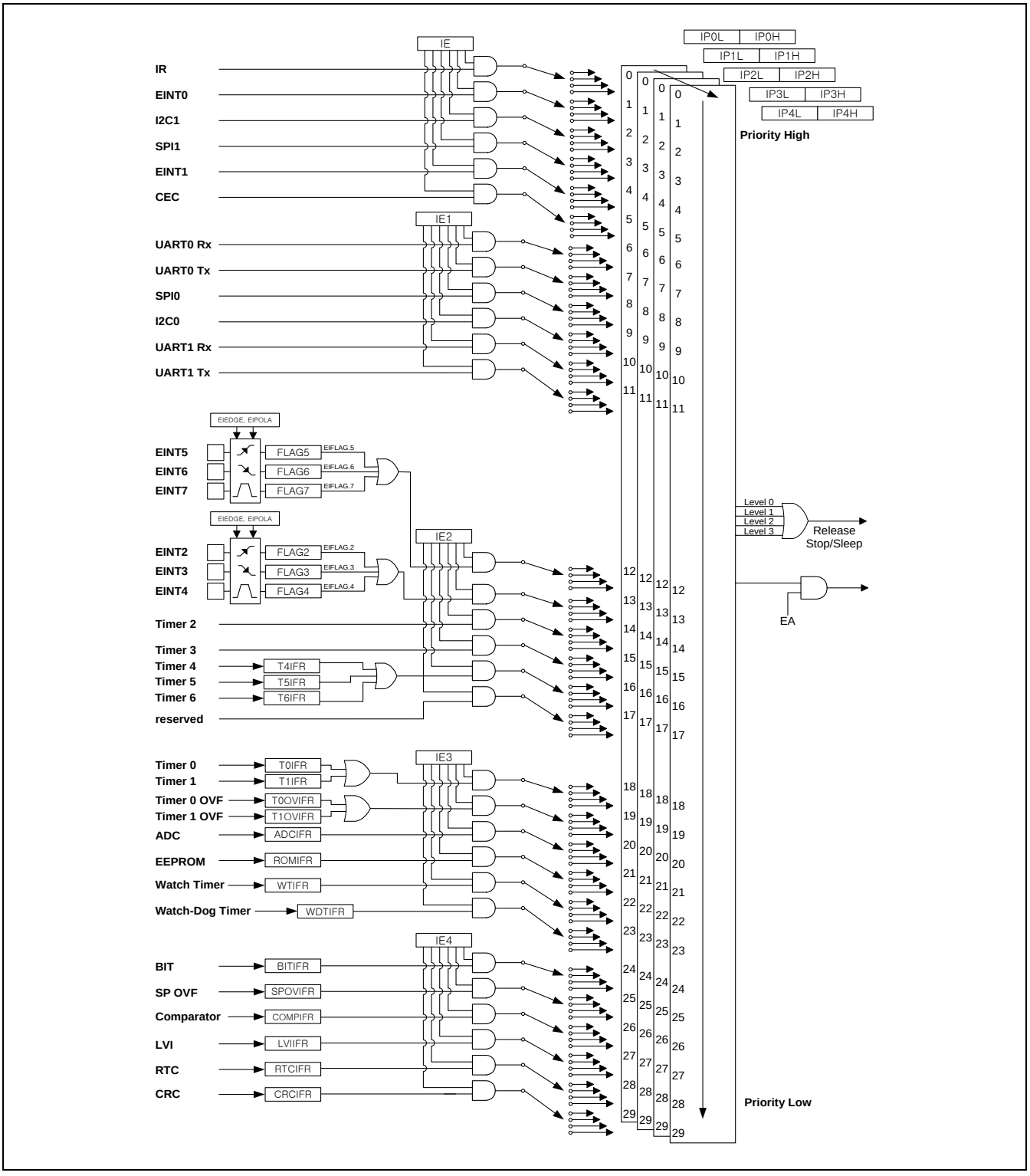


Figure 10.2 Block Diagram of Interrupt

NOTE)

1. The release signal may be generated by all interrupt sources which are enabled without reference to a priority level.
2. An interrupt request is delayed during data are written to IE, IE1, IE2, IE3, IE4, IP0L/H, IP1L/H, IP2L/H, IP3L/H, IP4L/H and PCON register.

10.4 Interrupt Vector Table

The interrupt controller supports 30 interrupt sources as shown in the Table 10-2. When interrupt is served, long call instruction (LCALL) is executed and program counter jumps to the vector address. All interrupt requests have their own priority order.

Interrupt Source	Symbol	Interrupt Enable Bit	Priority	Mask	Vector Address
Hardware RESET	RESETB	0	0	Non-Maskable	0000H
IR Interrupt	INT0	IE.0	1	Maskable	0003H
External Interrupt 0	INT1	IE.1	2	Maskable	000BH
I2C1 Interrupt	INT2	IE.2	3	Maskable	0013H
SPI1 Interrupt	INT3	IE.3	4	Maskable	001BH
External Interrupt 1	INT4	IE.4	5	Maskable	0023H
CEC Interrupt	INT5	IE.5	6	Maskable	002BH
UART0 Rx Interrupt	INT6	IE1.0	7	Maskable	0033H
UART0 Tx Interrupt	INT7	IE1.1	8	Maskable	003BH
SPI0 Interrupt	INT8	IE1.2	9	Maskable	0043H
I2C0 Tx Interrupt	INT9	IE1.3	10	Maskable	004BH
UART1 Rx Interrupt	INT10	IE1.4	11	Maskable	0053H
UART1 Tx Interrupt	INT11	IE1.5	12	Maskable	005BH
External Interrupt 5 – 7	INT12	IE2.0	13	Maskable	0063H
External Interrupt 2 – 3	INT13	IE2.1	14	Maskable	006BH
T2 Match Interrupt	INT14	IE2.2	15	Maskable	0073H
T3 Match Interrupt	INT15	IE2.3	16	Maskable	007BH
T4/5/6 Match Interrupt	INT16	IE2.4	17	Maskable	0083H
Reserved	INT17	IE2.5	18	Maskable	008BH
T0/1 Match Interrupt	INT18	IE3.0	19	Maskable	0093H
T0/1 Overflow Interrupt	INT19	IE3.1	20	Maskable	009BH
ADC Interrupt	INT20	IE3.2	21	Maskable	00A3H
EEPROM Interrupt	INT21	IE3.3	22	Maskable	00ABH
WT Interrupt	INT22	IE3.4	23	Maskable	00B3H
WDT Interrupt	INT23	IE3.5	24	Maskable	00BBH
BIT Interrupt	INT24	IE4.0	25	Maskable	00C3H
SP Overflow Interrupt	INT25	IE4.1	26	Maskable	00CBH
Comparator Interrupt	INT26	IE4.2	27	Maskable	00D3H
LVI Interrupt	INT27	IE4.3	28	Maskable	00DBH
RTC Interrupt	INT28	IE4.4	29	Maskable	00E3H
CRC Interrupt	INT29	IE4.5	30	Maskable	00EBH

Table 10.2 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.

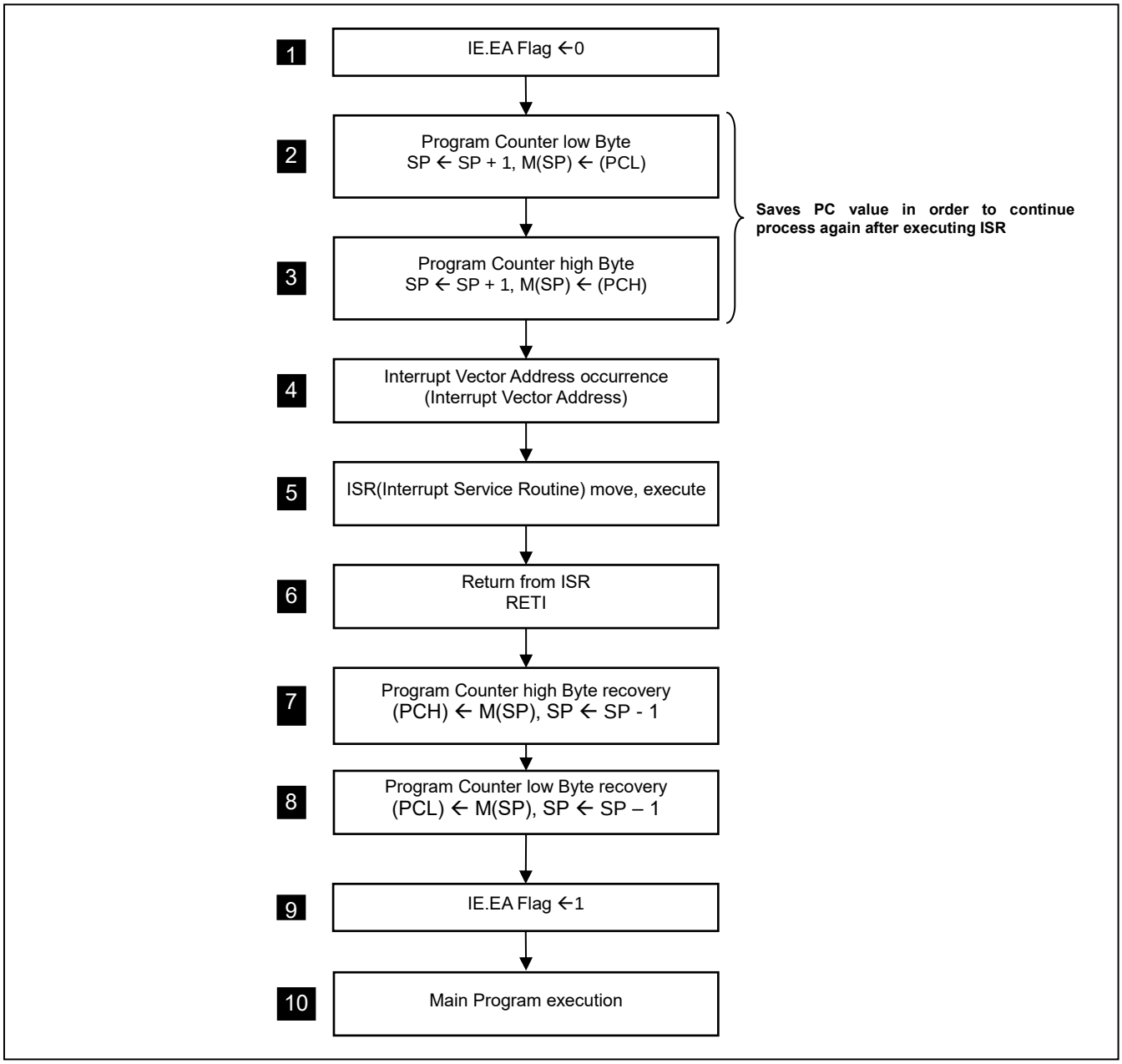


Figure 10.3 Interrupt Sequence Flow

10.6 Effective Timing after Controlling Interrupt Bit

Case a) Control Interrupt Enable Register (IE, IE1, IE2, IE3, IE4)

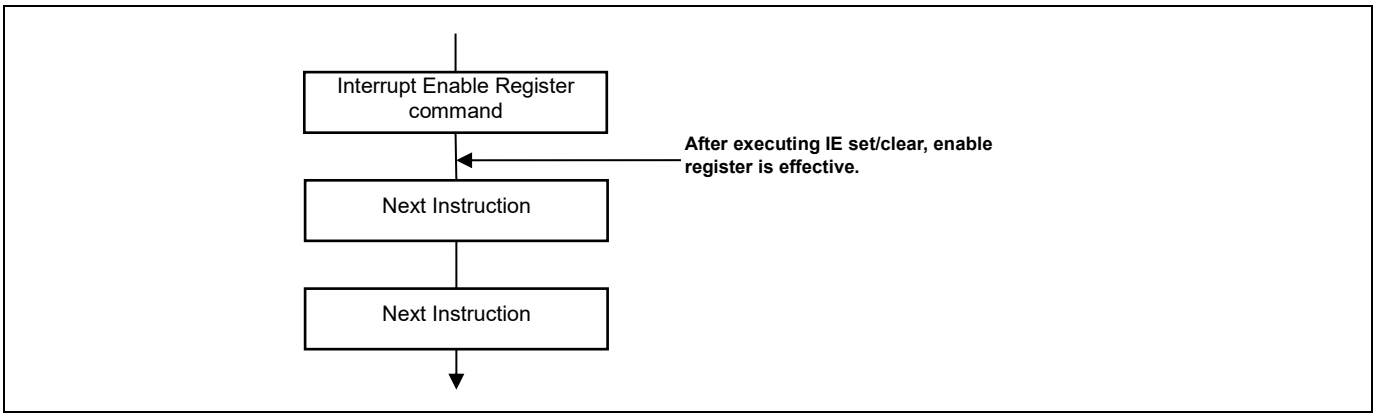


Figure 10.4 Effective Timing of Interrupt Enable Register

Case b) Interrupt flag Register

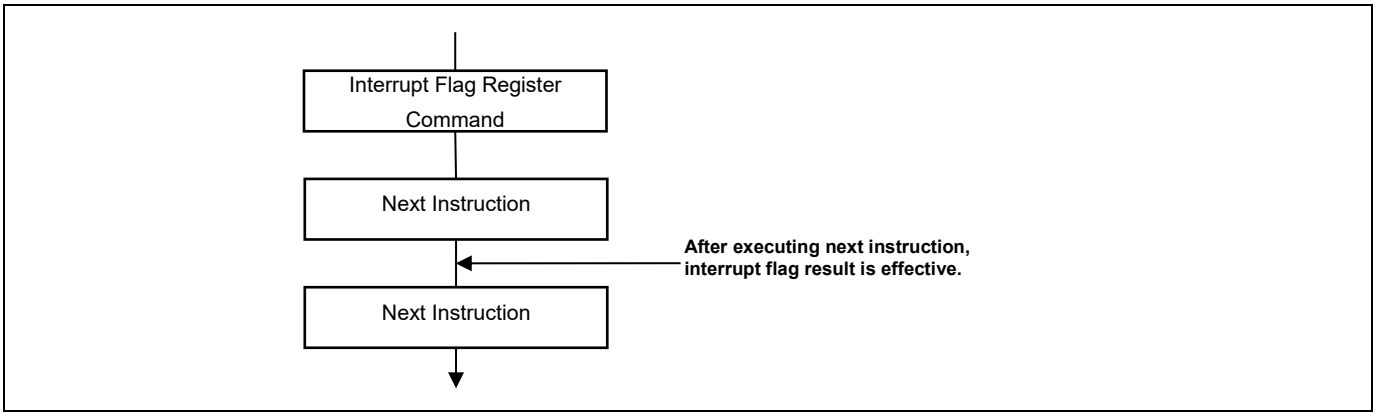


Figure 10.5 Effective Timing of Interrupt Flag Register

10.7 Multi Interrupt

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

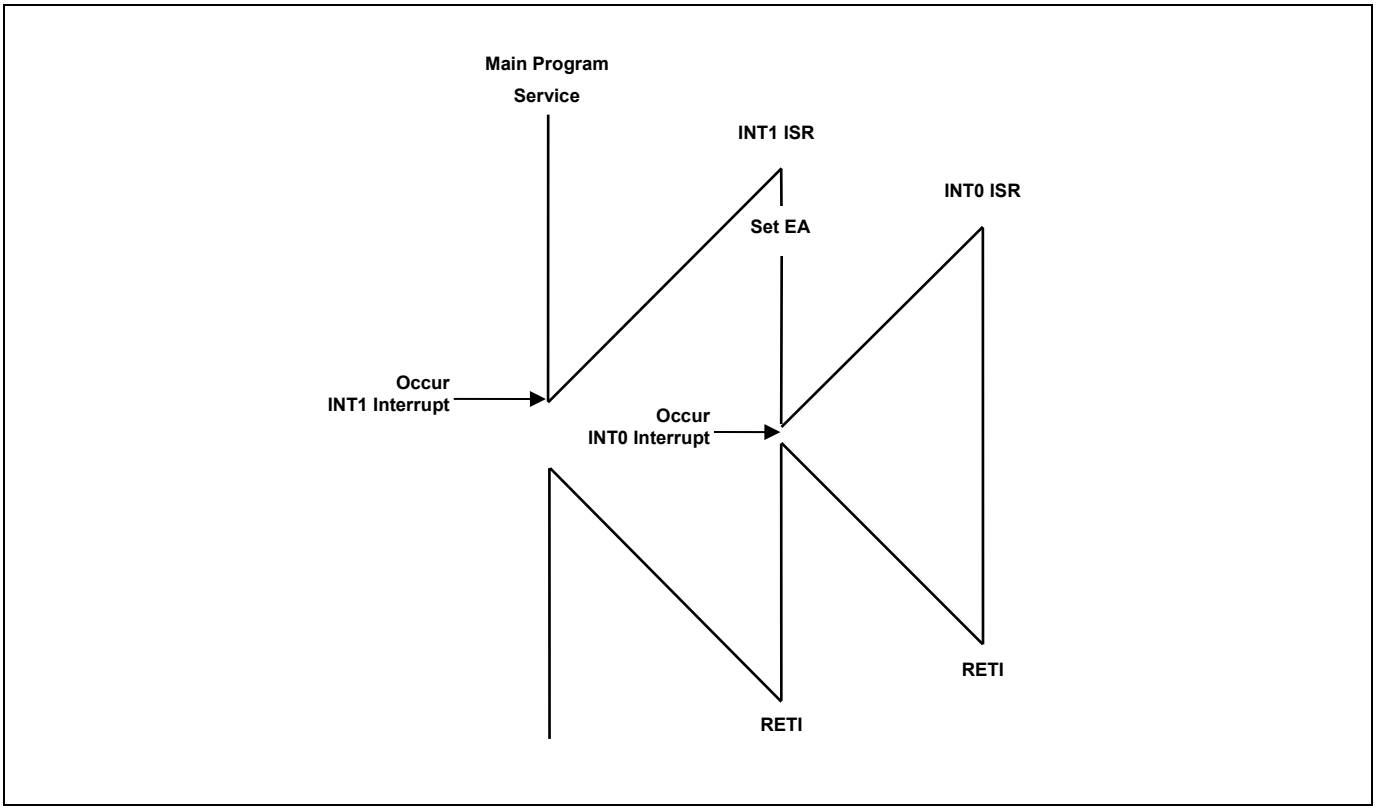


Figure 10.6 Effective Timing of Multi-Interrupt

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

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

10.8 Interrupt Enable Accept Timing

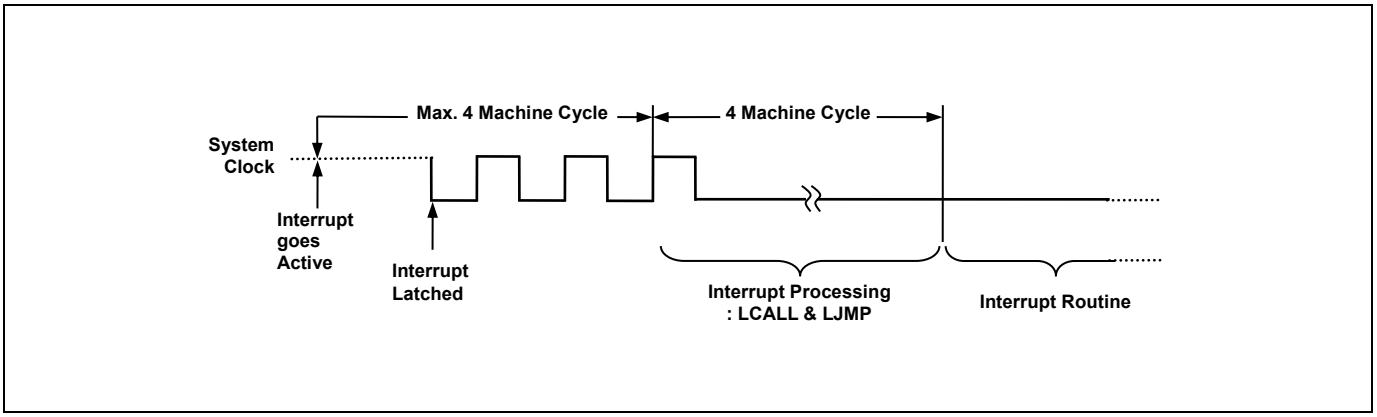


Figure 10.7 Interrupt Response Timing Diagram

10.9 Interrupt Service Routine Address

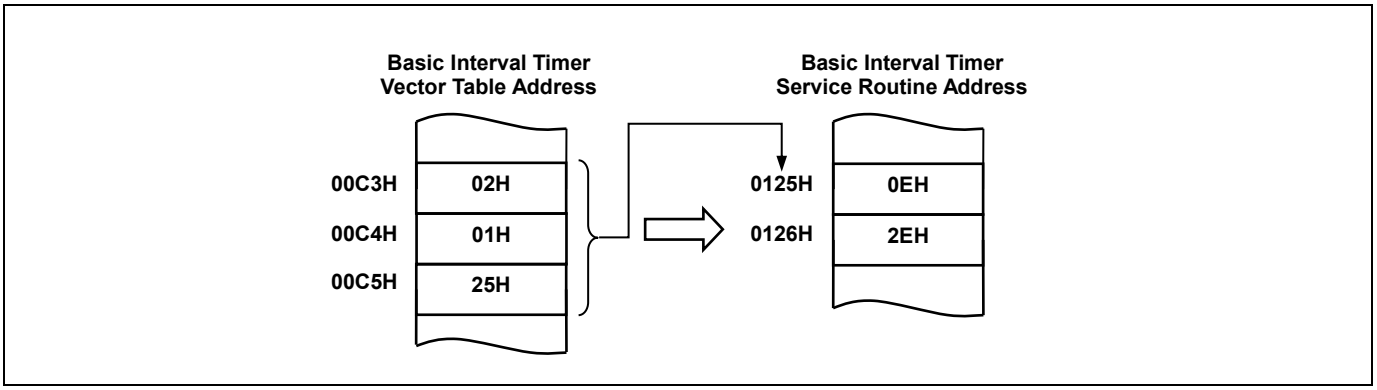


Figure 10.8 Correspondence between Vector Table Address and the Entry Address of ISR

10.10 Saving/Restore General-Purpose Registers

```
INTxx :  PUSH  PSW
         PUSH  DPL
         PUSH  DPH
         PUSH  B
         PUSH  ACC
         :
Interrupt_Processing:
         :
         POP  ACC
         POP  B
         POP  DPH
         POP  DPL
         POP  PSW
         RETI
```

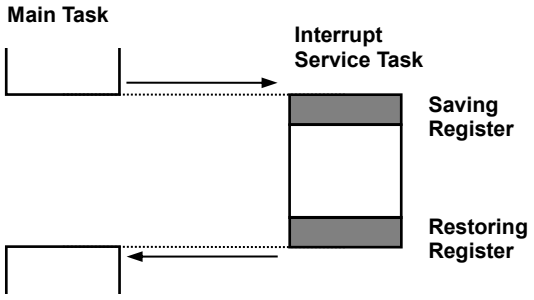


Figure 10.9 Saving/Restore Process Diagram and Sample Source

10.11 Interrupt Timing

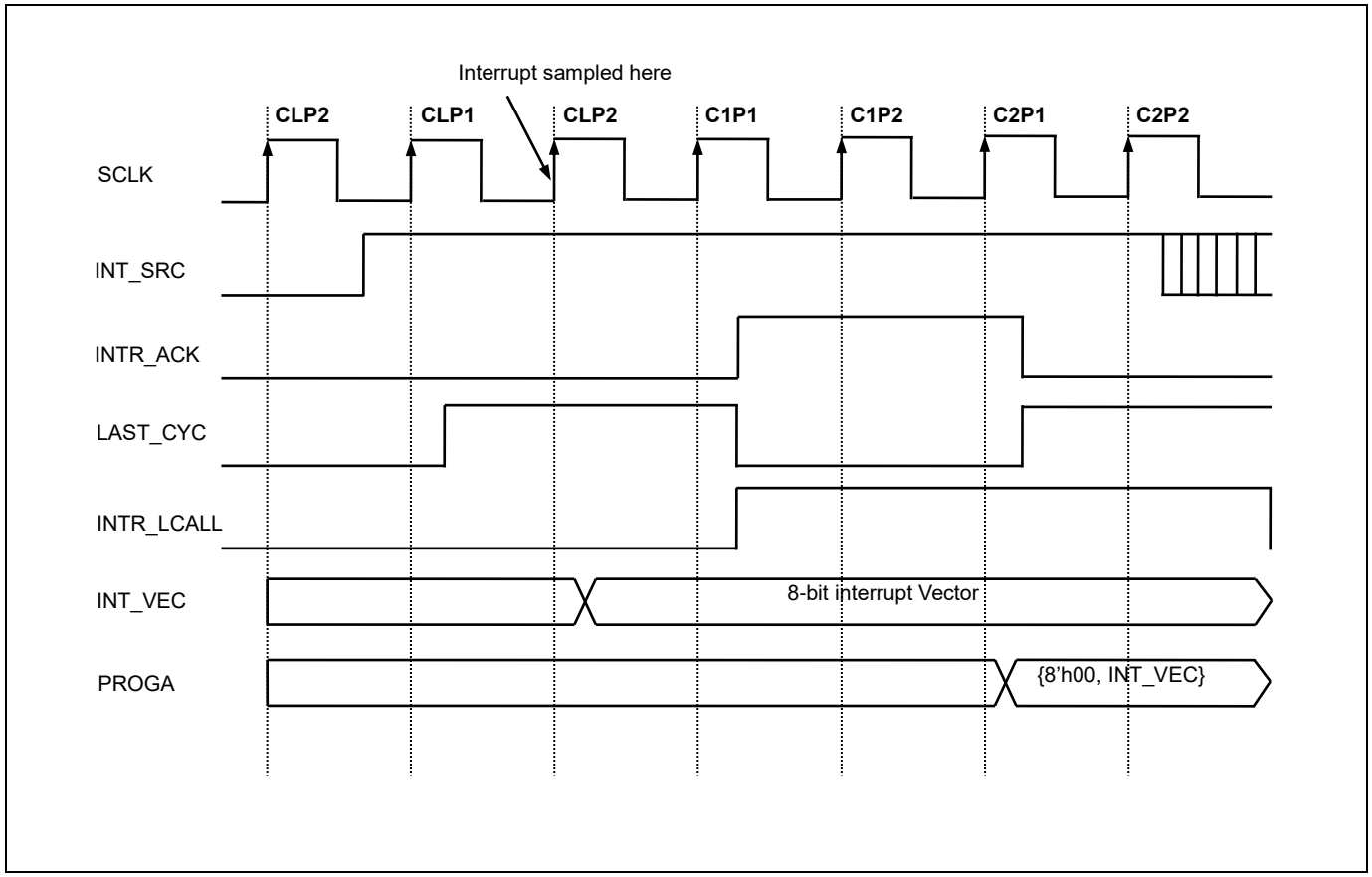


Figure 10.10 Timing Chart of Interrupt Acceptance and Interrupt Return Instruction

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

NOTE)

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

10.12 Interrupt Register Overview

10.12.1 Interrupt Enable Register (IE, IE1, IE2, IE3, IE4)

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

10.12.2 10.12.2 Interrupt Priority Register (IP0H/L, IP1H/L, IP2H/L, IP3H/L, IP4H/L)

The 30 interrupts are divided into 6 groups which have each 4 interrupt sources. An individual interrupt can be assigned 4 levels interrupt priority using interrupt priority register. Level 3 is the highest priority, while level 0 is the lowest priority. After a reset IP0H/L, IP1H/L, IP2H/L, and IP3H/L, are cleared to '00H'. If interrupts have the same priority level, lower number interrupt is served first.

10.12.3 External Interrupt Flag Register (EIFLAG)

The external interrupt flag register is set to '1' when the external interrupt generating condition is satisfied. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a '0' to it.

10.12.4 External Interrupt Edge Register (EIEDGE)

The External interrupt edge register determines which type of edge or level sensitive interrupt. Initially, default value is level. For level, write '0' to related bit. For edge, write '1' to related bit.

10.12.5 External Interrupt Polarity Register (EIPOLA)

According to EIEDGE register, the external interrupt polarity (EIPOLA) register has a different meaning. If EIEDGE is level type, EIPOLA is able to have Low/High level value. If EIEGDE is edge type, EIPOLA is able to have rising/falling edge value.

10.12.6 External Interrupt Both Edge Enable Register (EIBOTH)

When the external interrupt both edge enable register is written to '1', the corresponding external pin interrupt is enabled by both edges. Initially, default value is disabled.

10.12.7 External Interrupt Enable Register (EIENAB)

When the external interrupt enable register is written to '1', the corresponding external pin interrupt is enabled. The EIEDGE and EIPOLA register defines whether the external interrupt is activated on rising or falling edge or level sensed.

10.12.8 Register Map

Name	Address	Dir	Default	Description
IE	A8H	R/W	00H	Interrupt Enable Register
IE1	A9H	R/W	00H	Interrupt Enable Register 1
IE2	AAH	R/W	00H	Interrupt Enable Register 2
IE3	ABH	R/W	00H	Interrupt Enable Register 3
IE4	ACH	R/W	00H	Interrupt Enable Register 4
IP0L	8FA0H	R/W	00H	Interrupt Priority 0 Low Register
IP0H	8FA1H	R/W	00H	Interrupt Priority 0 High Register
IP1L	8FA2H	R/W	00H	Interrupt Priority 1 Low Register
IP1H	8FA3H	R/W	00H	Interrupt Priority 1 High Register
IP2L	8FA4H	R/W	00H	Interrupt Priority 2 Low Register
IP2H	8FA5H	R/W	00H	Interrupt Priority 2 High Register
IP3L	8FA6H	R/W	00H	Interrupt Priority 3 Low Register
IP3H	8FA7H	R/W	00H	Interrupt Priority 3 High Register
IP4L	8FA8H	R/W	00H	Interrupt Priority 4 Low Register
IP4H	8FA9H	R/W	00H	Interrupt Priority 4 High Register
EIENAB	A3H	R/W	00H	External Interrupt Enable Register
EIFLAG	A4H	R/W	00H	External Interrupt Flag Register
EIEDGE	A5H	R/W	00H	External Interrupt Edge Register
EIPOLA	A6H	R/W	00H	External Interrupt Polarity Register
EIBOTH	A7H	R/W	00H	External Interrupt Both Edge Register

Table 10.3 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), Interrupt Enable Register 2 (IE2), Interrupt Enable Register 3 (IE3) and Interrupt Enable Register 4 (IE4). For external interrupt, it consists of External Interrupt Flag Register (EIFLAG), External Interrupt Edge Register (EIEDGE), External Interrupt Polarity Register (EIPOLA) and External Interrupt Enable Register (EIENAB).

10.12.10 Register Description for Interrupt

IE (Interrupt Enable Register) : A8H

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

Initial value : 00H

EA Enable or disable all interrupt bits
- 0 All Interrupt disable
- 1 All Interrupt enable

INT5E Enable or disable CEC Interrupt
- 0 Disable
- 1 Enable

INT4E Enable or disable External Interrupt 1
- 0 Disable
- 1 Enable

INT3E Enable or disable SPI1 Interrupt
- 0 Disable
- 1 Enable

INT2E Enable or disable I2C1 Interrupt
- 0 Disable
- 1 Enable

INT1E Enable or disable External Interrupt 0
- 0 Disable
- 1 Enable

INT0E Enable or disable IR Interrupt
- 0 Disable
- 1 Enable

IE1 (Interrupt Enable Register 1) : A9H

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

Initial value : 00H

INT11E Enable or disable UART1 Tx Interrupt
0 Disable
1 Enable

INT10E Enable or disable UART1 Rx Interrupt
0 Disable
1 Enable

INT9E Enable or disable I2C0 Interrupt
0 Disable
1 Enable

INT8E Enable or disable SPI0 Interrupt
0 Disable
1 Enable

INT7E Enable or disable UART0 Tx Interrupt
0 Disable
1 Enable

INT6E Enable or disable UART0 Rx Interrupt
0 Disable
1 Enable

IE2 (Interrupt Enable Register 2) : AAH

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

Initial value : 00H

INT16E Enable or Disable Timer 4/5/6 match interrupt
0 Disable
1 Enable

INT15E Enable or disable Timer 3 match Interrupt
0 Disable
1 Enable

INT14E Enable or disable Timer 2 match Interrupt
0 Disable
1 Enable

INT13E Enable or Disable External interrupt 2 ~ 4 (EINT2 ~ EINT4)
0 Disable
1 Enable

INT12E Enable or Disable External interrupt 5 ~ 7 (EINT5 ~ EINT7)
0 Disable
1 Enable

IE3 (Interrupt Enable Register 3) : ABH

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

Initial value : 00H

INT23E Enable or disable WDT Interrupt
- 0 Disable
- 1 Enable

INT22E Enable or disable WT Interrupt
- 0 Disable
- 1 Enable

INT21E Enable or disable EEPROM Interrupt
- 0 Disable
- 1 Enable

INT20E Enable or disable ADC Interrupt
- 0 Disable
- 1 Enable

INT19E Enable or Disable Timer 0/1 overflow interrupt
- 0 Disable
- 1 Enable

INT18E Enable or Disable Timer 0/1 match interrupt
- 0 Disable
- 1 Enable

IE4 (Interrupt Enable Register 4) : ACH

7	6	5	4	3	2	1	0
-	-	INT29E	INT28E	INT27E	INT26E	INT25E	INT24E
R	R	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

INT29E Enable or disable CRC Interrupt
- 0 Disable
- 1 Enable

INT28E Enable or disable RTC Interrupt
- 0 Disable
- 1 Enable

INT27E Enable or disable LVI Interrupt
- 0 Disable
- 1 Enable

INT26E Enable or disable Comparator Interrupt
- 0 Disable
- 1 Enable

INT25E Enable or disable SP overflow Interrupt
- 0 Disable
- 1 Enable

INT24E Enable or disable BIT Interrupt
- 0 Disable
- 1 Enable

IP0L (Interrupt Priority 0 Low Register) : 8FA0H

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

Initial value : 00H

IP0H (Interrupt Priority 0 High Register) : 8FA1H

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

Initial value : 00H

IP0L[5:0], IP0H[5:0]

Select IE Interrupt Priority

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

IP1L (Interrupt Priority 1 Low Register) : 8FA2H

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

Initial value : 00H

IP1H (Interrupt Priority 1 High Register) : 8FA3H

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

Initial value : 00H

IP1L[5:0], IP1H[5:0]

Select IE1 Interrupt Priority

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

IP2L (Interrupt Priority 2 Low Register) : 8FA4H

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

Initial value : 00H

IP2H (Interrupt Priority 2 High Register) : 8FA5H

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

Initial value : 00H

IP2L[5:0], IP2H[5:0]]

Select IE2 Interrupt Priority

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

IP3L (Interrupt Priority 3 Low Register) : 8FA6H

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

Initial value : 00H

IP3H (Interrupt Priority 3 High Register) : 8FA7H

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

Initial value : 00H

IP3L[5:0], IP3H[5:0]]

Select IE3 Interrupt Priority

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

IP4L (Interrupt Priority 4 Low Register) : 8FA8H

7	6	5	4	3	2	1	0
–	–	IP4L5	IP4L4	IP4L3	IP4L2	IP4L1	IP4L0
–	–	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

IP4H (Interrupt Priority 4 High Register) : 8FA9H

7	6	5	4	3	2	1	0
–	–	IP4H5	IP4H4	IP4H3	IP4H2	IP4H1	IP4H0
–	–	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

IP4L[5:0], IP4H[5:0]]

Select IE4 Interrupt Priority

IP4Hx	IP4Lx	Description
0	0	level 0 (lowest)
0	1	level 1
1	0	level 2
1	1	level 3 (highest)

EIENAB (External Interrupt Enable Register) : A3H

7	6	5	4	3	2	1	0
ENAB7	ENAB6	ENAB5	ENAB4	ENAB3	ENAB2	ENAB1	ENAB0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

ENAB[7:0] Control External Interrupt for EINTx
0 Disable EINTx (default)
1 Enable EINTx

EXT_INT0 -> ENAB0,, EXT_INT7->ENAB7

EIFLAG (External Interrupt Flag Register) : A4H

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

Initial value : 00H

FLAG[7:0] If External Interrupt is occurred, the flag becomes '1'. The flag can be cleared by writing a '0' to bit
0 External Interrupt not occurred
1 External Interrupt occurred

EINT0 -> FLAG0,, EINT7->FLAG7

EIEDGE (External Interrupt Edge Register) : A5H

7	6	5	4	3	2	1	0
EDGE7	EDGE6	EDGE5	EDGE4	EDGE3	EDGE2	EDGE1	EDGE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

EDGE[7:0] Determines which type of edge or level sensitive interrupt may occur.
0 Level (default)
1 Edge

EINT0 -> EDGE0,, EINT7->EDGE7

EIPOLA (External Interrupt Polarity Register) : A6H

7	6	5	4	3	2	1	0
POLA7	POLA6	POLA5	POLA4	POLA3	POLA2	POLA1	POLA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

POLA[7:0] According to EIEDGE, External interrupt polarity register has a different means. If EIEDGE is level type, external interrupt polarity is able to have Low/High level value. If EIEGDE is edge type, external interrupt polarity is able to have rising/ falling edge value.

Level case:

0 When High level, Interrupt occurred (default)

1 When Low level, Interrupt occurred

Edge case:

0 When Rising edge, Interrupt occurred (default)

1 When Falling edge, Interrupt occurred

EINT0 -> POLA0,, EINT7->POLA7

EIBOTH (External Interrupt Both Edge Enable Register) : A7H

7	6	5	4	3	2	1	0
BOTH7	BOTH6	BOTH5	BOTH4	BOTH3	BOTH2	BOTH1	BOTH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

BOTH[7:0] Determines which type of interrupt may occur, EIBOTH or EIEDGE+EIPOLA. if EIBOTH is enable, EIEDGE and EIPOLA register value don't matter

0 Disable (default)

1 Enable

EINT0 -> BOTH0,, EINT7->BOTH7

11 Peripheral Hardware

11.1 Stack Pointer Management Unit

11.1.1 Overview

The SPMU register enables stack pointer interrupt. When stack pointer is larger than or less than the compare register, stack pointer flag is generated. If stack pointer interrupt is enabled and the global interrupt is enabled, program go to the interrupt service routine.

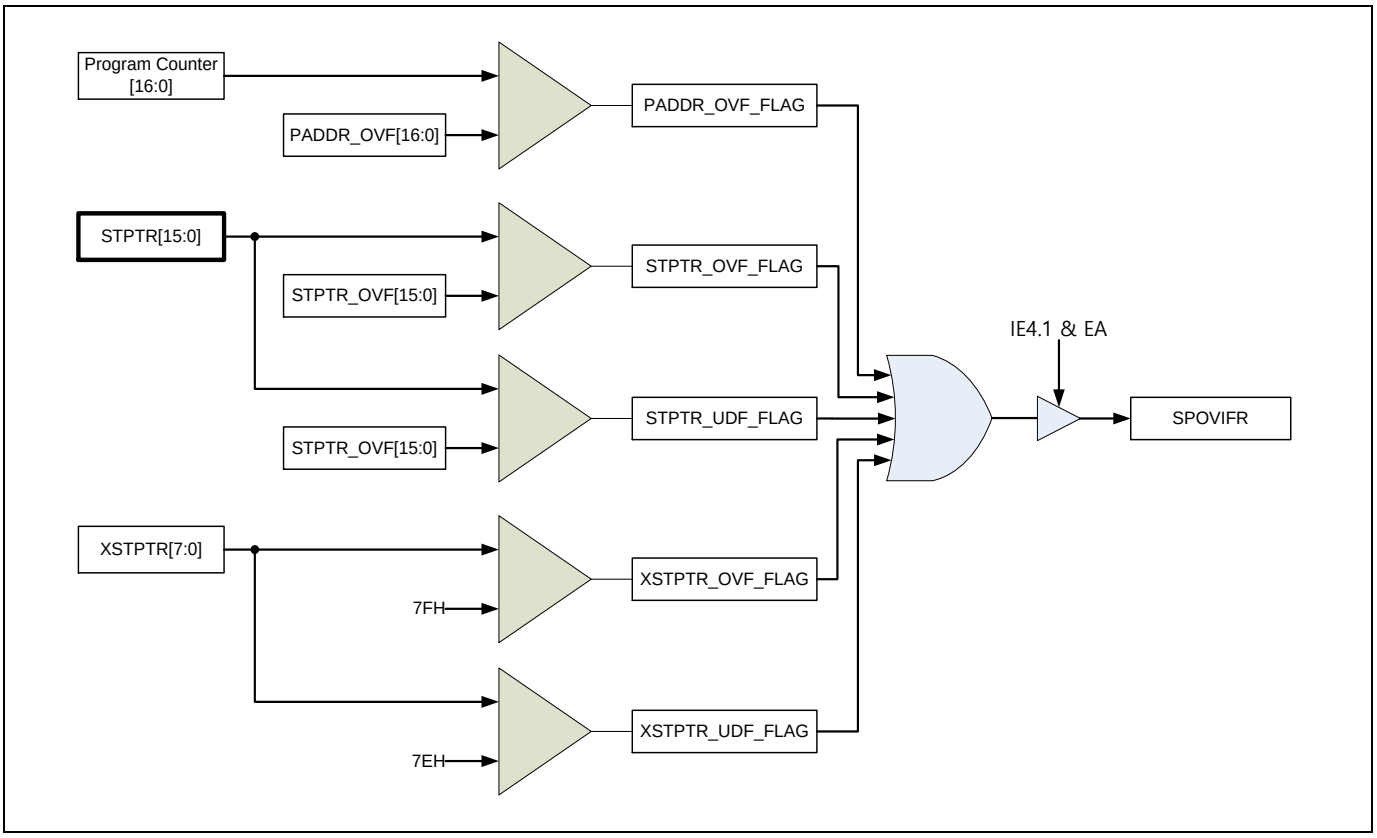


Figure 11.1 SPMU block diagram

11.1.2 Register Map

Name	Address	Direction	Default	Description
SPMUCR	8FD0H	R/W	00H	SPMU Control register
SPMUSR	8FD8H	R/W	00H	SPMU Flag register
PADDR_OVFH	8FD1H	R/W	00H	Program Address Overflow Data Register High
PADDR_OVFH	8FD2H	R/W	00H	Program Address Overflow Data Register High
PADDR_OVFH	8FD3H	R/W	00H	Program Address Overflow Data Register High
STPTR_OVFH	8FD4H	R/W	00H	Stack Pointer Overflow Data Register High
STPTR_OVFL	8FD5H	R/W	FEH	Stack Pointer Overflow Data Register Low:
STPTR_UDFH	8FD6H	R/W	00H	Stack Pointer Underflow Data Register High
STPTR_UDFL	8FD7H	R/W	07H	Stack Pointer Underflow Data Register Low
PADDRH	8FD9H	R	00H	Program Address Invalid Data Register High
PADDRM	8FDAH	R	00H	Program Address Invalid Data Register Middle
PADDRL	8FDBH	R	00H	Program Address Invalid Data Register Low
STPTRH	8FDCH	R	00H	Stack Pointer Register High
STPTRL	8FDDH	R	07H	Stack Pointer Register Low
XSTPTR	8FDEH	R	7FH	Memory Extension Stack Pointer

Table 11.1 SPMU Register Map

11.1.3 Register Description for SPMU

SPMUCR (SPMU Control register) : 8FD0H

7	6	5	4	3	2	1	0
-	-	-	XSTPTR_UDF_EN	XSTPTR_OVF_EN	STPTR_UDF_EN	STPTR_OVF_EN	PADDR_MNT_EN
-	-	-	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

XSTPTR_UDF_EN When memory extension stack pointer underflow occurs, this enables the generation of XSTPTR_UDF_FLAG
- 0 Disable
- 1 Enable

XSTPTR_OVF_EN When memory extension stack pointer overflow occurs, this enables the generation of XSTPTR_OVF_FLAG
- 0 Disable
- 1 Enable

STPTR_UDF_EN When stack pointer underflow occurs, this enables the generation of STPTR_UDF_FLAG
- 0 Disable
- 1 Enable

STPTR_OVF_EN When stack pointer overflow occurs, this enables the generation of STPTR_OVF_FLAG
- 0 Disable
- 1 Enable

PADDR_MNT_EN When program address is larger than the program address overflow data register (PADDR_OVF), this enables the generation of PADDR_FLAG
- 0 Disable
- 1 Enable

PADDR_OVFH (Program Address Overflow Data Register High) : 8FD1H

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

Initial value : 00H

PADDR_OVFH Program address overflow data register high

PADDR_OVFM (Program Address Overflow Data Register Middle) : 8FD2H

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

Initial value : 00H

PADDR_OVFM Program address overflow data register middle

PADDR_OVFL (Program Address Overflow Data Register Low) : 8FD3H

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

Initial value : 00H

PADDR_OVFL Program address overflow data register low

STPTR_OVFH (Stack Pointer Overflow Data Register High) : 8FD4H

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

Initial value : 00H

STPTR_OVFH Stack pointer overflow data register high

STPTR_OVFL (Stack Pointer Overflow Data Register Low) : 8FD5H

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

Initial value : FEH

SPWRH Stack Pointer overflow data register low

STPTR_UDFH (Stack Pointer Underflow Data Register High) : 8FD6H

7	6	5	4	3	2	1	0
STPTR UDFH							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

STPTR_UDFH Stack pointer underflow data register high

STPTR_UDFL (Stack Pointer Underflow Data Register Low) : 8FD7H

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

Initial value : 07H

STPTR_UDFL Stack pointer underflow data register low

SPMUSR (SPMU Flag register) : 8FD8H

7	6	5	4	3	2	1	0
-	-	-	XSTPTR_UDF_FLAG	XSTPTR_OVF_FLAG	STPTR_UDF_FLAG	STPTR_OVF_FLAG	PADDR_FLAG
-	-	-	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

XSTPTR_UDF_FLAG If memory extension stack pointer underflow is occurred, the flag becomes ‘1’. The flag can be cleared by writing a ‘0’ to bit
- 0 Memory extension stack pointer underflow is not occurred
- 1 Memory extension stack pointer underflow occurred

XSTPTR_OVF_FLAG If memory extension stack pointer overflow is occurred, the flag becomes ‘1’. The flag can be cleared by writing a ‘0’ to bit
- 0 Memory extension stack pointer overflow is not occurred
- 1 Memory extension stack pointer overflow occurred

STPTR_UDF_FLAG If stack pointer underflow is occurred, the flag becomes ‘1’. The flag can be cleared by writing a ‘0’ to bit
- 0 Stack pointer underflow is not occurred
- 1 Stack pointer underflow occurred

STPTR_OVF_FLAG If stack pointer overflow is occurred, the flag becomes ‘1’. The flag can be cleared by writing a ‘0’ to bit
- 0 Stack pointer overflow is not occurred
- 1 Stack pointer overflow occurred

PADDR_FLAG If program address overflow is occurred, the flag becomes ‘1’. The flag can be cleared by writing a ‘0’ to bit
- 0 Program address overflow is not occurred
- 1 Program address overflow occurred

PADDRH (Program Address Invalid Data Register High) : 8FD9H

7	6	5	4	3	2	1	0
PADDRH							
R	R	R	R	R	R	R	R

Initial value : 00H

PADDRH Program address overflow data register high

PADDRM (Program Address Invalid Data Register Middle) : 8FDAH

7	6	5	4	3	2	1	0
PADDRM							
R	R	R	R	R	R	R	R

Initial value : 00H

PADDRM Program address overflow data register middle

PADDRL (Program Address Invalid Data Register Low) : 8FDBH

7	6	5	4	3	2	1	0
PADDRL							
R	R	R	R	R	R	R	R

Initial value : 00H

PADDRL Program address overflow data register low

STPTRH (Stack Pointer Register High) : 8FDCH

7	6	5	4	3	2	1	0
STPTRH							
R	R	R	R	R	R	R	R

Initial value : 00H

SPWRH Stack pointer register high

STPTRL (Stack Pointer Register Low) : 8FDDH

7	6	5	4	3	2	1	0
STPTRL							
R	R	R	R	R	R	R	R

Initial value : 07H

SPWRL Stack pointer register low

XSTPTR (Memory Extension Stack Pointer) : 8FDEH

7	6	5	4	3	2	1	0
XSTPTR							
R	R	R	R	R	R	R	R

Initial value : 7FH

XSTPTR Memory extension stack pointer register

11.2 Clock Generator

11.2.1 Overview

As shown in Figure 11.2Table 11.2, the clock generator produces the basic clock pulses which provide the system clock to be supplied to the CPU and the peripheral hardware. The system colock source get from three kinds of resources

- Internal RC OSC (16MHz)
  - INT-RC OSC/1 (16MHz)
  - INT-RC OSC/2 (8MHz, Default system clock)
  - INT-RC OSC/4 (4MHz)
  - INT-RC OSC/8 (2MHz)
  - INT-RC OSC/16 (1MHz)
- Internal WDT RING OSC (256kHz)
- External crystal at Xin and Xout pin (1~ 16MHz)
- Sub crystal at SUBxin and SUBxout pins (32.768kHz, the resource of real time clock especially)

The default system clock after power on is RC OSC Oscillator(8Mhz). It is clock resource to have quickly stable oscillation at POR or RESET active and provide system clock source to CPU and an other peripheral block. It is also clock source of BIT, WDT and ports de-bounce that control system stabilization. After the system operation start with RC OSC, an application software change system clock source to Main or SUB OSC with clock control registers.

11.2.2 Block Diagram

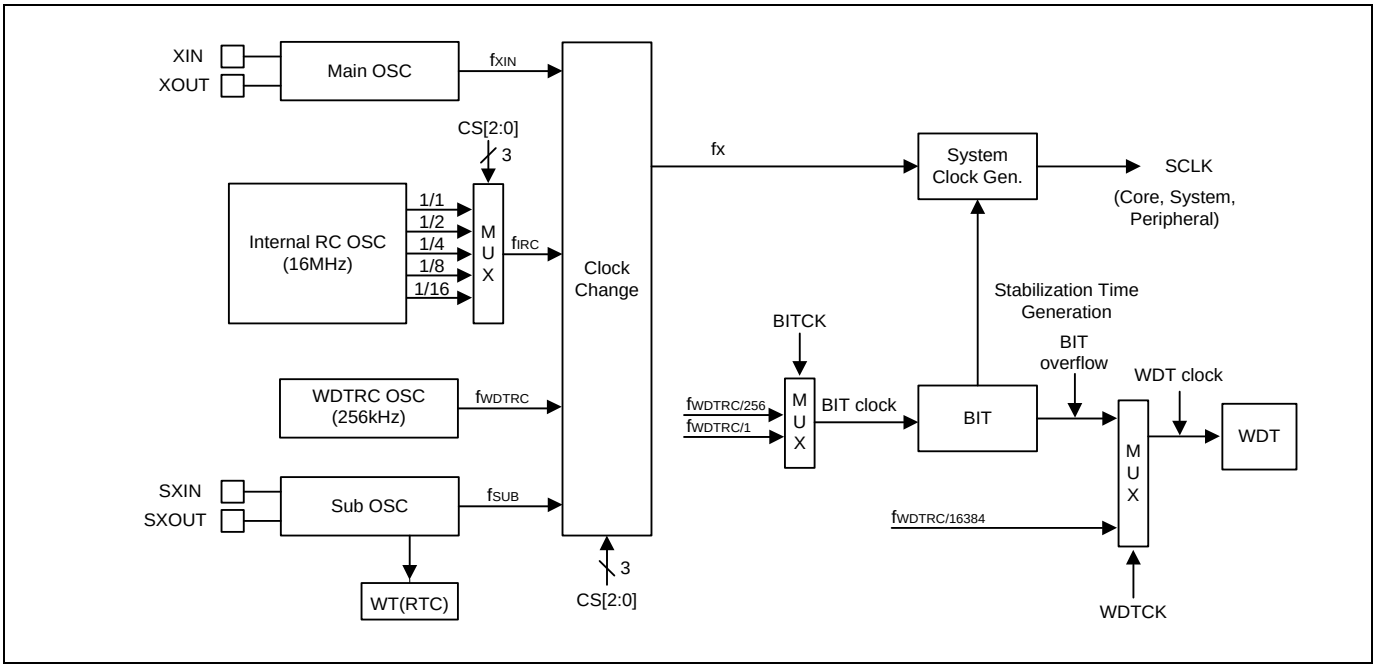


Figure 11.2 Clock Generator Block Diagram

11.2.3 CPU clock and Peripherals Clock

Almost A97C450 blocks (CPU and Peripherals) are operating by System clock but a few of Peripherals block source are directly supplied from clock source.

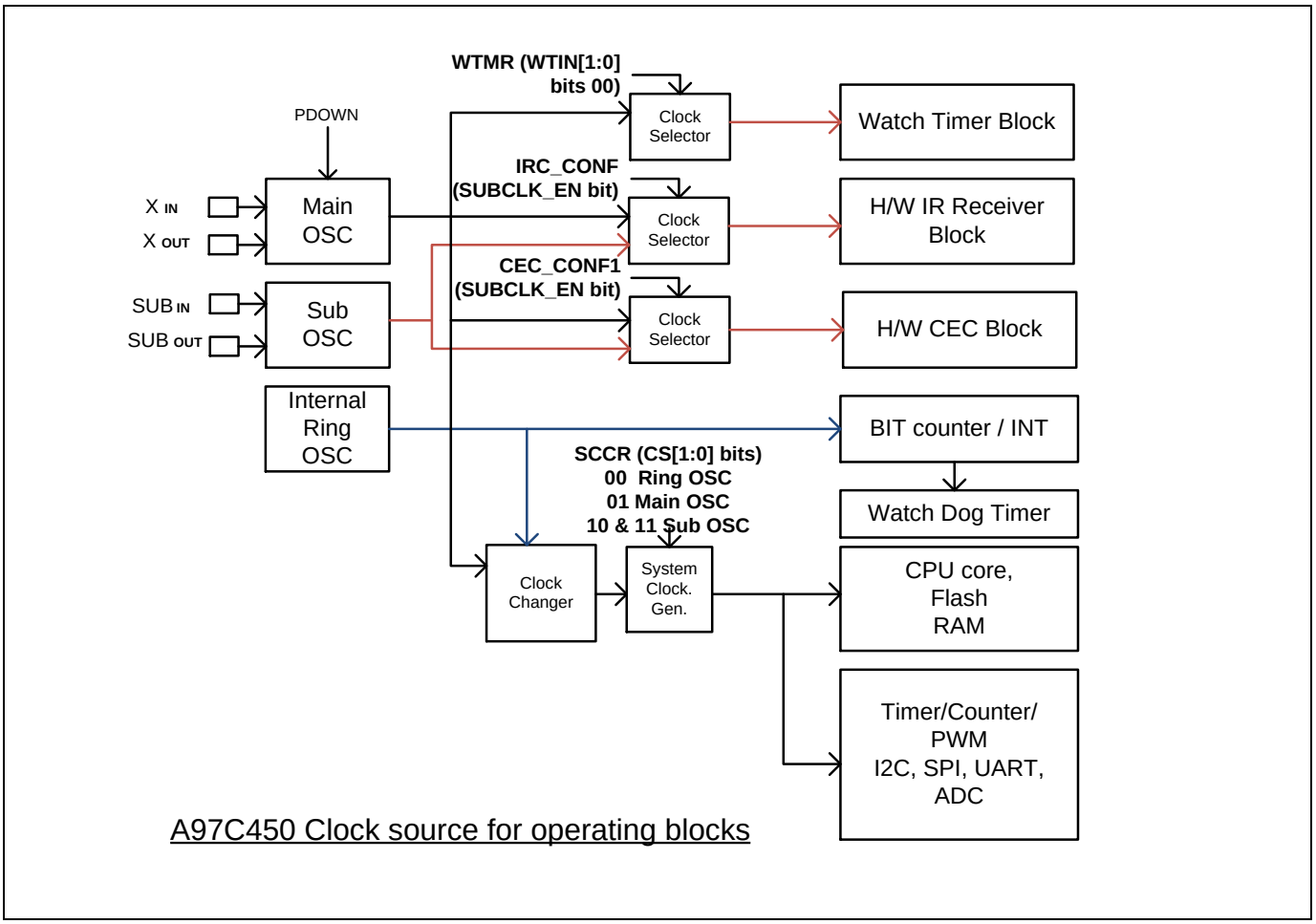


Figure 11.3 Clock source for operating block

The peripheral block that is not operated by system clock source(Fsclk)

- WDT RING OSC (256kHz internal)
  Bit counter block
  Watch Dog Timer block
- SUB clock (32.768Khz)
  Watch Timer and RTC block
  CEC block (optional SUBCLK_EN bit on CEC_CONF1 register (XSFR address : 8F62h)
  H/W IR clock (optional SUBCLK_EN bit on IRC_CONF register (XSFR address : 8F4Ch)
  (In case of Power down mode (STOP mode), wake up source)
- XTAL (only power Stop mode, and enable Main OSC block(XSTOP = '0')
  Watch Timer and RTC block
  CEC block
  H/W IR clock

11.2.4 Register Map

Name	Address	Direction	Default	Description
SCCR	8AH	R/W	41H	System and Clock Control Register

Table 11.2 Clock Generator Register Map

11.2.5 Clock Generator Register Description

The Clock Control Register selects clock source for system operation. This register operation depends on oscillation resource and system configure option register(FUSE_CONFIG).

NOTE)
It should to set SXINEN bit or XINENA bit on FUSE_CONFIG register to use External clock source(Crystal Oscillator (1~16MHz) or/and Sub-Clock Crystal Oscillator (32.768 KHz)).

Especially for Main Crystal Oscillator application, the bit (XOFF) on SCCR has to be "0" (X-Tal Oscillation enable) before changing clock souce.

NOTE)
Before changing system clock source with the bits (CS[1:0]), please check SXINEN, XINENA [bit 4, 5] on FUSE_CONFIG and XOFF [bit 2] on SCCR.

11.2.6 Register Description for Clock Generator

SCCR (System and Clock Control Register) : 8AH

7	6	5	4	3	2	1	0
STOP1	IRCEN	CBYS	XSTOP	SSTOP	CS2	CS1	CS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 41H

STOP1 The operation of RING Oscillation at stop mode
- 0 Ring-Oscillator is disabled at stop mode (=STOP2)
- 1 Ring-Oscillator is enabled at stop mode (=STOP1)

IRCEN Enable IRC 16MHz
- 0 Disable IRC 16MHz
- 1 Enable IRC 16MHz

CBYS It control methods of system clock change.
If this bit is '0', clock change occure on BIT interrupt generation on STOP condition. If this bit is '1', the clock change take place on setting CS[1:0]
NOTE) when clear this bit, keep other bits in SCCR.
- 0 Clock changed by hardware during stop mode (default)
- 1 Clock changed by software

XSTOP Stop main x-tal.
- 0 Do not stop main x-tal.
- 1 Stop main x-tal.

SSTOP Stop sub-xtal.
- 0 Do not stop sub-xtal.
- 1 Stop sub-xtal.

CS[2:0] It select system clock
NOTE) With CBYS bit, reflection point is decided

CS2	CS1	CS0	Description
0	0	0	fIRC (16 MHz)
0	0	1	fIRC/2 (8 MHz)
0	1	0	fIRC/4 (4 MHz)
0	1	1	fIRC/8 (2 MHz)
1	0	0	fIRC/16 (1 MHz)
1	0	1	fXIN Main Clock (1~16MHz)
1	1	0	fSUB clock(32.768kHz)
1	1	1	fRING(256kHz)

Note 1. If XINENA bit in FUSE_CONF is to '0', XOFF fixed to '1'

Note 2. By writing 'A5h' to AUTHORITY register, SCCR changing is possible.

11.3 Basic Interval Timer

11.3.1 Overview

The A97C450 has one 8-bit basic interval timer that is free-run and can't stop. Block diagram is shown in Figure 11.4. In addition, the basic interval timer generates the time base for watchdog timer counting. It also provides a basic interval timer interrupt (BITF).

The A97C450 has these basic interval timer (BIT) features:

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

11.3.2 Block Diagram

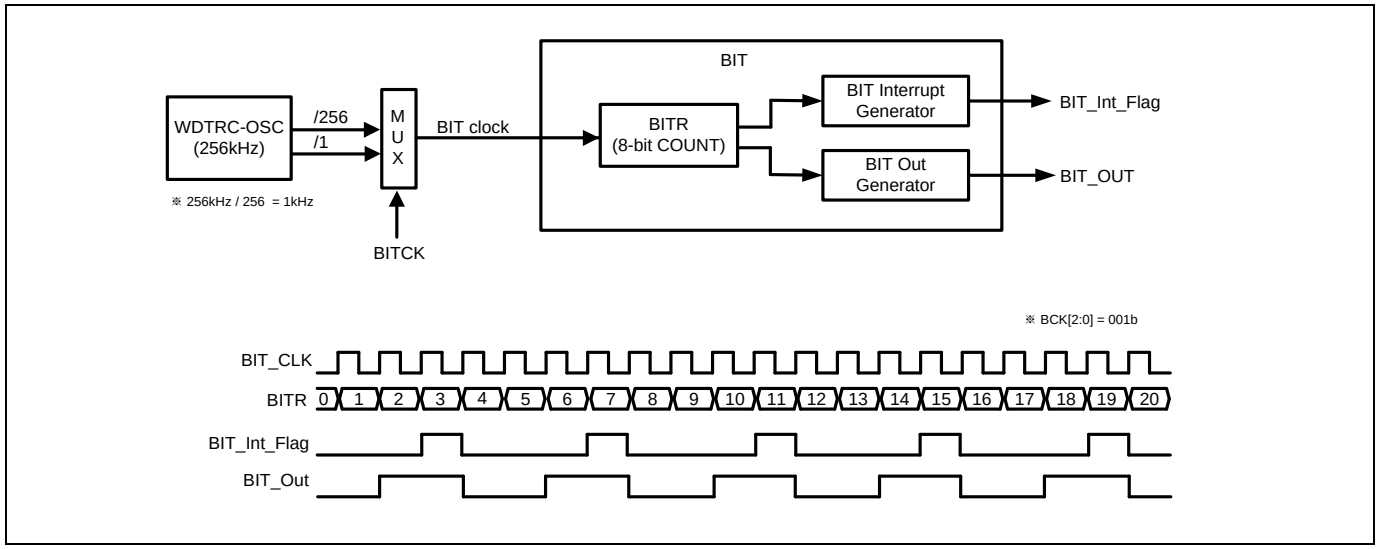


Figure 11.4 Basic Interval Timer Block Diagram

11.3.3 Register Map

Name	Address	Direction	Default	Description
BCCR	8BH	R/W	03H	BIT Clock Control Register
BITR	8CH	R	00H	Basic Interval Timer Register

Table 11.3 Basic Interval Timer Register Map

11.3.4 Basic Interval Timer Register Description

The Bit Interval Timer Register consists of BIT Clock control register (BCCR) and Basic Interval Timer register (BITR). If BCLR bit set to '1', BITR becomes '0' and then counts up. After 1 machine cycle, BCLR bit is cleared as '0' automatically.

11.3.5 Register Description for Basic Interval Timer

BCCR (BIT Clock Control Register) : 8BH

7	6	5	4	3	2	1	0
BITF	-	-	BITCK	BCLR	BCK2	BCK1	BCK0
R/W	R	R	R/W	R/W	R/W	R/W	R/W

Initial value : 03H

BITF — When BIT Interrupt occurs, this bit becomes '1'. For clearing bit, write '0' to this bit.
- 0 no generation
- 1 generation

BITCK — Select BIT clock source
- 0 f_{WDTRC} / 256
- 1 f_{WDTRC} / 1

BCLR — If BCLR Bit is written to '1', BIT Counter is cleared as '0'
- 0 Free Running
- 1 Clear Counter

BCK[2:0] — Select BIT overflow period (BIT Clock ≒ 1 KHz)

BCK2	BCK1	BCK0	
0	0	0	2msec (BIT Clock * 2)
0	0	1	4msec
0	1	0	8msec
0	1	1	16msec (default)
1	0	0	32msec
1	0	1	64msec
1	1	0	128msec
1	1	1	256msec

Note 1. By writing 'A5h' to AUTHORITY register, BCCR changing is possible.

BITR (Basic Interval Timer Register) : 8CH

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

Initial value : 00H

BIT[7:0] BIT Counter

11.4 Watch Dog Timer

11.4.1 Overview

The watchdog timer rapidly detects the CPU malfunction such as endless looping caused by noise or the like, and resumes the CPU to the normal state. The watchdog timer signal for detecting malfunction can be selected either a reset CPU or an interrupt request. When the watchdog timer is not being used for malfunction detection, it can be used as a timer to generate an interrupt at fixed intervals. It is possible to use free running 8-bit timer mode (WDTRSON='0') or watch dog timer mode (WDTRSON='1') as setting WDTMR[6] bit. If writing WDTMR[5] to '1', WDT counter value is cleared and counts up. After 1 machine cycle, this bit has '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 the CPU in accordance with the bit WDTRSON.

The clock source of Watch Dog Timer is BIT overflow output. The interval of watchdog timer interrupt is decided by BIT overflow period and WDTR set value. The equation is as below

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

11.4.2 WDT Interrupt Timing Waveform

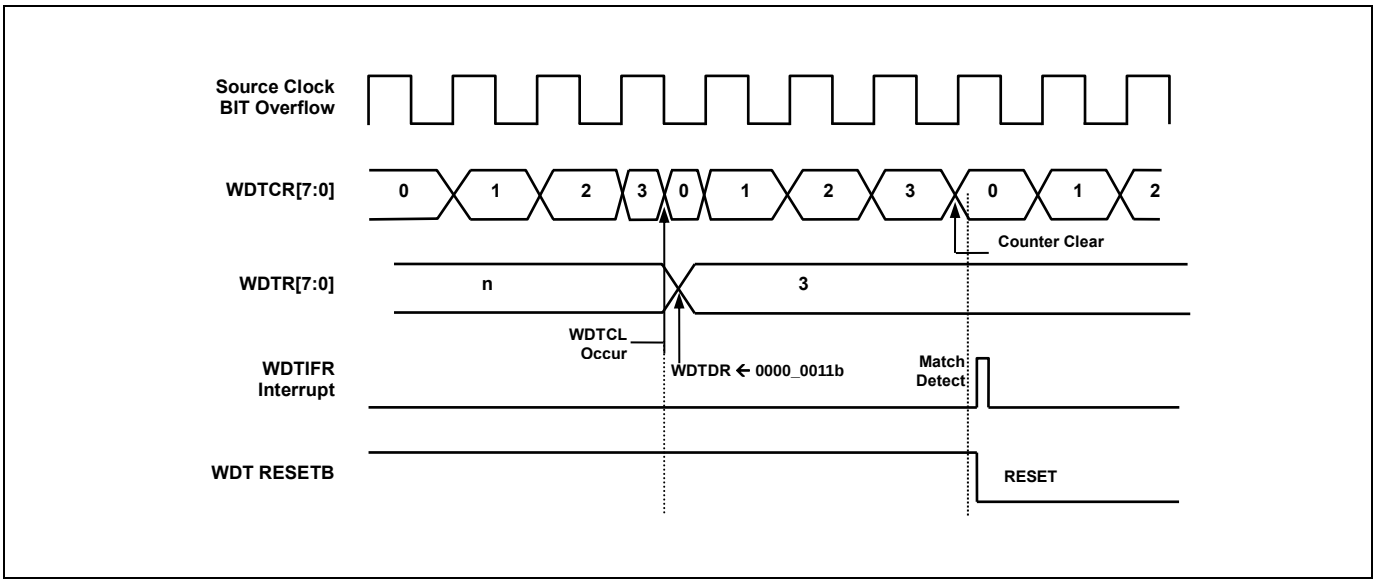


Figure 11.5 Watch Dog Timer Interrupt Timing Waveform

11.4.3 Block Diagram

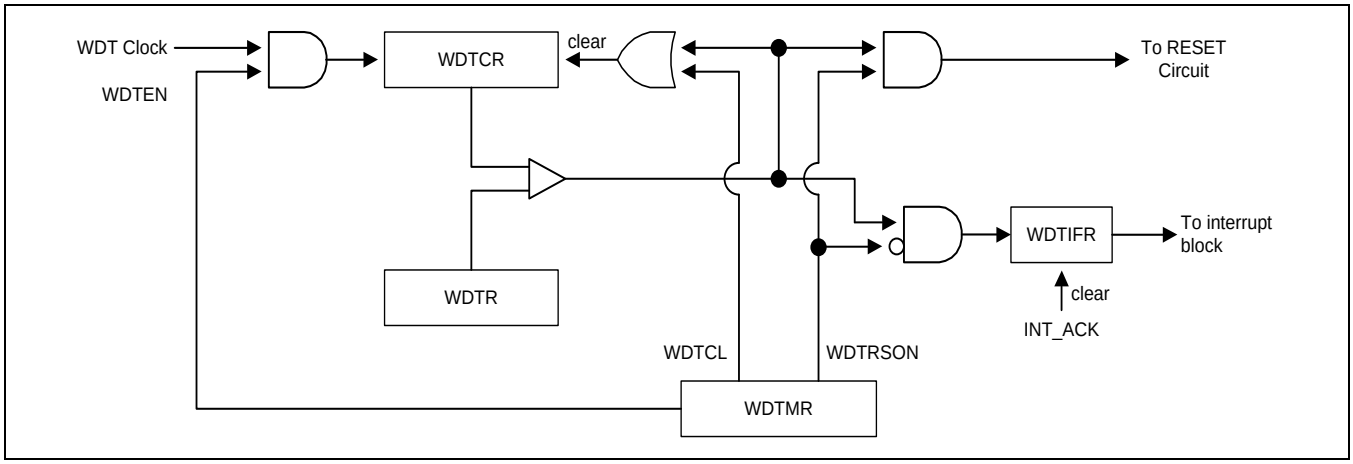


Figure 11.6 Watch Dog Timer Block Diagram

11.4.4 Register Map

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

Table 11.4 Watch Dog Timer Register Map

11.4.5 Watch Dog Timer Register Description

The Watch dog timer (WDT) Register consists of Watch Dog Timer Register (WDTR), Watch Dog Timer Counter Register (WDTCR) and Watch Dog Timer Mode Register (WDTMR).

11.4.6 Register Description for Watch Dog Timer

WDTR (Watch Dog Timer 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=(BIT Interrupt Interval) x(WDTR Value+1)

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

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

WDTMR (Watch Dog Timer Mode Register) : 8DH

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

Initial value : 00H

WDTEN Control WDT operation
0 disable
1 enable

WDTRSON Control WDT Reset operation
0 Free Running 8-bit timer
1 Watch Dog Timer Reset ON

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

WDTCK Control WDT Clock Selection Bit
0 BIT overflow for WDT clock
1 WDTRC / 16384 for WDT clock

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

11.5 Watch Timer

11.5.1 Overview

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

11.5.2 Block Diagram

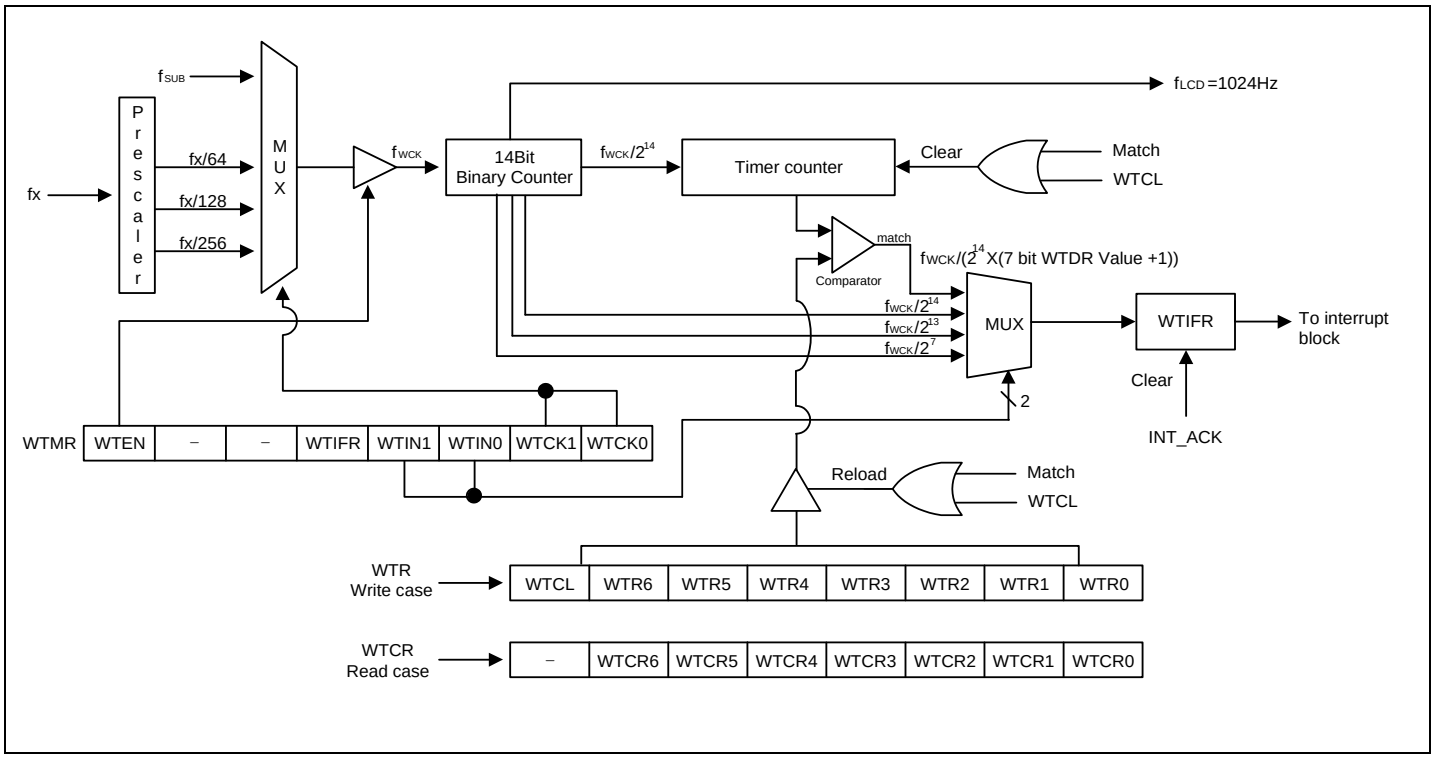


Figure 11.7 Watch Timer Block Diagram

11.5.3 Register Map

Name	Address	Direction	Default	Description
WTMR	9DH	R/W	00H	Watch Timer Mode Register
WTR	9EH	W	7FH	Watch Timer Register
WTCR	9EH	R	00H	Watch Timer Counter Register

Table 11.5 Watch Timer Register Map

11.5.4 Watch Timer Register Description

The watch timer register (WT) consists of Watch Timer Mode Register (WTMR), Watch Timer Counter Register (WTCR) and Watch Timer Register (WTR). As WTMR is 6-bit writable/readable register, WTMR can control the clock source (WTCK), interrupt interval (WTIN) and function enable/disable (WTEN). Also there is WT interrupt flag bit (WTIFR).

11.5.5 Register Description for Watch Timer

WTMR (Watch Timer Mode Register) : 9DH

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

Initial value : 00H

WTEN Control Watch Timer
- 0 disable
- 1 enable

WTIFR When WT Interrupt occurs, this bit becomes '1'. For clearing bit, write '0' to this bit or auto clear by INT_ACK signal.
- 0 WT Interrupt no generation
- 1 WT Interrupt generation

WTIN[1:0] Determine interrupt interval

WTIN1	WTIN0	description
0	0	fwck/2048
0	1	fwck/8192
1	0	fwck/16384
1	1	fwck/16384 x (7bit WT Value)

WTCK[1:0] Determine Source Clock

WTCK1	WTCK0	description
0	0	fsub
0	1	f_{XIN}/256
1	0	f_{XIN}/128
1	1	f_{XIN}/64

Remark: f_{XIN}– Main clock oscillation frequency
fsub- Sub clock oscillation frequency
fwck- selected Watch Timer clock

WTR (Watch Timer Register: Write Case) : 9EH

7	6	5	4	3	2	1	0
WTCL	WTR6	WTR5	WTR4	WTR3	WTR2	WTR1	WTR0
W	W	W	W	W	W	W	W

Initial value : 7FH

WTCL Clear WT Counter
0 Free Run
1 Clear WT Counter (auto clear after 1 Cycle)

WTR[6:0] Set WT period
WT Interrupt Interval=(fwck/2^14) x(7bit WT Value+1)

NOTE) To guarantee proper operation, it is greater than 01H to write WTR.

WTCR (Watch Timer Counter Register: Read Case) : 9EH

7	6	5	4	3	2	1	0
	WTCR6	WTCR5	WTCR4	WTCR3	WTCR2	WTCR1	WTCR0
-	R	R	R	R	R	R	R

Initial value : 00H

WTCR[6:0] WT Counter

11.6 Timer 0/1

11.6.1 Overview

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

It has three operating modes:

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

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

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

In the capture mode, by EINT0/1, the data is captured into input capture data register (TnCDR). In timer/counter mode, whenever counter value is equal to TnDR, TnO port toggles. Also the timer 0/1 outputs PWM waveform through PWMnO port in the PWM mode.

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

Table 11.6 Timer 0/1 Operating Modes

11.6.2 8-Bit Timer/Counter Mode

The 8-bit timer/counter mode is selected by control register as shown in Figure 11.8.
The 8-bit timer have counter and data register. The counter register is increased by internal or external clock input. Timer 0/1 can use the input clock with one of 2, 4, 8, 32, 128, 512 and 2048 prescaler division rates (TnCK[2:0]). When the value of TnCNT and TnDR is identical in timer 0/1, a match signal is generated and the interrupt of Timer n occurs. TnCNT value is automatically cleared by match signal. It can be also cleared by software (TnCC).
The external clock (ECn) counts up the timer at the rising edge. If the ECn is selected as a clock source by TnCK[2:0], EC0/EC1 port is automatically set to the input port.

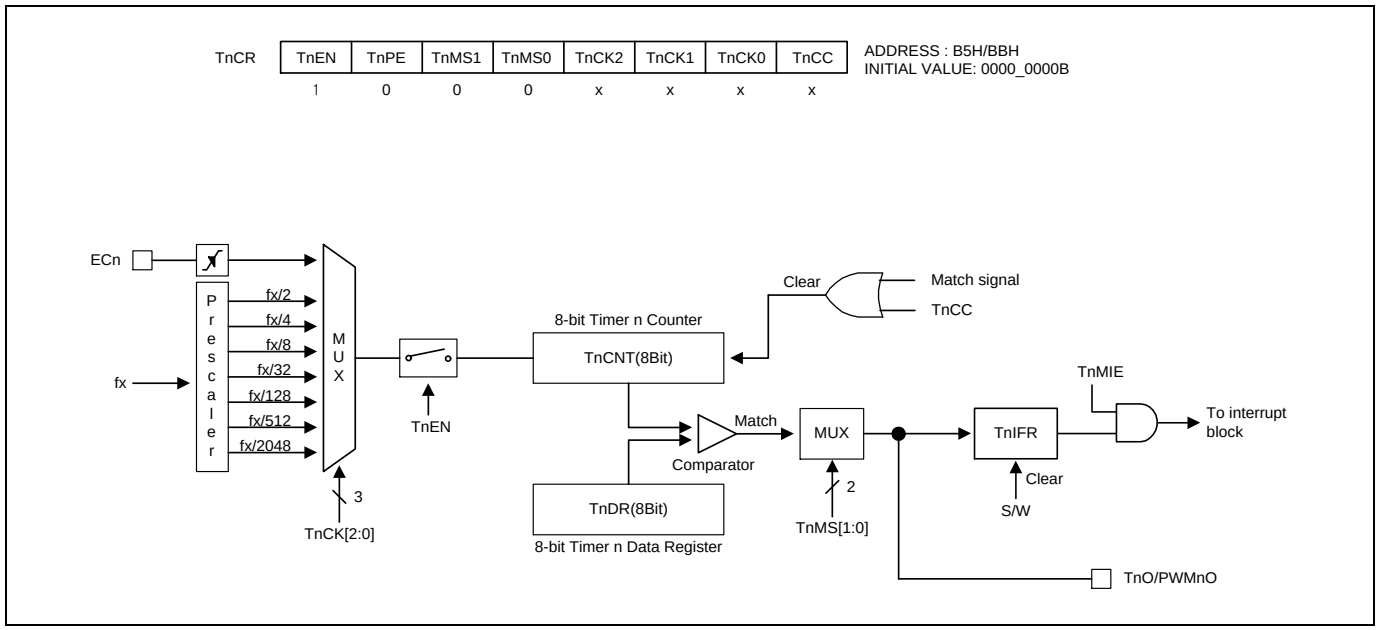


Figure 11.8 8-Bit Timer/Counter Mode for Timer 0/1 (Where n = 0 and 1)

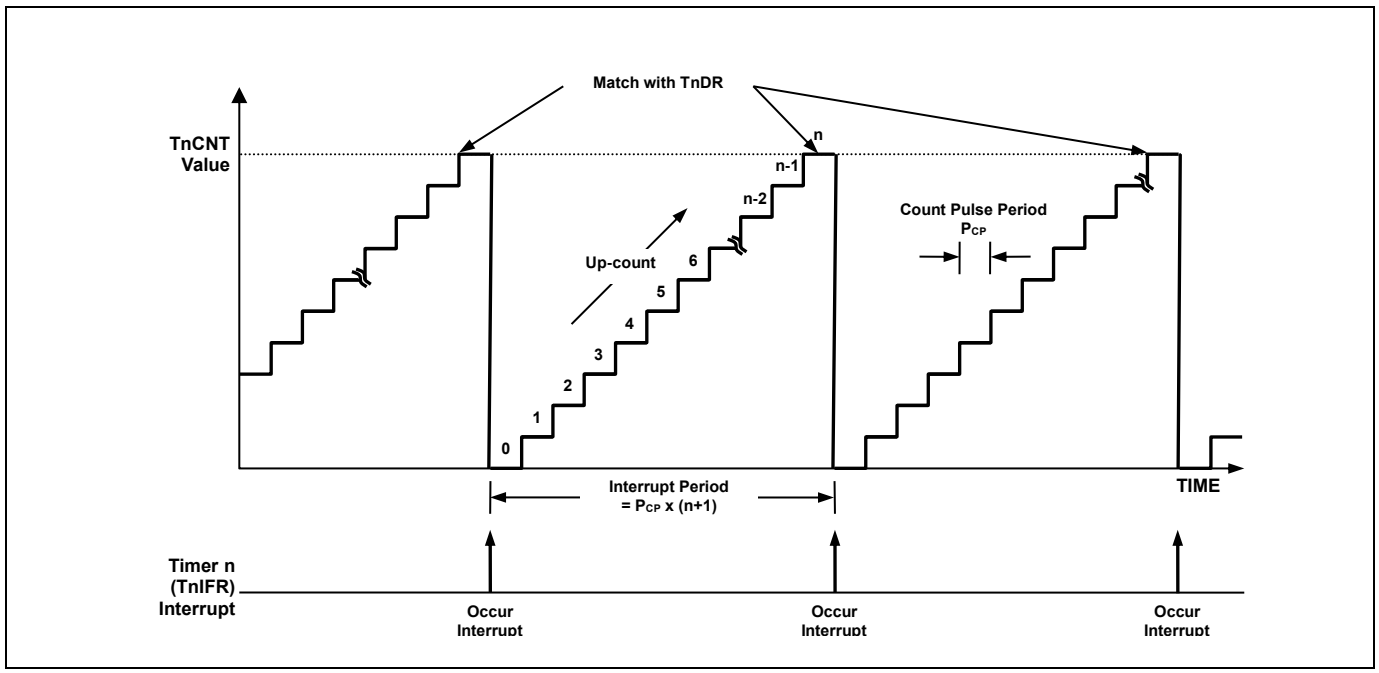


Figure 11.9 8-Bit Timer/Counter 0/1 Example (Where n = 0 and 1)

11.6.3 8-Bit PWM Mode

The timer 0/1 has a high speed PWM (Pulse Width Modulation) function. In PWM mode, TnO/PWMnO pin outputs up to 8-bit resolution PWM output. The PWM output pin is configured by setting the TnPE bit to 1. In the 8-bit timer/counter mode, a match signal is generated when the counter value is identical to the value of TnDR. When the value of TnCNT and TnDR is identical in timer n, a match signal is generated and the interrupt of timer 0/1 occurs. In PWM mode, the match signal does not clear the counter. Instead, it runs continuously, overflowing at “FFH” and then continues incrementing from “00H”. The timer 0/1 overflow interrupt is generated whenever a counter overflow occurs. TnCNT value is cleared by software (TnCC) bit.

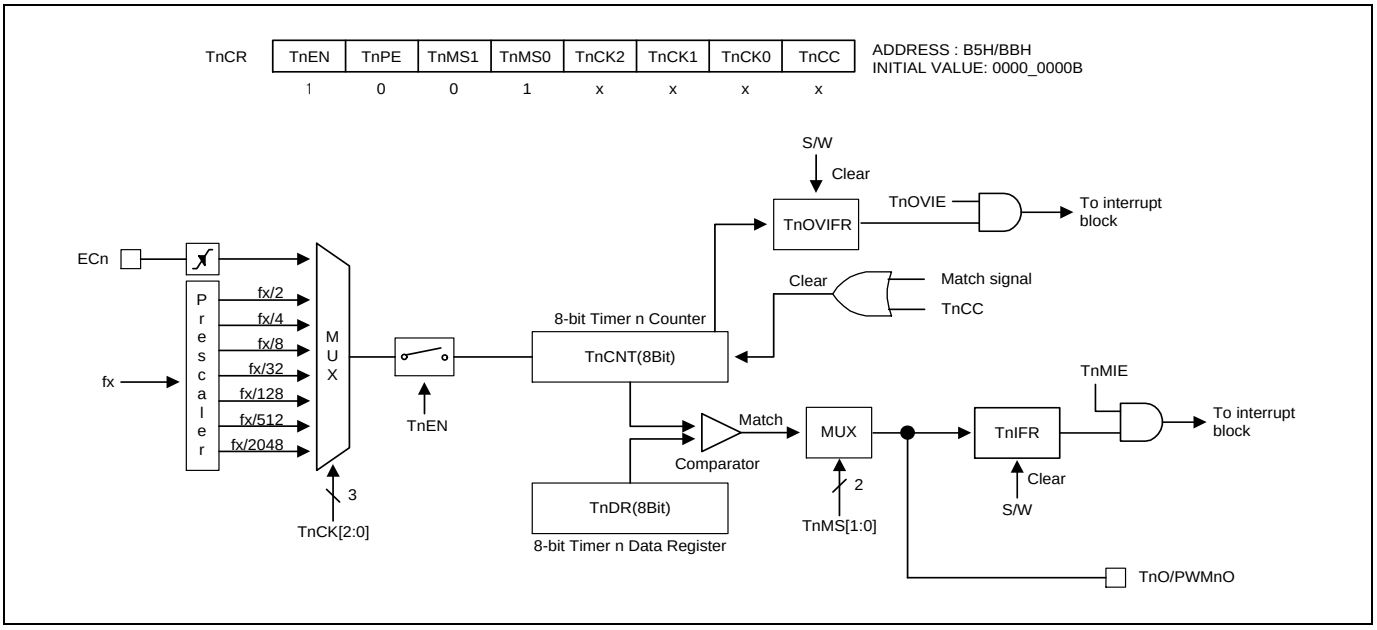


Figure 11.10 8-Bit PWM Mode for Timer 0/1 (Where n = 0 and 1)

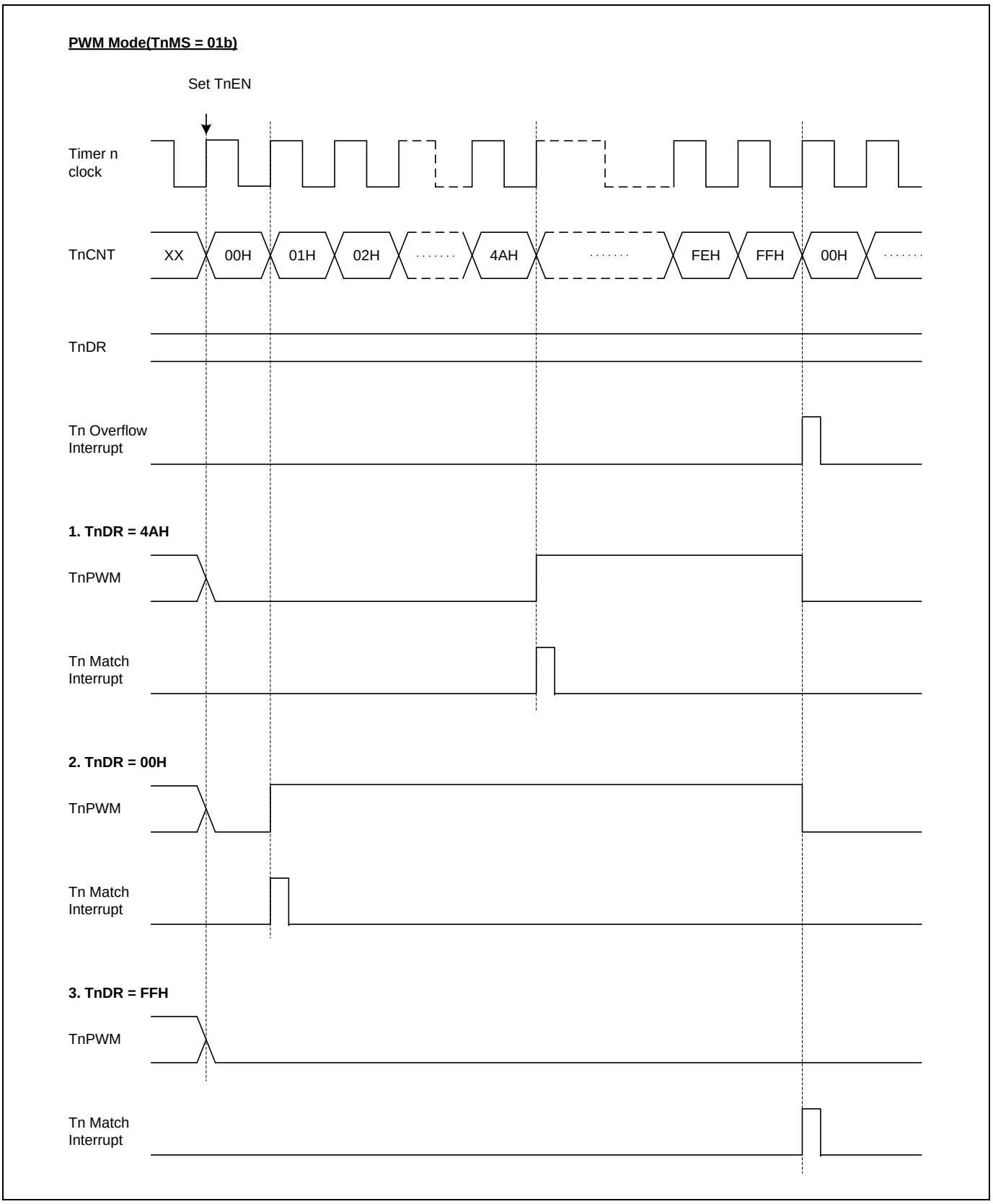


Figure 11.11 PWM Output Waveforms in PWM Mode for Timer 0/1 (Where n = 0 and 1)

11.6.4 8-Bit Capture Mode

The timer 0/1 capture mode is set by TnMS[1:0] as '1x'. The clock source can use the internal/external clock. Basically, it has the same function as the 8-bit timer/counter mode and the interrupt occurs when TnCNT is equal to TnDR. TnCNT value is automatically cleared by match signal and it can be also cleared by software (TnCC). This timer interrupt in capture mode is very useful when the pulse width of captured signal is wider than the maximum period of timer.

The capture result is loaded into TnCDR. In the timer n capture mode, timer n output (TnO) waveform is not available. According to EIEDGE, EIPOLA registers setting, the external interrupt EINTn function is chosen. Of course, the EINTn pin must be set to an input port.

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

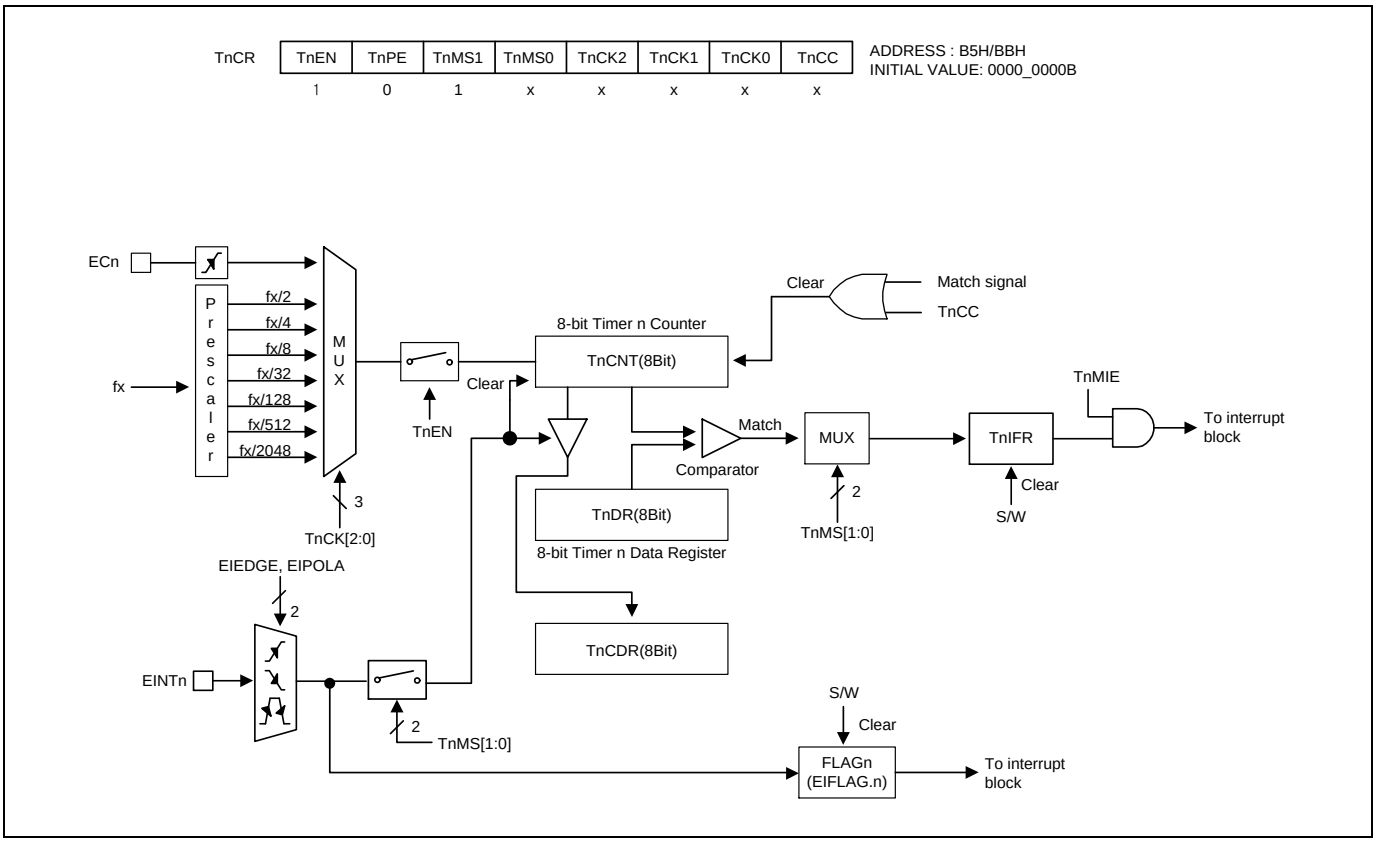


Figure 11.12 8-Bit Capture Mode for Timer 0/1 (Where n = 0 and 1)

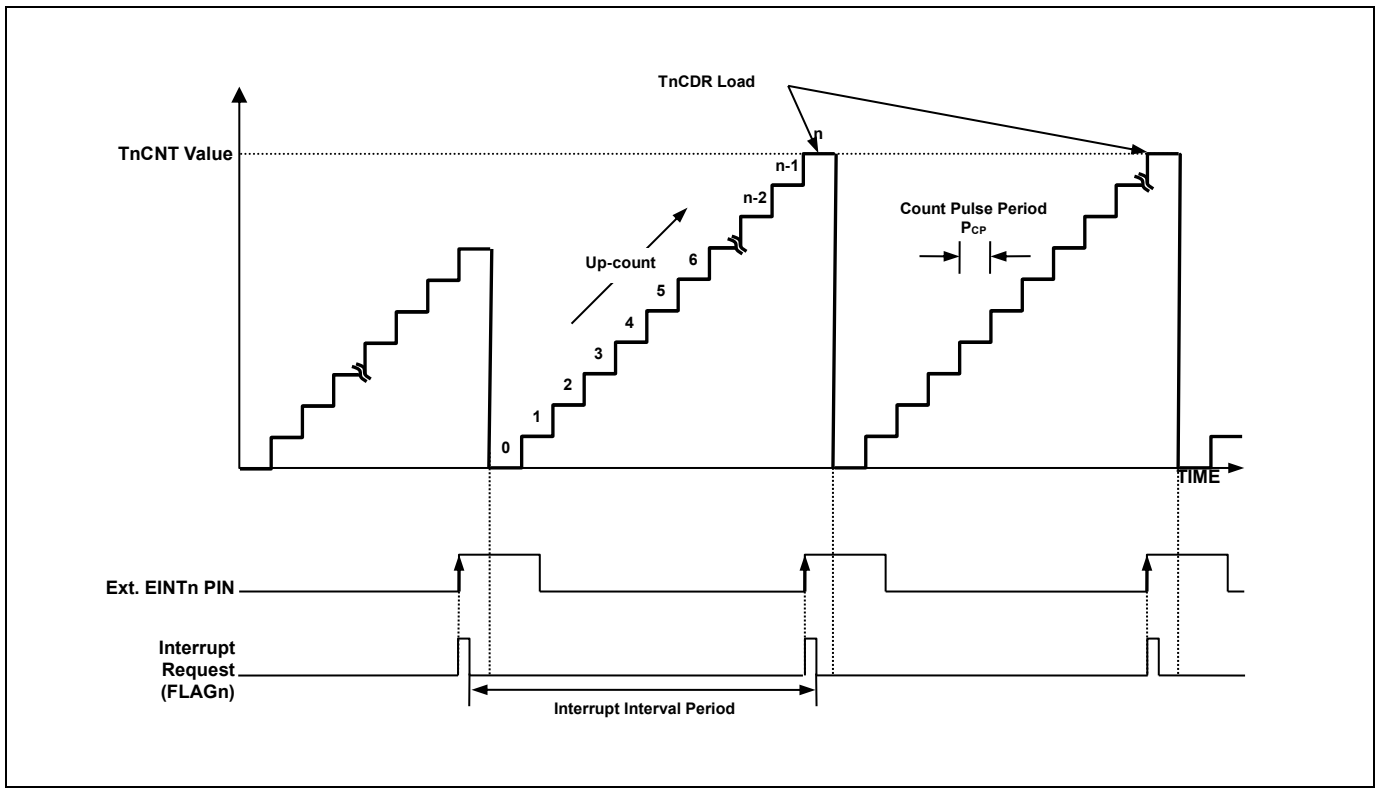


Figure 11.13 Input Capture Mode Operation for Timer 0/1/2 (Where n = 0, 1 and 2)

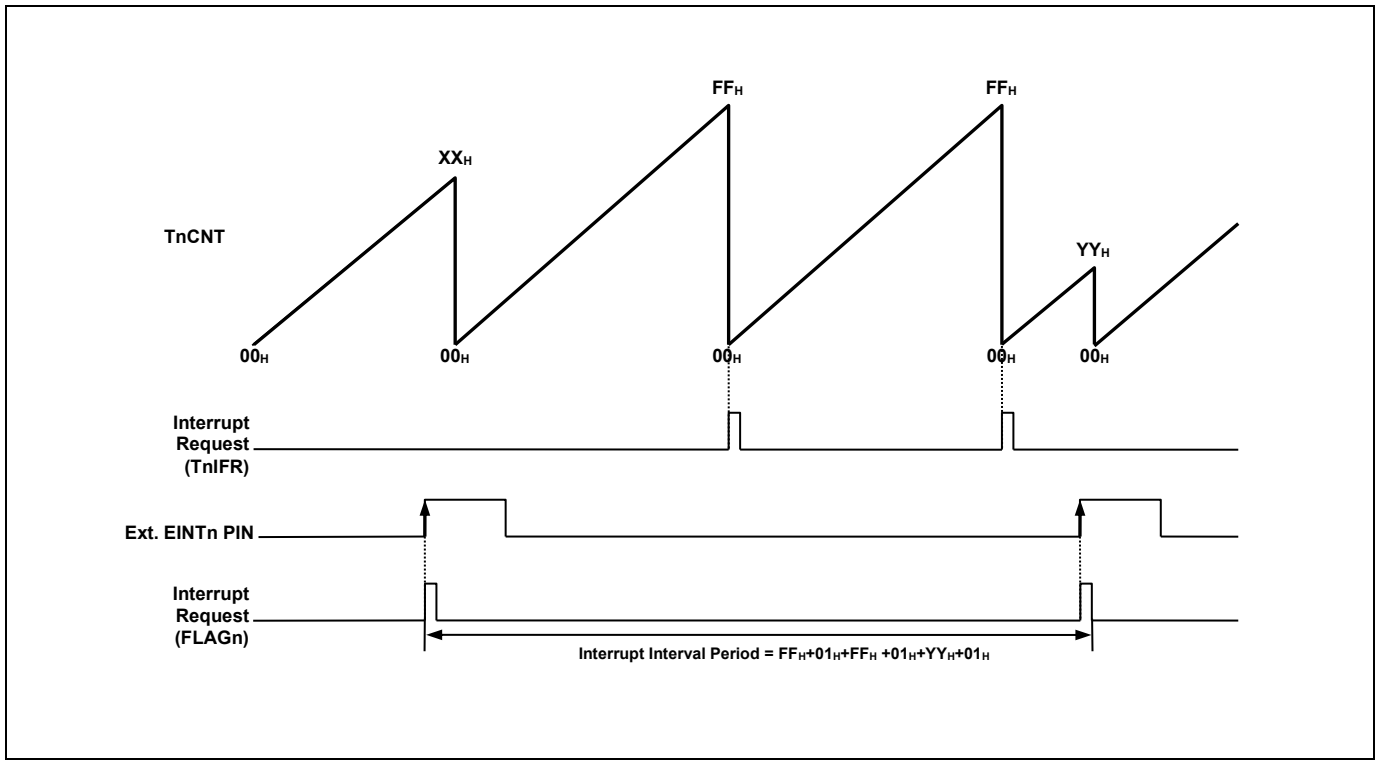


Figure 11.14 Express Timer Overflow in Capture Mode (Where n = 0 and 1)

11.6.5 Block Diagram

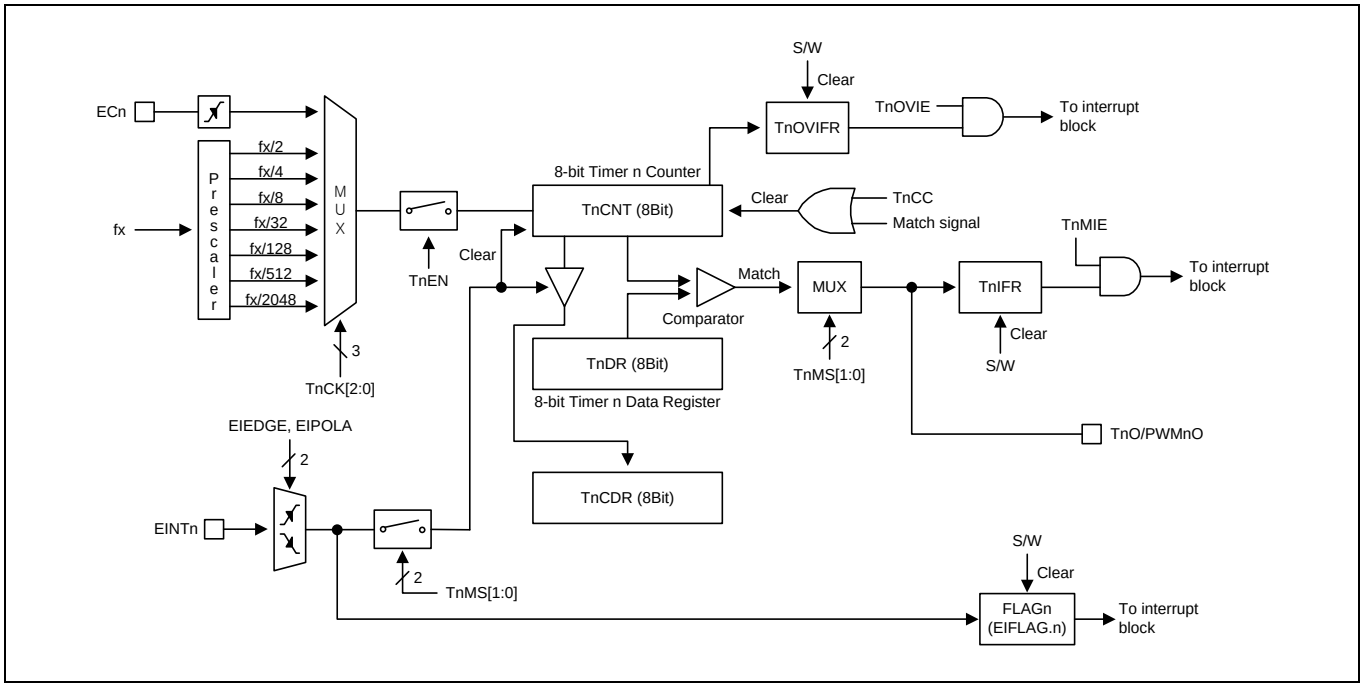


Figure 11.15 8-Bit Timer 0/1/2 Block Diagram (Where n = 0 and 1)

11.6.6 Register Map

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

Table 11.7 Timer 0/1 Register Map

11.6.7 Timer/Counter 0 Register Description

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

11.6.8 Register Description for Timer/Counter 0/1

TnCNT (Timer n Counter Register) : B3H/B7H, n= 0 and 1

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

Initial value : 00H

TnCNT[7:0] Tn Counter

TnDR (Timer n Data Register : Write only when it is capture mode) : B4H/BAH, n= 0 and 1

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

Initial value : FFH

TnDR[7:0] Tn Data

TnCDR (Timer n Capture Data Register : Capture mode only) : B4H/BAH, n= 0 and 1

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

Initial value : 00H

TnCDR[7:0] Tn Capture Data

TnCR (Timer n Control Register) : B5H/BBH, n= 0 and 1

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

Initial value : 00H

TnEN Control Timer n

0 Timer n disable

1 Timer n enable

TnPE Configure Timer n PWM output port.

0 GPIO

1 PWM output

TnMS[1:0] Control Timer n Operation Mode

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

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

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

TnCC Clear timer n Counter

0 No effect

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

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

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

TINTCR(Timer Interrupt Control Register) : B2H

7	6	5	4	3	2	1	0
-	-	T1MIE	T0MIE	-	-	T1OVIE	T0OVIE
-	-	R/W	R/W	-	-	R/W	R/W

Initial value : 00H

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

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

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

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

TIFLAG0(Timer Interrupt Flag Register) : B6H

7	6	5	4	3	2	1	0
-	T4IFR	T4IFR	T4IFR	T1OVIFR	T1IFR	T0OVIFR	T0IFR
-	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

T6IFR When Timer 6 match interrupt occurs, this bit becomes '1'. The flag is cleared only by writing a '0' to the bit. So, the flag should be cleared by software. Writing "1" has no effect.

0 Timer 6 match interrupt no generation

1 Timer 6 match interrupt generation

T5IFR When Timer 5 match interrupt occurs, this bit becomes '1'. The flag is cleared only by writing a '0' to the bit. So, the flag should be cleared by software. Writing "1" has no effect.

0 Timer 5 match interrupt no generation

1 Timer 5 match interrupt generation

T4IFR When Timer 4 match interrupt occurs, this bit becomes '1'. The flag is cleared only by writing a '0' to the bit. So, the flag should be cleared by software. Writing "1" has no effect.

0 Timer 4 match interrupt no generation

1 Timer 4 match interrupt generation

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

0 Timer 1 overflow interrupt no generation

1 Timer 1 overflow interrupt generation

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

0 Timer 1 match interrupt no generation

1 Timer 1 match interrupt generation

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

0 Timer 0 overflow interrupt no generation

1 Timer 0 overflow interrupt generation

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

0 Timer 0 match interrupt no generation

1 Timer 0 match interrupt generation

11.7 Timer 2/3

11.7.1 Overview

The 16-bit timer 2/3 consists of multiplexer, timer 2/3 A data register high/low, timer 2/3 B data register high/low and timer 2/3 control register high/low (TnADRH, TnADRL, TnBDRH, TnBDRL, TnCRH, TnCRL).

It has four operating modes:

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

The timer/counter 2/3 can be clocked by an internal or an external clock source (ECn). The clock source is selected by clock selection logic which is controlled by the clock selection bits (TnCK[2:0]).
- TIMER n clock source: fX/1, 2, 4, 8, 64, 512, 2048 and ECn

In the capture mode, by EINT1n, the data is captured into input capture data register (TnBDRH/TnBDRL). Timer 2/3 outputs the comparison result between counter and data register through TnO port in timer/counter mode. Also Timer 2/3 outputs PWM wave form through PWMnO port in the PPG mode.

TnEN	TnMS[1:0]	TnCK[2:0]	Timer 2/3
1	00	XXX	16 Bit Timer/Counter Mode
1	01	XXX	16 Bit Capture Mode
1	10	XXX	16 Bit PPG Mode (one-shot mode)
1	11	XXX	16 Bit PPG Mode (repeat mode)

Table 11.8 Timer 2/3 Operating Modes

11.7.2 16-bit Timer/Counter Mode

The 16-bit timer/counter mode is selected by control register as shown in Figure 11.14.
The 16-bit timer have counter and data register. The counter register is increased by internal or external clock input. Timer 2/3 can use the input clock with one of 1, 2, 4, 8, 64, 512 and 2048 prescaler division rates (TnCK[2:0]). When the value of TnCNTH, TnCNTL and the value of TnADRH, TnADRL are identical in Timer 2/3 respectively, a match signal is generated and the interrupt of Timer 2/3 occurs. The TnCNTH, TnCNTL value is automatically cleared by match signal. It can be also cleared by software (TnCC).
The external clock (ECn) counts up the timer at the rising edge. If the ECn is selected as a clock source by TnCK[2:0], ECn port is automatically set to the input port.

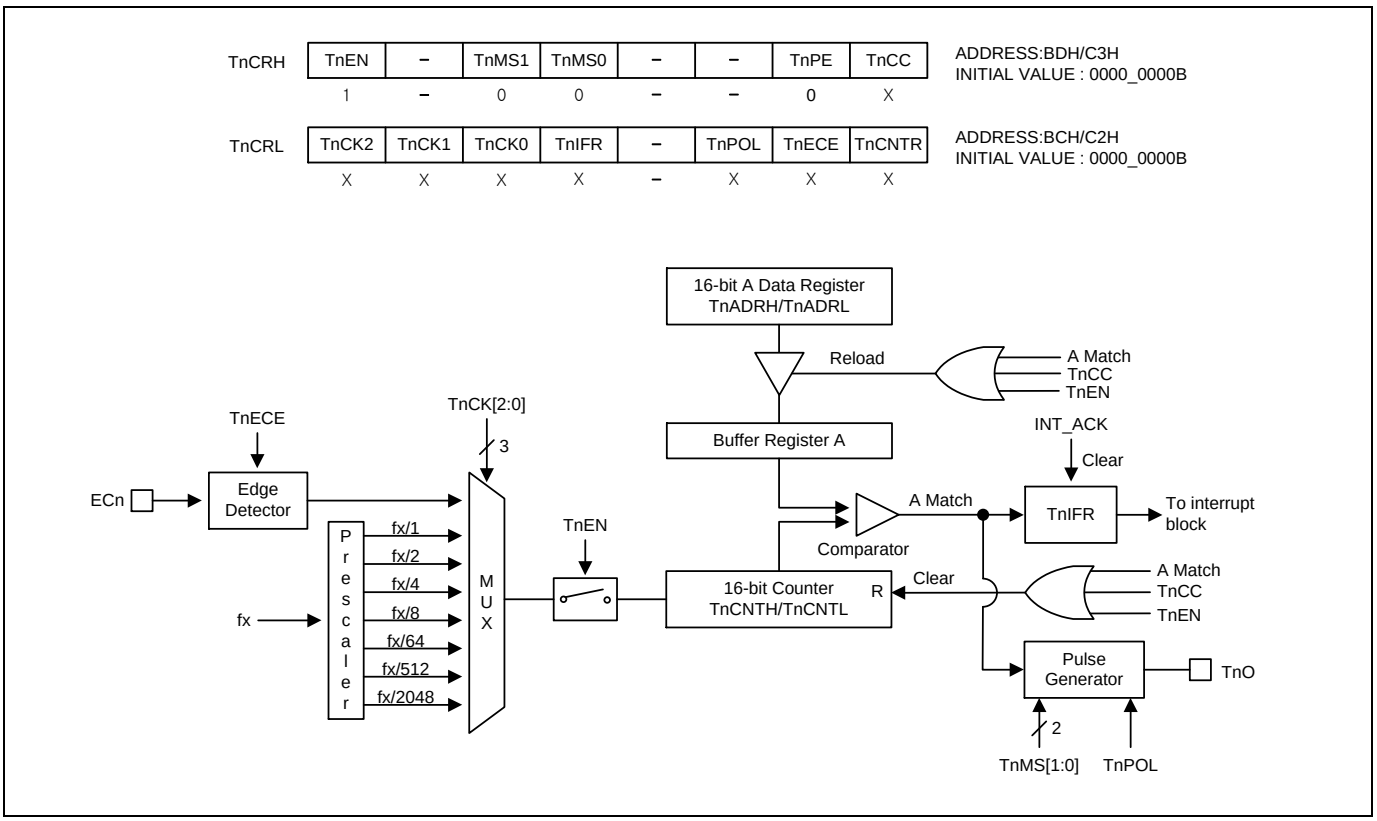


Figure 11.16 16-bit Timer/Counter Mode for Timer 2/3 (where n= 2 and 3)

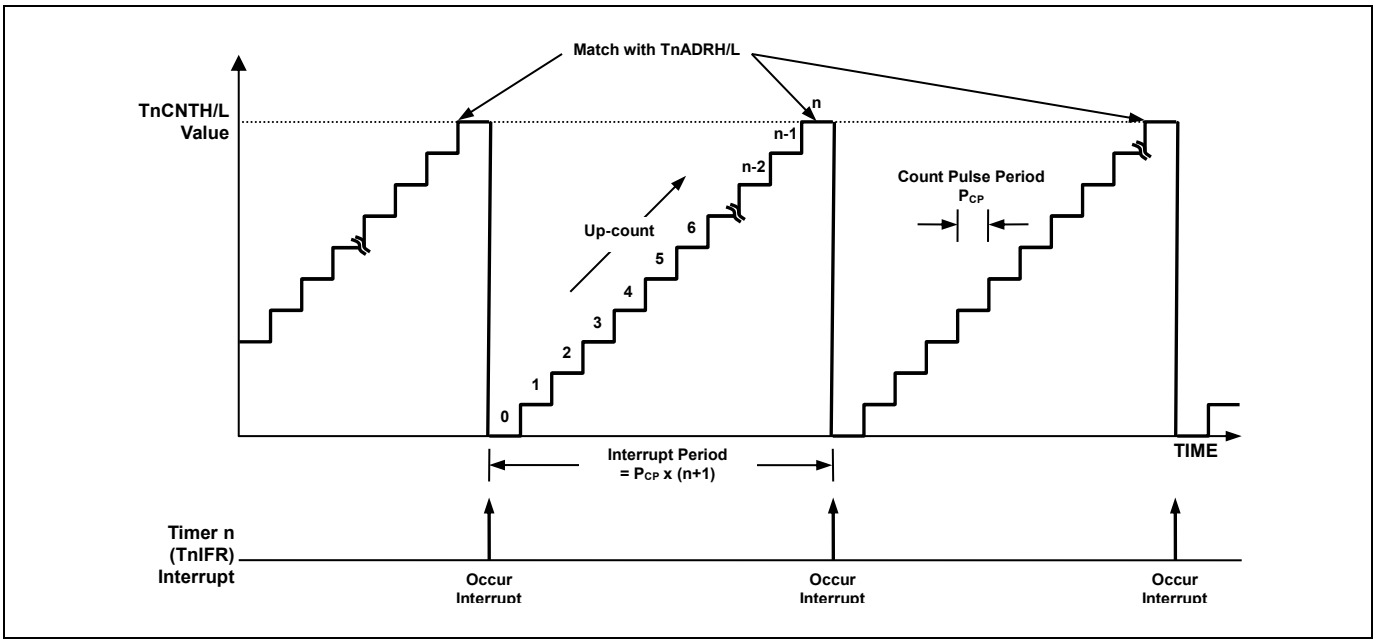


Figure 11.17 16-bit Timer/Counter 2/3 Example (where n= 2 and 3)

11.7.3 16-bit Capture Mode

The 16-bit timer 2/3 capture mode is set by TnMS[1:0] as '01'. The clock source can use the internal/external clock. Basically, it has the same function as the 16-bit timer/counter mode and the interrupt occurs when TnCNTH/TnCNTL is equal to TnADRH/TnADRL. The TnCNTH, TnCNTL values are automatically cleared by match signal. It can be also cleared by software (TnCC).

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

The capture result is loaded into TnBDRH/TnBDRL.

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

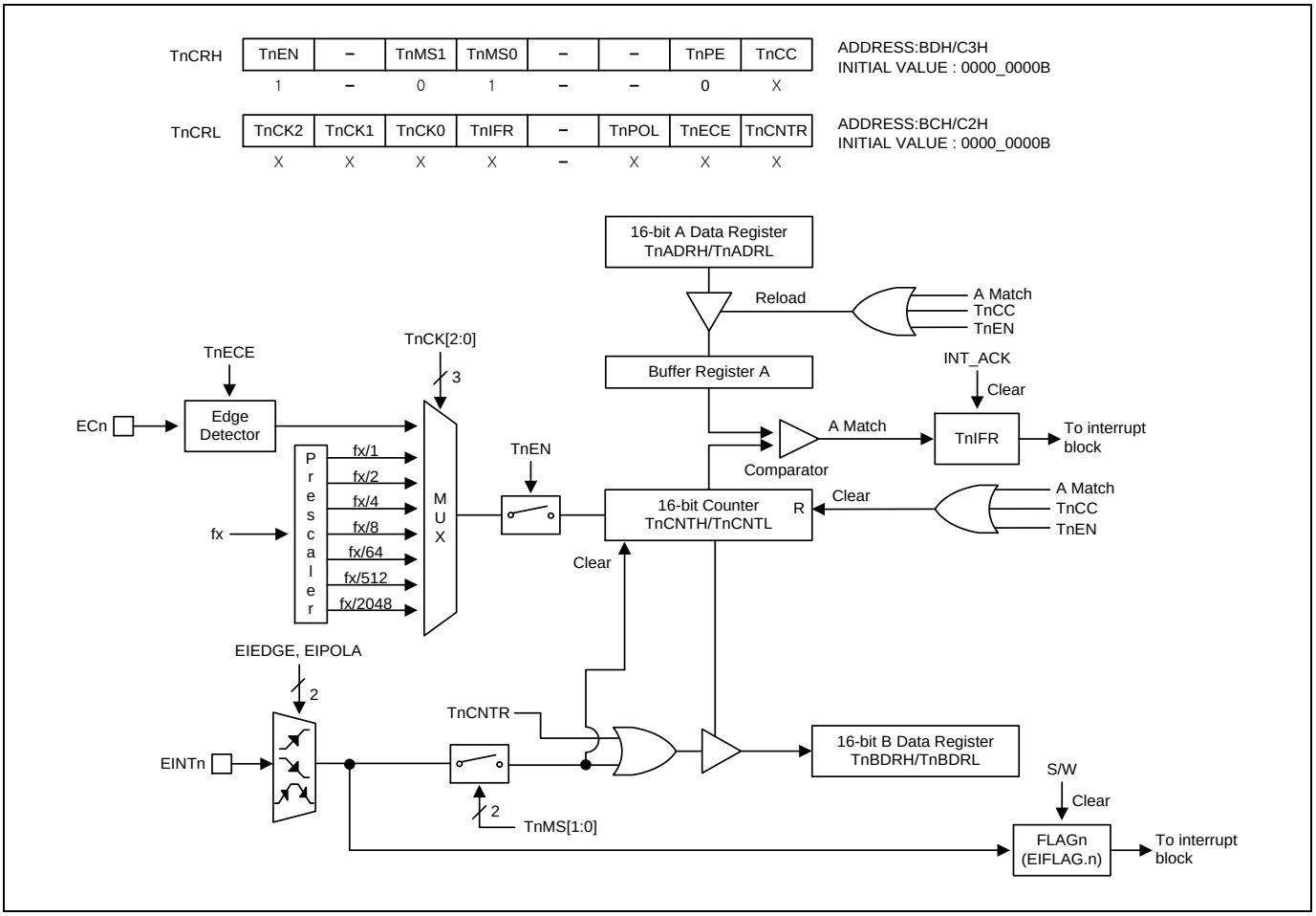


Figure 11.18 16-bit Capture Mode for Timer 2/3 (where n= 2 and 3)

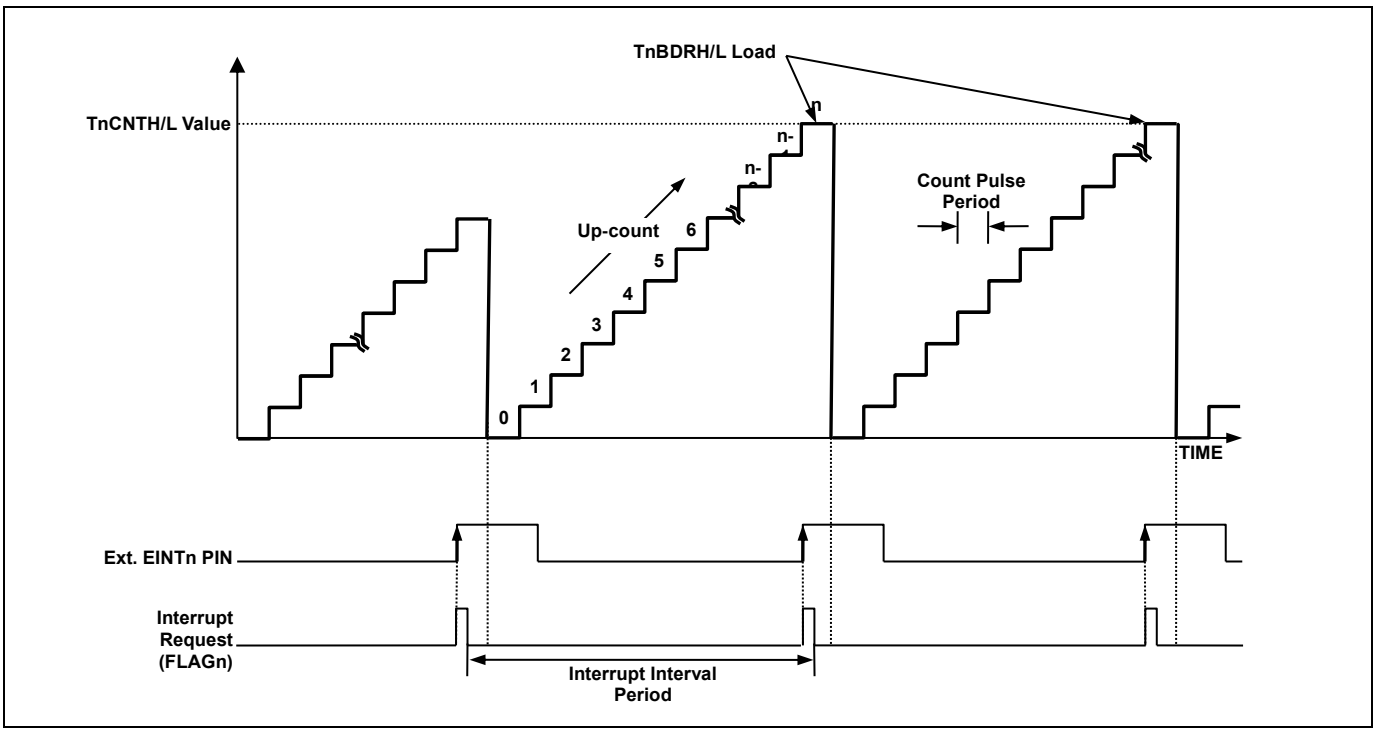


Figure 11.19 Input Capture Mode Operation for Timer 2/3 (where n= 2 and 3)

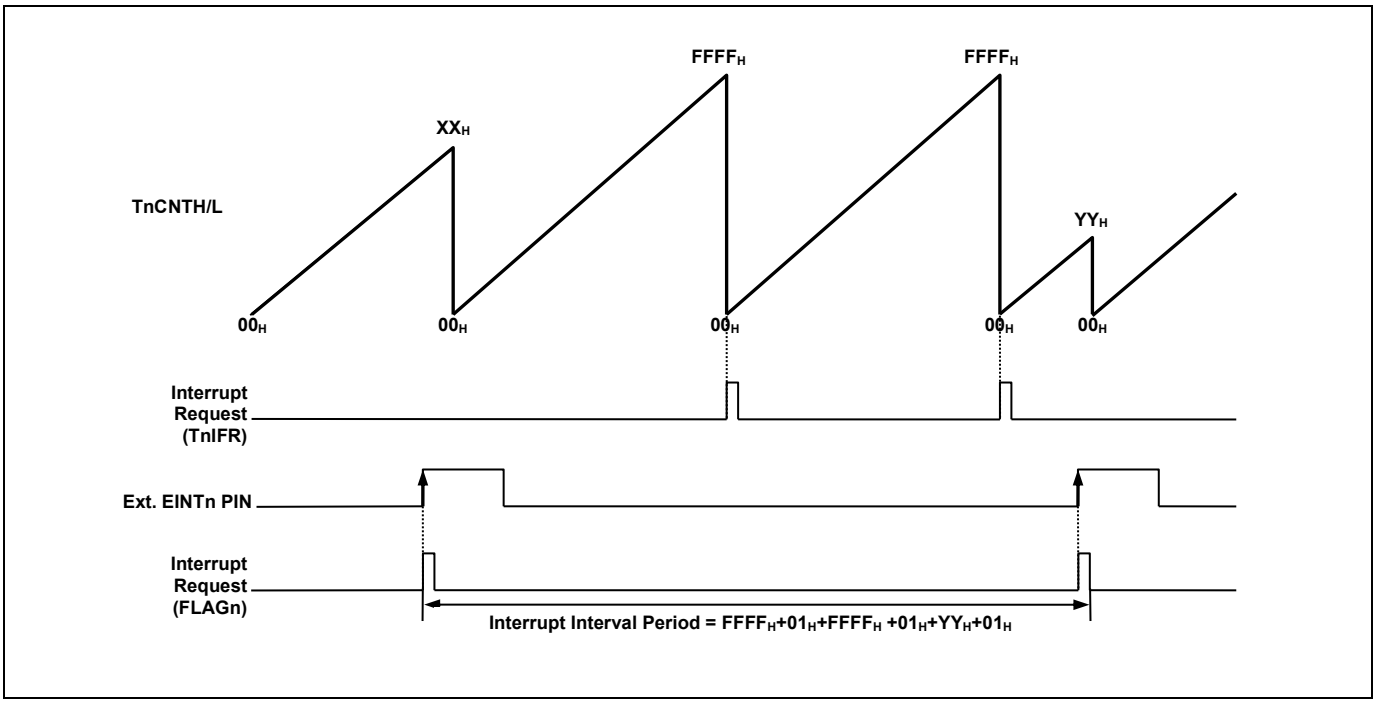
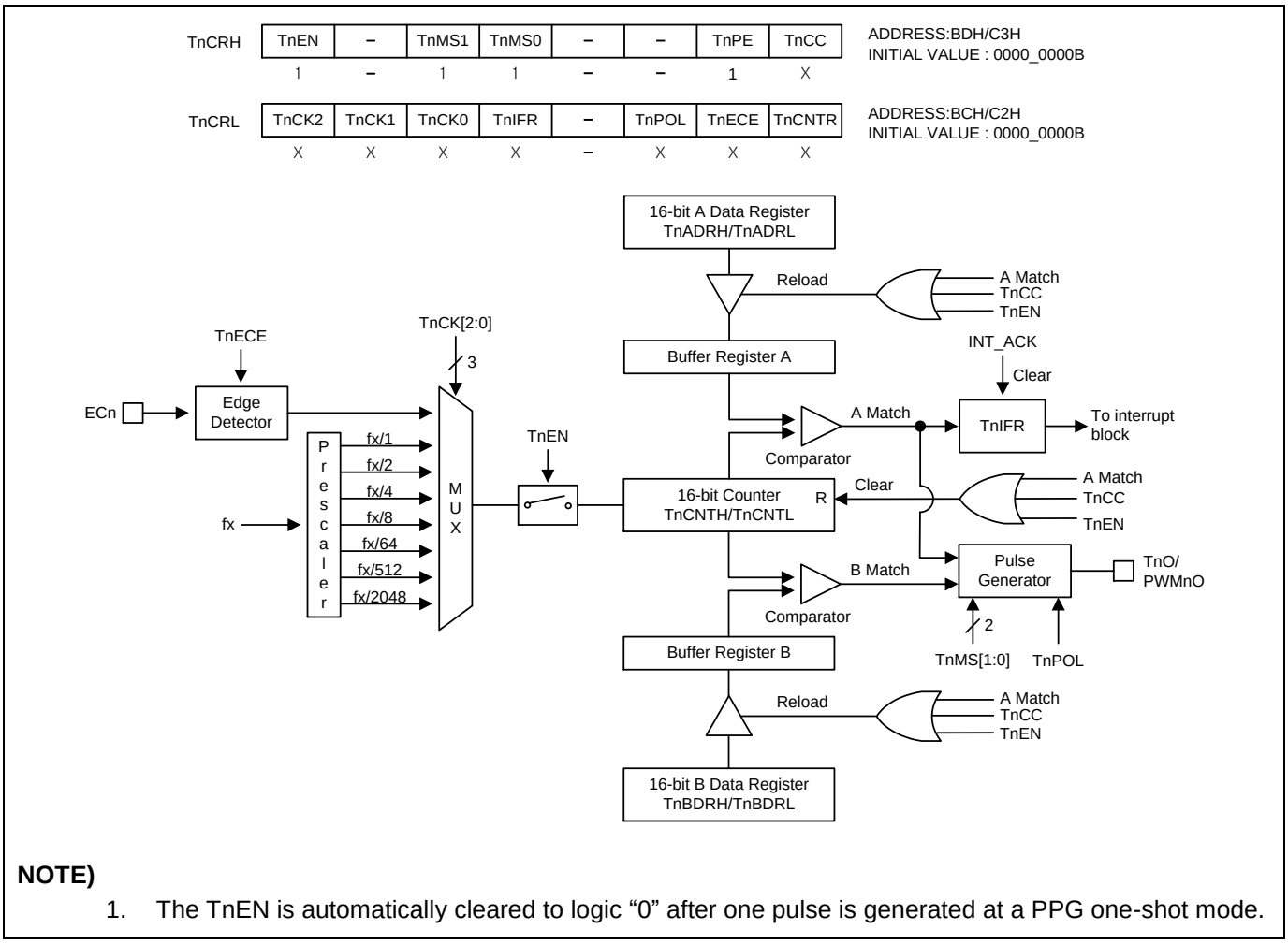


Figure 11.20 Express Timer Overflow in Capture Mode (where n= 2 and 3)

11.7.4 16-bit PPG Mode

The timer 2/3 has a PPG (Programmable Pulse Generation) function. In PPG mode, TnO/PWMnO pin outputs up to 16-bit resolution PWM output. This pin should be configured as a PWM output by setting TnPE bit to '1'. The period of the PWM output is determined by the TnADRH/TnADRL. And the duty of the PWM output is determined by the TnBDRH/TnBDRL.



NOTE)

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

Figure 11.21 16-bit PPG Mode for Timer 2/3 (where n= 2 and 3)

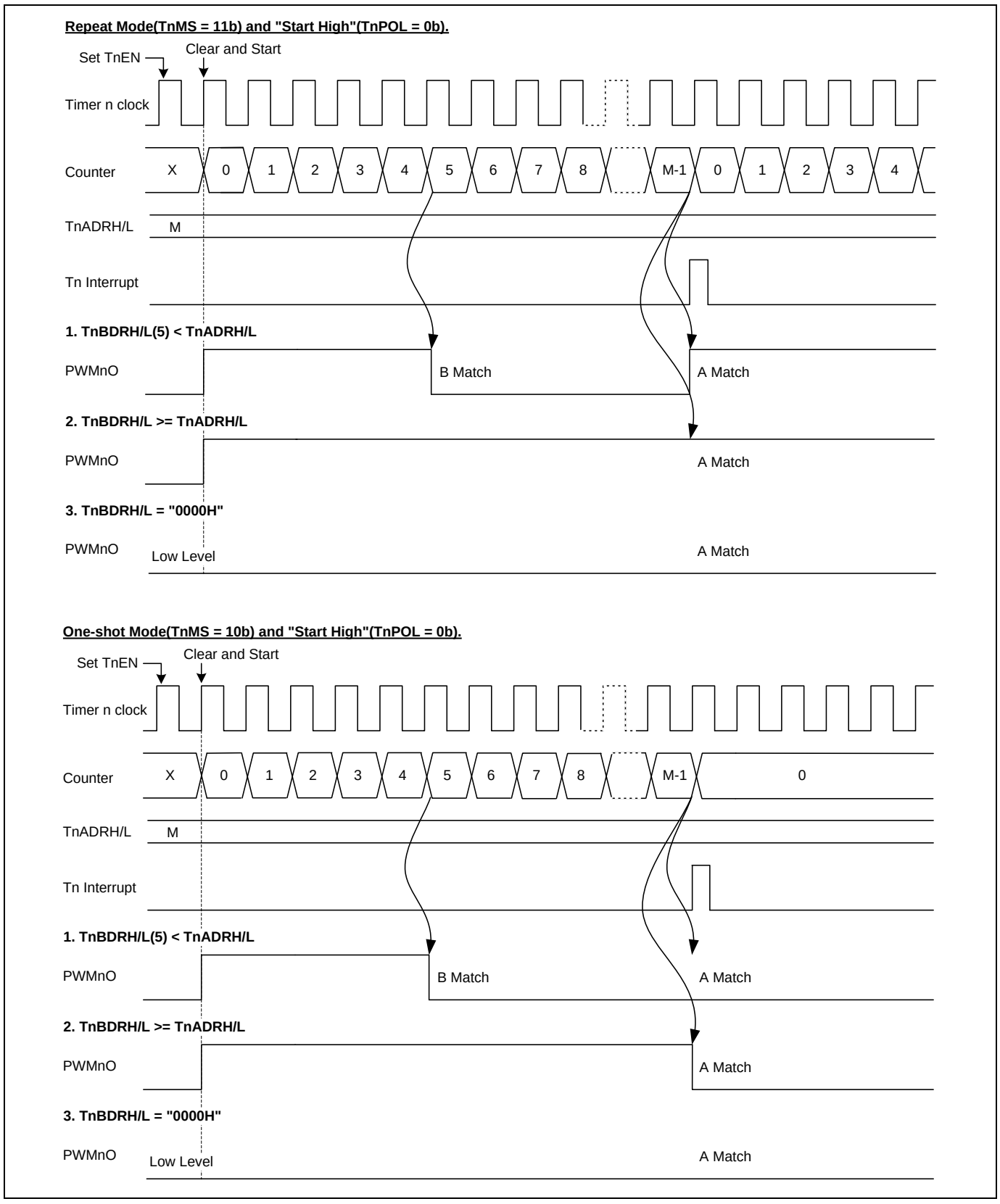


Figure 11.22 16-bit PPG Mode Timming chart for Timer 2/3 (where n= 2 and 3)

11.7.5 Block Diagram

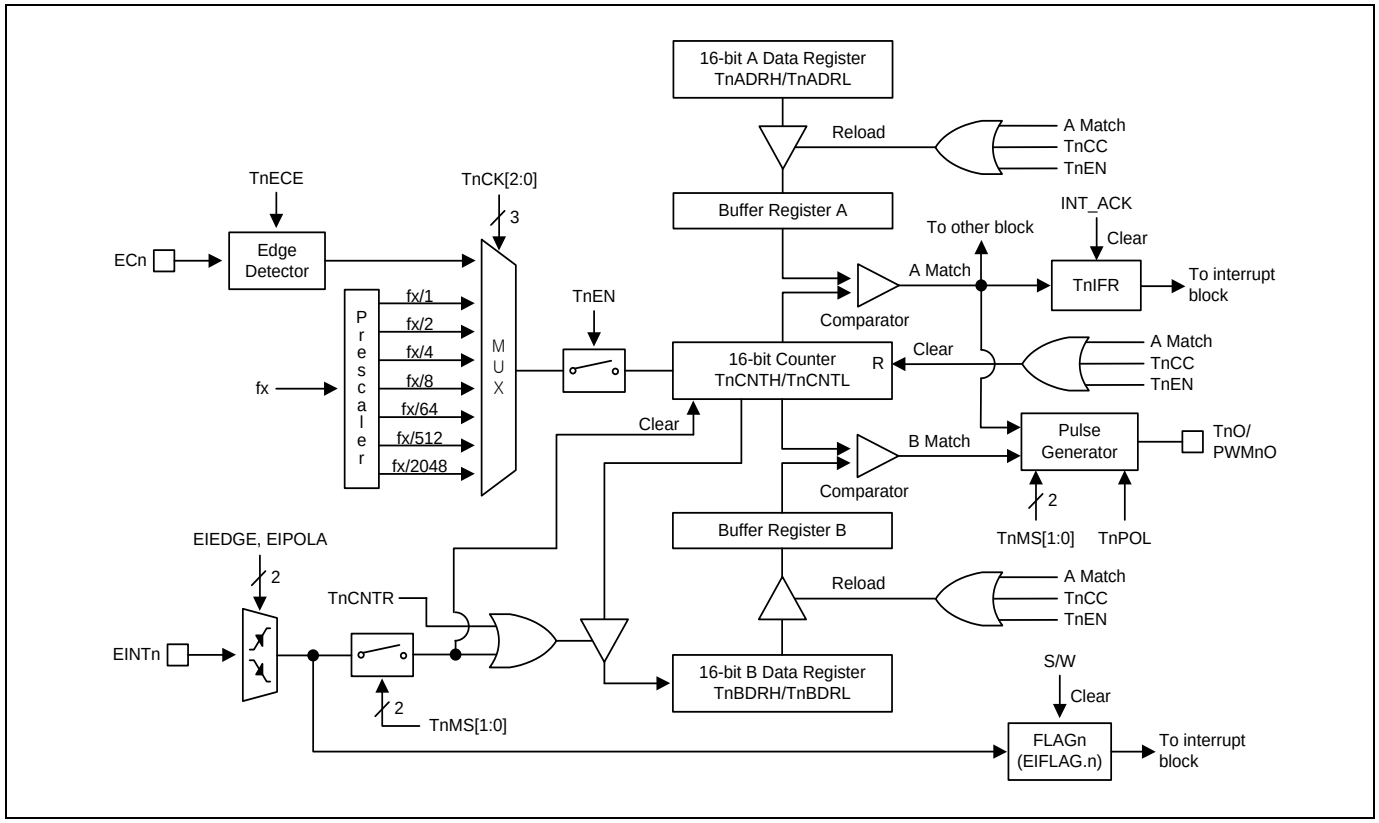


Figure 11.23 16-bit Timer 2/3 Block Diagram (where n= 2 and 3)

11.7.6 Register Map

Name	Address	Direction	Default	Description
TnADRH	BFH/C5H	R/W	FFH	Timer n A Data High Register
TnADRL	BEH/C4H	R/W	FFH	Timer n A Data Low Register
TnBDRH	C1H/C7H	R/W	FFH	Timer n B Data High Register
TnBDRL	C0H/C6H	R/W	FFH	Timer n B Data Low Register
TnCRH	BDH/C3H	R/W	00H	Timer n Control High Register
TnCRL	BCH/C2H	R/W	00H	Timer n Control Low Register

Table 11.9 Timer 2/3 Register Map

11.7.7 Timer/Counter 2/3 Register Description

The timer/counter 2/3 register consists of timer 2/3 A data high register (TnADRH), timer 2/3 A data low register (TnADRL), timer 2/3 B data high register (TnBDRH), timer 2/3 B data low register (TnBDRL), timer 2/3 control high register (TnCRH) and timer 2/3 control low register (TnCRL).

11.7.8 Register Description for Timer/Counter 2/3

TnADRH (Timer n A data High Register) : BFH/C5H, n= 2 and 3

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

Initial value : FFH

TnADRH[7:0] Tn A Data High Byte

TnADRL (Timer n A Data Low Register) : BEH/C4H, n= 2 and 3

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

Initial value : FFH

TnADRL[7:0] Tn A Data Low Byte

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

TnBDRH (Timer n B Data High Register) : C1H/C7H, n= 2 and 3

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

Initial value : FFH

TnBDRH[7:0] Tn B Data High Byte

TnBDRL (Timer n B Data Low Register) : C0H/C6H, n= 2 and 3

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

Initial value : FFH

TnBDRL[7:0] Tn B Data Low Byte

TnCRH (Timer n Control High Register) : BDH/C3H, n= 2 and 3

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

Initial value : 00H

TnEN Control Timer n

0 Timer n disable

1 Timer n enable (Counter clear and start)

TnMS[1:0] Control Timer n Operation Mode

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

TnPE Configure Timer n PWM output port.

0 GPIO

1 PWM output

TnCC Clear Timer n Counter

0 No effect

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

TnCRL (Timer n Control Low Register) : BCH/C2H, n= 2 and 3

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

Initial value : 00H

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

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

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

0 Tn Interrupt no generation

1 Tn Interrupt generation

TnPOL TnO/PWMnO Polarity Selection

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

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

TnECE Timer n External Clock Edge Selection

0 External clock falling edge

1 External clock rising edge

TnCNTR Timer n Counter Read Control

0 No effect

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

11.8 Timer 4/5/6

11.8.1 Overview

The 16-bit timer 4/5/6 consists of multiplexer, timer 4/5/6 A data register high/low, timer 4/5/6 B data register high/low and timer 4/5/6 control register high/low (TnADRH, TnADRL, TnBDRH, TnBDRL, TnCRH, TnCRL).

It has four operating modes:

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

The timer/counter 4/5/6 can be clocked by an internal or an external clock source (ECn). The clock source is selected by clock selection logic which is controlled by the clock selection bits (TnCK[2:0]).
- TIMER n clock source: fX/1, 2, 4, 8, 64, 512, 2048 and ECn

In the capture mode, by EINT1n, the data is captured into input capture data register (TnBDRH/TnBDRL). Timer 4/5/6 outputs the comparison result between counter and data register through TnO port in timer/counter mode. Also Timer 4/5/6 outputs PWM wave form through PWMnO port in the PPG mode.

TnEN	TnMS[1:0]	TnCK[2:0]	Timer 4/5/6
1	00	XXX	16 Bit Timer/Counter Mode
1	01	XXX	16 Bit Capture Mode
1	10	XXX	16 Bit PPG Mode (one-shot mode)
1	11	XXX	16 Bit PPG Mode (repeat mode)

Table 11.10 Timer 4/5/6 Operating Modes

11.8.2 16-bit Timer/Counter Mode

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

The 16-bit timer have counter and data register. The counter register is increased by internal or external clock input. Timer 4/5/6 can use the input clock with one of 1, 2, 4, 8, 64, 512 and 2048 prescaler division rates (TnCK[2:0]). When the value of TnCNTH, TnCNTL and the value of TnADRH, TnADRL are identical in Timer 4/5/6 respectively, a match signal is generated and the interrupt of Timer 4/5/6 occurs. The TnCNTH, TnCNTL value is automatically cleared by match signal. It can be also cleared by software (TnCC).

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

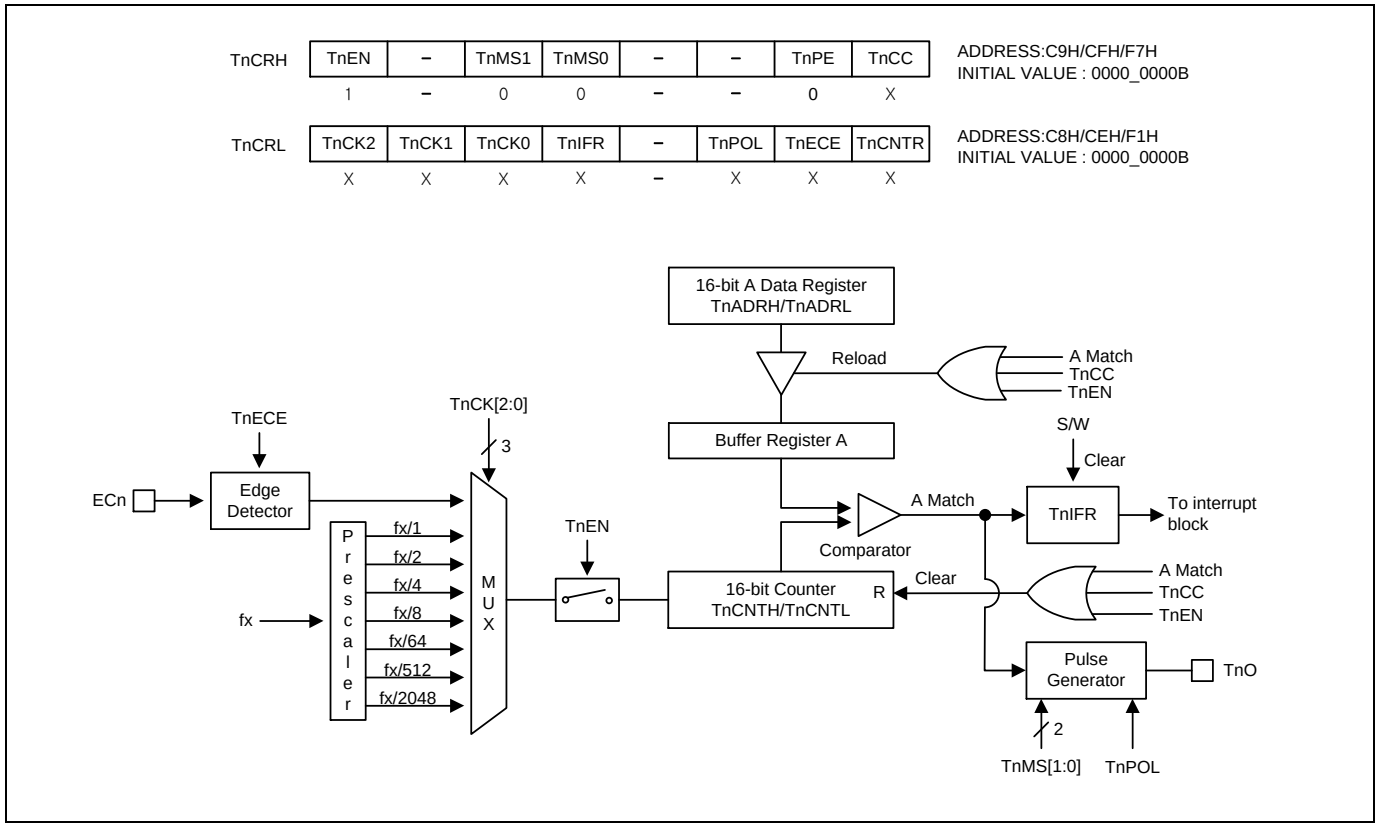


Figure 11.24 16-bit Timer/Counter Mode for Timer 4/5/6 (where n= 4, 5 and 6)

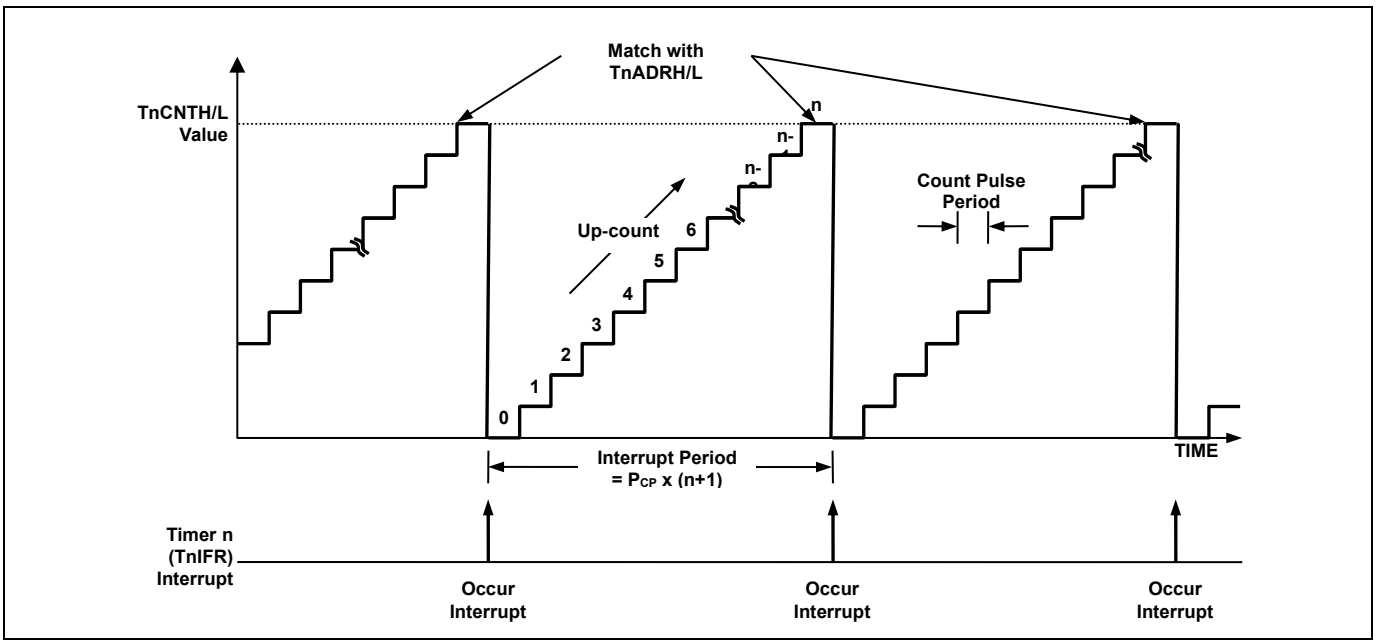


Figure 11.25 16-bit Timer/Counter 4/5/6 Example (where n= 4, 5 and 6)

11.8.3 16-bit Capture Mode

The 16-bit timer 4/5/6 capture mode is set by TnMS[1:0] as '01'. The clock source can use the internal/external clock. Basically, it has the same function as the 16-bit timer/counter mode and the interrupt occurs when TnCNTH/TnCNTL is equal to TnADRH/TnADRL. The TnCNTH, TnCNTL values are automatically cleared by match signal. It can be also cleared by software (TnCC).

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

The capture result is loaded into TnBDRH/TnBDRL.

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

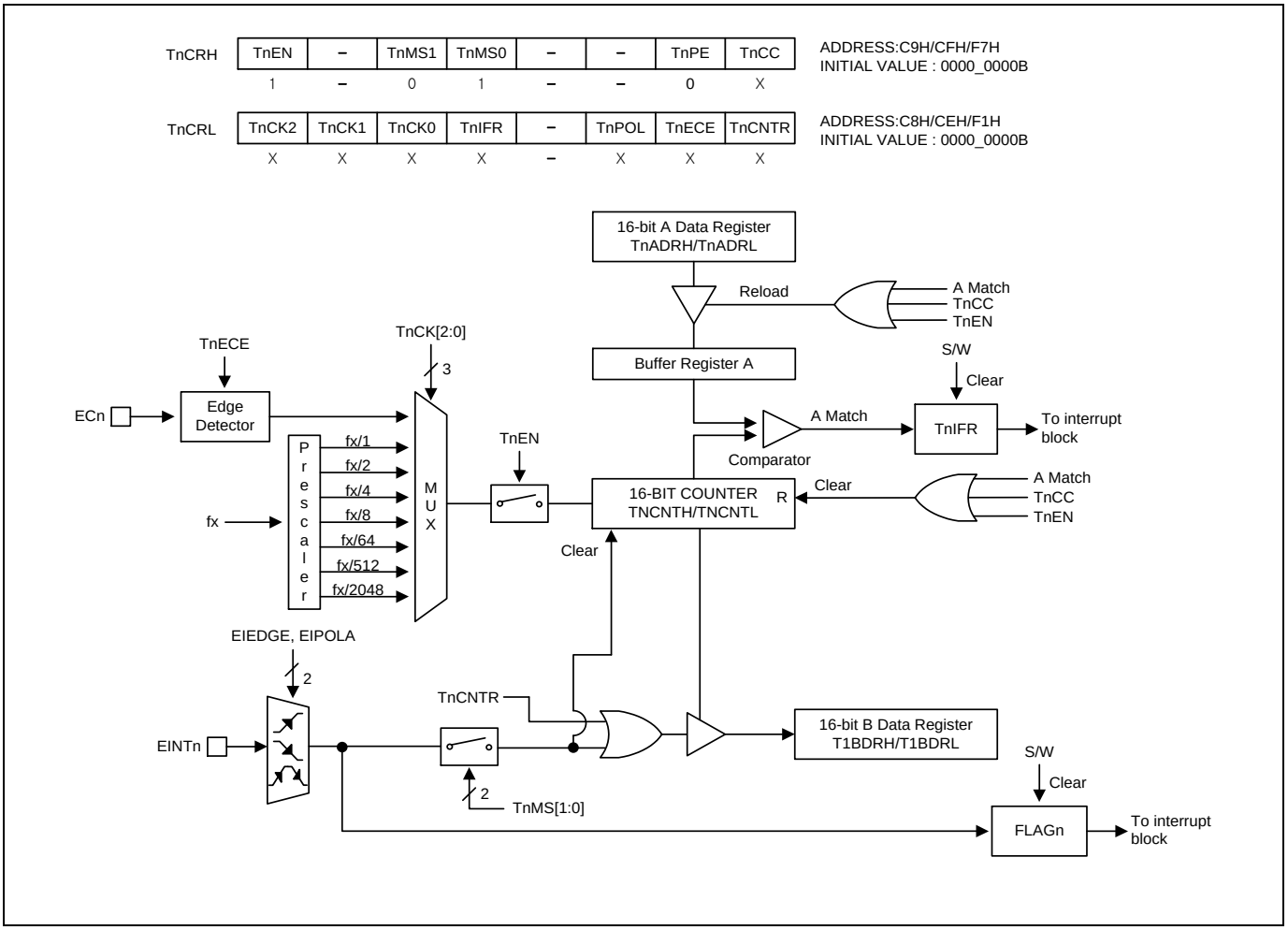


Figure 11.26 16-bit Capture Mode for Timer 4/5/6 (where n= 4, 5 and 6)

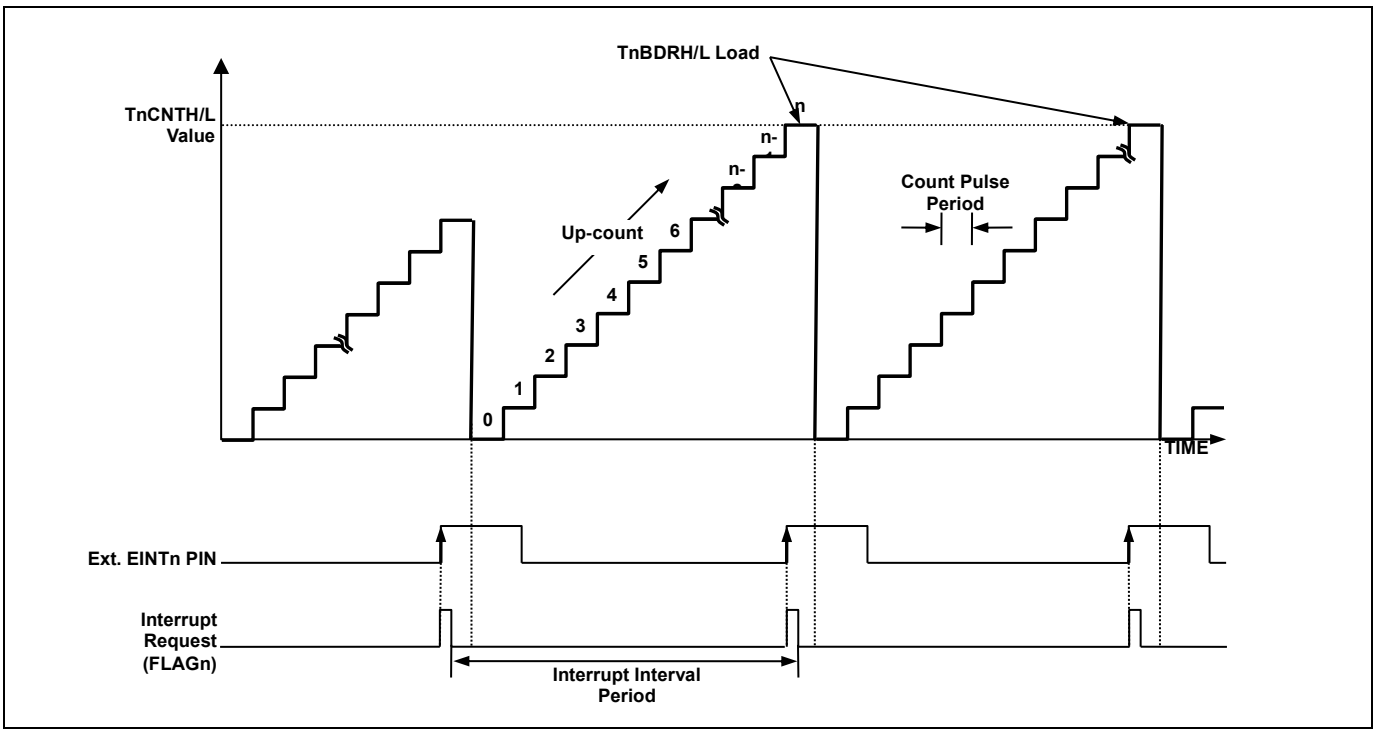


Figure 11.27 Input Capture Mode Operation for Timer 4/5/6 (where n= 4, 5 and 6)

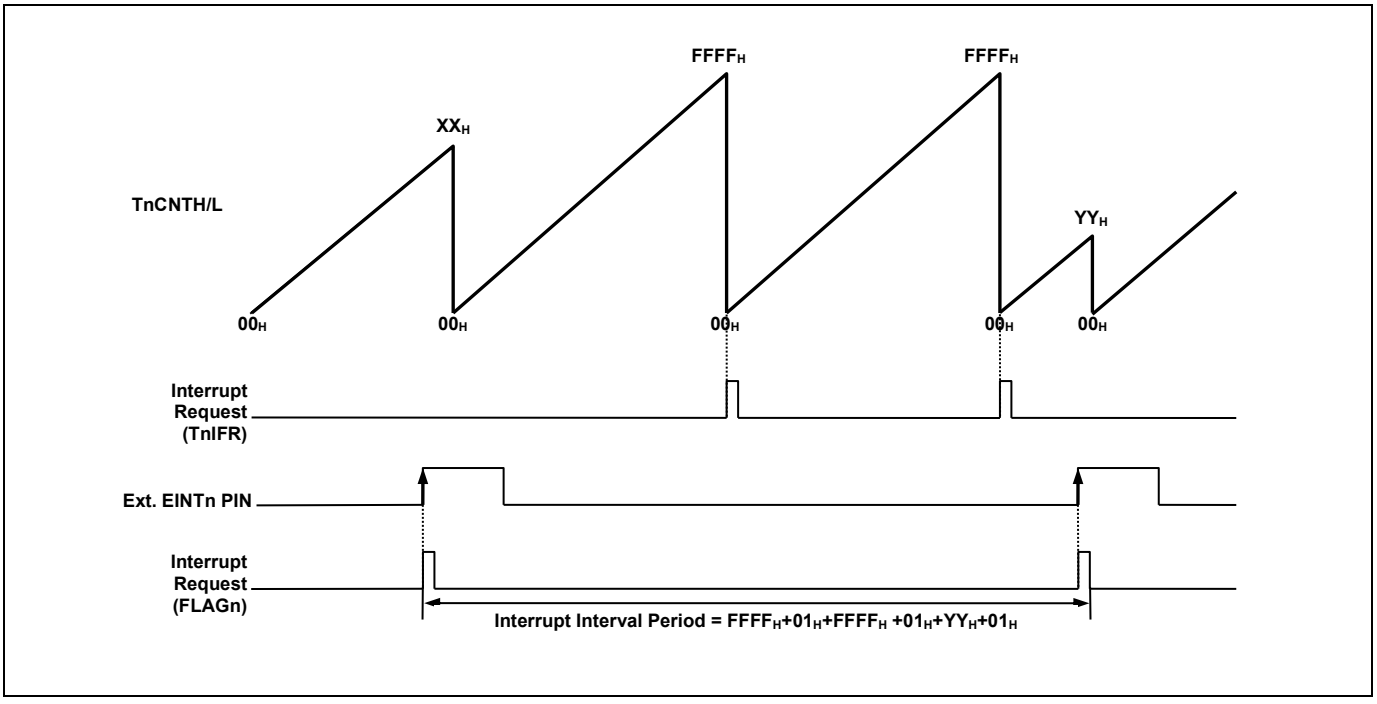
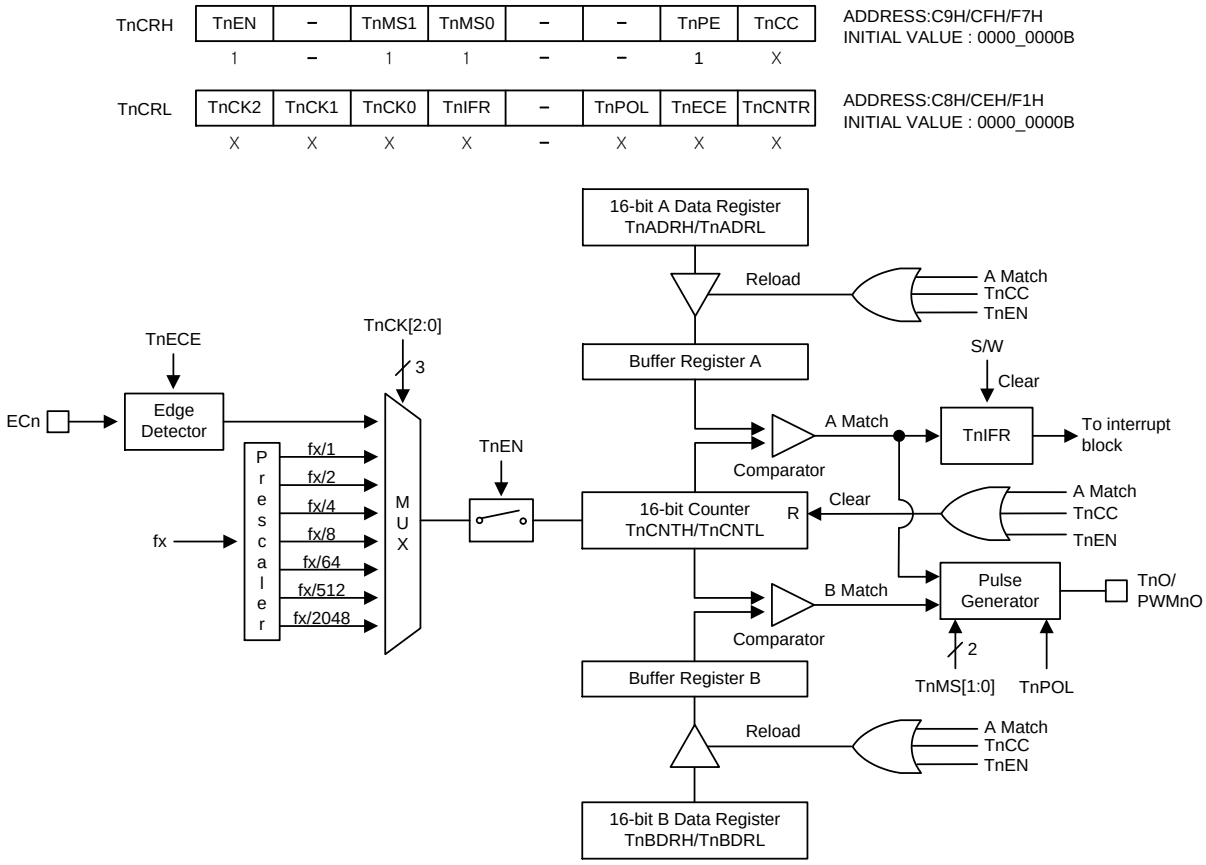


Figure 11.28 Express Timer Overflow in Capture Mode (where n= 4, 5 and 6)

11.8.4 16-bit PPG Mode

The timer 4/5/6 has a PPG (Programmable Pulse Generation) function. In PPG mode, TnO/PWMnO pin outputs up to 16-bit resolution PWM output. This pin should be configured as a PWM output by setting TnPE bit to '1'. The period of the PWM output is determined by the TnADRH/TnADRL. And the duty of the PWM output is determined by the TnBDRH/TnBDRL.



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

Figure 11.29 16-bit PPG Mode for Timer 4/5/6 (where n= 4, 5 and 6)

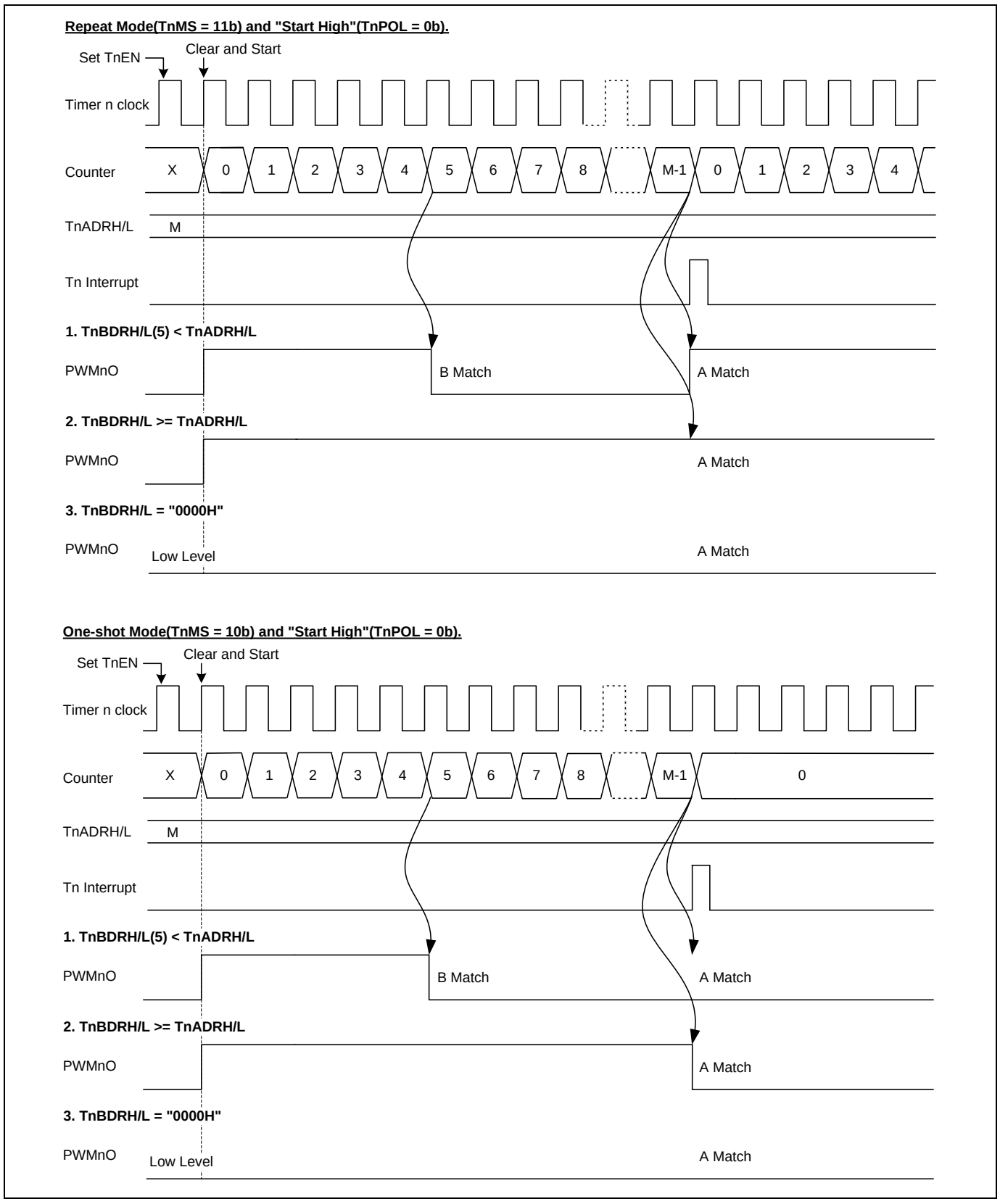


Figure 11.30 16-bit PPG Mode Timming chart for Timer 4/5/6 (where n= 4, 5 and 6)

11.8.5 Block Diagram

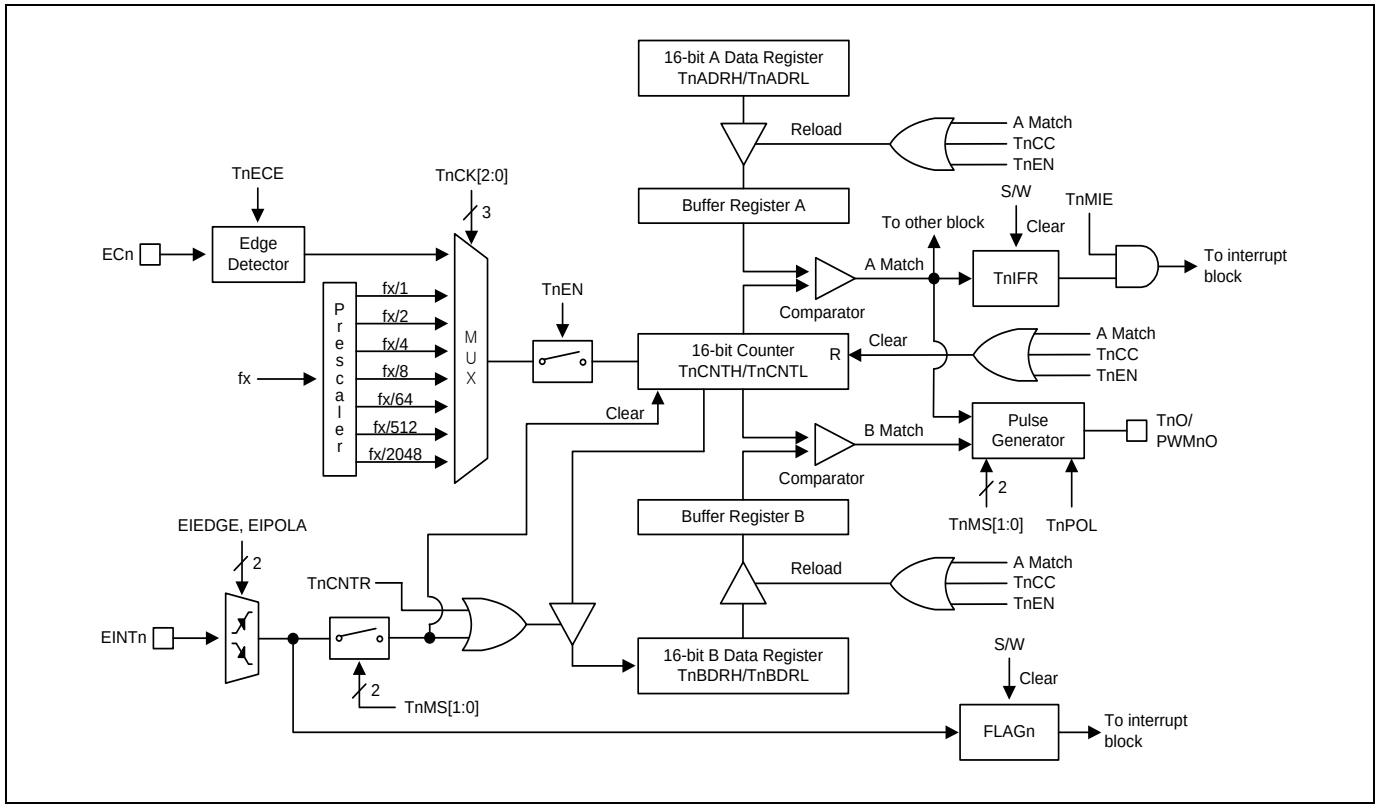


Figure 11.31 16-bit Timer 4/5/6 Block Diagram (where n= 4, 5 and 6)

11.8.6 Register Map

Name	Address	Direction	Default	Description
TnADRH	CBH/D3H/EFH	R/W	FFH	Timer n A Data High Register
TnADRL	CAH/D2H/EEH	R/W	FFH	Timer n A Data Low Register
TnBDRH	CDH/D5H/F9H	R/W	FFH	Timer n B Data High Register
TnBDRL	CCH/D4H/F8H	R/W	FFH	Timer n B Data Low Register
TnCRH	C9H/CFH/F7H	R/W	00H	Timer n Control High Register
TnCRL	C8H/CEH/F1H	R/W	00H	Timer n Control Low Register
TIFLAG	B6H	R/W	00H	Timer Interrupt Flag Register

Table 11.11 Timer 4/5/6 Register Map

11.8.7 Timer/Counter 4/5/6 Register Description

The timer/counter 4/5/6 register consists of timer 4/5/6 A data high register (TnADRH), timer 4/5/6 A data low register (TnADRL), timer 4/5/6 B data high register (TnBDRH), timer 4/5/6 B data low register (TnBDRL), timer 4/5/6 control high register (TnCRH), timer 4/5/6 control low register (TnCRL), and timer interrupt flag register(TIFLAG1).

11.8.8 Register Description for Timer/Counter 4/5/6

TnADRH (Timer n A data High Register) : DDH/E5H/EDH/105BH (SFR/SFR/SFR/XSFR), n= 4, 5 and 6

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

Initial value : FFH

TnADRH[7:0] Tn A Data High Byte

TnADRL (Timer n A Data Low Register) : DCH/E4H/ECH/105AH (SFR/SFR/SFR/XSFR), n= 4, 5 and 6

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

Initial value : FFH

TnADRL[7:0] Tn A Data Low Byte

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

TnBDRH (Timer n B Data High Register) : DFH/E7H/EFH/105DH (SFR/SFR/SFR/XSFR), n= 4, 5 and 6

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

Initial value : FFH

TnBDRH[7:0] Tn B Data High Byte

TnBDRL (Timer n B Data Low Register) : **DEH/E6H/EEH/105CH (SFR/SFR/SFR/XSFR), n= 4, 5 and 6**

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

Initial value : FFH

TnBDRL[7:0] Tn B Data Low Byte

TnCRH (Timer n Control High Register) : DBH/E3H/EBH/1059H (SFR/SFR/SFR/XSFR), n= 4, 5 and 6

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

Initial value : 00H

TnEN Control Timer n

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

TnMIE Enable or Disable Timer n Match Interrupt

0	Disable
1	Enable

TnMS[1:0] Control Timer n Operation Mode

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

TnPE Configure Timer n PWM output port.

0	GPIO
1	PWM output

TnCC Clear Timer n Counter

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

NOTE) Refer to the timer interrupt flag register (TIFLAG) for the T4/T5/T6 interrupt flags.

TnCRL (Timer n Control Low Register) : DAH/E2H/EAH/1058H (SFR/SFR/SFR/XSFR), n= 4, 5 and 6

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

Initial value : 00H

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

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

TnPOL TnO/PWMnO Polarity Selection

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

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

TnECE Timer n External Clock Edge Selection

0 External clock falling edge

1 External clock rising edge

TnCNTR Timer n Counter Read Control

0 No effect

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

11.9 Buzzer Driver

11.9.1 Overview

The Buzzer consists of 8 Bit Counter and BUZDR (Buzzer Data Register), BUZCR (Buzzer Control Register). The Square Wave (122.07Hz~250 KHz, @16MHz) gets out of P33/BUZ pin. BUZDR (Buzzer Data Register) controls the Buzzer frequency (look at the following expression). In the BUZCR (Buzzer Control Register), BUCK[1:0] selects source clock divided from prescaler.

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

BUZDR[7:0]	Buzzer Frequency (kHz)			
	BUZCR[2:1]=00	BUZCR[2:1]=01	BUZCR[2:1]=10	BUZCR[2:1]=11
0000_0000	250kHz	125kHz	62.5kHz	31.25kHz
0000_0001	125kHz	62.5kHz	31.25kHz	15.624kHz
…	…	…	…	…
1111_1101	984.252Hz	492.126Hz	246.062Hz	123.03Hz
1111_1110	980.392Hz	490.196Hz	245.098Hz	122.548Hz
1111_1111	976.562Hz	488.282Hz	244.140Hz	122.07Hz

Table 11.12 Buzzer Frequency at 16 MHz

11.9.2 Block Diagram

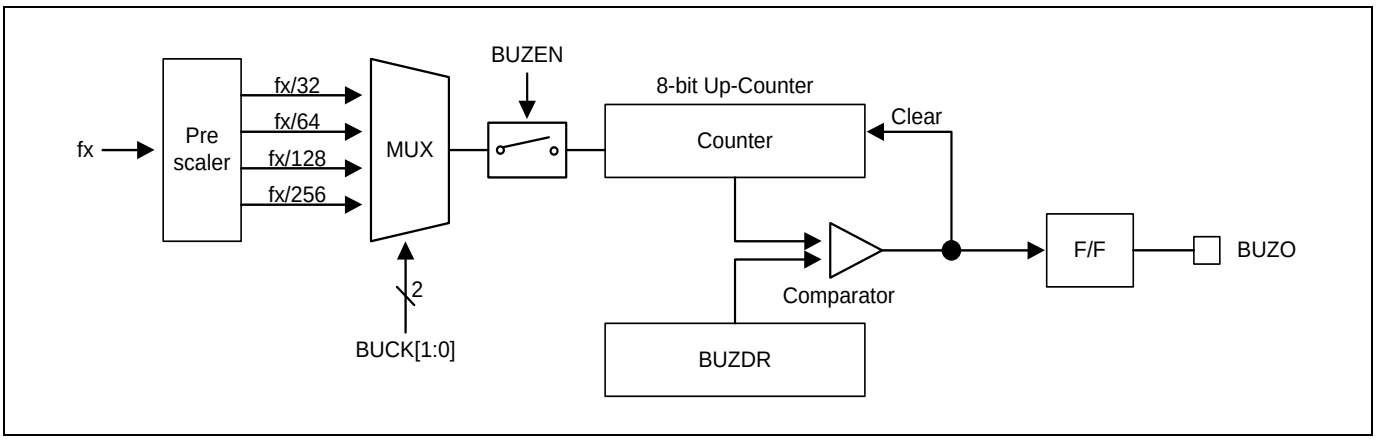


Figure 11.32 Buzzer Driver Block Diagram

11.9.3 Register Map

Name	Address	Direction	Default	Description
BUZDR	8FH	R/W	FFH	Buzzer Data Register
BUZCR	9FH	R/W	00H	Buzzer Control Register

Table 11.13 Buzzer Driver Register Map

11.9.4 Buzzer Driver Register Description

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

11.9.5 Register Description for Buzzer Driver

BUZDR (Buzzer Data Register): 8FH

7	6	5	4	3	2	1	0
BUZDR7	BUZDR6	BUZDR5	BUZDR4	BUZDR3	BUZDR2	BUZDR1	BUZDR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: FFH

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

BUZCR (Buzzer Control Register): 97H

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

Initial value: 00H

BUCK[1:0] Buzzer Driver Source Clock Selection

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

BUZEN Buzzer Driver Operation Control

0	Buzzer Driver disable
1	Buzzer Driver enable

NOTE)

1. fx: System clock oscillation frequency.

11.10 USART

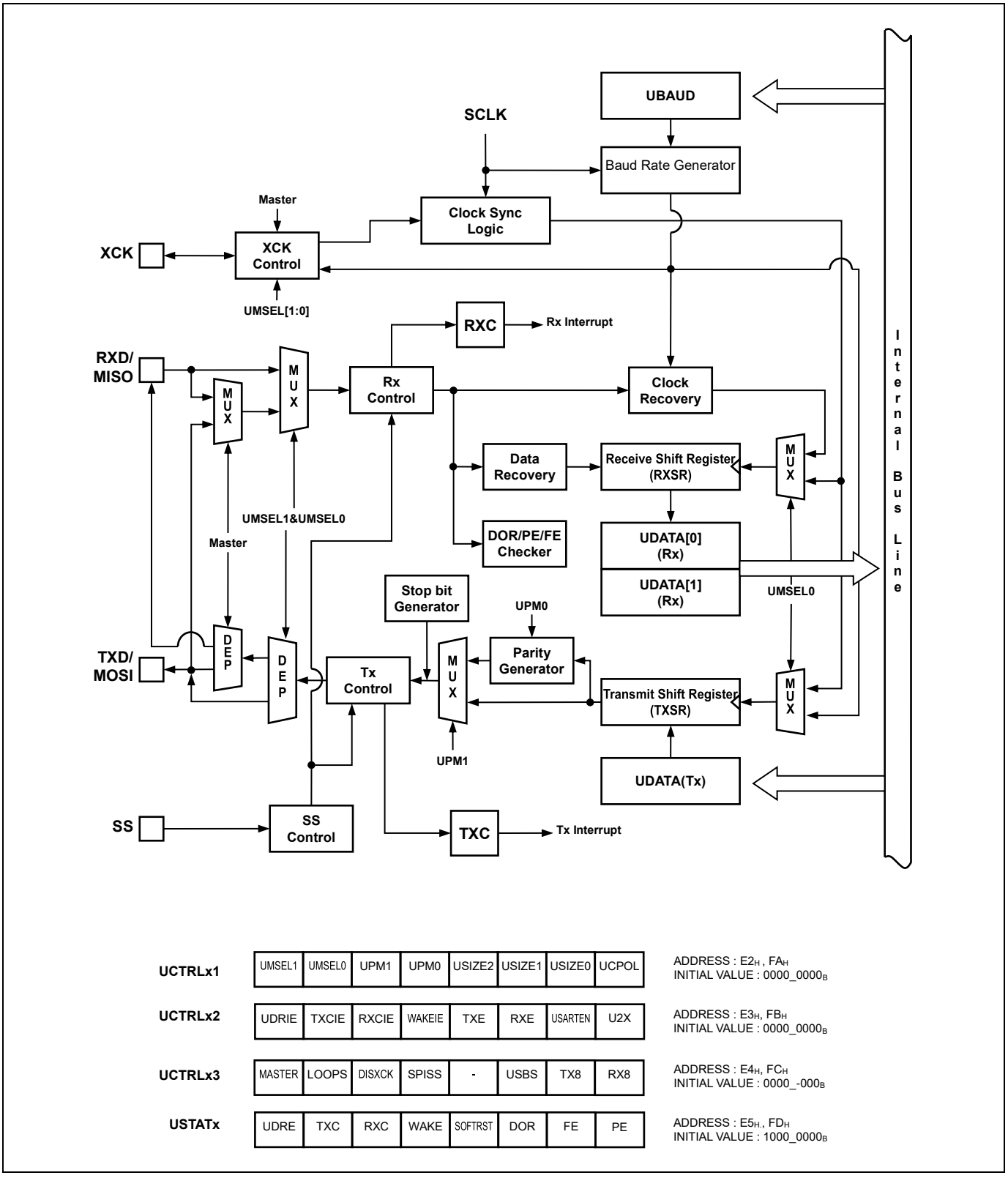
11.10.1 Overview

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 (UDATAx) and control logic. The Receiver supports the same frame formats as the Transmitter and can detect Frame Error, Data OverRun and Parity Errors.

11.10.2 Block Diagram



UCTRLx1	UMSEL1	UMSEL0	UPM1	UPM0	USIZE2	USIZE1	USIZE0	UCPOL	ADDRESS : $E2_H$, FA_H INITIAL VALUE : 0000_0000_B
UCTRLx2	UDRIE	TXCIE	RXCIE	WAKEIE	TXE	RXE	USARTEN	U2X	ADDRESS : $E3_H$, FB_H INITIAL VALUE : 0000_0000_B
UCTRLx3	MASTER	LOOPS	DISXCK	SPISS	-	USBS	TX8	RX8	ADDRESS : $E4_H$, FC_H INITIAL VALUE : 0000_-000_B
USTATx	UDRE	TXC	RXC	WAKE	SOFTRST	DOR	FE	PE	ADDRESS : $E5_{H.}$, FD_H INITIAL VALUE : 1000_0000_B

Figure 11.33 USART Block Diagram

11.10.3 Clock Generation

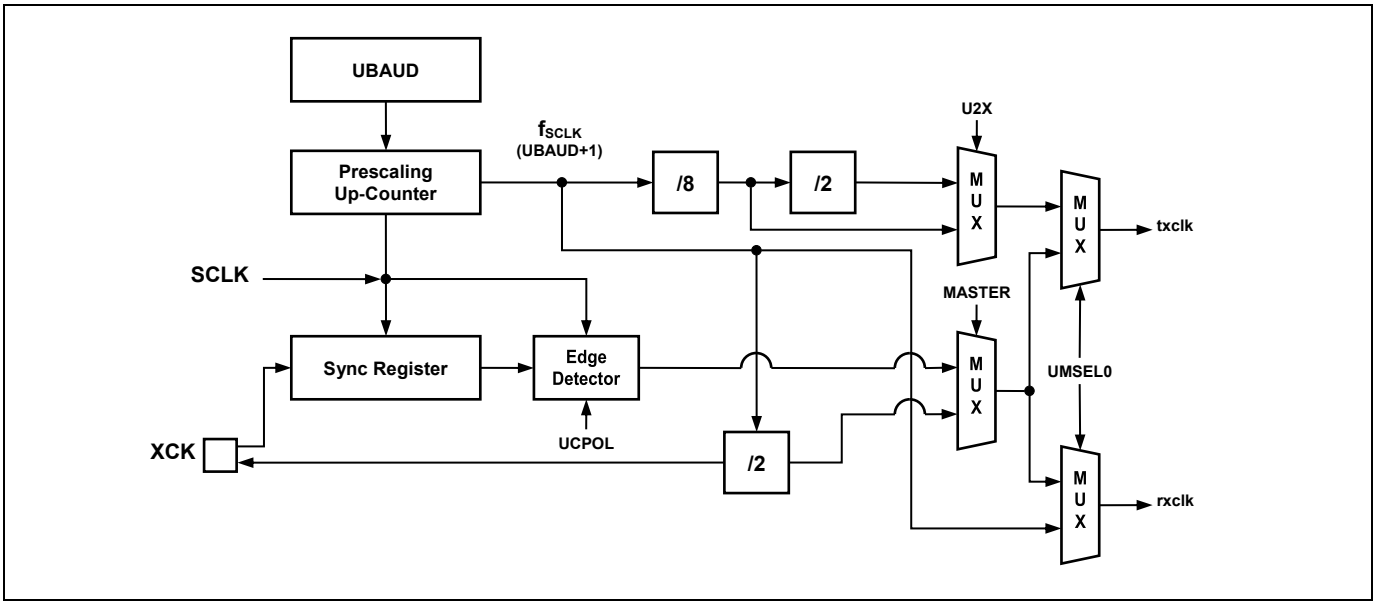


Figure 11.34 Clock Generation Block Diagram

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 UCTRLx1 register selects between asynchronous and synchronous operation. Asynchronous Double Speed mode is controlled by the U2X bit in the UCTRLx2 register. The MASTER bit in UCTRLx2 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 11.14 Equations for Calculating Baud Rate Register Setting

11.10.4 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).

11.10.5 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 UCTRLx1 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.

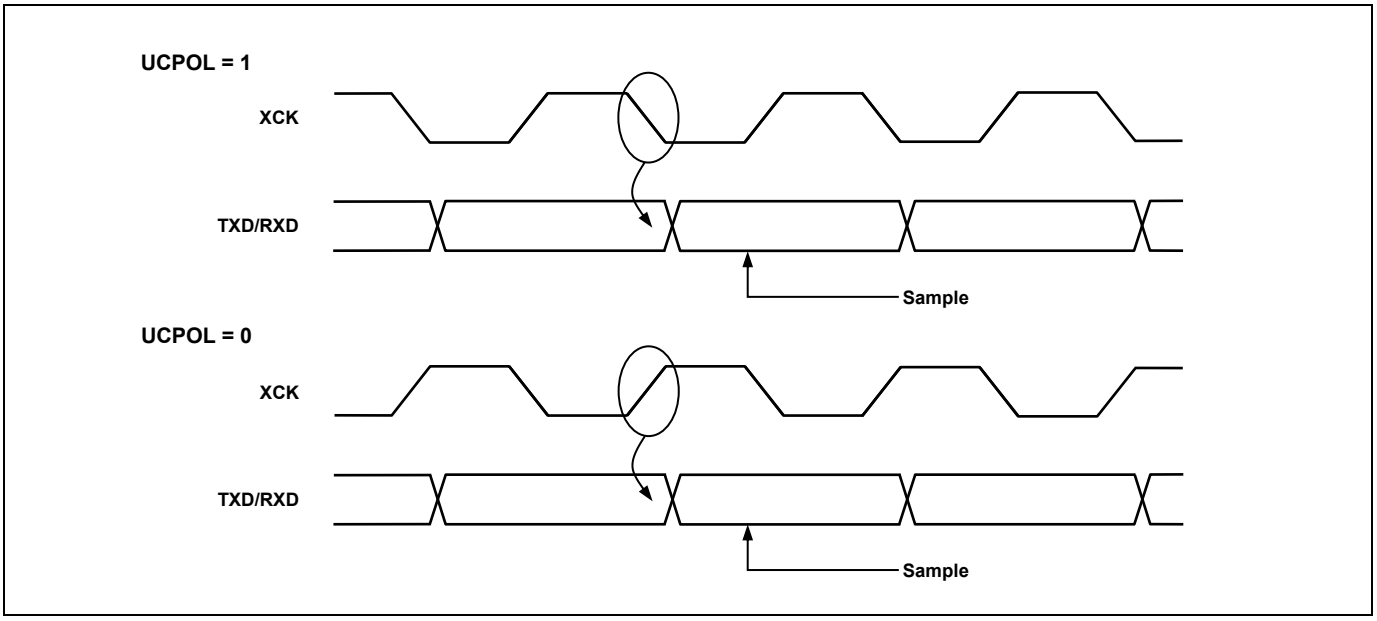


Figure 11.35 Synchronous Mode XCKn Timing

11.10.6 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.

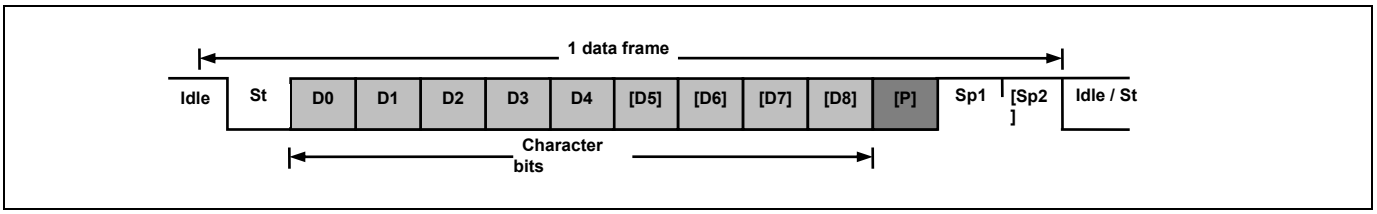


Figure 11.36 frame format

1 data frame consists of the following bits

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

The frame format used by the USART is set by the USIZE[2:0], UPM[1:0] and USBS bits in UCTRLx1 register. The Transmitter and Receiver use the same setting.

11.10.7 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

11.10.8 USART Transmitter

The USART Transmitter is enabled by setting the TXE bit in UCTRLx1 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 UCTRLx3 register.

11.10.8.1 Sending Tx data

A data transmission is initiated by loading the transmit buffer (UDATAx 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 UCTRLx3 register before loading transmit buffer (UDATA register).

11.10.8.2 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.

11.10.8.3 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.

11.10.8.4 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.

11.10.9 USART Receiver

The USART Receiver is enabled by setting the RXE bit in the UCTRLx1 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 UCTRLx3 register.

11.10.9.1 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 UDATAx register.

If 9-bit characters are used (USIZE[2:0] = 7), the ninth bit is stored in the RX8 bit position in the UCTRLx3 register. The 9th bit must be read from the RX8 bit before reading the low 8 bits from the UDATAx register. Likewise, the error flags FE, DOR, PE must be read before reading the data from UDATAx register. This is because the error flags are stored in the same FIFO position of the receive buffer.

11.10.9.2 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 UCTRLx2 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 USTATx 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 UDATAx register, read the USTATx 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.

11.10.9.3 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.

11.10.9.4 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.

11.10.9.5 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.

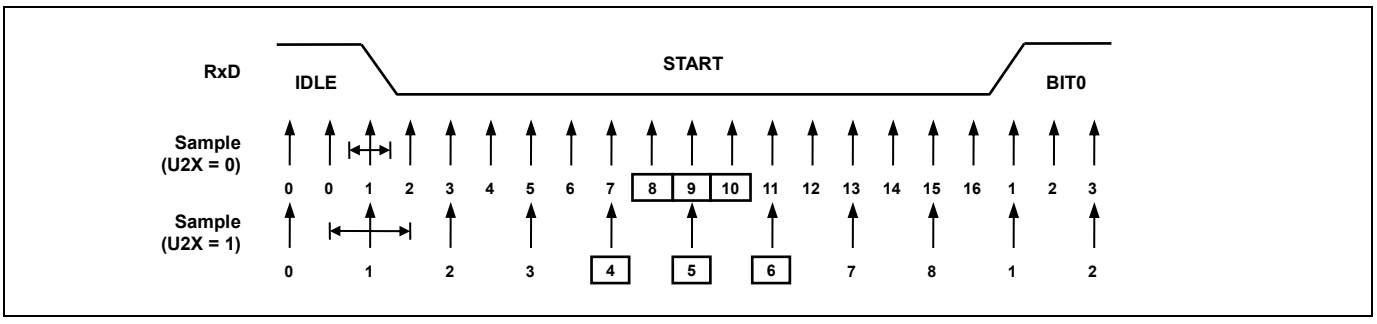


Figure 11.37 Start bit Sampling

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.

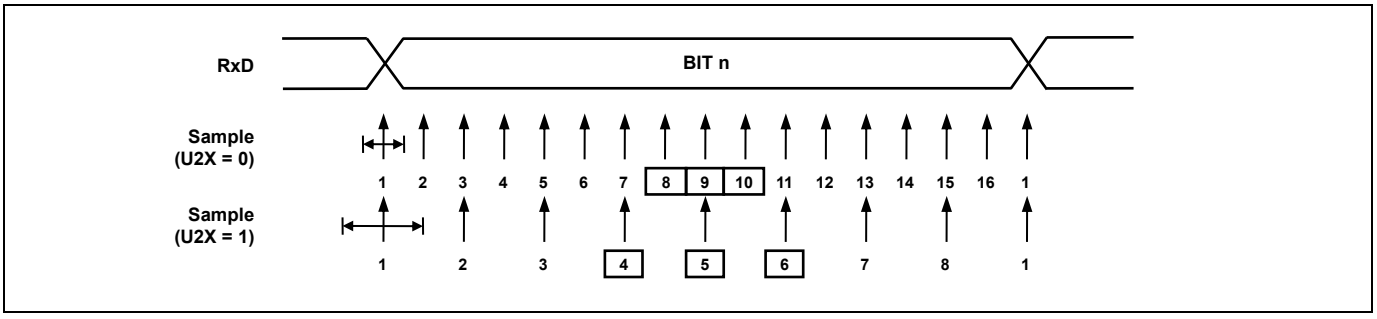


Figure 11.38 Sampling of Data and Parity bit

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).

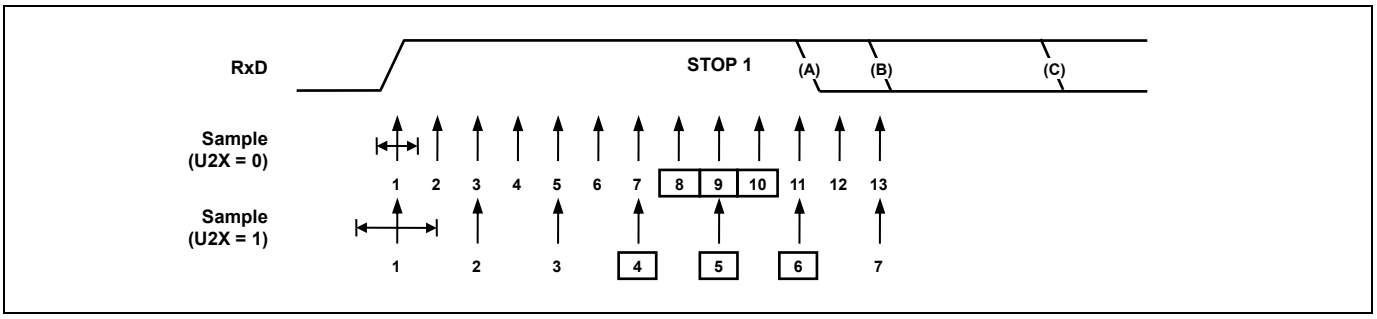


Figure 11.39 Stop bit Sampling and Next Start bit Sampling

11.10.10 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.

11.10.10.1 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 UCTRLx1 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 11.15 CPOL Functionality

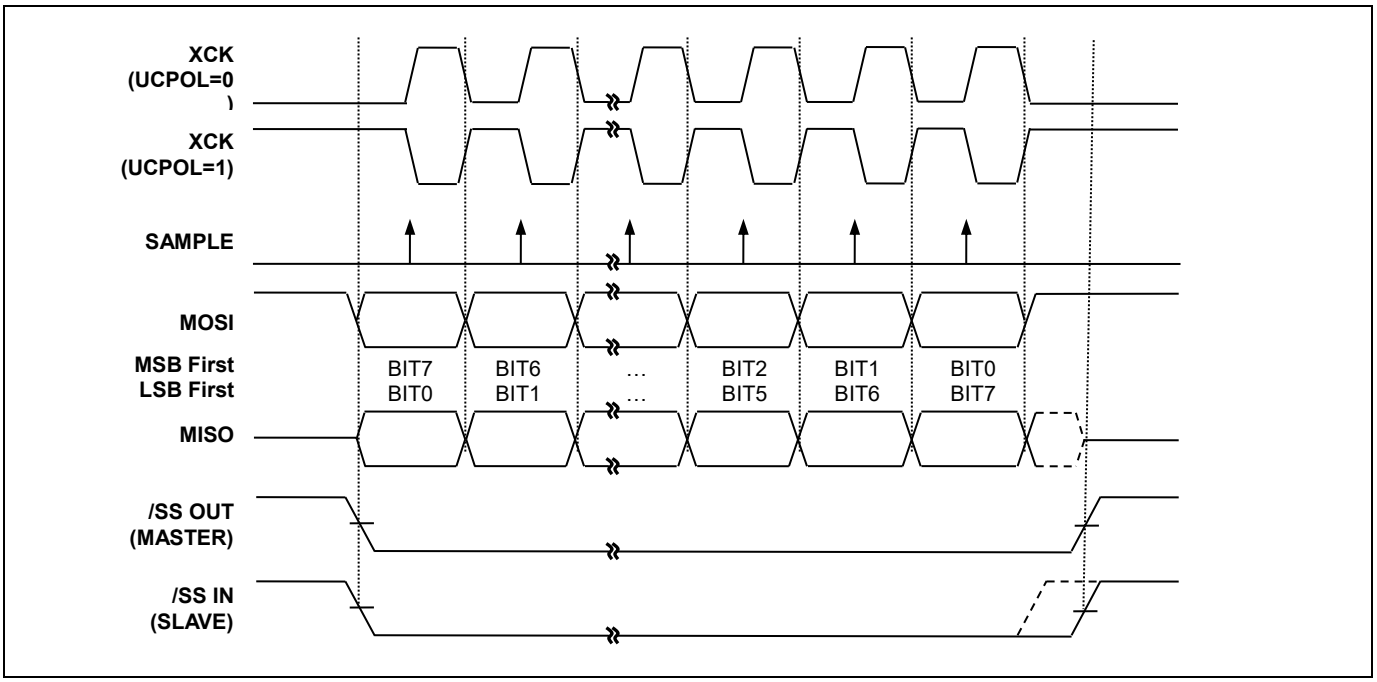


Figure 11.40 SPI Clock Formats when UCPHA=0

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.

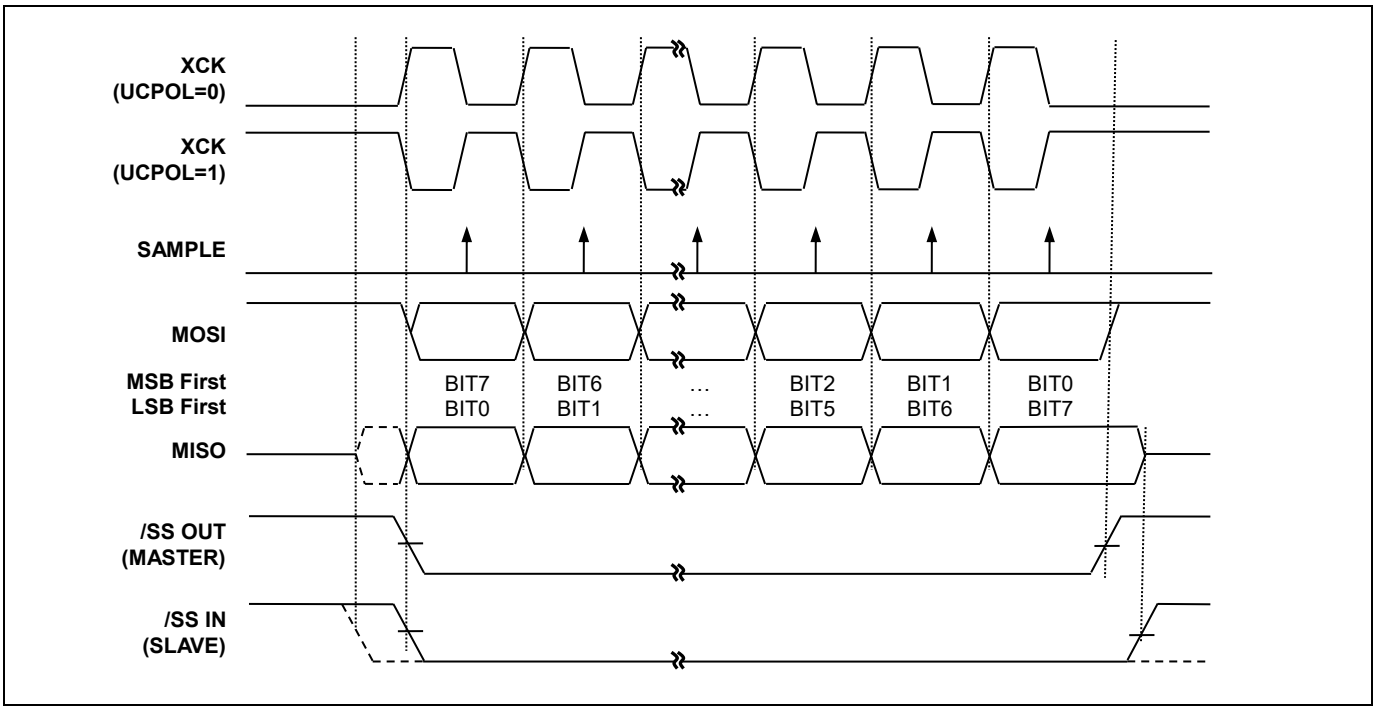


Figure 11.41 SPI Clock Formats when UCPHA=1

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.

11.10.11 Register Map

Name	Address	Direction	Default	Description
UCTRL01	E2H	R/W	00H	USART 0 Control 1 Register 0
UCTRL02	E3H	R/W	00H	USART 0 Control 2 Register 0
UCTRL03	E4H	R/W	00H	USART 0 Control 3 Register 0
UCTRL04	8FC0H	R/W	00H	USART 0 Control 4 Register 0
FPCR0	8FC1H	R/W	00H	USART 0 Floating Point Counter
USTAT0	E5H	R	80H	USART Status Register 0
UBAUD0	E6H	R/W	FFH	USART Baud Rate Generation Register 0
UDATA0	E7H	R/W	FFH	USART Data Register 0
UCTRL11	FAH	R/W	00H	USART 1 Control 1 Register 1
UCTRL12	FBH	R/W	00H	USART 1 Control 2 Register 1
UCTRL13	FCH	R/W	00H	USART 1 Control 3 Register 1
UCTRL14	8FC8H	R/W	00H	USART 1 Control 4 Register 0
FPCR1	8FC9H	R/W	00H	USART 1 Floating Point Counter
USTAT1	FDH	R	80H	USART Status Register 1
UBAUD1	FEH	R/W	FFH	USART Baud Rate Generation Register 1
UDATA1	FFH	R/W	FFH	USART Data Register 1

Table 11.16 USART Register Map

11.10.12 USART Register Description

USART module consists of USART Control 1 Register (UCTRLx1), USART Control 2 Register (UCTRLx2), USART Control 3 Register (UCTRLx3), USART Control 4 Register (UCTRLx4), USART Floating Point Counter (FPCRx), USART Status Register (USTATx), USART Data Register (UDATAx), and USART Baud Rate Generation Register (UBAUDx).

11.10.13 Register Description for USART

UCTRLx1 (USART Control 1 Register) E2H, FAH

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.

USIZE 2	USIZE 1	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)

UCTRLx2 (USART Control 2 Register) E3H, FBH

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

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

UCTRLx3 (USART Control 3 Register) E4H, FCH

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'

UCTRLx4 (USART Control 4 Register) 8FC0H, 8FC8H

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

Initial value : 00_H

FPCREN Enable baudrate compensation
- 0 Disable
- 1 Enable

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

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

FPCRx (USART Floating Point Counter) 8FC1H, 8FC9H

7	6	5	4	3	2	1	0
FPCR0	FPCR0	FPCR0	FPCR0	FPCR0	FPCR0	FPCR0	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

USTATx (USART Status Register) E5H

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)

2. 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.

UBAUDx (USART Baud-Rate Generation Register) E6H, FEH

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.

UDATAx (USART Data Register) E7H, FFH

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 : FF_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.

11.10.14 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%

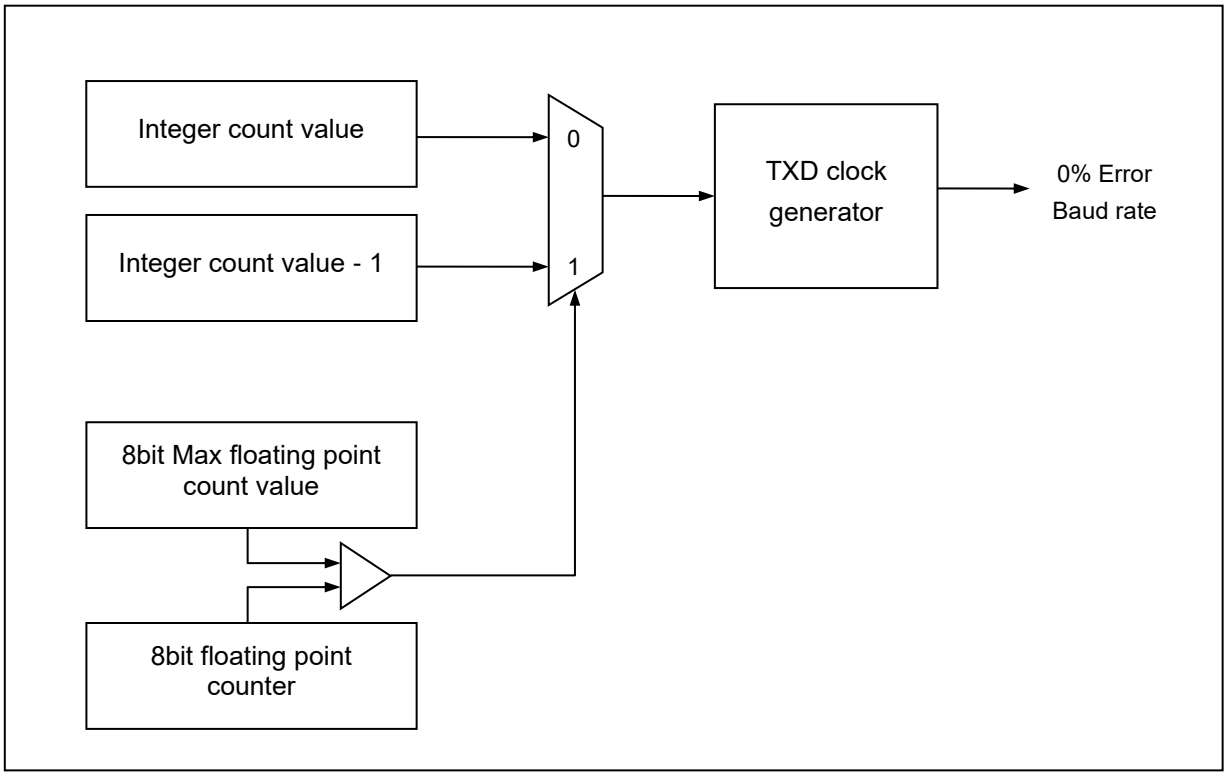
Baud Rate	fOSC=8.00MHz				fOSC=11.0592MHz				fOSC=14.7456MHz			
	U2X=0		U2X=1		U2X=0		U2X=1		U2X=0		U2X=1	
	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR
2400	207	0.2%	-	-	-	-	-	-	-	-	-	-
4800	103	0.2%	207	0.2%	143	0.0%	-	-	191	0.0%	-	-
9600	51	0.2%	103	0.2%	71	0.0%	143	0.0%	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 11.17 Examples of UBAUD Settings for Commonly Used Oscillator Frequencies

11.10.15 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), a calculated integer count value must be 16.36 value (BAUD+1 = 16000000/(16×57600) = 17.36). Here, BAUD which can be set is the nearest big integer number 17. To realize 0% error of baud rate, floating point counter value must be 164 ((17-16.36) x 256 ≒ 164). Namely you have to write the 164(decimal number) in USART_FPCR and 17(decimal number) in USART_BAUD.



11.11 SPI

11.11.1 Overview

There is Serial Peripheral Interface (SPI) one channel in A97C450. The SPI allows synchronous serial data transfer between the external serial devices. It can do Full-duplex communication by 4-wire (MOSI, MISO, SCK, SS), support Master/Slave mode, can select serial clock (SCK) polarity, phase and whether LSB first data transfer or MSB first data transfer.

11.11.2 Block Diagram

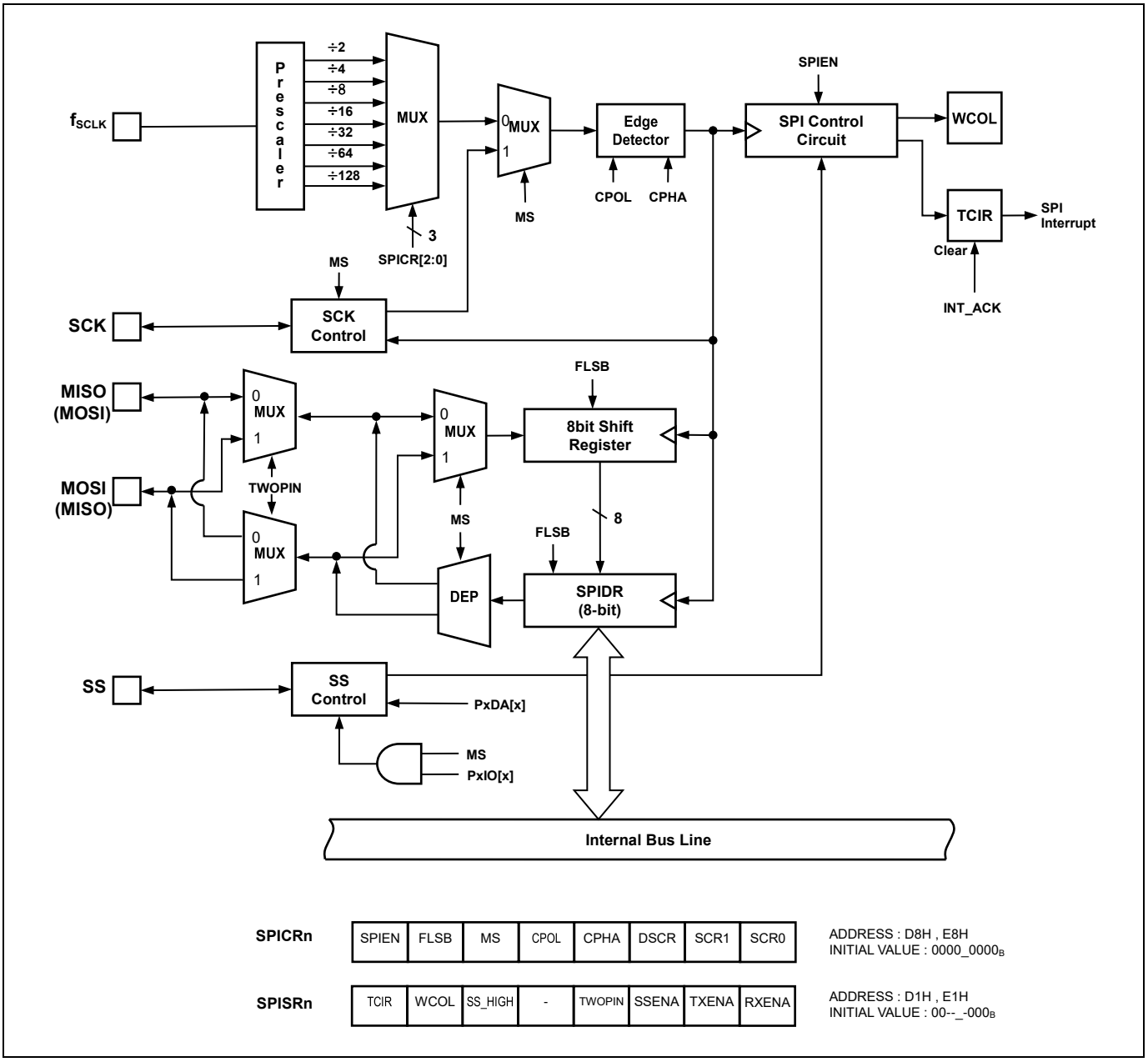


Figure 11.42 SPI Block Diagram

11.11.3 Data Transmit / Receive Operation

User can use SPI for serial data communication by following step

1. Select SPI operation mode(master/slave, polarity, phase) by control register SPICR.
2. When the SPI is configured as a Master, it selects a Slave by SS signal (active low). When the SPI is configured as a Slave, it is selected by SS signal incoming from Master
3. When the user writes a byte to the data register SPIDR, SPI will start an operation.
4. In this time, if the SPI is configured as a Master, serial clock will come out of SCK pin. And Master shifts the eight bits into the Slave (transmit), Slave shifts the eight bits into the Master at the same time (receive). If the SPI is configured as a Slave, serial clock will come into SCK pin. And Slave shifts the eight bits into the Master (transmit), Master shifts the eight bits into the Slave at the same time (receive).
5. When transmit/receive is done, TCIR (Transmit Complete or Interrupt Request) bit will be set. If the SPI interrupt is enabled, an interrupt is requested. And TCIR bit is cleared by hardware when executing the corresponding interrupt. If SPI interrupt is disable, TCIR bit is cleared when user read the status register SPISR, and then access (read/write) the data register SPIDR.

NOTE)

3. If you want to use both transmit and receive, set the TXENA, RXENA bit of SPISR, and if user want to use only either transmit or receive, clear the TXENA or RXENA. In this case, user can use disabled pin by GPIO freely.

11.11.4 SS pin function

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

NOTE)

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

11.11.5 Timing Waveform

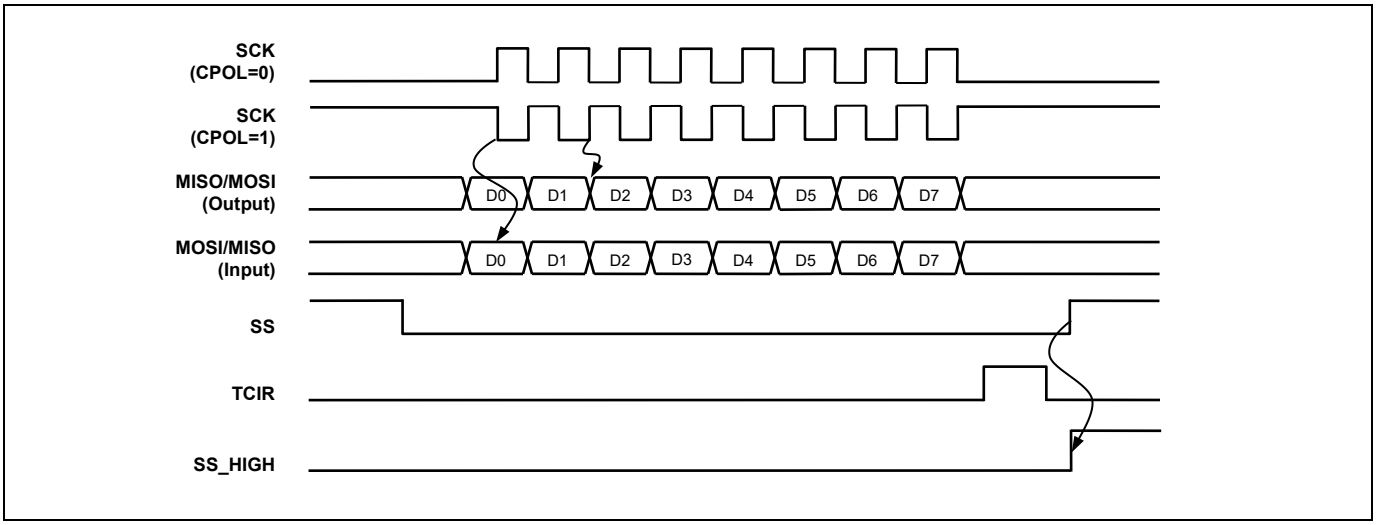


Figure 11.43 SPI Transmit/Receive Timing Diagram at CPHA = 0

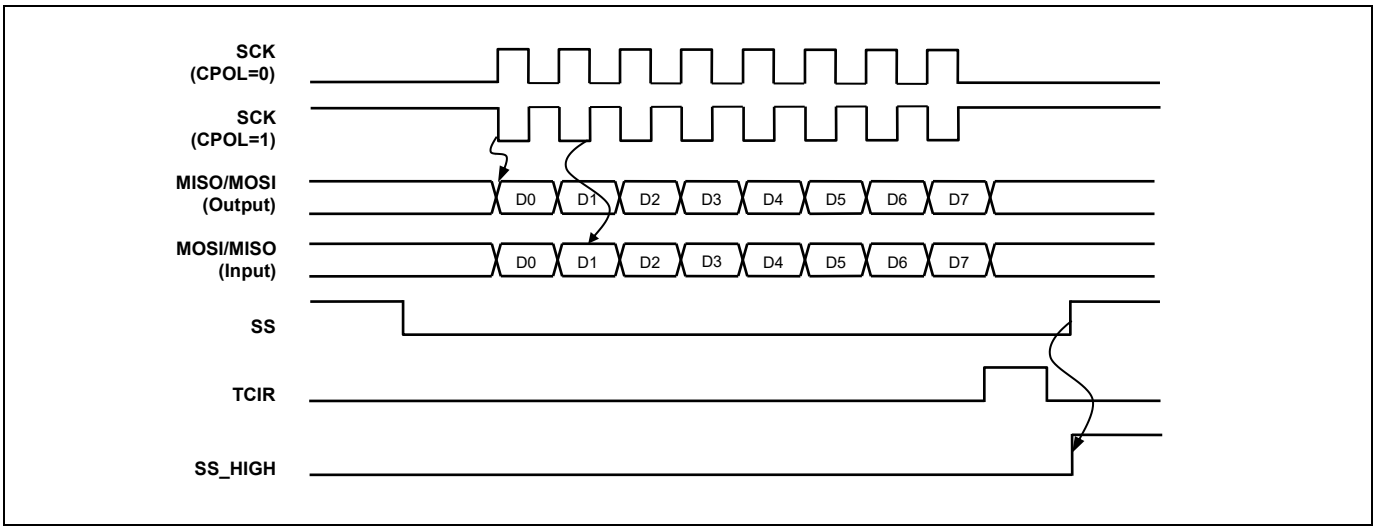


Figure 11.44 SPI Transmit/Receive Timing Diagram at CPHA = 1

11.11.6 Register Map

Name	Address	Direction	Default	Description
SPICR0	D8H	R/W	0H	SPI Control Register 0
SPIDR0	D9H	R/W	0H	SPI Data Register 0
SPISR0	D1H	-	0H	SPI Status Register 0
SPICR1	E8H	R/W	0H	SPI Control Register 1
SPIDR1	E9H	R/W	0H	SPI Data Register 1
SPISR1	E1H	-	0H	SPI Status Register 1

Table 11.18 SPI Register Map

11.11.7 SPI Register description

The SPI Register consists of SPI Control Register (SPICR), SPI Status Register (SPISR) and SPI Data Register (SPIDR)

11.11.8 Register Description for SPI

SPICRn (SPI Control Register) : D8H/E8H

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

Initial value : 00H

SPIEN — This bit controls the SPI operation
- 0 SPI Disable
- 1 SPI Enable

FLSB — This bit selects the data transmission sequence
- 0 MSB First
- 1 LSB First

MS — This bit selects whether Master or Slave mode
- 0 Slave mode
- 1 Master mode

CPOL, CPHA — These two bits control the serial clock (SCK) mode

Clock Polarity (CPOL) bit determine SCK's value at idle mode

Clock Phase (CPHA) bit determine if data is sampled on the leading or trailing edge of SCK. Refer to Figure 0-32, Figure 0-33

CPOL	CPHA	Leading Edge	Trailing Edge
0	0	Sample (Rising)	Setup (Falling)
0	1	Setup (Rising)	Sample (Falling)
1	0	Sample (Falling)	Setup (Rising)
1	1	Setup (Falling)	Sample (Rising)

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

fx– Main system clock oscillation frequency.

DSCR	SCR1	SCR0	SCK frequency
0	0	0	fx/4
0	0	1	fx/16
0	1	0	fx/64
0	1	1	fx/128
1	0	0	fx/2
1	0	1	fx/8
1	1	0	fx/32
1	1	1	fx/64

SPIDRn (SPI Data Register) : D9H/E9H

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

Initial value : 00H

SPIDR [7:0] SPI data register.
Although you only use reception, user must write any data in here to start the SPI operation.

SPISRn (SPI Status Register) : D1H/E1H

7	6	5	4	3	2	1	0
TCIR	WCOL	SS_HIGH	-	TWOPIN	SSENA	TXENA	RXENA
R	R	R/W	-	R/W	R/W	R/W	R/W

Initial value : 00H

TCIR When a serial data transmission is complete, the TCIR bit is set. If the SPI interrupt is enabled, an interrupt is requested. And TCIR bit is cleared by hardware when executing the corresponding interrupt. If SPI interrupt is disable, TCIR bit is cleared when user read the status register SPISR0, and then access (read/write) the data register SPIDR0.

0 Interrupt cleared

1 Transmission Complete and Interrupt Requested

WCOL This bit is set if the data register SPIDR0 is written during a data transfer. This bit is cleared when user read the status register SPISR0, and then access (read/write) the data register SPIDR0.

0 No collision

1 Write Collision

SS_HIGH When SS pin is configured as input(master or slave), if ‘HIGH’ signal come into SS pin, this flag bit will be set at the SS rising time. And you can clear it by writing ‘0’.

You can write only zero.

0 Flag is cleared

1 Flag is set

TWOPIN This bit controls the 2 pin operation.

0 Disable

1 Enable

SSENA This bit controls the SS pin operation

0 Disable

1 Enable

TXENA This bit controls a data transfer operation

0 Disable

1 Enable

RXENA This bit controls a data reception operation

0 Disable

1 Enable

Note that if the MS is set to ‘0’, when TWOPIN is set to ‘0’, port 03 is set to MISO and if the MS is set to ‘0’, when TWOPIN is set to ‘1’, port 02 is set to MOSI. But if the MS is set to ‘1’, when TWOPIN is set to ‘0’, port 03 is set to MOSI and if the MS is set to ‘1’, when TWOPIN is set to ‘1’, port 02 is set to MISO.

11.12 I2C

11.12.1 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

11.12.2 Block Diagram

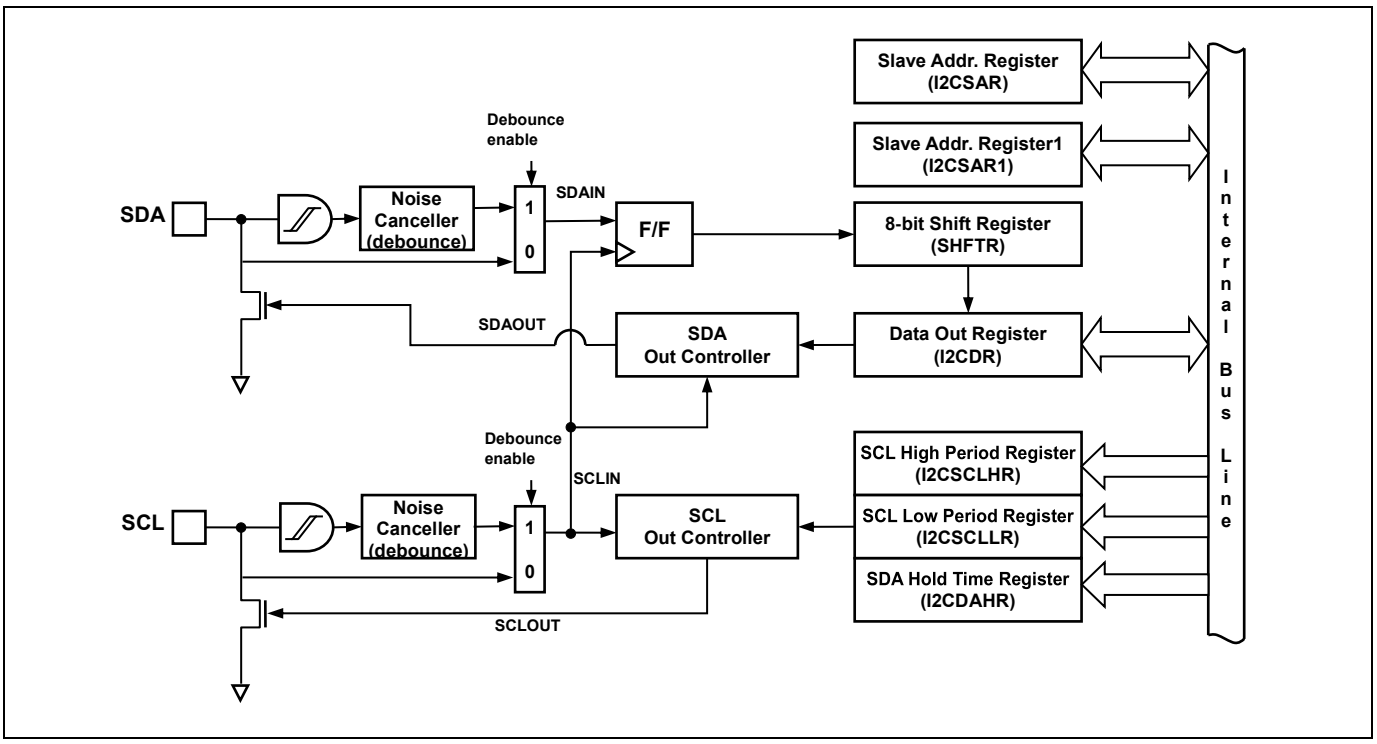


Figure 11.45 I2C Block Diagram

11.12.3 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.

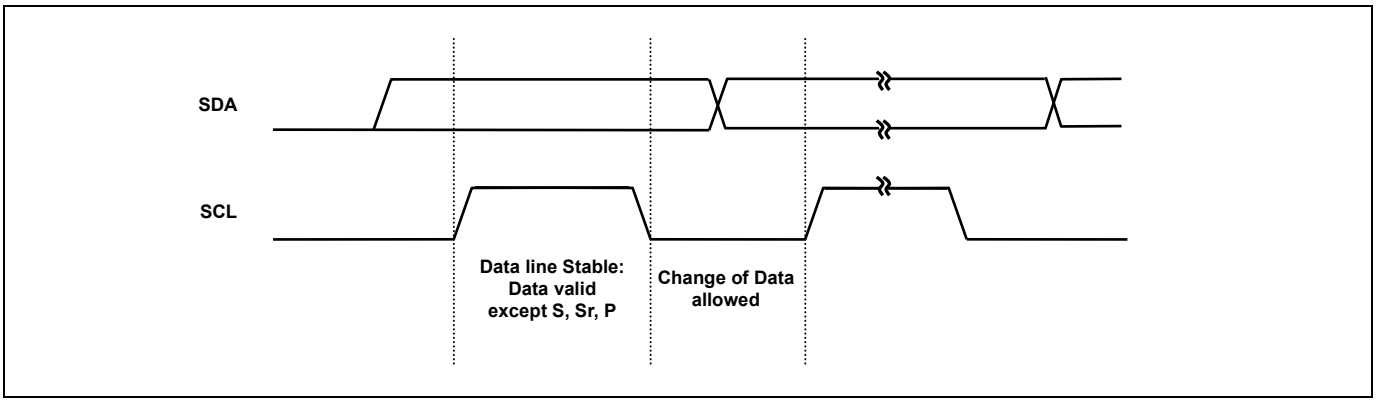


Figure 11.46 Bit Transfer on the I2C-Bus

11.12.4 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.

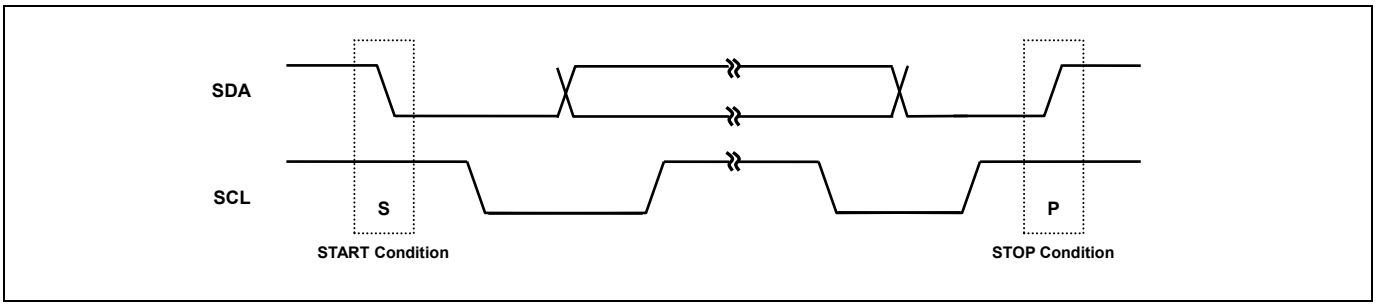


Figure 11.47 START and STOP Condition

11.12.5 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.

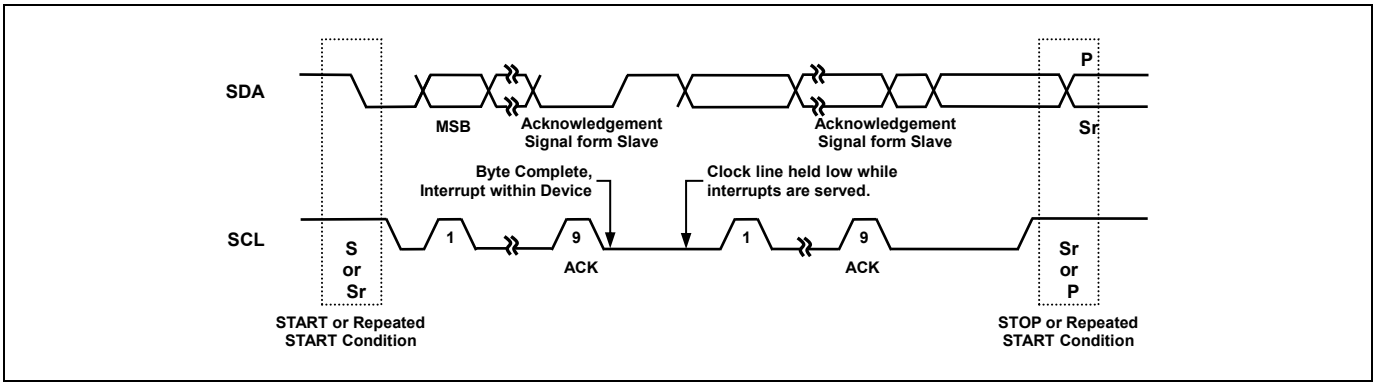


Figure 11.48 STOP or Repeated START Condition

11.12.6 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.

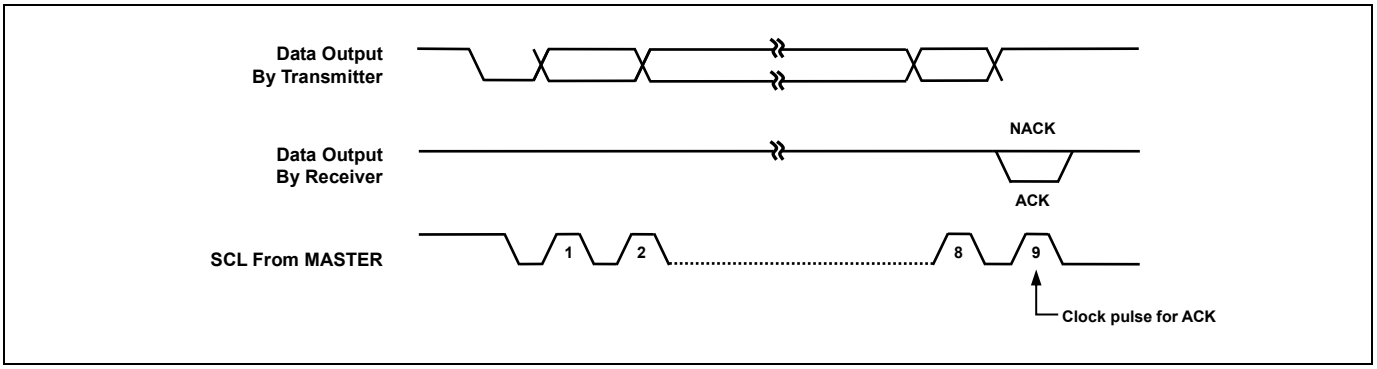


Figure 11.49 Acknowledge on the I2C-Bus

11.12.7 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.

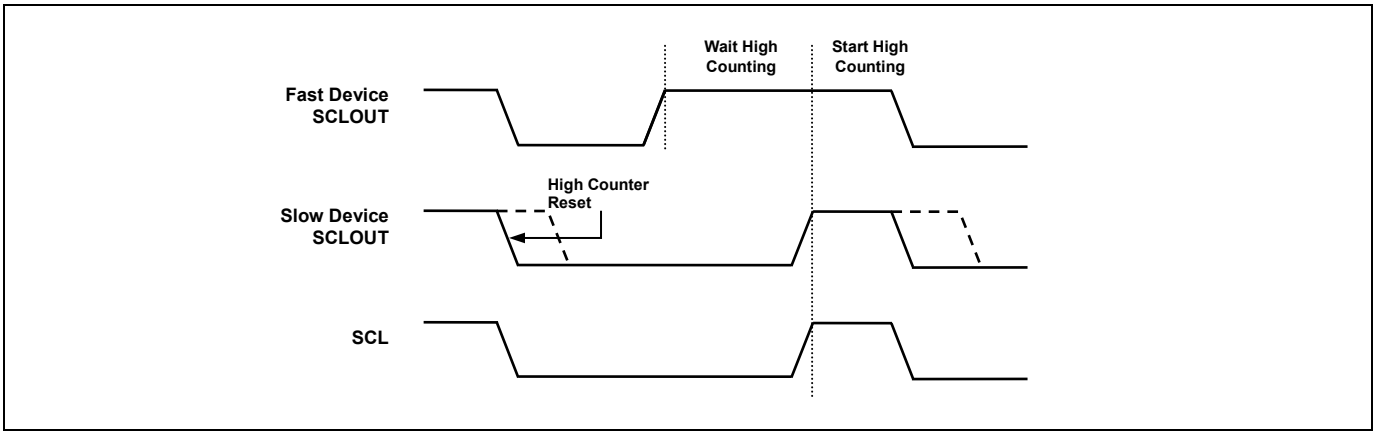


Figure 11.50 Clock Synchronization during Arbitration Procedure

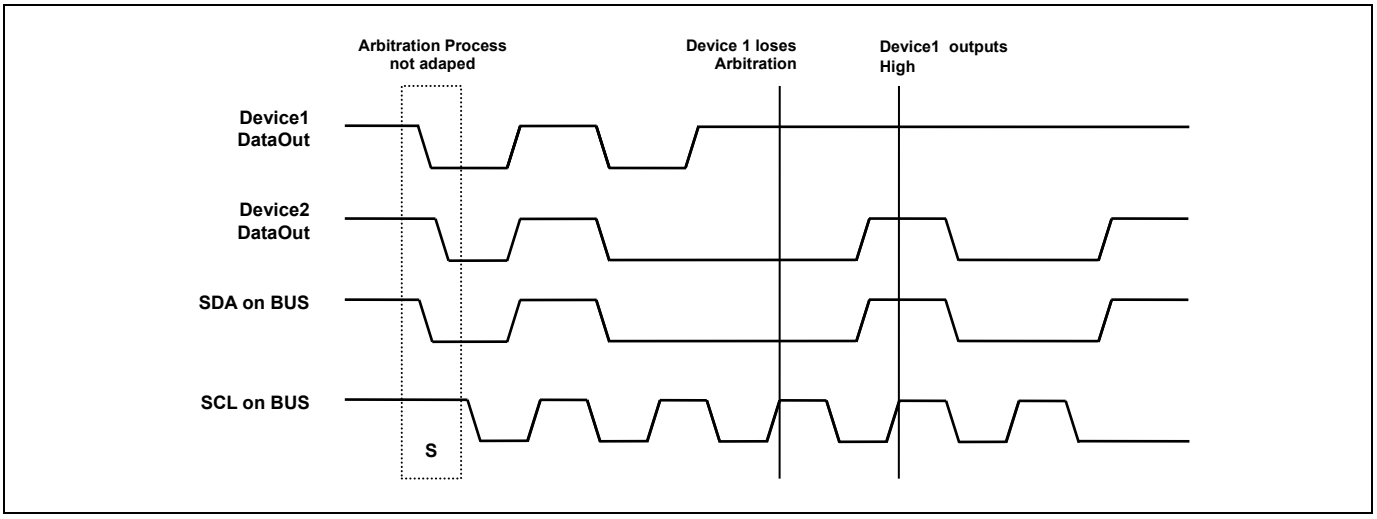


Figure 11.51 Arbitration Procedure of Two Masters

11.12.8 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.

11.12.8.1 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 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 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.

11.12.8.2 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.

   1) Master continues receiving data from slave. To do this, set ACKEN bit in I2CMR to ACK 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) 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.

11.12.8.3 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.

11.12.8.4 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.

11.12.9 Register Map

Name	Address	Dir	Default	Description
I2CMR00	DAH	R/W	00H	I2C0 Mode Control Register 0
I2CSR0	DBH	R	00H	I2C0 Status Register
I2CSCLLR0	DCH	R/W	3FH	I2C0 SCL Low Period Register
I2CSCLHR0	DDH	R/W	3FH	I2C0 SCL High Period Register
I2CSDAHR0	DEH	R/W	01H	I2C0 SDA Hold Time Register
I2CDR0	DFH	R/W	FFH	I2C0 Data Register
I2CSAR00	D7H	R/W	00H	I2C0 Slave Address 0 Register
I2CSAR01	D6H	R/W	00H	I2C0 Slave Address 1 Register
I2CMR01	8F1EH	R/W	00H	I2C0 Mode Control Register 1
I2CMR10	8F20H	R/W	00H	I2C1 Mode Control Register 0
I2CSR1	8F21H	R	00H	I2C1 Status Register
I2CSCLLR1	8F22H	R/W	3FH	I2C1 SCL Low Period Register
I2CSCLHR1	8F23H	R/W	3FH	I2C1 SCL High Period Register
I2CSDAHR1	8F24H	R/W	01H	I2C1 SDA Hold Time Register
I2CDR1	8F25H	R/W	FFH	I2C1 Data Register
I2CSAR10	8F26H	R/W	00H	I2C1 Slave Address 1 Register
I2CSAR11	8F27H	R/W	00H	I2C1 Slave Address Register
I2CMR11	8F1FH	R/W	00H	I2C1 Mode Control Register 1

Table 11.19 I2C Register Map

11.12.10 I2C Register Description

I2C Registers are composed of I2C Mode Control Register 0 (I2CMRn0), I2C Mode Control Register 1 (I2CMRn1), I2C Status Register (I2CSRn), SCL Low Period Register (I2CSCLLRn), SCL High Period Register (I2CSCLHRn), SDA Hold Time Register (I2CSDAHRn), I2C Data Register (I2CDRn), I2C Slave Address 0 Register (I2CSARn1) and I2C Slave Address 1 Register (I2CSARn1).

11.12.11 Register Description for I2C

I2CMRn0 (I²C Mode Control Register 0) : DAH, 8F20H

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

I2CSRn (I²C Status Register) : DBH, 8F21H

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 sources of interrupt. When an I²C interrupt occurs except for STOP interrupt, the SCL line is held LOW. To release SCL, write arbitrary value to I2CSR. When I2CSR is written, the TEND, STOP, SSEL, LOST and RXACK bits are cleared.

NOTE 2) The writing to I2CSR can be done in software regardless of the state of I²C. It should not be written to the I2CSR during transmit or receive sequence in order to ensure correct I²C operation.

I2CSCLLRn (SCL Low Period Register) : DCH, 8F22H

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.

I2CSCLHRn (SCL High Period Register) : DDH, 8F23H

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.

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

I2CSDAHRn (SDA Hold Time Register) : DEH, 8F24H

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.

I2CDRn (I²C Data Register) : DFH, 8F25H

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.

I2CSARn0 (I²C Slave Address Register 0) : D7H. 8F26H

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

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

I2CSARn1 (I2C Slave Address Register 1) : D6H, 8F27H

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

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

I2CMRn1 (I2C Mode Control Register 1) : 8F1EH, 8F1FH

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	DIS_SDAH
-	-	-	-	-	-	-	RW

Initial value : 00H

DIS_SDAH Disable SDA hold time

0 Enable SDA hold time.

When I2CSR is written, the MSB of the transmitted data of I2CDR is on the output to SDA port before three system clocks from the rising edge of SCL. The other bits after the MSB bit are outputted to SDA port in the timing controlled by I2CSDAHR from the falling edge of SCL.

1 Disable SDA hold time.

When I2CDR is written, the MSB of the transmitted data of I2CDR is on the output to SDA port directly. The other bits after the MSB bit are outputted after the falling edge of SCL irrespective of I2CSDAHR.

11.13 12-bit A/D Converter

11.13.1 Overview

The analog-to-digital converter (A/D) allows conversion of an analog input signal to 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. Also internal timer, external generating event, comparator, the trigger of timer1pwm and etc. can start ADC regardless of interrupt occurrence.

ADC Conversion Time = ADCLK * 60 cycles

After STBY bit is reset (ADC power enable) and it is restarted, during some cycle, ADC conversion value may have an inaccurate value.

When using ports as ADC input port, it is recommended to set corresponding PSR2, PSR3 registers to prevent current leakage or unexpected function, because analog value enters to digital circuit.

NOTE)

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

11.13.2 Block Diagram

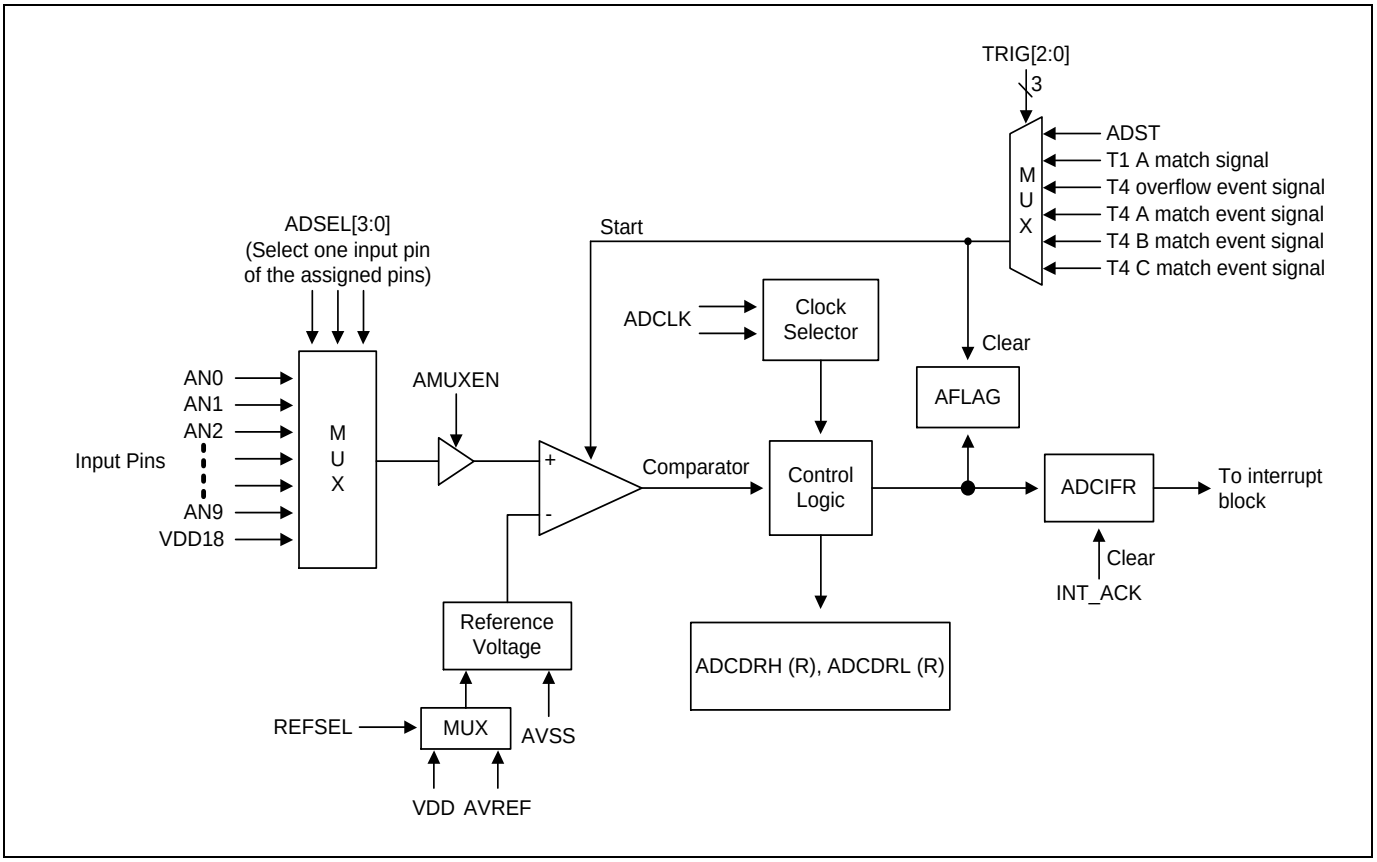


Figure 11.52 12-bit ADC Block Diagram

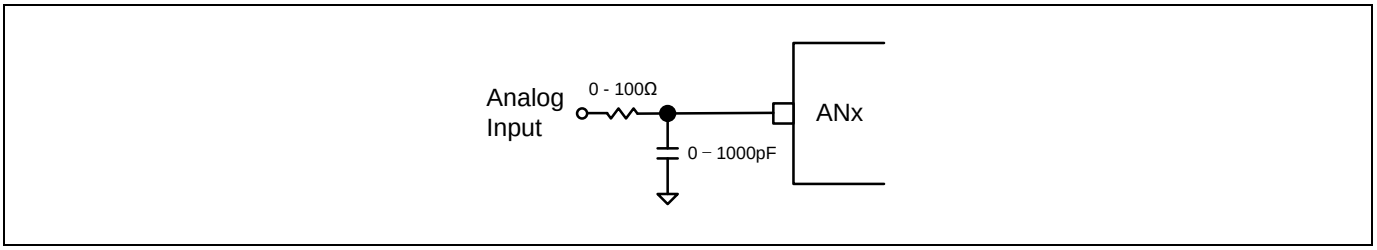


Figure 11.53 A/D Analog Input Pin with Capacitor

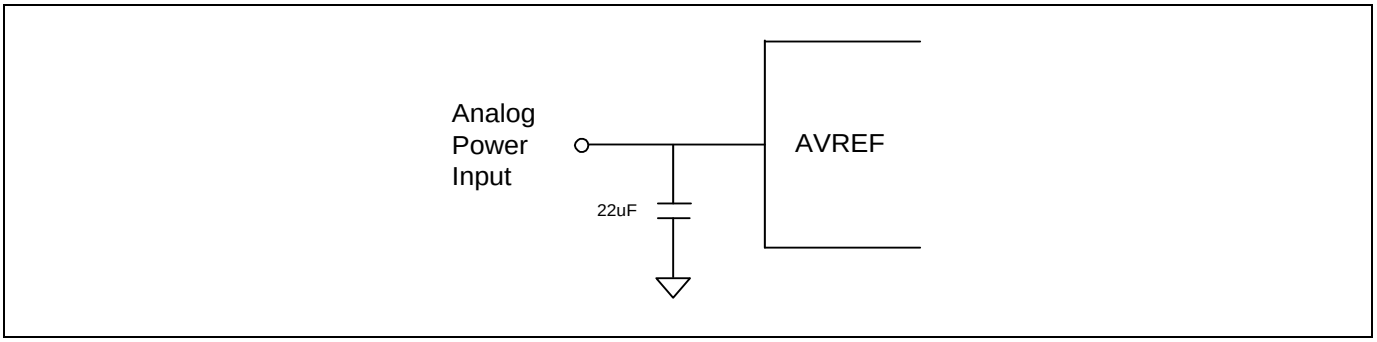


Figure 11.54 A/D Power (AVREF) Pin with Capacitor

11.13.3 ADC Operation

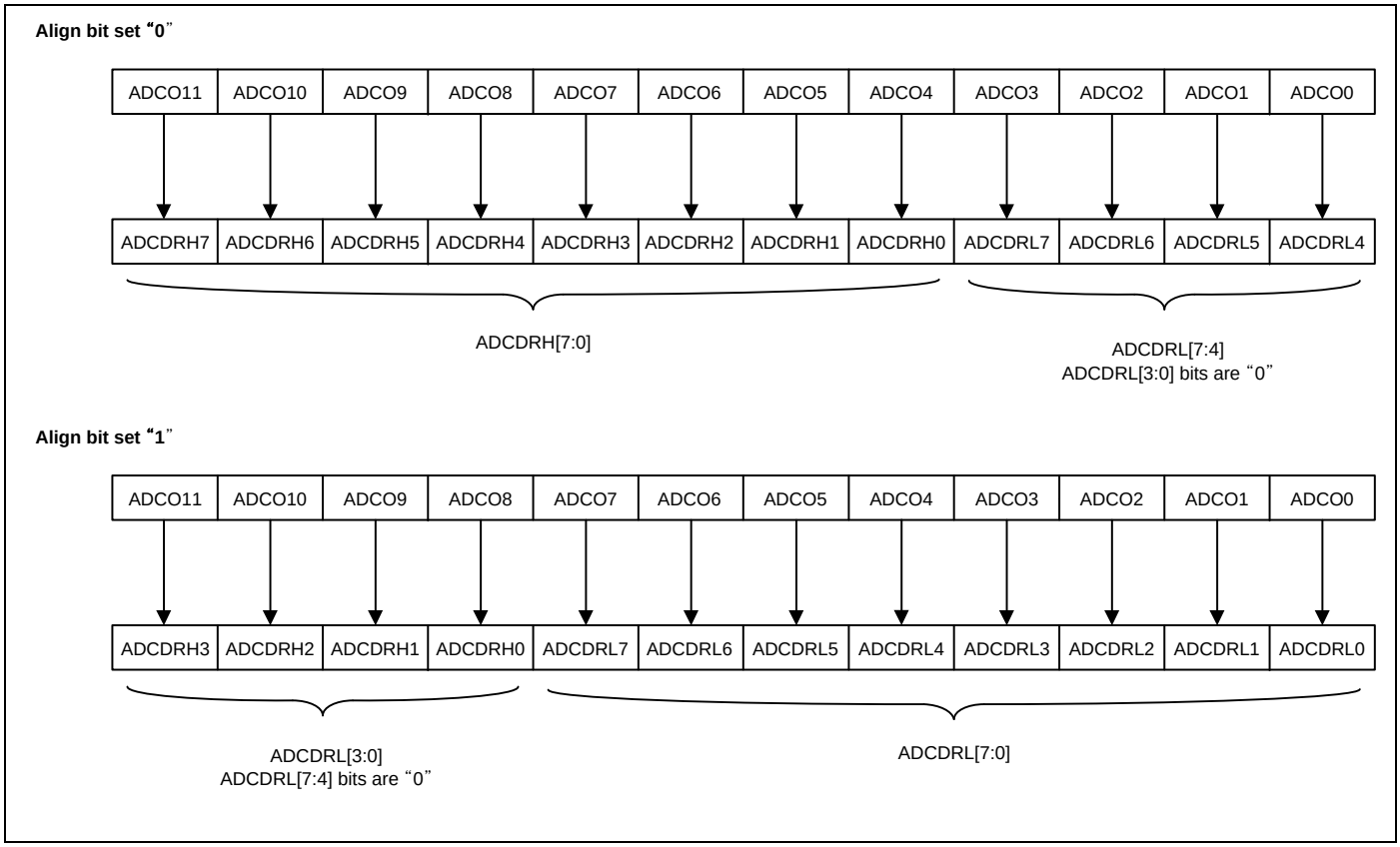


Figure 11.55 ADC Operation for Align Bit

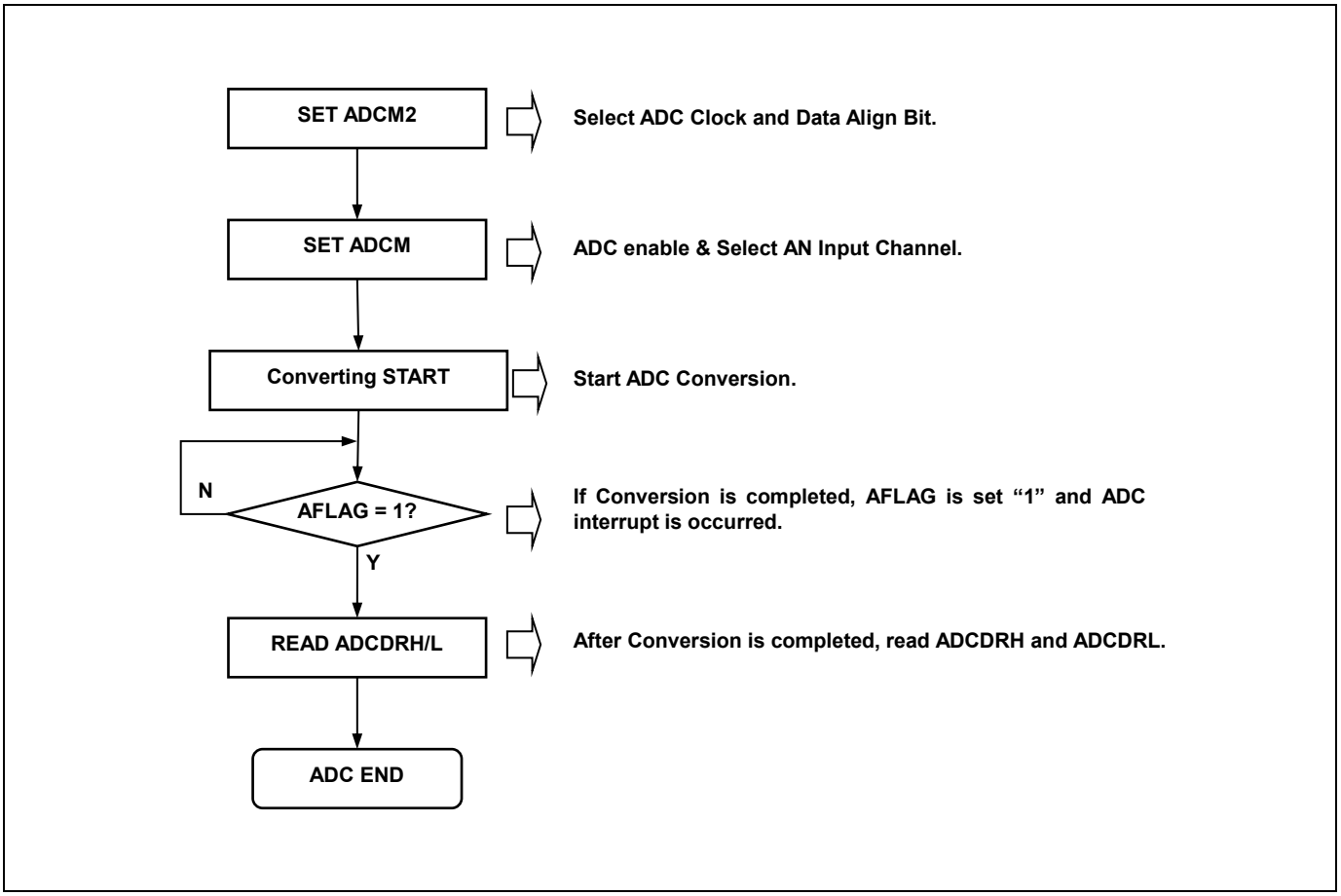


Figure 11.56 A/D Converter Operation Flow

11.13.4 Register Map

Name	Address	Direction	Default	Description
ADCM	9AH	R/W	8FH	A/D Converter Mode Register
ADCM2	92H	R/W	02H	A/D Converter Mode 2 Register
ADCRL	9BH	R	-	A/D Converter Result Low Register
ADCRH	9CH	R	-	A/D Converter Result High Register

Table 11.20 ADC Register Map

11.13.5 ADC Register Description

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 (ADCM2)..

NOTE) When STBY bit is set to '1', ADCM2 is read.

If ADC enables, it is possible only to write ADCM2. When reading, ADCRL is read.

11.13.6 Register Description for ADC

ADCM (A/D Converter Mode Register) : 9AH

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 (Vref=VDD)
- 1 External Reference(AVREF, AN0 disable)

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	Channel7(AN8)
1	0	0	1	Channel7(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	VDD18 (Test Only)

ADCM2 (A/D Converter Mode Register) : 92H

7	6	5	4	3	2	1	0
EXTRG	TSEL2	TSEL1	TSEL0	ALIGN	CKSEL2	CKSEL1	CKSEL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 02H

EXTRG A/D External Trigger

A/D conversion Start by external Trigger, and Stop by clearing this bit

0 A/D conversion Stop and External Trigger disable (default)

1 External Trigger enable

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

TSEL2	TSEL1	TSEL0	Description
0	0	0	Ext. Interrupt0
0	0	1	Ext. Interrupt1
0	1	0	Comparator interrupt
0	1	1	TIMER2 A-Match interrupt
1	0	0	TIMER3 A-Match interrupt
1	0	1	TIMER4 A-Match interrupt
1	1	0	TIMER5 A-Match interrupt
1	1	1	TIMER6 A-Match interrupt

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[2:0] A/D Converter Clock selection

CKSEL2	CKSEL1	CKSEL0	ADC Clock	ADC VDD
0	0	0	fx/2	Test Only
0	0	1	fx/4	3V~5V
0	1	0	fx/8 (default)	3V~5V
0	1	1	fx/16	3V~5V
1	0	0	fx/32	2.7V~3V
1	0	1	fx/64	2.4V~2.7V
1	1	0	fx/64	2.4V~2.7V
1	1	1	fx/64	2.4V~2.7V

NOTE) 1. fx : system clock

2. ADC clock must be used Max. 3MHz greater than 4V.

3. ADC clock must be used Max. 2MHz less than 4V.

ADCRL (A/D Converter Result Low Register) : 9BH

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

ADCRH (A/D Converter Result High Register) : 9CH

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

11.14 HDMI™- CEC Controller

11.14.1 Introduction

The A97C450 embeds a HDMI™-CEC interface controller hardware supporting the CEC standard specification (HDMI Specification Version 1.3a). The Consumer electronics control (CEC) is the appendix supplement 1 to the HDMI standard. This protocol including the physical layer and link layer provides high-level communication control between various audio and video products. In this section, we explain the hardware stack including PHY layer and link layer.
The CEC block consists of clock control block, TX/RX buffer (40 data buffers about each of TX buffers 20, RX buffers 20) and data control block. CEC Data transmitting and receiving process are executing by H/W block. The following functionalities are supported by CEC H/W block:

- Bit timing control (One, Zero tolerance)
- Control signal free time by H/W and Message time constraints by S/W application
- Issue end of message bit (EOM) by H/W
- Generate ACK/NACK signal depended on Message characteristic (broadcasting or direct address)
- Frame Re-transmissions
- Bit/Line error Handling
- Arbitration by end of transmission, and independently processing arbitration with RX process
  (does not lose RX Data even if device loses arbitration process)

11.14.2 HDMI™-CEC register map

Name	Address	Direction	Default	Description
CEC_PRES1	8F60H	R/W	00H	Clock prescaler register (High byte)
CEC_PRES0	8F61H	R/W	00H	Clock prescaler register (Low byte)
CEC_CONF1	8F62H	R/W	00H	CEC Block configuration register 1
CEC_CONF0	8F63H	R/W	00H	CEC Block configuration register 0
CEC_GCTRL	8F64H	R/W	00H	Signal free time control register
CEC_ICTRL	8F65H	R/W	00H	TX data control register
CEC_FCTRL	8F66H	W	00H	RX data count register
CEC_ISTAT	8F67H	R	00H	TX block state register
CEC_FSTAT	8F68H	R	00H	RX block state register
CEC_ICLR	8F69H	W	00H	TX block state clear register
CEC_FCLR	8F6AH	W	00H	RX block status clear register
CEC_TXH	8F6BH	R/W	00H	TX Header data buffer
CEC_TXD_1 ~ CEC_TXD_19	8F6CH ~ 8F7EH	R/W	00H	TX Data buffers
CEC_RXH	8F7FH	R	-	RX Header Data buffer
CEC_RXD_1 ~ CEC_RXD_19	8F80H ~ 8F92H	R	-	RX Data buffer

Table 11.21 HDMI CEC Register Map

11.14.3 HDMI-CEC Registers description

The HDMI-CEC Registers are defined at xdata SFR address area.

CEC_PRESx (CEC Prescaler Register) : 8F60H, 8F61H

7	6	5	4	3	2	1	0
PRES1_7	PRES1_6	PRES1_5	PRES1_4	PRES1_3	PRES1_2	PRES1_1	PRES1_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

7	6	5	4	3	2	1	0
PRES0_7	PRES0_6	PRES0_5	PRES0_4	PRES0_3	PRES0_2	PRES0_1	PRES0_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

CEC_PRESx The basic clock of CEC block requests 32.768KHz. Input source clock(System clock) changes to CEC base clock(32.68kHz) by setting prescaler register.

CEC_PRESx = (F_{SCLK} / 32.768kHz) / 2 – 1

F_{SCLK} = System clock

CEC_CONF1 (CEC Configuration Register 1) : 8F62H

7	6	5	4	3	2	1	0
INT_ERR_TX_EN	INT_ERR_ACK_EN	-	-	-	-	SUBCLK_EN	CEC_EN
R/W	R/W	-	-	-	-	R/W	R/W

Initial value : 00H

INT_ERR_TX_EN	At finding an error on CEC line, enable or disable CEC Interrupt event
	0 Disable CEC Interrupt at TX error
	1 Enable CEC Interrupt at TX error
INT_ERR_ACK_EN	At ACK error, enable or disable CEC Interrupt event
	0 Disable CEC Interrupt
	1 Enable CEC Interrupt
SUBCLK_EN	It selects CEC block clock source to use Sub OSC or Main system clock
	0 Select Main system clock as CEC clock source
	1 Select Sub OSC
CEC_EN	Enable / Disable CEC block active (In case of disable, CEC wake up is not available)
	0 Inactive CEC block
	1 Active CEC block

CEC_CONF0 (CEC Configuration Register 0) : 8F63H

7	6	5	4	3	2	1	0
-	TX_RETRY2	TX_RETRY1	TX_RETRY0	Dev_Add3	Dev_Add2	Dev_Add1	Dev_Add0
R/W	-	-	-	R/W	R/W	R/W	R/W

Initial value : 00H

TX_RETRY[6:4] Under uncertain condition of CEC line, if a CEC follower do not response ACK it decides Re-transmission times after 4 *2.4 Msec.

Dev_Add[3:0] Own Device Address (self-logical address)
CEC Device logical Address (0 ~ 15)

CEC_GCTRL (Signal free time control register) : 8F64H

7	6	5	4	3	2	1	0
SFT_7_10	SFT_7_10	SFT_7_10	SFT_5_7	SFT_5_7	SFT_3_5	SFT_3_5	LINE_BUSY
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Initial value : 00H

CEC_GCTRL Before attempting to transmit or re-transmit a frame, a device shall ensure that the CEC line has been inactive for a number of bit periods. This signal free time is defined as the time since the start of the final bit of the previous frame. It define signal free times.

SFT_7_10[2:0] When present initiator wants to send another frame immediately after its previous frame, it decide waiting time before sending.
Default signal free time (7x2.4) ms
Waiting time = (7+0.5 x SFT_7_10)x2.4 ms

SFT_5_7[1:0] When new initiator wants to send a frame, it decide waiting time before sending.
Default signal free time (5x2.4)ms
Waiting time =(5+0.5 * SFT_5_7)x2.4 ms

SFT_3_5[1:0] When previous attempt to send frame unsuccessful, it decide waiting time before sending.
Default signal free time (3 x 2.4) ms
Waiting time =(3+0.5 * SFT_3_5)x2.4 ms

LINE_BUSY If it is high state, CEC line is busy (occupying CEC line by any CEC device)
0 Free CEC line
1 Line busy

CEC_ICTRL (CEC TX data control register) : 8F65H

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

Initial value : 00H

TX_DSIZE TX data buffers on CEC block have 19. It decides transmission data sizes except header byte.

0 Just transmit header byte.

n Max TX data size is 19

TX_MODE If it is set, it starts to transmit TX data after writing TX data on TX data buffers and TX Data size.

It is cleared by H/W after finishing data transmission completely

(It is possible to use a busy signal for transmission line)

CEC_FCTRL (CEC RX data count register) : 8F66H

7	6	5	4	3	2	1	0
-	-	-	RX_DSISE	RX_DSISE	RX_DSISE	RX_DSISE	RX_DSISE
-	-	-	-	R	R	R	R

Initial value : 00H

RX_DSIZE RX data buffers on CEC block have 19. The value of register is RX data size received from initiator.

0 Just receive header byte.

n Max RX data size is 19

CEC_ISTAT (CEC TX block state register) : 8F67H

7	6	5	4	3	2	1	0
-	-	-	INT_ERR_TX_OTHER	INT_ERR_ACK	INT_ARBIT	INT_TX_FAIL	INT_TX_DONE
-	-	-	-	R	R	R	R

Initial value : 00H

INT_ERR_TX_OTHER — After finishing transmission it indicates an error from a follower or CEC line. It is cleared by setting CLR_INT_ERR_TX_OTHER bit on CEC_ICLR register
- 0 No error
- 1 An error is invoked

INT_ERR_ACK — After finishing transmission, it indicates to have right ACK/NACK or not. It is cleared by setting CLR_INT_ERR_ACK bit on CEC_ICLR register
- 0 Getting right ACK/NACK
- 1 An ACK/NACK error is invoked

INT_ARBIT — It indicates to have an arbitration process and the device has lost CEC line occupation. If the device loses CEC line right, S/W has to start re-transmit data after waiting signal free time.
It is cleared by setting CLR_INT_ARBIT bit on CEC_ICLR register
- 0 no lost CEC line occupation
- 1 lost arbitration

INT_TX_FAIL — It indicates a transmission fail after finishing re-transmissions (defined by TX_RETRY).
It is cleared by setting CLR_INT_TX_FAIL bit on CEC_ICLR register
- 0 No error transmission
- 1 Completely transmission fail

INT_TX_DONE — It indicates a transmission to be completely done.
It is cleared by setting CLR_INT_TX_DONE bit on CEC_ICLR register
- 0 No meaning
- 1 Set after a frame transmission to be done.

CEC_FSTAT (CEC RX block state register) : 8F68H

7	6	5	4	3	2	1	0
-	-	-	-	INT_ERR_RX_OTHER	INT_ERR_BIT	INT_ERR_OVF	INT_RX_DONE
R	-	-	-	R	R	R	R

Initial value : XXH

INT_ERR_RX_OTHER — While receiving RX data and it has any error by other device or abnormal line condition, it indicates an error (insert 3.6ms low time). It is cleared by setting CLR_INT_ERR_RX_OTHER bit on CEC_FCLR register
- 0 No error
- 1 An error is invoked

INT_ERR_BIT — It indicates low period of RX data to be long.
- 0 No error
- 1 Low period of RX is long.

INT_ERR_OVF — It indicates to overflow RX buffers(more than receiving 19 data)
It is cleared by setting CLR_INT_ERR_OVF bit on CEC_FCLR register
- 0 No overflow
- 1 RX buffer overflow

INT_RX_DONE — It indicates to finish RX reception without any problem.
It is cleared by setting CLR_INT_RX_DONE bit on CEC_FCLR register
- 0 No meaning
- 1 Completely received RX Data (directly or broadcasting data)

CEC_ICLR (CEC TX block state clear register) : 8F69H

7	6	5	4	3	2	1	0
-	-	-	CLR_INT_ERR_TX_OTHER	CLR_INT_ERR_ACK	CLR_INT_ARBIT	CLR_INT_TX_FAIL	CLR_INT_TX_DONE
-	-	-	W	W	W	W	W

Initial value : 00H

CLR_INT_ERR_TX_OTHER	It clear INT_ERR_TX_OTHER bit on CEC_ISTAT register	
	0	-
	1	Clear INT_ERR_TX_OTHER bit
CLR_INT_ERR_ACK	It clear INT_ERR_ACK bit on CEC_ISTAT register	
	0	-
	1	Clear INT_ERR_ACK bit
CLR_INT_ARBIT	It clear INT_ARBIT bit on CEC_ISTAT register	
	0	-
	1	Clear INT_ARBIT bit
CLR_INT_TX_FAIL	It clear INT_TX_FAIL bit on CEC_ISTAT register	
	0	-
	1	Clear INT_TX_FAIL bit
CLR_INT_TX_DONE	It clear INT_TX_DONE bit on CEC_ISTAT register	
	0	-
	1	Clear INT_TX_DONE bit

CEC_FCLR (CEC RX block state clear register) : 8F6AH

7	6	5	4	3	2	1	0
-	-	-	-	CLR_INT_ERR_RX_OTHER	CLR_INT_ERR_BIT	CLR_INT_ERR_OVF	CLR_INT_RX_DONE
W	-	-	-	W	W	W	W

Initial value : XXH

CLR_INT_ERR_RX_OTHER	It clear INT_ERR_RX_OTHER bit on CEC_FSTAT register	
	0	-
	1	Clear INT_ERR_RX_OTHER bit
CLR_INT_ERR_BIT	It clear INT_ERR_BIT bit on CEC_FSTAT register	
	0	-
	1	Clear INT_ERR_BIT bit
CLR_INT_ERR_OVF	It clear INT_ERR_OVF bit on CEC_FSTAT register	
	0	-
	1	Clear INT_ERR_OVF bit
CLR_INT_RX_DONE	It clear INT_RX_DONE bit on CEC_FSTAT register	
	0	-
	1	Clear INT_RX_DONE bit

CEC_TXH (CEC TX Header data buffer) : 8F6BH

7	6	5	4	3	2	1	0
TXH_7	TXH_6	TXH_5	TXH_4	TXH_3	TXH_2	TXH_1	TXH_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : XXH

CEC_TXH [7:4] Initiator logical address 4 bit

CEC_TXH [3:0] Follower logical address 4 bit (or 0x0F broadcasting address)

CEC_TXD_X (CEC TX Data buffers) : 8F6CH ~ 8F7EH

7	6	5	4	3	2	1	0
TXD_7	TXD_6	TXD_5	TXD_4	TXD_3	TXD_2	TXD_1	TXD_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

CEC_ TXD CEC TX data buffers(TX CEC Op code, parameters)

CEC_RXH (CEC RX Header data buffer) : 8F7FH

7	6	5	4	3	2	1	0
TXH_7	TXH_6	TXH_5	TXH_4	TXH_3	TXH_2	TXH_1	TXH_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

CEC_RXH [7:4] destination logical address 4 bit (0xF0 broadcasting address)

CEC_TXH [3:0] initiator logical address 4 bit

CEC_RXD_X (CEC RX Data buffers) : 8F80H ~8F92H

7	6	5	4	3	2	1	0
RXD_7	RXD_6	RXD_5	RXD_4	RXD_3	RXD_2	RXD_1	RXD_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

CEC_RXD CEC RX data buffers(RX CEC Op code, parameters)

The following diagram describes response state for TX block state register when CEC line has no follower device or CEC devices do retry

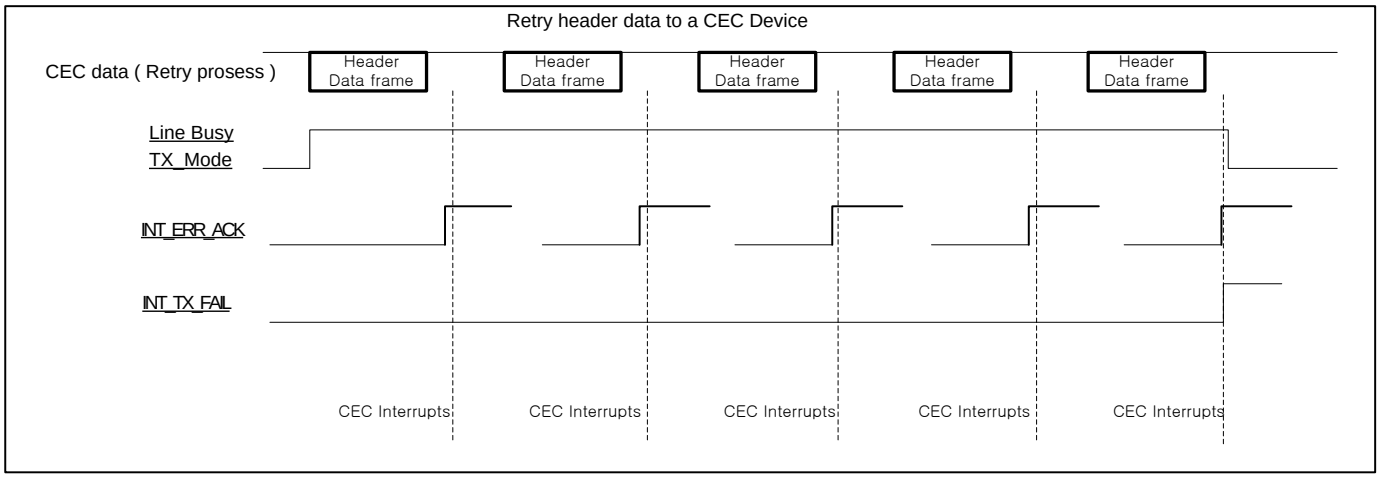


Figure 11.57 A/D Analog Input Pin Connecting Capacitor

11.15 IR Controller

11.15.1 Introduction

The A97C450 embeds the hardware IR receiver with noise filter. Remote infra-red receivers are very sensitive to optical signals and may result in wrong data decoding. The noise filter will improve noise immunity of remote control signal. And then, the probabilities of wrong data decoding and the wake-up by noise on Stop mode are decreased.

11.15.2 IR Controller Register map

Name	Address	Direction	Default	Description
IRC_PRES1	8F48H	R/W	00H	IRC Prescaler High Register
IRC_PRES0	8F49H	R/W	00H	IRC Prescaler Low Register
IRC_FRMP1	8F4AH	R/W	00H	IRC Frame Period Parameter High Register
IRC_FRMP0	8F4BH	R/W	00H	IRC Frame Period Parameter Low Register
IRC_CONF	8F4CH	R/W	00H	IRC Configuration Register
IRC_FLAG	8F4DH	R/W	00H	IRC Status Register
IRC_EDGE1	8F4EH	R	00H	IRC Edge Counter Data High Register
IRC_EDGE0	8F4FH	R	02H	IRC Edge Counter Data Low Register

Table 11.22 IR Controller Register map

11.15.3 IR Controller Registers description

The IR Controller consists of IRC Prescaler Register (IRC_PRES1, IRC_PRES0), IRC Frame Time Data Register (IRC_FRMP1, IRC_FRMP0), IRC Configuration Register (IRC_CONF), IRC Status Register (IRC_FLAG), and IRC Edge Capture Data Register (IRC_EDGE1, IRC_EDGE0).

IRC_PRES1 (IRC Prescaler Register) : 8F48H

7	6	5	4	3	2	1	0
PRES1_7	PRES1_6	PRES1_5	PRES1_4	PRES1_3	PRES1_2	PRES1_1	PRES1_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

IRC_PRES0 (IRC Prescaler Register) : 8F49H

7	6	5	4	3	2	1	0
PRES0_7	PRES0_6	PRES0_5	PRES0_4	PRES0_3	PRES0_2	PRES0_1	PRES0_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

IRC_PRES1/ IRC_PRES0 The registers are the prescaler value to set IR timer counter clock of IR pulse. It is written by software.

IRC_PRESx = (F_{SCLK} / Target frequency) / 2 - 1

Fsclk is system clock frequency. Target frequency is a base clock of IRC Edge Counter and IRC Frame Period Parameter

Ex) In case of IRC_EDGE1/0 base clock = 0.03msec (32.768kHz), F_{SCLK} = 16MHz , **IRC_PRES1/0 = 243 ((**16MHz/32.768kHz))/2-1)

IRC_FRMP1 (IRC Frame Period Parameter Register) : 8F4AH

7	6	5	4	3	2	1	0
FRMP1_7	FRMP1_6	FRMP1_5	FRMP1_4	FRMP1_3	FRMP1_2	FRMP1_1	FRMP1_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

IRC_FRMP0 (IRC Frame Period Parameter Register) : 8F4BH

7	6	5	4	3	2	1	0
FRMP0_7	FRMP0_6	FRMP0_5	FRMP0_4	FRMP0_3	FRMP0_2	FRMP0_1	FRMP0_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

IRC_FRAM1/ IRC_FRMP0 The both registers are used to make IR frame periods. These values are written by software. The counter step depends on IRC Prescaler Register. If IRC frame period has more than the value of IRC_FRMPx, it invokes interrupt event (The interrupt vector address is equal to IR INT0).

IR frame periods = Frame time * Target frequency

If the values of IRC_FRAM1 and IRC_FRMP0 are 0FFH, frame interrupt event isn't invoked and interrupt is disabled.

IRC_CONF (IRC Configuration Register) : 8F4CH

7	6	5	4	3	2	1	0
AWAKE_EN	REDGE_EN	SEDGE_EN	FLT_SEL[2]	FLT_SEL[1]	FLT_SEL[0]	SUBCLK_EN	IRC_EN
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W

Initial value : 00H

AWAKE_EN Asynchronous wake-up enable. This is used at STOP mode in case of using sub-oscillator as IRC clock source. If this bit is set, STOP mode can be waked up by INT_WAKE interrupt of IRC_CTRL.
- 0 disable asynchronous wake-up
- 1 enable asynchronous wake-up

RDGE_EN This is rising or falling edge interrupt enable bit.
- 0 enable falling edge interrupt
- 1 enable rising edge interrupt

SEDGE_EN This decide the edge interrupt type that is single edge (rising or falling state) interrupt or both edge (rising and falling state).
- 0 enable both edge interrupt
- 1 enable single edge interrupt

FLT_SEL These are noise filter selection bits.
- 0 No noise filter
- 1 Noise: <1 cycle, Signal: >2 cycles
- 2 Noise: <2 cycle, Signal: >3 cycles
- 3 Noise: <3 cycle, Signal: >4 cycles
- 4 Noise: <4 cycle, Signal: >5 cycles
- 5 Noise: <5 cycle, Signal: >6 cycles
- 6 Noise: <6 cycle, Signal: >7 cycles
- 7 Noise: <7 cycle, Signal: >8 cycles

SUBCLK_EN Sub-oscillator clock select. If this bit is set, IRC clock source is sub-oscillator clock.
- 0 The system clock is used as IRC clock source
- 1 Sub-oscillator clock is used as IRC clock source

IRC_EN Enable IRC block
- 0 IRC block is inactive
- 1 IRC block is active

IRC_FLAG (IRC status Register) : 8F4DH

7	6	5	4	3	2	1	0
INT_AWAKE	-	-	-	INT_FRM	INT_EDGE	IRC_RX	TIMER_EN
R	-	-	-	R	R	R	R/W

Initial value : 00H

INT_AWAKE This bit is used to indicate when IRC is waked up from STOP mode in case of using sub-oscillator clock as IRC clock source. This bit can be invoked when AWAKE_EN of IRC_CONF is set.

0 Asynchronous wake-up is not invoked

1 Asynchronous wake-up is invoked

INT_FRM It indicate the interrupt event by IRC frame counter**. It is cleared by software.**

0 Frame time interrupt is not invoked

1 Frame time interrupt is invoked

INT_EDGE This is single edge interrupt flag bit. **It is cleared by software.**

0 Edge interrupt is not invoked

1 Edge interrupt is invoked

IRC_RX The state of IR signal line is indicated

0 IR line is low

1 IR line is high

TIMER_EN This bit is used to control frame and edge timer. If IRC_RX has event, TIMER_EN is set by hardware. If frame is finished, **TIMER_EN is cleared by software for next frame.**

0 Frame and edge timer is cleared and disable.

1 Frame and edge timer is enable

IRC_EDGE1 (IRC Edge Counter Register) : 8F4EH

7	6	5	4	3	2	1	0
EDGE1_7	EDGE1_6	EDGE1_5	EDGE1_4	EDGE1_3	EDGE1_2	EDGE1_1	EDGE1_0
R	R	R	R	R	R	R	R

Initial value : 00H

IRC_EDGE0 (IRC Edge Counter Register) : 8F4FH

7	6	5	4	3	2	1	0
EDGE0_7	EDGE0_6	EDGE0_5	EDGE0_4	EDGE0_3	EDGE0_2	EDGE0_1	EDGE0_0
R	R	R	R	R	R	R	R

Initial value : 00H

IRC_EDGE1/ IRC_EDGE0 The both registers are IR capture counter between IR signal edge events. At any case, IRC_EDGE1/0 is only used depended on IRC Prescaler value(IRC_PRES1/0)

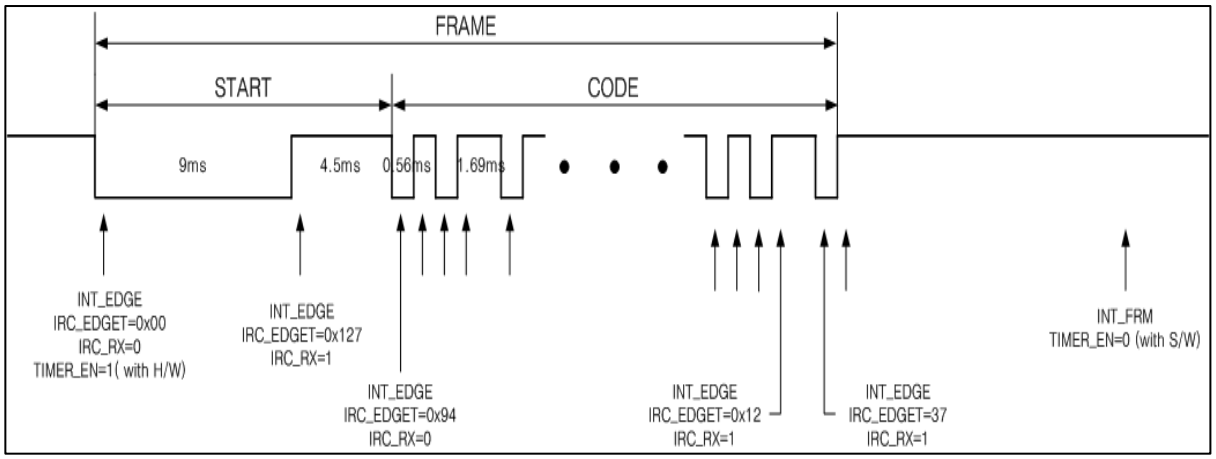


Figure 11.58 IR Receive Timing

11.16 Comparator

11.16.1 Overview

The A97C450 contains 2 analog comparators which can be easily used to compare the external input voltage with internal reference voltage. ADC and Comparator have the same input. When the comparator input voltage is smaller than the reference voltage comparator output status is '0' and interrupt flag is generated. Each comparator interrupt flag is assigned to one interrupt vector. Comparator input debounce clock and length can be set.

11.16.2 Block Diagram

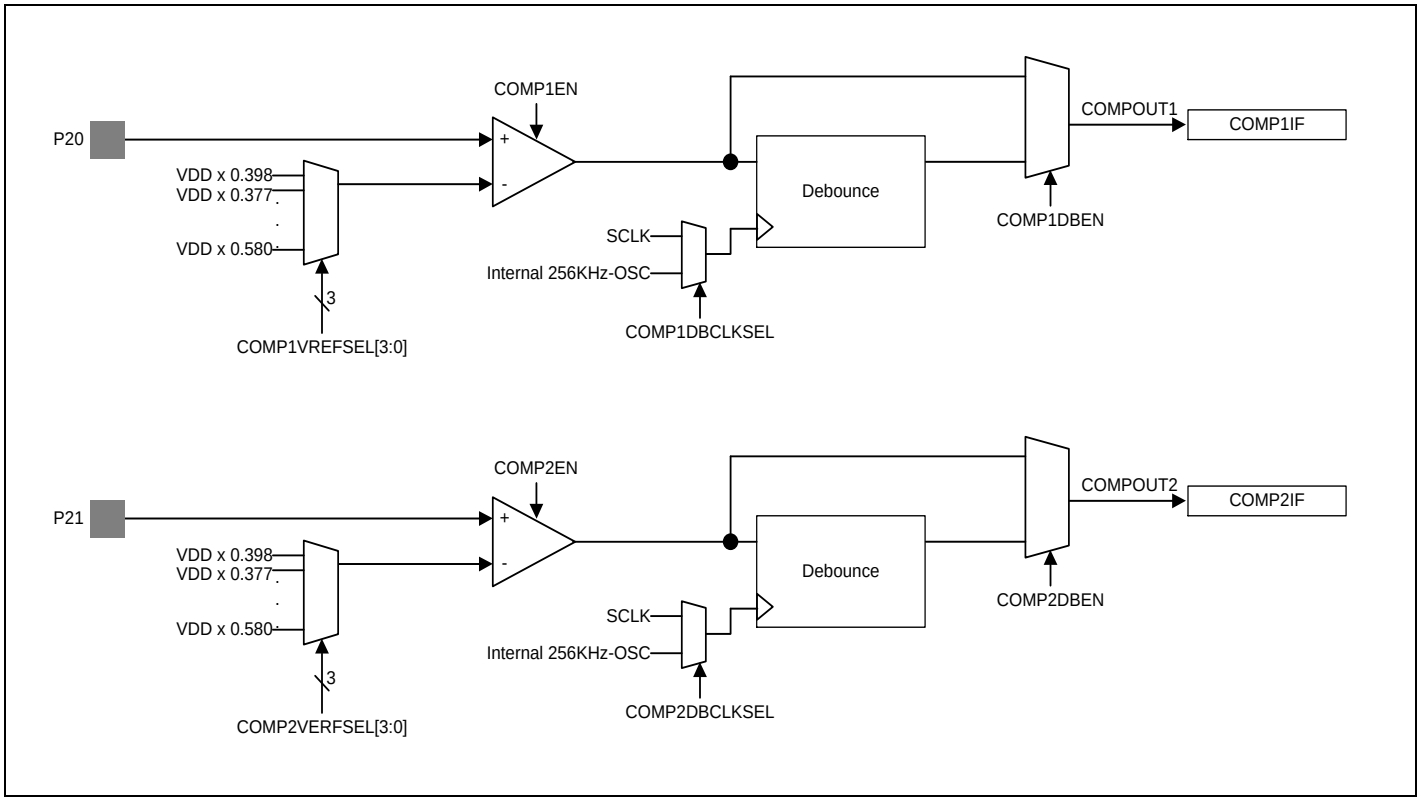


Figure 11.59 Buzzer Driver Block Diagram

11.16.3 Register Map

Name	Address	Direction	Default	Description
COMPCON	8F30H	R/W	00H	Comparator Control Register
COMPVREFSEL	8F31H	R/W	00H	Comparator Voltage Reference Select Register
COMPST	8F32H	R	11H	Comparator Output Read Register
COMPDBCNT	8F33H	R/W	FFH	Comparator Output Debounce Register

Table 11.23 Comparator Register Map

11.16.4 Comparator Register description

COMPCON (Comparator Control Register) : 8F30H

7	6	5	4	3	2	1	0
COMP2IF	COMP2DBCLKSEL	COMP2DBEN	COMP2EN	COMP1IF	COMP1DBCLKSEL	COMP1DBEN	COMP1EN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FFH

COMP2IF	Comparator 2 interrupt flag. The flag is cleared only by writing a '1' to the bit. So, the flag should be cleared by software. Writing "0" has no effect.	
	0	Comparator 2 interrupt not occurred
	1	Comparator 2 interrupt occured
COMP2DBCLKSEL	Select debounce clock of comparator 2 (DBCLK2)	
	0	DBCLK2 = SCLK
	1	DBCLK2 = Internal 256KHz-OSC
COMP2DBEN	Debounce of comparator 2 output control	
	0	Comparator 2 output debounce disable
	1	Comparator 2 output debounce enable
COMP2EN	Comparator 2 operation control	
	0	Comparator 2 disable
	1	Comparator 2 enable
COMP1IF	Comparator 1 interrupt flag. The flag is cleared only by writing a '1' to the bit. So, the flag should be cleared by software. Writing "0" has no effect.	
	0	Comparator 1 interrupt not occurred
	1	Comparator 1 interrupt occured
COMP1DBCLKSEL	Select debounce clock of comparator 1 (DBCLK1)	
	0	DBCLK1 = SCLK
	1	DBCLK1 = Internal 256KHz-OSC
COMP1DBEN	Debounce of comparator 1 output control.	
	0	Comparator 1 output debounce disable
	1	Comparator 1 output debounce enable
COMP1EN	Comparator 1 operation control	
	0	Comparator 1 Disable
	1	Comparator 1 Enable

COMPVREFSEL (Comparator Voltage Reference Select Register) : 8F31H

7	6	5	4	3	2	1	0
COMP2VREFSEL[3:0]				COMP1VREFSEL[3:0]			
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

COMP2VREFSEL[3:0]	Select comparator 2 reference voltage	
	0000	VDD x 0.348
	0001	VDD x 0.377
	0010	VDD x 0.406
	0011	VDD x 0.435
	0100	VDD x 0.464
	0101	VDD x 0.493
	0110	VDD x 0.522
	0111	VDD x 0.551
	1000	VDD x 0.580
	1001	Not used
	1010	Not used
	1011	Not used
	1100	Not used
	1101	Not used
	1110	Not used
	1111	Not used
COMP1VREFSEL[3:0]	Select comparator 1 reference voltage	
	0000	VDD x 0.348
	0001	VDD x 0.377
	0010	VDD x 0.406
	0011	VDD x 0.435
	0100	VDD x 0.464
	0101	VDD x 0.493
	0110	VDD x 0.522
	0111	VDD x 0.551
	1000	VDD x 0.580
	1001	Not used
	1010	Not used
	1011	Not used
	1100	Not used
	1101	Not used
	1110	Not used
	1111	Not used

COMPST (Comparator Output Read Register) : 8F32H

7	6	5	4	3	2	1	0
-	-	-	COMPOUT2	-	-	-	COMPOUT1
-	-	-	R/W	-	-	-	R/W

Initial value : 11H

COMPOUT2	Comparator 2 output status	
	0	Comparator 2 output low
	1	Comparator 2 output high (default)
COMPOUT1	Comparator 1 output status	
	0	Comparator 1 output low
	1	Comparator 1 output high (default)

COMPDBCNT (Comparator Output Debounce Register) : 8F33H

7	6	5	4	3	2	1	0
-	COMP2DBCNT			-	COMP1DBCNT		
-	R/W	R/W	R/W	-	R/W	R/W	R/W

Initial value : FFH

COMP2DBCNT[2:0]	Select comparator 2 output debounce length Debounce length = COMP2DBCNT x 2 x 1/f_{dbclk2}
COMP1DBCNT[2:0]	Select comparator 1 output debounce length Debounce length = COMP1DBCNT x 2 x 1/f_{dbclk1}

11.17 RTC (Real-time Clock)

11.17.1 Overview

The real-time clock has the following features.

- Using the SUB clock as RTC clock source (f_{RTC})
- Having counters of year, month, week, day, hour, minute, and second, and can count up to 99 years.
- Constant-period interrupt function (period: 0.5 seconds, 1 second, 1 minute, 1 hour, 1 day, 1 month)
- Alarm interrupt function (alarm: week, hour, minute)
- Pin output function of 1Hz

The real-time clock interrupt signal (RTCIFR) can be utilized for wakeup from STOP.

11.17.2 Block Diagram

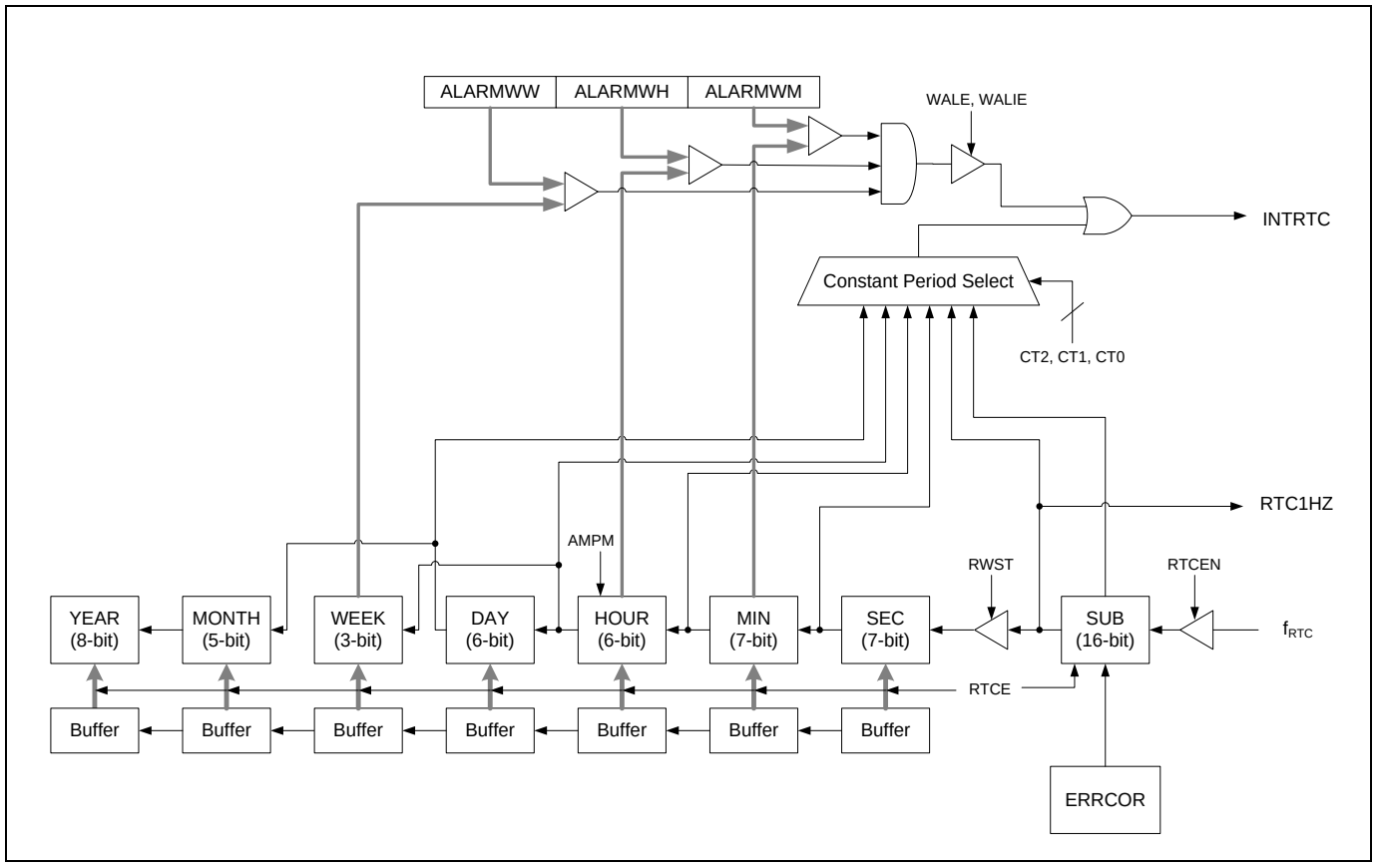


Figure 11.60 RTC block diagram

11.17.3 RTC Operation

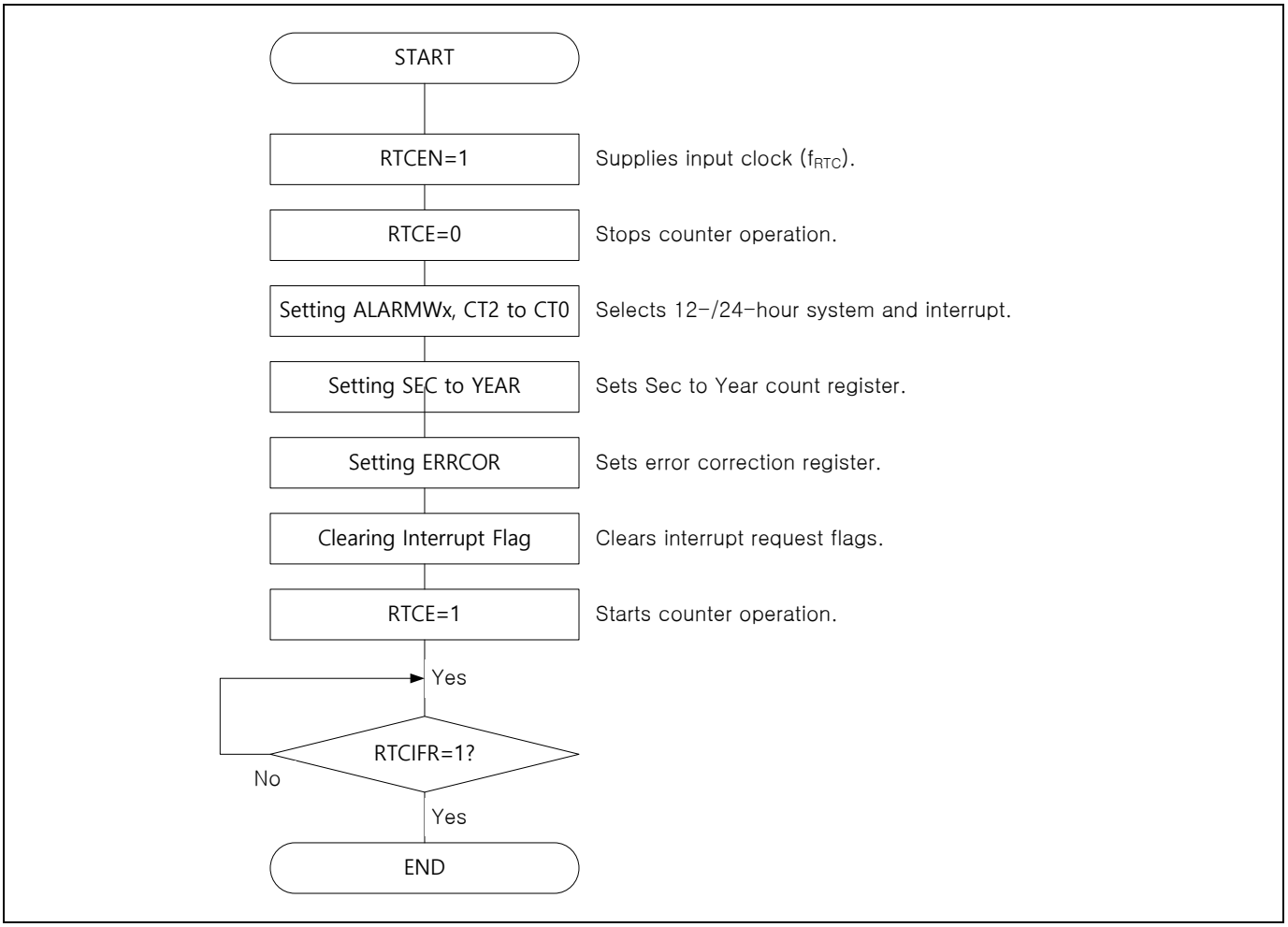


Figure 11.61 Procedure for Starting Operation of RTC

1. First set the RTCEN bit to 1, while oscillation of the count clock (SUB-clock) is stable.
2. Set up the ERRCOR register only if the error must be corrected. For details about how to calculate the correction value, see error correction register (ERRCOR) description.

11.17.4 Shifting to Power-Down mode after starting operation

Perform one of the following processing when shifting to power down mode immediately after setting the RTCE bit to 1. However, after setting the RTCE bit to 1, this processing is not required when shifting to power down mode after the RTC interrupt has occurred.

1. Shifting to HALT/STOP mode when at least two cycles of the count clock (fRTC) have elapsed after setting the RTCE bit to 1 (see Example 1).
2. Checking by polling the RWST bit to become 1, after setting the RTCE bit to 1 and then setting the RWAIT bit to 1. Afterward, setting the RWAIT bit to 0 and shifting to HALT/STOP mode after checking again by polling that the RWST bit has become 0 (see Example 2).

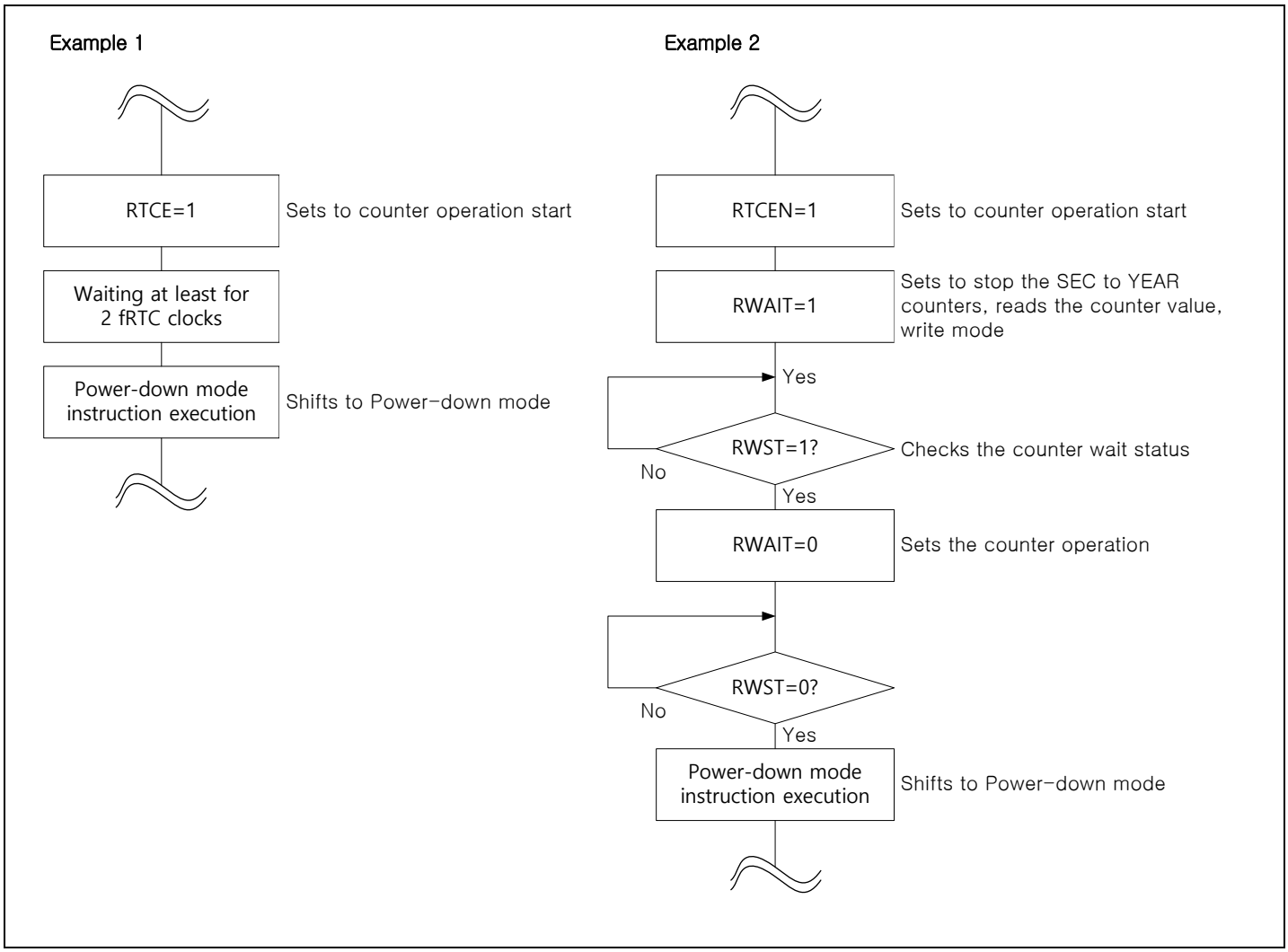


Figure 11.62 Procedure for Starting Operation of RTC

11.17.5 Reading/writing RTC

Read or write the counter after setting 1 to RWAIT first.

Set RWAIT to 0 after completion of reading or writing the counter.

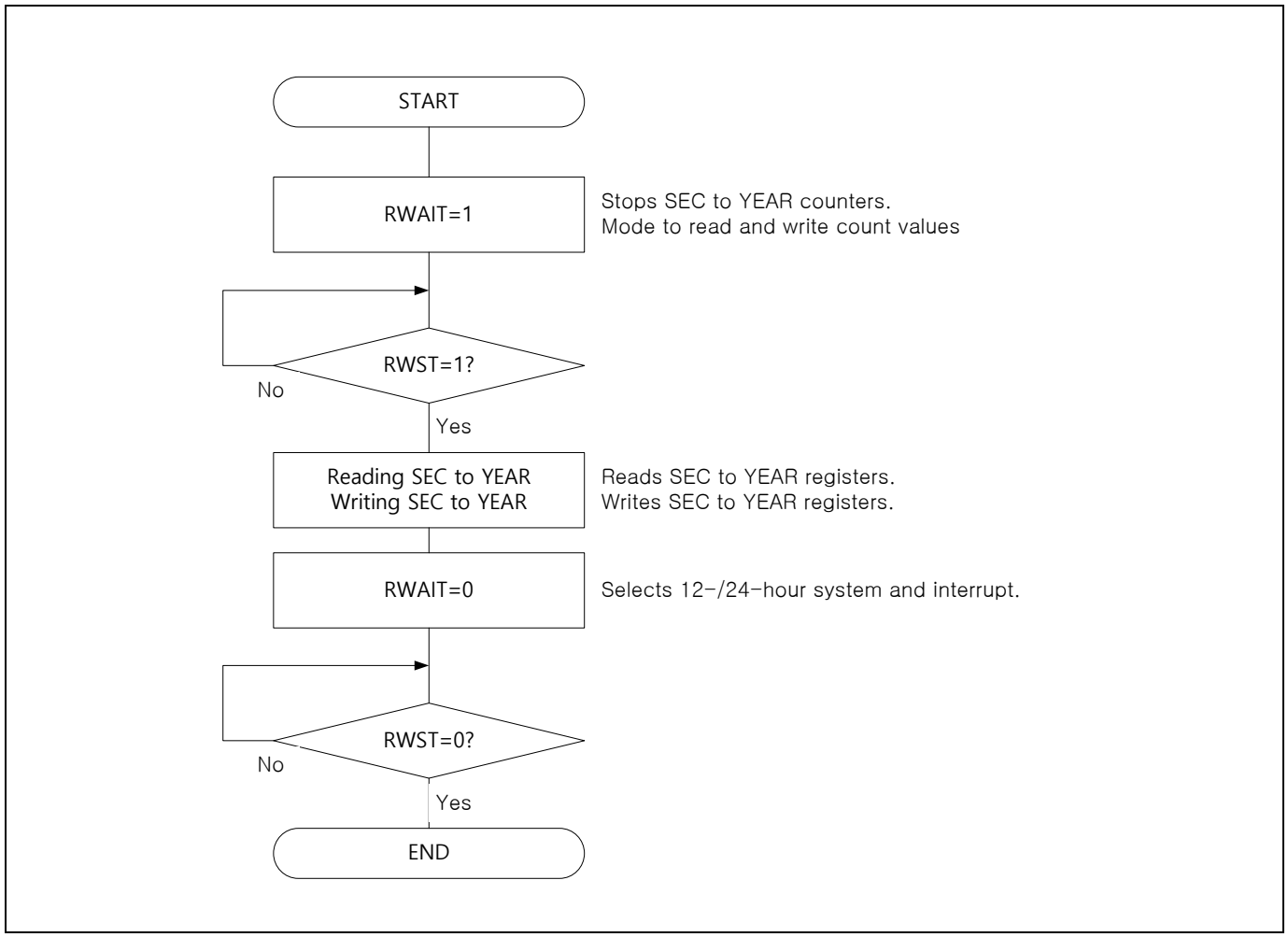


Figure 11.63 Procedure for Reading RTC

NOTE)

1. Be sure to confirm that RWST = 0 before setting STOP mode.
2. Complete the series of process of setting the RWAIT bit to 1 to clearing the RWAIT bit to 0 within 1 second.
3. The second count register (SEC), minute count register (MIN), hour count register (HOUR), week count register (WEEK), day count register (DAY), month count register (MONTH), and year count register (YEAR) may be read/written in any sequence.
4. All the registers do not have to read and only some registers may be read or written.
5. When changing the values of the SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR register while the counter operates (RTCE = 1), rewrite the values of the MIN register after disabling interrupt servicing INTRTC by using the interrupt mask flag register. Furthermore, clear the WAFG, RIFG and RTCIF flags after rewriting the MIN register. <<

11.17.6 Setting alarm of RTC

Set time of alarm after setting 0 to WALIE (alarm operation invalid.) first.

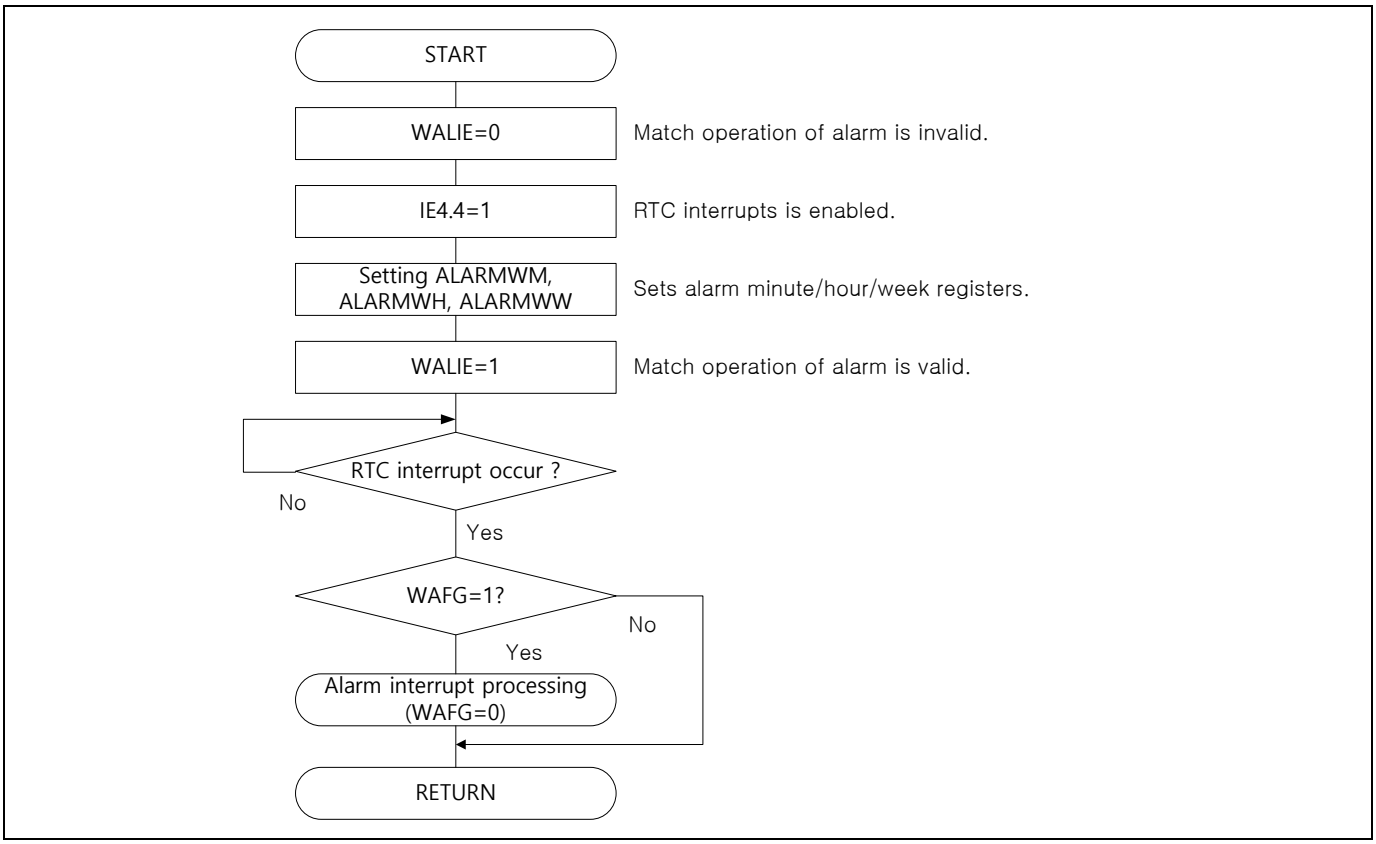


Figure 11.64 Procedure for Reading RTC

The alarm week register (ALARMWW), alarm hour register (ALARMWH), and alarm week register (ALARMWW) may be written in any sequence.

Fixed-cycle interrupts and alarm match interrupts use the same interrupt source (RTC interrupt). When using these two types of interrupts at the same time, which interrupt occurred can be judged by checking the fixed-cycle interrupt status flag (RIFG) and the alarm detection status flag (WAFG) upon RTC interrupt occurrence.

Time of Alarm	Day (ALARMWW)							12-Hour Display		24-Hour Display	
	Sunday WW 0	Monday WW 1	Tuesday WW 2	Wednesday WW 3	Thursday WW 4	Friday WW 5	Saturday WW 6	Hour (ALARMWH)	Minute (ALARMWM)	Hour (ALARMWH)	Minute (ALARMWM)
Every day, 0:00 a.m.	1	1	1	1	1	1	1	12	00	00	00
Every day, 1:30 a.m.	1	1	1	1	1	1	1	01	30	01	30
Every day, 11:59 a.m	1	1	1	1	1	1	1	11	59	11	59
Monday through Friday, 0:00 p.m	0	1	1	1	1	1	0	32	00	12	00
Sunday, 1:30 p.m	1	0	0	0	0	0	0	21	30	13	30
Monday, Wednesday, 11:59 p.m	0	1	0	1	0	0	0	31	59	23	59

Table 11.24 Example of setting the alarm

11.17.7 Error Correction of RTC

The watch can be corrected with high accuracy when it is slow or fast, by setting a value to the error correction register. The correction value is 2, 4, 6, 8, … 120, 122, 124 or -2, -4, -6, -8, … -120, -122, -124.
The target frequency is the frequency resulting after correction performed by using the error correction register.

<Example of calculating the correction value>
The correction value used when correcting the count value of the internal counter (16-bit) is calculated by using the following expression. Set the DEV bit to 0 when the correction range is 63.1 ppm or less, or 63.1 ppm or more.

- When DEV = 0 : Correction value = Number of correction counts in 1 minute ÷ 3
  = (f_{RTC} ÷ 32768 - 1) x (32768 x 60 ÷ 3)
- When DEV = 1 : Correction value = Number of correction counts in 1 minute
  = (f_{RTC} ÷ 32768 - 1) x (32768 x 60)

The correction value is the calculated by using bits 6 to 0 of the error correction register (ERRCOR).
(When F6 = 0) Correction value = {(F5, F4, F3, F2, F1, F0) - 1} x 2
(When F6 = 1) Correction value = {(/F5, /F4, /F3, /F2, /F1, /F0) + 1} x 2

When (F6, F5, F4, F3, F2, F1, F0) is (*, 0, 0, 0, 0, 0, *), watch error correction is not performed.
"*" is 0 or 1. /F5 to /F0 are bit-inverted values (000011 when 111100).

<Correction example>

Example of correcting from 32772.3 Hz to 32768 Hz (32772.3 Hz – 131.2 ppm)

1. Measuring the oscillation frequency

   The f_{RTC} is measured by outputting to RTC1HZ pin when the ERRCOR register is set to its initial value (00H).

2. Calculating the correction value

   Assume the target frequency to be 32768 Hz (32772.3Hz–131.2ppm) and DEV to be 0, because the correctable range of –131.2 ppm is –63.1 ppm or lower.

   Correction value = $(f_{RTC} \div 32768 - 1) \times 32768 \times 60 \div 3$

   $= (32772.3 \div 32768 - 1) \times 32768 \times 60 \div 3 = 86$

3. Calculating the values to be set to (F6 to F0)

   If the correction value is 0 or larger (when slowing), assume F6 to be 0.

   Calculate (F5, F4, F3, F2, F1, F0) from the correction value.

   $\{ (F5, F4, F3, F2, F1, F0) - 1\} \times 2 = 86$

   $(F5, F4, F3, F2, F1, F0) = (1, 0, 1, 1, 0, 0)$

   Consequently, when correcting from 32772.3 Hz to 32768 Hz (32772.3 Hz – 131.2 ppm), setting the error correction register such that DEV is 0 and the correction value is 86 (bits 6 to 0 of the ERROR register : 0101100) results in 32768 Hz (0 ppm).

11.17.8 Register Map

Name	Address	Direction	Default	Description
RTCC0	8FB1H	R/W	00H	RTC Control Register 0
RTCC1	8FB2H	R/W	00H	RTC Control Register 1
ERRCOR	8FB3H	R/W	00H	Watch Error Correction Register
SEC	8FB4H	R/W	00H	Second Count Register
MIN	8FB5H	R/W	00H	Minute Count Register
HOUR	8FB6H	R/W	12H	Hour Count Register
DAY	8FB7H	R/W	01H	Day Count Register
WEEK	8FB8H	R/W	00H	Week Count Register
MONTH	8FB9H	R/W	01H	Month Count Register
YEAR	8FBAH	R/W	00H	Year Count Register
ALARMWM	8FBBH	R/W	00H	Alarm Minute Register
ALARMWH	8FBCH	R/W	12H	Alarm Hour Register
ALARMWW	8FBDH	R/W	00H	Alarm Week Register

Table 11.25 RTC Register Map

11.17.9 RTC Register description

RTCC0 (RTC Control Register 0) : 8FB1H

7	6	5	4	3	2	1	0
RTCE	-	RTCO1HZEN	-	AMPM	CT2	CT1	CT0
R/W	-	R/W	-	R/W	R/W	R/W	R/W

Initial value : 00H

RTCE RTC operation control. (RWST, RWAIT)
- 0 Stops counter operation.

RTCO1HZEN RTCO1HZ pin output control.
- 0 Disables output of the RTCO1HZ pin (1Hz).

AMPM Selection of 1/-24-hour system
- 0 12-hour system (a.m. and p.m. are displayed.)
- 1 24-hour system

CT Constant-period interrupt (INTRTC) selection

CT2	CT1	CT0	description
0	0	0	Does not use constant-period interrupt function.
0	0	1	Once per 0.5 s (synchronized with second count up)
0	1	0	Once per 1 s (same time as second count up)
0	1	1	Once per 1 m (second 00 of every minute)
1	0	0	Once per 1 hour (minute 00 and second 00 of every hour)
1	0	1	Once per 1 day (hour 00, minute 00, and second 00 of every day)
1	1	0	Once per 1 month (Day 1, hour 00 a.m., minute 00, and second 00 of every month)
1	1	1	reserved

NOTE)

1. AMPM can be changed only in WAIT mode. If the AMPM bit value is changed, the values of the hour count register (HOUR) change according to the specified time system in Table 11.26.

RTCC1 (RTC Control Register 1) : 8FB2H

7	6	5	4	3	2	1	0
RTCEN	SUB_ON_STOP	WALIE	WAFG	RIFG	-	RWST	RWAIT
R/W	R/W	R/W	R/W	R/W	-	R	R/W

Initial value : 00H

RTCEN	Control RTC input clock supply.	
	0	Stop RTC input clock supply. SFR used by the RTC cannot be written. The RTC is in the reset status.
	1	Enable RTC input clock supply. SFR used by the RTC can be read and written.
SUB_ON_STOP	SUB clock enable or disable at stop mode	
	0	SUB clock disable at stop mode
	1	SUB clock enable at stop mode
WALIE	Alarm operation control	
	0	Match operation is invalid.
	1	Match operation is valid..
WAFG	Alarm detection status flag. This flag is cleared when “0” is written to it. Writing “1” to it is invalid.	
	0	Alarm mismatch
	1	Detection of matching of alarm
RIFG	Constant-period interrupt status flag. This flag is cleared when “0” is written to it. Writing “1” to it is invalid.	
	0	Constant-period interrupt is not generated.
	1	Constant-period interrupt is generated.
RWST	Wait-mode status flag of real-time clock	
	0	Counter is operating.
	1	Mode to write counter value
RWAIT	Wait control of real-time clock	
	0	Sets counter operation.
	1	Stops SEC to YEAR counters. Mode to write counter value

NOTE)

2. If RTCEN = 0, writing to a control register of the real-time clock or interval timer is ignored. Even if the registers are read, only default values are read. (SEC to YEAR, ALARMWx, ERRCOR)
3. RWST indicates whether the setting of the RWAIT bit is valid. Before writing the counter value, confirm that the value of this flag is 1.
4. RWAIT controls the operation of the counter. Be sure to write “1” to it to write the counter value. As the sub-count register is continuing to run, complete writing within one second and turn back to 0. When the sub-count register overflowed while RWAIT = 1, it keeps the event of overflow until RWAIT = 0, then counts up. However, when it wrote a value to second count register, it will not keep the overflow event.
5. Remark Fixed-cycle interrupts and alarm match interrupts use the same interrupt source (RTCIFR). When using these two types of interrupts at the same time, which interrupt occurred can be judged by checking the fixed-cycle interrupt status flag (RIFG) and the alarm detection status flag (WAFG) upon INTRTC occurrence

SEC (Second count register) :

The SEC register is an 8-bit register that takes a value of 0 to 59 (decimal) and indicates the count value of seconds. It counts up when the sub-count register overflows.

When data is written to this register, it is written to a buffer and then to the counter when RWST bit is changed to '0'. Set a decimal value of 00 to 59 to this register in BCD code. If a value that is not a BCD code is written, it is ignored. If a value outside the range is written, the reset value is set.

7	6	5	4	3	2	1	0
-	SEC6	SEC5	SEC4	SEC3	SEC2	SEC1	SEC0
-	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

MIN (Minute count register) :

The MIN register is an 8-bit register that takes a value of 0 to 59 (decimal) and indicates the count value of minutes. It counts up when the second counter overflows.

When data is written to this register, it is written to a buffer and then to the counter when RWST bit is changed to '0'. Even if the second count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 00 to 59 to this register in BCD code. If a value that is not a BCD code is written, it is ignored. If a value outside the range is written, the reset value is set.

7	6	5	4	3	2	1	0
-	MIN6	MIN5	MIN4	MIN3	MIN2	MIN1	MIN0
-	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

HOUR (Hour count register) :

The HOUR register is an 8-bit register that takes a value of 00 to 23 or 01 to 12 and 21 to 32 (decimal) and indicates the count value of hours. It counts up when the minute counter overflows.

When data is written to this register, it is written to a buffer and then to the counter when RWST bit is changed to '0'. Even if the minute count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Specify a decimal value of 00 to 23, 01 to 12, or 21 to 32 by using BCD code according to the time system specified using bit 3 (AMPM) of real-time clock control register 0 (RTCC0).

If the AMPM bit value is changed, the values of the HOUR register and ALARMWH register change according to the specified time system.

If a value that is not a BCD code is written, it is ignored. If a value outside the range is written, the reset value is set.

Reset signal clears this register to 12H.

However, the value of this register is 00H if the AMPM bit (bit 3 of the RTCC0 register) is set to 1 after reset.

7	6	5	4	3	2	1	0
-	-	HOUR5	HOUR4	HOUR3	HOUR2	HOUR1	HOUR0
-	-	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 12H

In 12-hour display, the fifth bit of the HOUR register displays 0 for AM and 1 for PM.

This table below shows the relationship between the setting value of the AMPM bit, the hour count register (HOUR) value, and time.

24-Hour Display (AMPM=1)		12-Hour Display (AMPM=0)	
Time	HOUR Register	Time	00H
0	00H	12 a.m.	12H
1	01H	1 a.m.	01H
2	02H	2 a.m.	02H
3	03H	3 a.m.	03H
4	04H	4 a.m.	04H
5	05H	5 a.m.	05H
6	06H	6 a.m.	06H
7	07H	7 a.m.	07H
8	08H	8 a.m.	08H
9	09H	9 a.m.	09H
10	10H	10 a.m.	10H
11	11H	11 a.m.	11H
12	12H	12 p.m.	32H
13	13H	1 p.m.	21H
14	14H	2 p.m.	22H
15	15H	3 p.m.	23H
16	16H	4 p.m.	24H
17	17H	5 p.m.	25H
18	18H	6 p.m.	26H
19	19H	7 p.m.	27H
20	20H	8 p.m.	28H
21	21H	9 p.m.	29H
22	22H	10 p.m.	30H
23	23H	11 p.m.	31H

Table 11.26 Time system of HOUR

DAY (Day count register) :

The DAY register is an 8-bit register that takes a value of 1 to 31 (decimal) and indicates the count value of days.

It counts up when the hour counter overflows.

This counter counts as follows.

- 01 to 31 (January, March, May, July, August, October, December)
- 01 to 30 (April, June, September, November)
- 01 to 29 (February, leap year)
- 01 to 28 (February, normal year)

When data is written to this register, it is written to a buffer and then to the counter when RWST bit is changed to '0'. Even if the hour count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 01 to 31 to this register in BCD code. If a value that is not a BCD code is written, it is ignored. If a value outside the range is written, the reset value is set.

7	6	5	4	3	2	1	0
-	-	DAY5	DAY4	DAY3	DAY2	DAY1	DAY0
-	-	RW	RW	RW	RW	RW	RW

Initial value : 01H

WEEK (Week count register) :

The WEEK register is an 8-bit register that takes a value of 0 to 6 (decimal) and indicates the count value of weekdays. It counts up in synchronization with the day counter.

When data is written to this register, it is written to a buffer and then to the counter when RWST bit is changed to '0'. Set a decimal value of 00 to 06 to this register in BCD code. If a value that is not a BCD code is written, it is ignored. If a value outside the range is written, the reset value is set.

7	6	5	4	3	2	1	0
-	-	-	-	-	WEEK2	WEEK1	WEEK0
-	-	-	-	-	R/W	R/W	R/W

Initial value : 00H

The value corresponding to the month count register (MONTH) or the day count register (DAY) is not stored in the week count register (WEEK) automatically. After reset release, set the week count register as follow.

Day	
Sunday	00H
Monday	01H
Tuesday	02H
Wednesday	03H
Thursday	04H
Friday	05H
Saturday	06H

Table 11.27 Week count register setting

MONTH (Month count register) :

The MONTH register is an 8-bit register that takes a value of 1 to 12 (decimal) and indicates the count value of months. It counts up when the day counter overflows.

When data is written to this register, it is written to a buffer and then to the counter when RWST bit is changed to '0'. Even if the day count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 01 to 12 to this register in BCD code. If a value that is not a BCD code is written, it is ignored. If a value outside the range is written, the reset value is set.

7	6	5	4	3	2	1	0
-	-	-	MONTH4	MONTH3	MONTH2	MONTH1	MONTH0
-	-	-	R/W	R/W	R/W	R/W	R/W

Initial value : 01H

YEAR (Year count register) :

The YEAR register is an 8-bit register that takes a value of 0 to 99 (decimal) and indicates the count value of years.
It counts up when the month count register overflows.
Values 00, 04, 08, …, 92, and 96 indicate a leap year.
When data is written to this register, it is written to a buffer and then to the counter when RWST bit is changed to '0'. Even if the MONTH register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 00 to 99 to this register in BCD code. If a value that is not a BCD code is written, it is ignored.

7	6	5	4	3	2	1	0
YEAR7	YEAR6	YEAR5	YEAR4	YEAR3	YEAR2	YEAR1	YEAR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

ERRCOR (Watch error correction register) :

This register is used to correct the watch with high accuracy when it is slow or fast by changing the value that overflows from the sub-count register to the second count register (SEC) (reference value: 7FFFH).

7	6	5	4	3	2	1	0
DEV	F6	F5	F4	F3	F2	F1	F0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

DEV Setting of watch error correction timing
- 0 Corrects watch error when the second digits are at 00, 20, or 40 (every 20 seconds).
- 1 Corrects watch error only when the second digits are at 00 (every 60 seconds).

F6~F0 Setting of watch error correction value
- F6=0 Decreases by {(F5, F4, F3, F2, F1, F0) – 1} × 2.
- F6=1 Increases by {(/F5, /F4, /F3, /F2, /F1, /F0) + 1} × 2.

When (F6, F5, F4, F3, F2, F1, F0) = (*, 0, 0, 0, 0, 0, *), the watch error is not corrected.
* is 0 or 1. /F5 to /F0 are the inverted values of the corresponding bits (000011 when 111100).
Range of correction value:
(when F6 = 0) 2, 4, 6, 8, … , 120, 122, 124
(when F6 = 1) –2, –4, –6, –8, … , –120, –122, –124

The range of value that can be corrected by using the watch error correction register (ERRCOR) is shown below.

	DEV = 0 (correction every 20 seconds)	DEV = 1 (correction every 60 seconds)
Correctable range	−189.2 ppm to 189.2 ppm	-63.1 ppm to 63.1 ppm
Maximum excludes quantization error	±1.53 ppm	±0.51 ppm
Minimum resolution	±3.05 ppm	±1.02 ppm

Table 11.28 The range of correction

If a correctable range is –63.1 ppm or lower and 63.1 ppm or higher, set 0 to DEV.

ALARMWM (Alarm minute register) :

This register is used to set minutes of alarm. Set a decimal value of 00 to 59 to this register in BCD code. If a value outside the range is set, the alarm is not detected.

7	6	5	4	3	2	1	0
-	WM6	WM5	WM4	WM3	WM2	WM1	WM0
-	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

ALARMWH (Alarm hour register) :

This register is used to set hours of alarm. Reset signal generation clears this register to 12H.
However, the value of this register is 00H if the AMPM bit (bit 3 of the RTCC0 register) is set to 1 after reset.
Set a decimal value of 00 to 23, 01 to 12, or 21 to 32 to this register in BCD code. If a value outside the range is set, the alarm is not detected.
The values of the ALARMWH register change according to the specified time system when AMPM bit is changing.

7	6	5	4	3	2	1	0
-	-	WH5	WH4	WH3	WH2	WH1	WH0
-	-	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 12H

ALARMWW (Alarm week register) :

This register is used to set date of alarm.

7	6	5	4	3	2	1	0
-	WW6	WW5	WW4	WW3	WW2	WW1	WW0
-	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

11.18 BISC

11.18.1 Overview

The A97C450 has built in self calibration circuit for internal 16MHz RC-OSC. If the BISC counter is less than the reference counter the trim value sets to slow value, if the BISC counter is larger than the reference counter the trim value sets to fast value. Reference counter input clock is sub-OSC. Refer to the following example.

1. Enables reference clock source.
2. Set the BISC counter register value (INTCNTH, INTCNTL).
3. Set the reference counter register value (XTLCNTH, XTLCNTL).
   The overflow period of XTLCNTH,XTLCNTL must be the same as the expected overflow period of INTCNTH,INTCNTL.
4. Enable built in self calibration by set CAL_EN of BISCCON register.

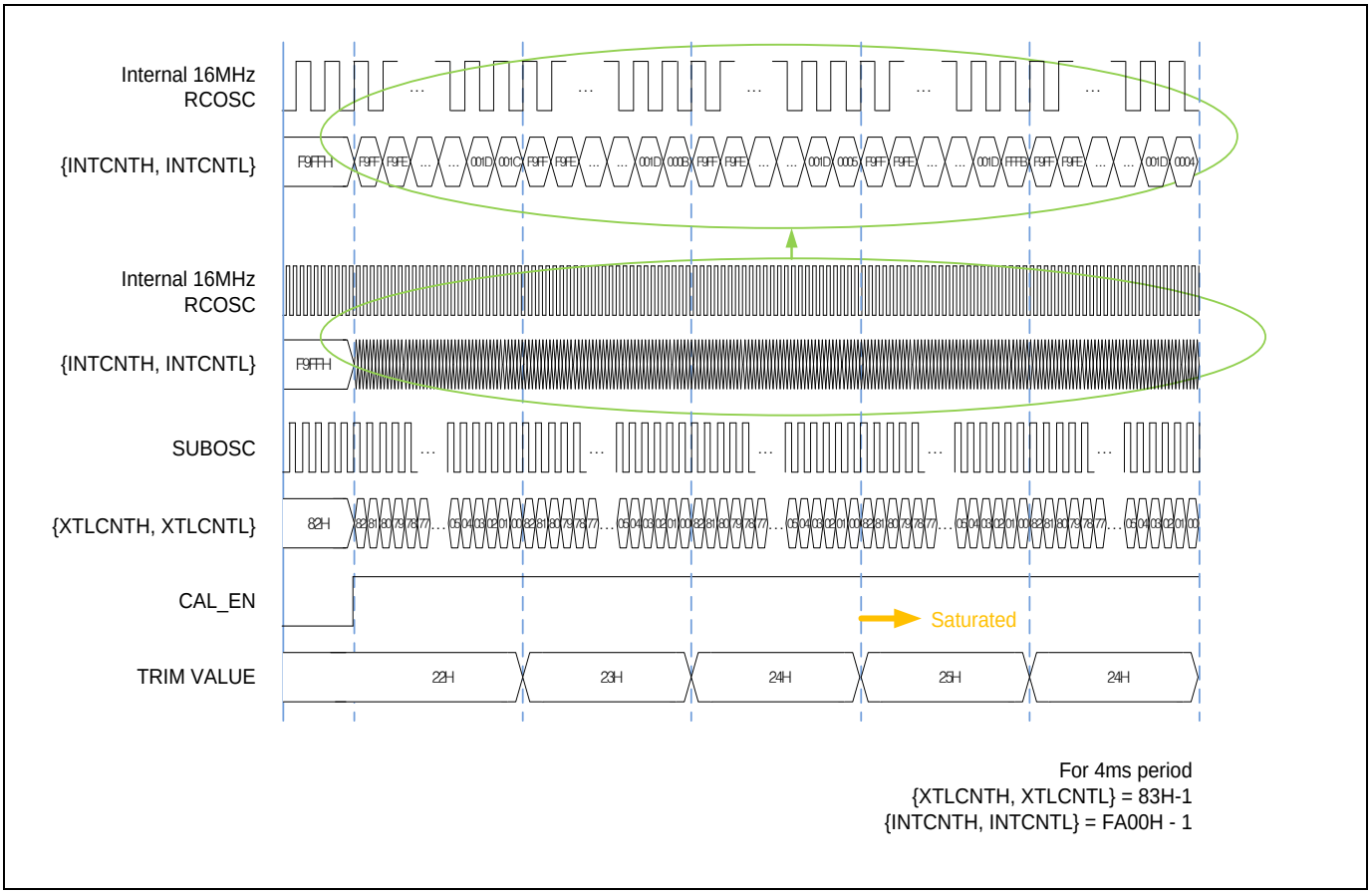


Figure 11.65 BISC waveform

11.18.2 Register Map

Name	Address	Direction	Default	Description
BISCCON	8F28H	R	00H	BISC Control Register
INTCNTH	8F29H	R/W	00H	BISC Counter Register High
INTCNTL	8F2AH	R/W	00H	BISC Counter Register Low
XTLCNTH	8F2BH	R/W	00H	BISC Reference Counter Register High
XTLCNTL	8F2CH	R/W	00H	BISC Reference Counter Register Low

Table 11.29 BISC Register Map

11.18.3 BISC Register description

BISCCON (BISC Control Register) : 8F28H

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

Initial value : 00H

SUB_SEL Select reference clock.
0 Don't'use
1 Select sub-crystal oscillator

CAL_EN Enable built in self calibration of internal RC-OSC
0 BISC disable
1 BISC enable

INTCNTH (BISC Counter Register High) : 8F29H

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

Initial value : 00H

INTCNTL (BISC Counter Register Low) : 8F2AH

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

Initial value : 00H

INTCNT[15:0] BISC counter register, the clock of this register is internal RC-OSC.

XTLCNTH (BISC Reference Counter Register High) : 8F2BH

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

Initial value : 00H

XTLCNTL (BISC Reference Counter Register Low) : 8F2CH

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

Initial value : 00H

XTLCNT[15:0] BISC reference counter register, the clock of this register is sub-OSC.

11.19 CRC

11.19.1 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. When using CRC for CRC_TYPE 5 and CRC_TYPE 7, add 256 to the XRAM address and use it.

(For IXRAM data CRC, use the physical address. : IRAM (0~0xFF), XRAM (0x100~0x10FF)

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 11.30 CRC mode

11.19.2 Block Diagram

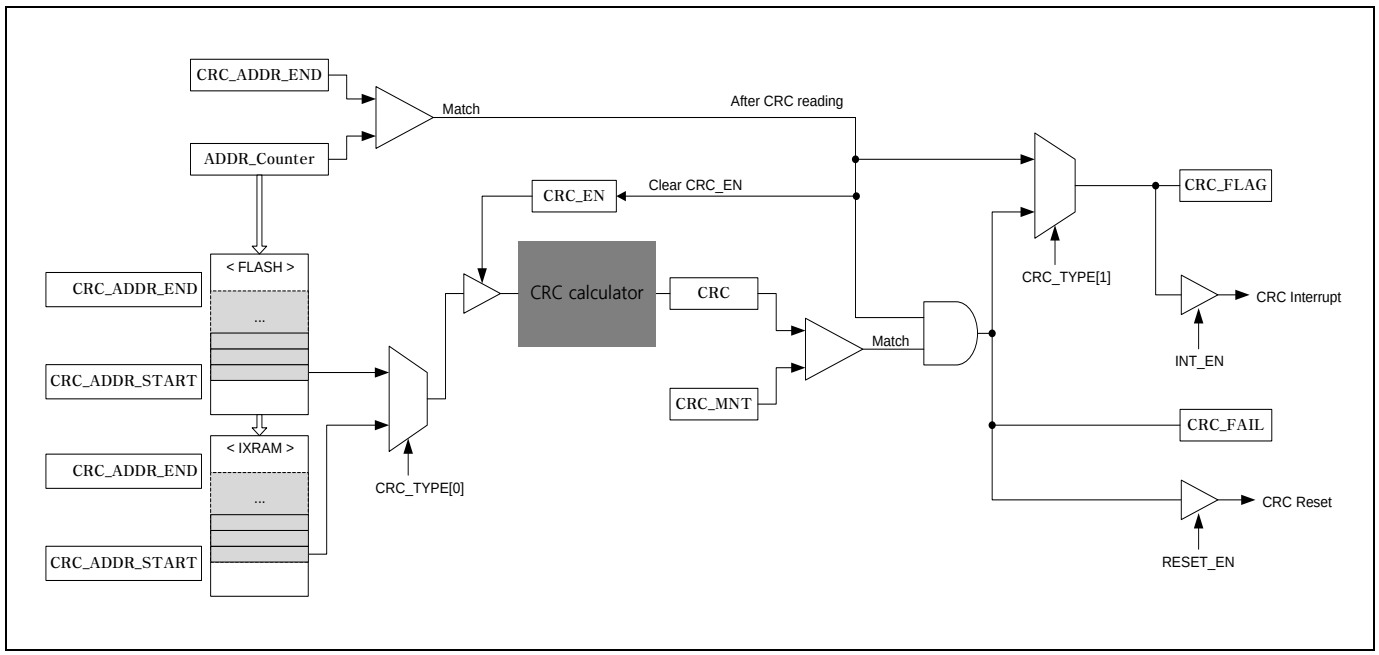


Figure 11.66 CRC block diagram

11.19.3 Register Map

Name	Address	Direction	Default	Description
CRC_CON	8FE0H	R/W	00H	CRC Control Register
CRC_H	8FE2H	R/W	00H	CRC High Register
CRC_L	8FE3H	R/W	00H	CRC Low Register
CRC_MNT_H	8FE4H	R/W	00H	CRC Monitor High Register
CRC_MNT_L	8FE5H	R/W	00H	CRC Monitor Low Register
CRC_ADDR_START_H	8FE9H	R/W	00H	CRC Start Address High Register
CRC_ADDR_START_M	8FEAH	R/W	00H	CRC Start Address Middle Register
CRC_ADDR_START_L	8FEBH	R/W	00H	CRC Start Address Low Register
CRC_ADDR_END_H	8FECH	R/W	00H	CRC End Address High Register
CRC_ADDR_END_M	8FEDH	R/W	00H	CRC End Address Middle Register
CRC_ADDR_END_L	8FEEH	R/W	00H	CRC End Address Low Register

Table 11.31 CRC Register Map

11.19.4 CRC Register description

CRC_CON (CRC Control Register) : 8FE0H

7	6	5	4	3	2	1	0
CRC_FLAG	CRC_INTEN	CRC_RESETEN	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	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

INT_EN Enable CRC interrupt
0 CRC interrupt disable
1 CRC interrupt enable

RESET_EN Enable CRC reset
0 CRC reset disable
1 CRC reset enable

CRC_EN Enable CRC operation, it is cleared automatically after the CRC monitoring 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.
* This value can be changed only when CRC_EN=0.
0xx Not used
100 Specified flash data
101 Specified IXRAM data
110 Specified flash data, validate CRC value
111 Specified IXRAM data, validate CRC value

CRC_H (CRC High Register) : 8FE2H

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) : 8FE3H

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) : 8FE4H

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) : 8FE5H

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_H (CRC Start Address High Register) : 8FE9H

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	CRC_ADDR_START[16]
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

CRC_ADDR_START_M (CRC Start Address Middle Register) : 8FEAH

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) : 8FEBH

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[10]
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_H (CRC End Address High Register) : 8FECH

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	CRC_ADDR_END [16]
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

CRC_ADDR_END_M (CRC End Address Middle Register) : 8FEDH

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) : 8FEEH

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 [10]
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

11.19.5 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 A97C450 has five power-down modes to minimize the power consumption of the device. In power down mode, power consumption is reduced considerably. The device provides five kinds of power saving functions, IDLE, SLEEP1, SLEEP2, STOP1 and STOP2 mode. In five modes, program is stopped.

12.2 Peripheral Operation in IDLE/STOP Mode

Peripheral	IDLE, SLEEP1 Mode	SLEEP2, STOP1, STOP2 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	Operates Continuously
Watch Timer	Operates Continuously	Stop (Only operate in sub clock mode)
TimerP0~1	Operates Continuously	Halted (Only when the External Clock Mode is Enable, Timer operates Normally)
ADC	Operates Continuously	Stop
BUZ	Operates Continuously	Stop
SPI/SCI	Operates Continuously	Only operate with external clock
I2C	Operates Continuously	Stop
Main OSC (1~16MHz)	Oscillation	Stop
Sub OSC (32.768kHz)	Oscillation	Depending on the settings
Internal RC OSC (256kHz)	Oscillation	Depending on the settings
I/O Port	Retain	Retain
Control Register	Retain	Retain
Address Data Bus	Retain	Retain
Release Method	By RESET, all Interrupts	By RESET, Timer Interrupt (External clock), SIO (External clock), External Interrupt, UART by ACK, I2C (slave mode), WT (sub clock),WDT, BIT(only when RCOSC256Khz enabled). H/W CEC, IR

Table 12.1 Peripheral Operation during Power Down Mode.

12.3 IDLE, SLEEP1 mode

The power control register is set to '01h' to enter the IDLE and SLEEP1 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 this mode. If using reset, because the device becomes initialized state, the registers have reset value.

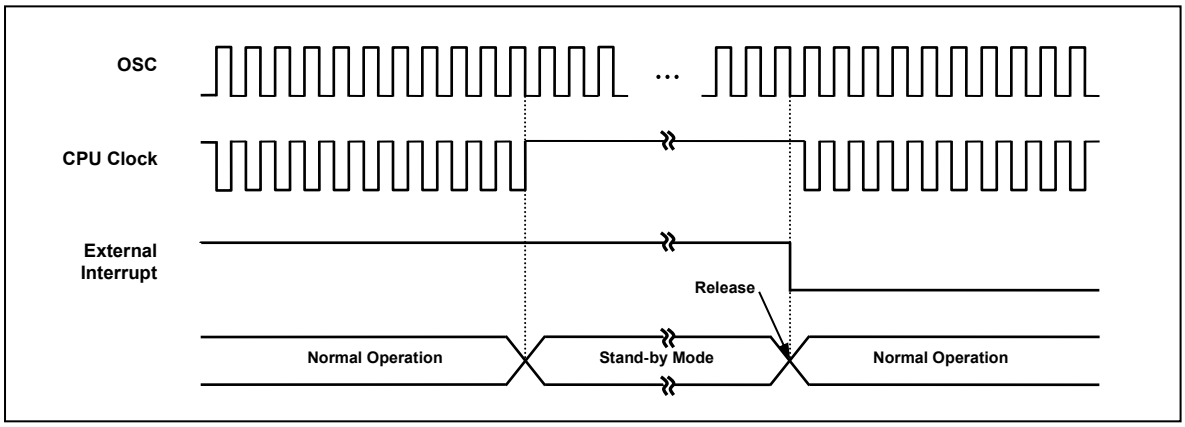


Figure 12.1 IDLE Mode Release Timing by External Interrupt

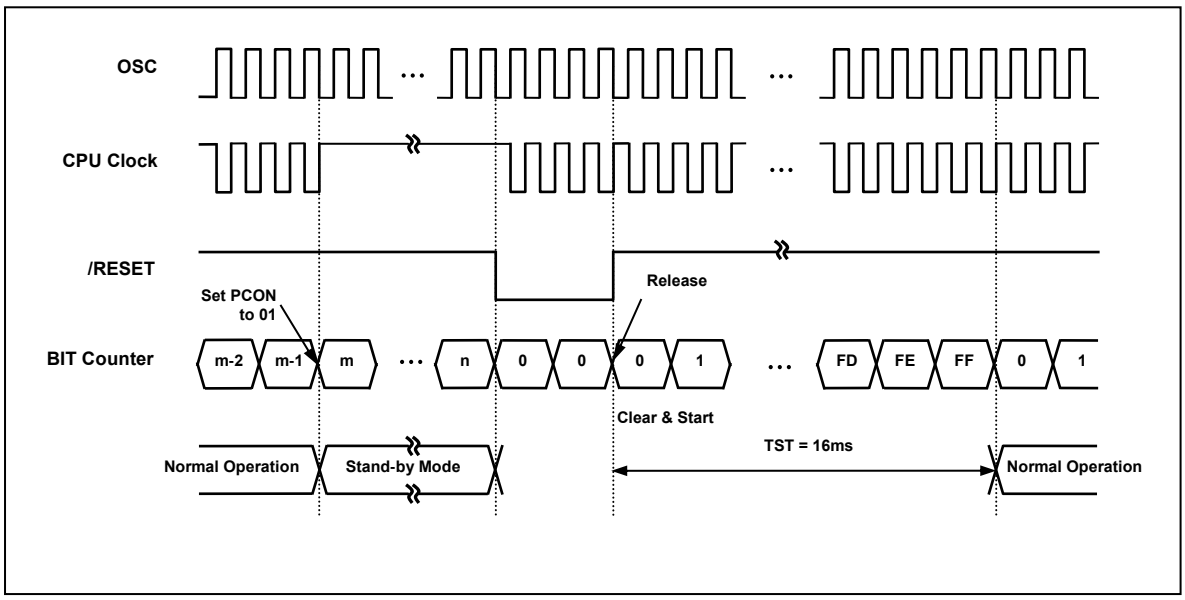


Figure 12.2 IDLE and SLEEP1 Mode Release Timing by RESETB

(Ex) MOV PCON, #0000_0001b; setting of IDLE and SLEEP1 mode :

12.4 STOP mode (STOP1, STOP2, SLEEP2)

In the stop mode, the main oscillator(internal 16MHz-OSC) and internal 256KHz-OSC stop. With the clock frozen, all functions are stopped, but the on-chip RAM and control registers are held.

The sources for exit from STOP mode are 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 12.3 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 prescaler divide ratio to have long enough time (more than 1msec). This guarantees that oscillator has started and stabilized.

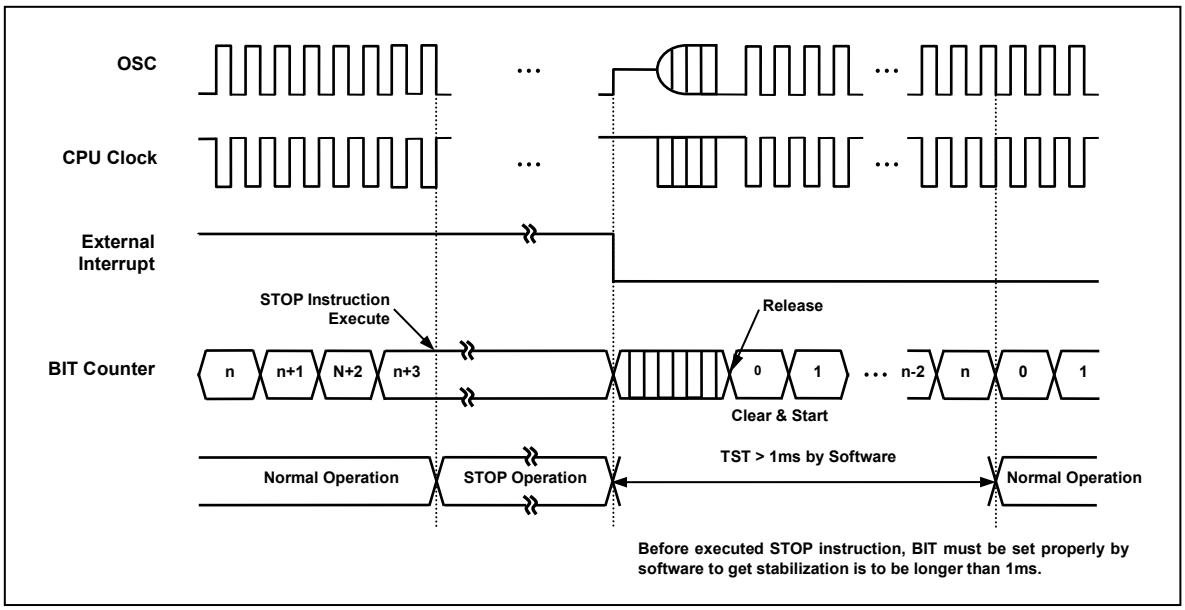


Figure 12.3 STOP Mode Release Timing by External Interrupt

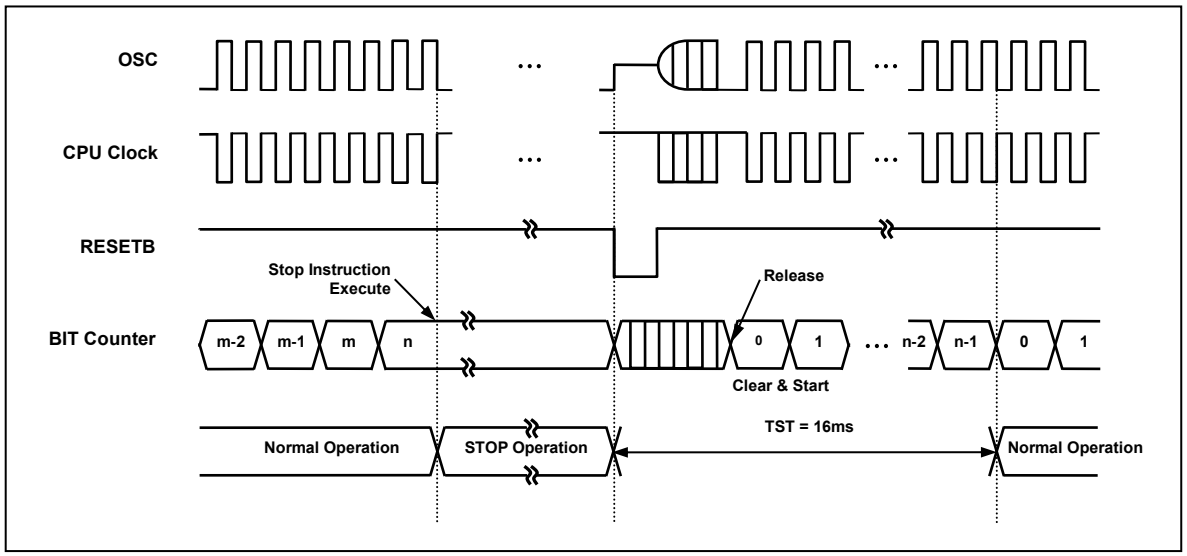


Figure 12.4 Mode Release Timing by RESETB

12.5 Release Operation of Power-Down Mode

After five Power-Down mode is released, the operation begins according to content of related interrupt register just before Power-Down mode start (Figure 12.5). It is released by only interrupt which each interrupt enable flag = `1`, and jump to the relevant interrupt service routine when EA='1' or next instruction.

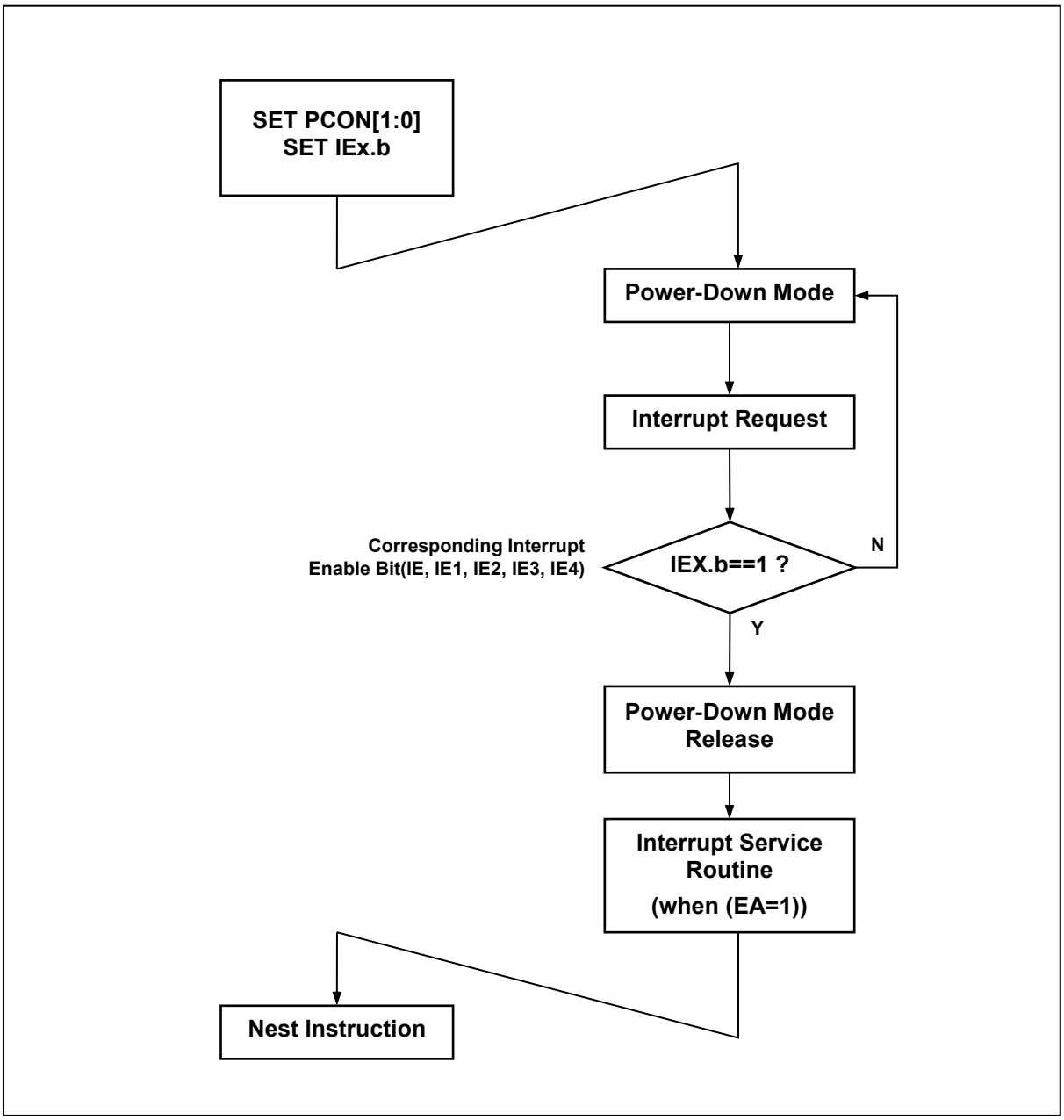


Figure 12.5 STOP1, 2 Mode Release Flow

12.5.1 Register Map

Name	Address	Direction	Default	Description
PCON	87H	R/W	00H	Power Control Register
VDCMOD	B8H	R/W	01H	VDC Mode Control Register

Table 12.2 Power Down Operation Register Map

12.5.2 Power Down Operation Register description

The Power Down Operation Register consists of the Power Control Register (PCON) and VDC Mode control Register (VDCMOD).

12.5.3 Register description for Power Down Operation

VDCMOD (VDC Mode Control Register) : B8H

7	6	5	4	3	2	1	0
-	-	-	-	VDCMOD3	VDCMOD2	VDCMOD1	VDCMOD0
-	-	-	-	R/W	R/W	R/W	R/W

Initial value : 01H

PCON (Power Control Register) : 87H

7	6	5	4	3	2	1	0
-	-	-	-	-	-	BIT1	BIT0
-	-	-	-	-	-	R/W	R/W

Initial value : 00H

VDCMOD, PCON — Select Power Down Mode

VDCMOD	PCON	Description
00H	X	Not Power Down Mode
00H	01H	IDLE
08H	01H	SLEEP1(Low Power VDC Mode)
04H	03H	SLEEP2
02H	03H	STOP1(Low Power VDC Mode)
01H	03H	STOP2(Low Power VDC Mode)

NOTE)

1. To enter IDLE mode and SLEEP1 mode, PCON must be set to ‘01H’.
2. To enter STOP1, STOP2, SLEEP2, PCON must be set to ‘03H’.
   (In this power down mode, PCON register is cleared automatically after interrupt or reset.)
3. The different thing in Power Down mode is VDC operating mode and clock operation of internal 16MHz-OSC at each STOP mode.
4. If WDTRCONSTOP is set to ‘0’, the internal 256kHz RCOSC stops at STOP2, STOP1 and SLEEP2 modes and works again to count up wakeup time when stop wakeup interrupt occurs. If WDTRCONSTOP is set to ‘1’, the internal 256kHz RCOSC does not stop at STOP2, STOP1 and SLEEP2 modes
5. It is not supported except the above 5 modes.
6. Two ‘NOP’ instructions are required after PCON is set
7. By writing ‘A5h’ to AUTHORITY register, it is possible to change VDCMOD.

13 RESET

13.1 Overview

The A97C450 can be reset by some reset sources. The followings are the hardware reset values.

On Chip Hardware	Initial Value
Program Counter (PC)	0000h
Accumulator	00h
Stack Pointer (SP)	07h
Peripheral Clock	On
Control Register	Peripheral Registers refer
Low Voltage Reset	Enable

Table 13.1 Reset state

13.2 Reset source

The A97C450 has five types of reset generation procedures. The followings are the reset sources.

- External RESET pin
- Power ON RESET (POR)
- WDT Overflow Reset (In the case of WDTEN = `1`)
- Low voltage Reset (In the case of LVREN = `1 `)
- Low voltage indicator Reset (In the case of LVILS ≠ `00 `)
- OCD Reset

13.3 Block Diagram

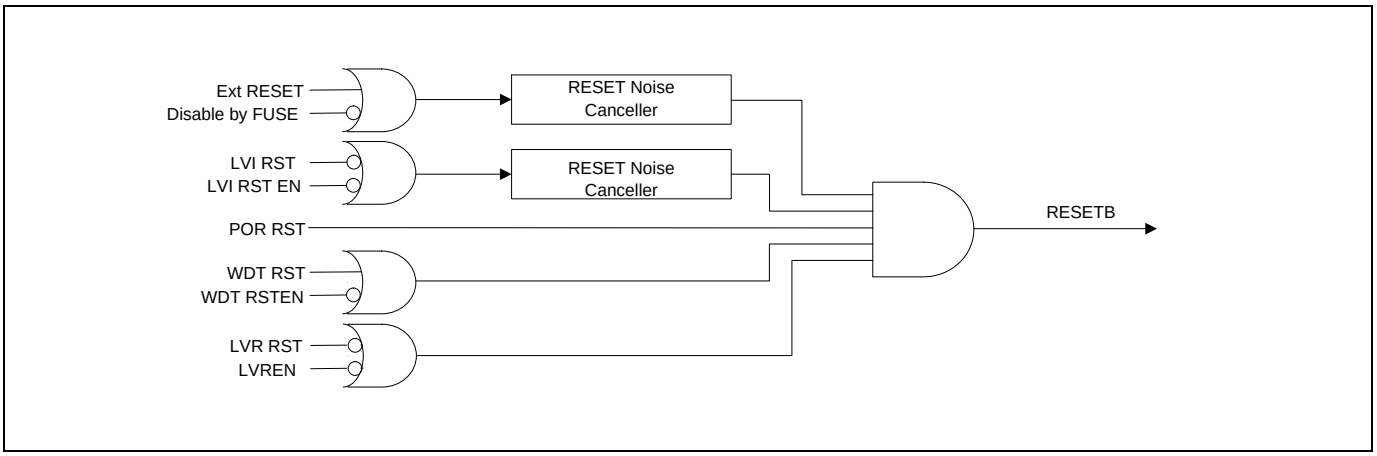


Figure 13.1 RESET Block Diagram

13.4 Power ON RESET

When rising device power, the POR (Power ON Reset) has a function to reset the device. If using POR, it executes the device RESET function instead of the RESET IC or the RESET circuits. And external RESET pin is able to use as Normal I/O pin.

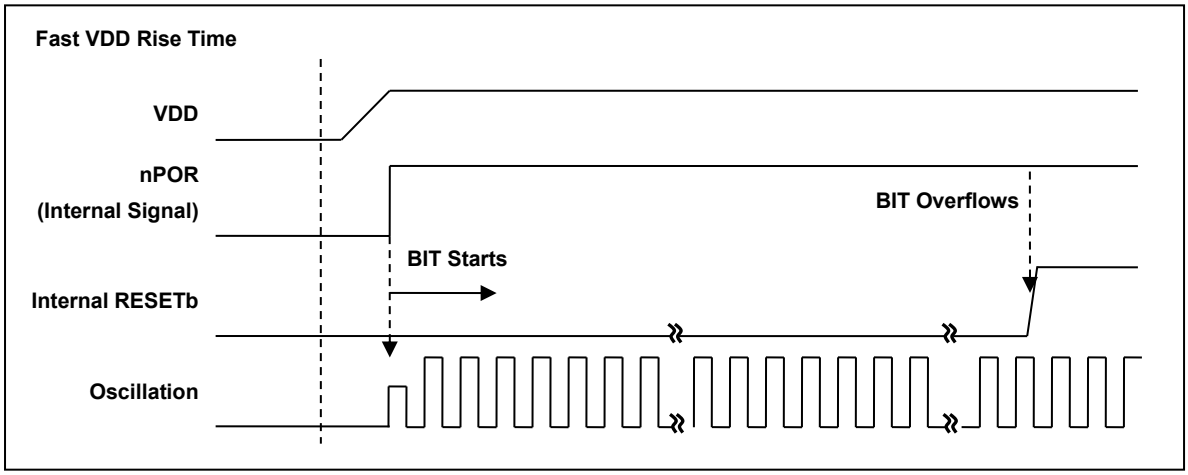


Figure 13.2 Fast VDD rising time

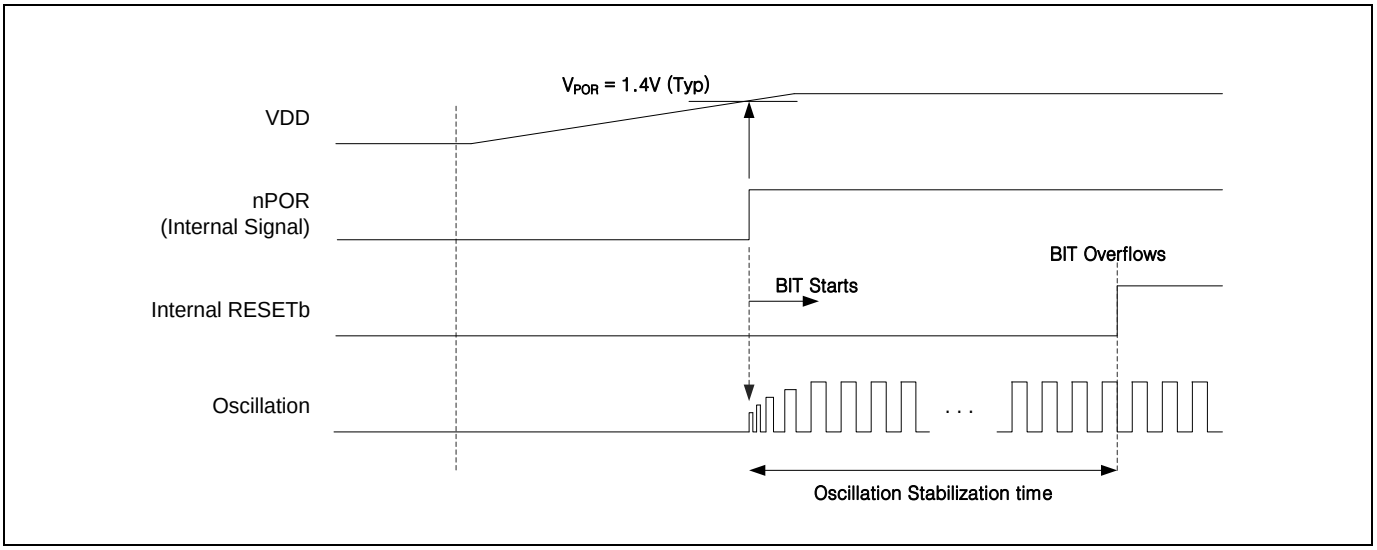


Figure 13.3 Internal RESET Release Timing On Power-Up

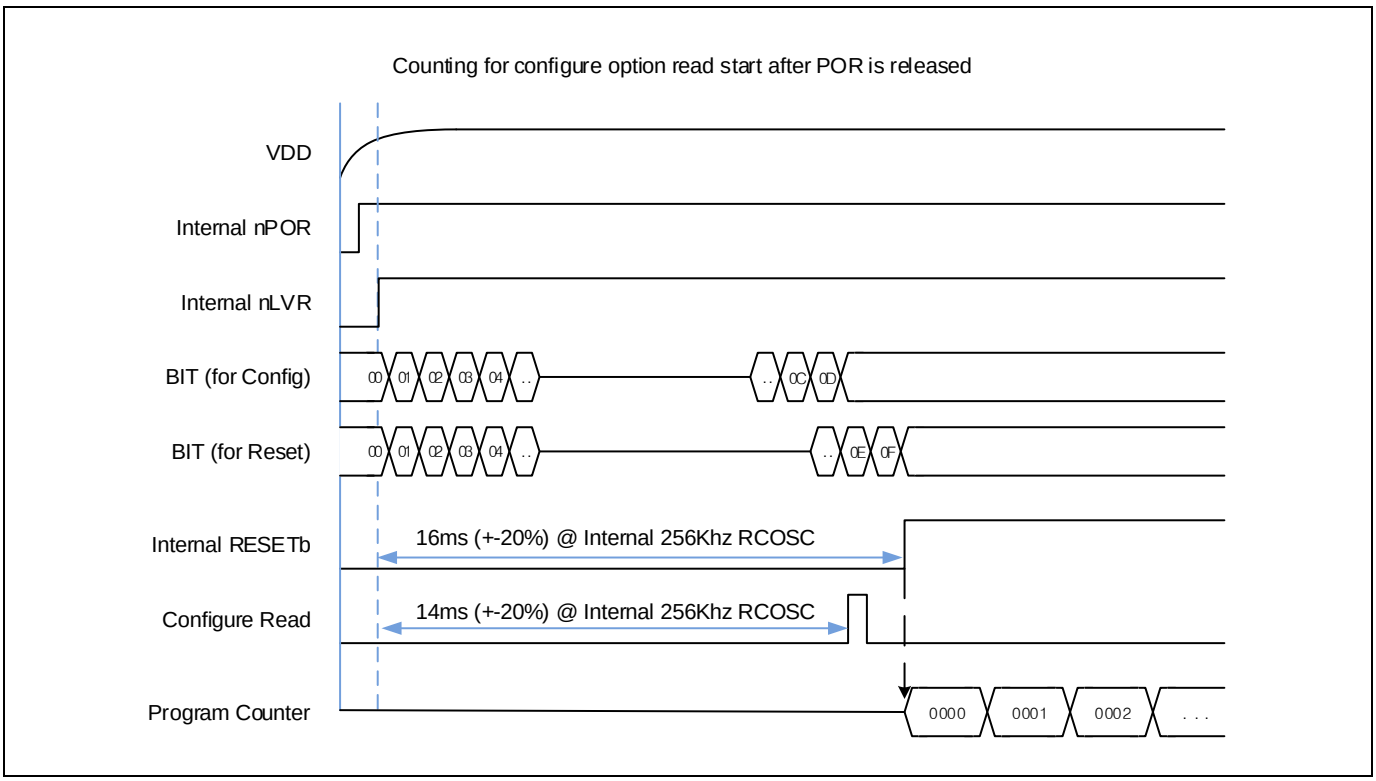


Figure 13.4 Configuration timing when Power-on

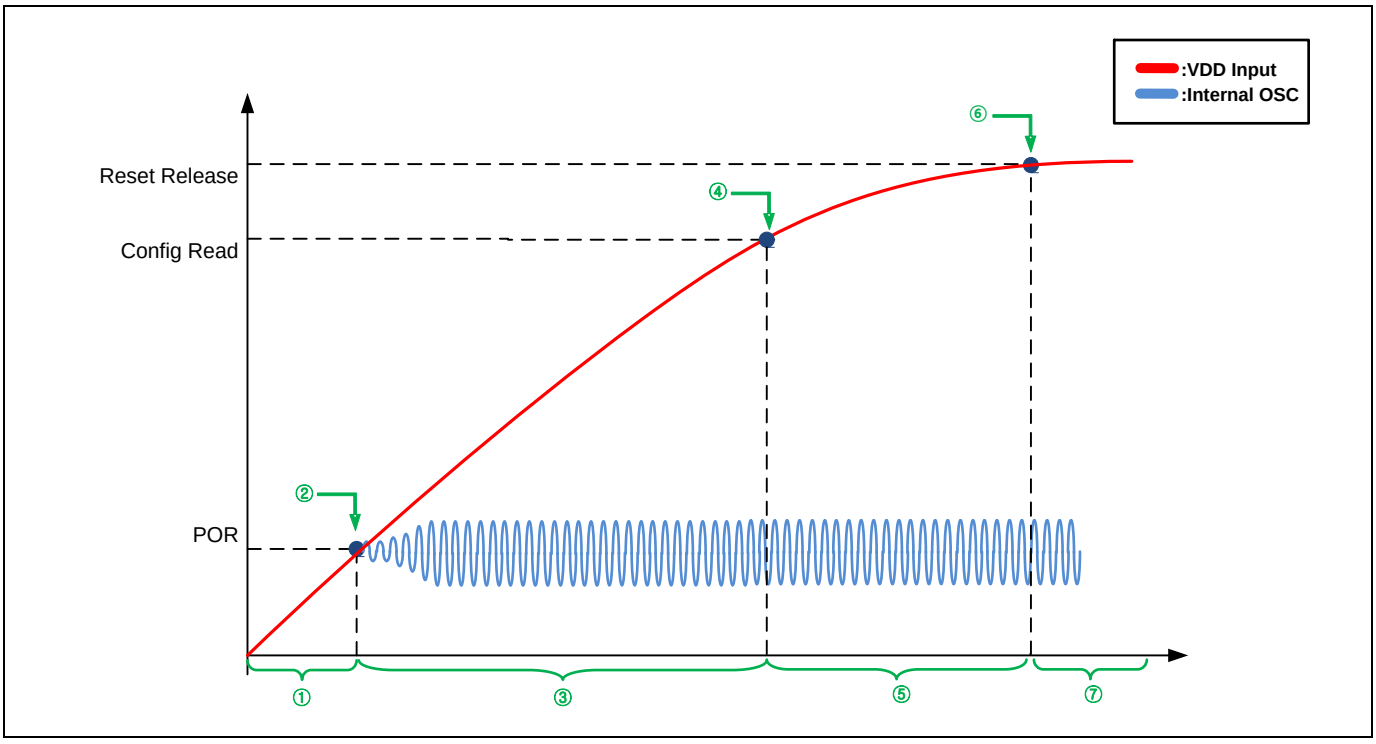


Figure 13.5 Boot Process Waveform

Process	Description	Remarks
①	-No Operation	
②	-1st POR level Detection -Internal OSC (256KHz) ON	-about 1.4V ~ 1.5V
③	- (INT-OSC256KHz/256×Eh Delay section (=14ms) -VDD input voltage must rise over than flash operating voltage for Config read	-Slew Rate >= 0.025V/ms
④	- Config read point	-about 1.5V ~ 1.6V -Config Value is determined by Writing Option
⑤	- Rising section to Reset Release Level	-16ms point after POR or Ext_reset release
⑥	- Reset Release section (BIT overflow) i) after16ms, after External Reset Release (External reset) ii) 16ms point after POR (POR only)	- BIT is used for Peripheral stability
⑦	-Normal operation	

Table 13.2 Boot Process Description

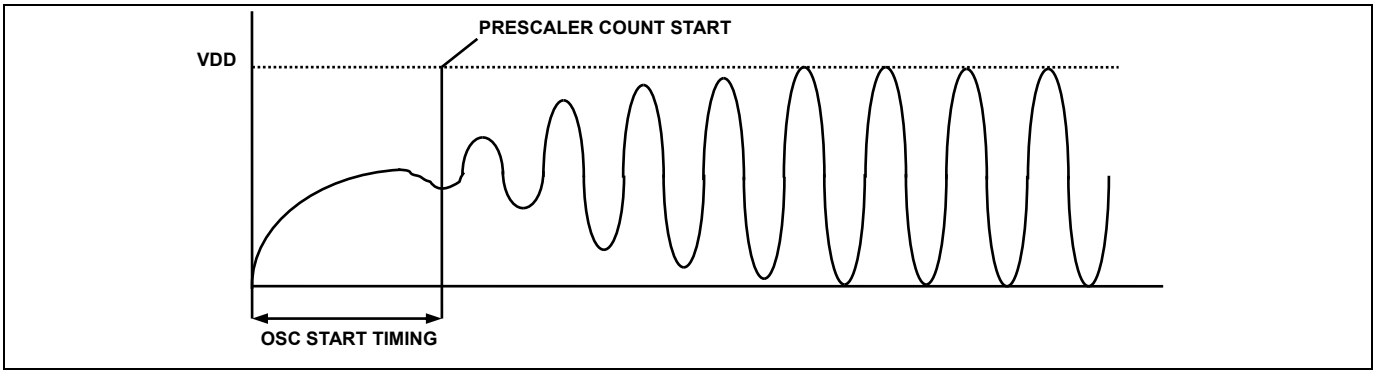


Figure 13.6 Oscillator generating waveform example

13.5 External RESETB Input

The External RESETB is the input to a Schmitt trigger. A reset in accomplished by holding the reset pin low for at least 16us over, within the operating voltage range and oscillation stable, it is applied, and the internal state is initialized. In stop mode, however, the reset is applied as soon as the reset pin goes low. 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.

The Figure 13.7 is the Noise canceller time diagram. It has the Noise cancel value of about 10us (@VDD=5V) to the low input of System Reset.

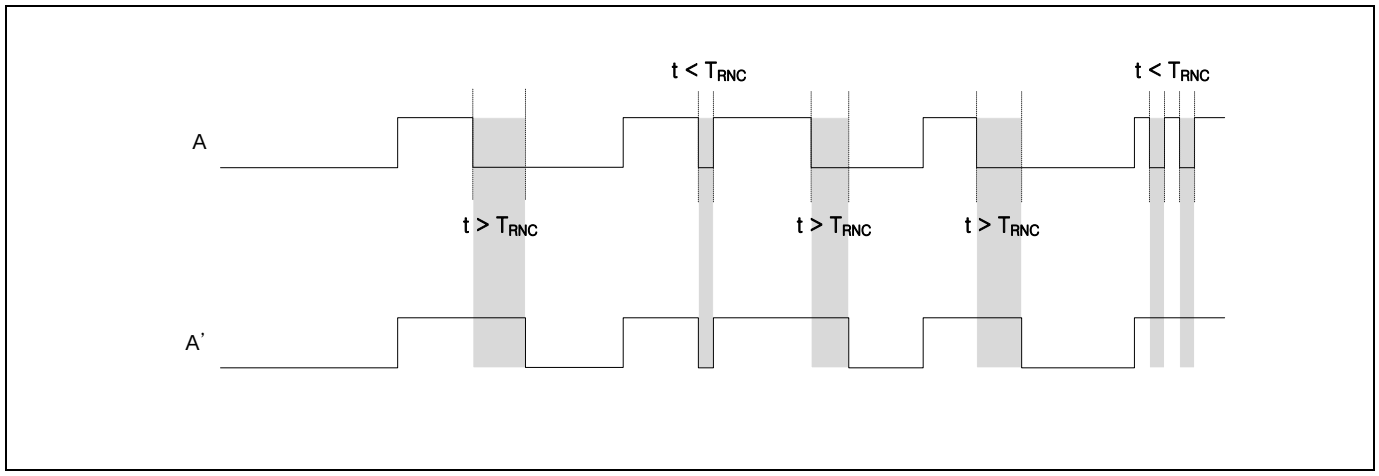


Figure 13.7 Reset noise canceller time diagram

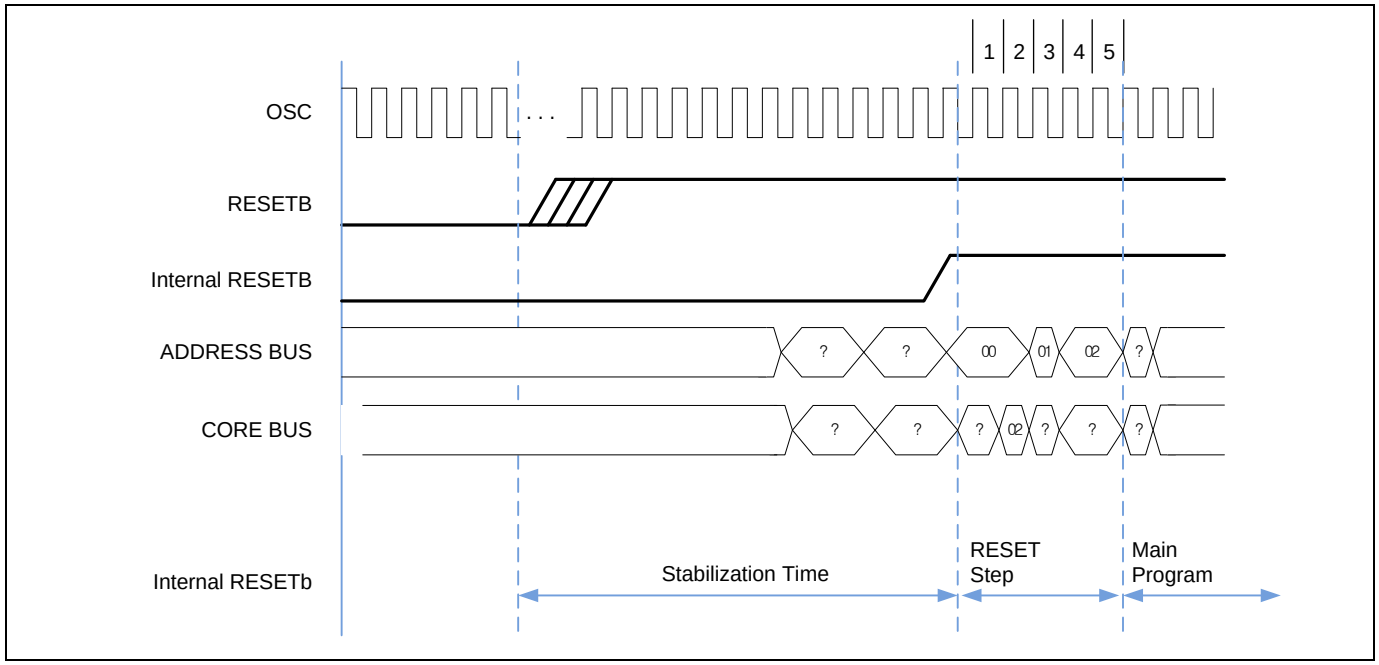


Figure 13.8 Timing Diagram after RESET

13.6 Low Voltage Reset and Low Voltage Indicator Processor

The A97C450 has an On-chip low voltage reset(LVR) and indicator(LVI) circuit for monitoring the VDD level during operation by comparing it to a fixed trigger level. The LVR trigger level is 1.6V and it is enabled by default. The LVR can be disable in the STOP mode by register setting. The trigger level for the LVI can be selected by LVILS[1:0] bit to be 2.4V, 2.8V or 3.5V.

In the STOP mode, this will contribute significantly to the total current consumption. So to minimize the current consumption, the LVR and LVI is set to off by software.

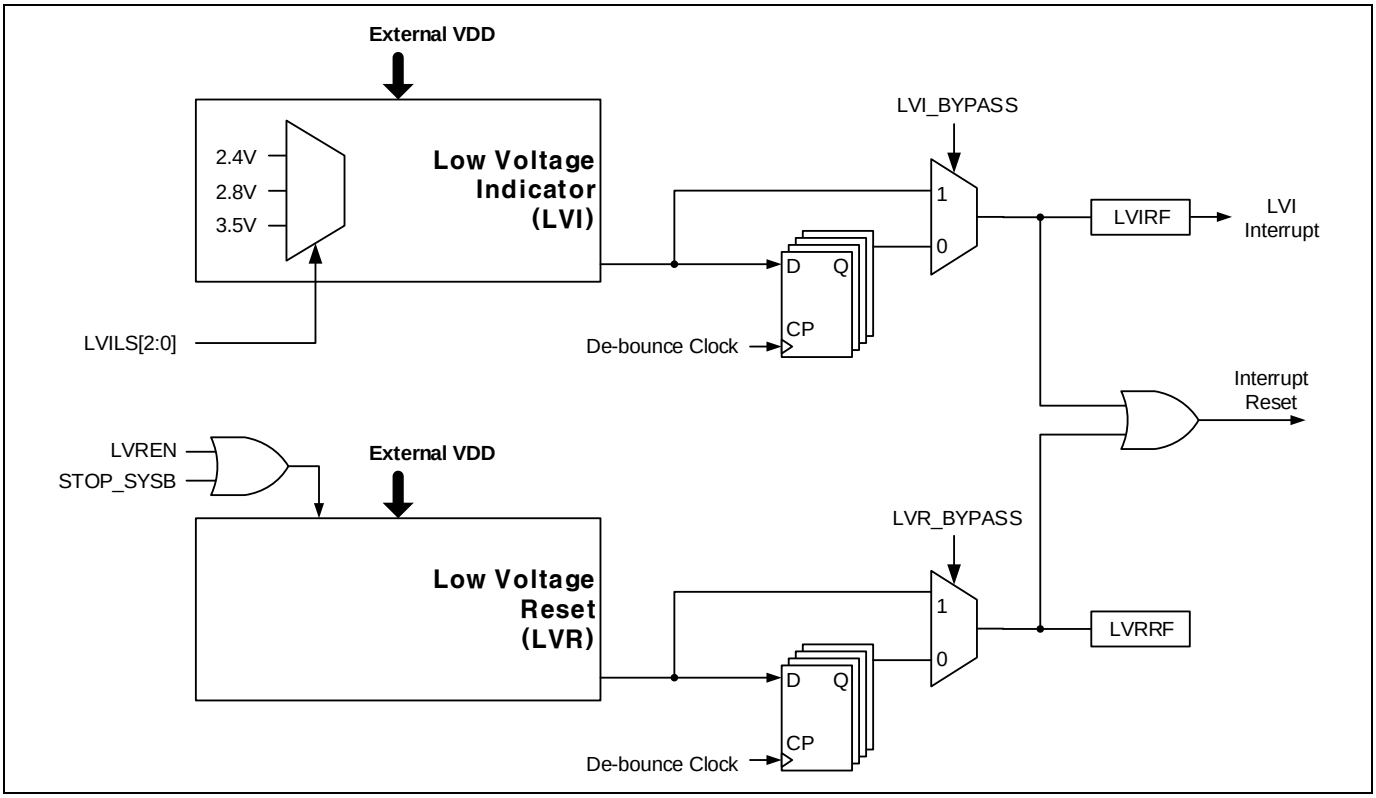


Figure 13.9 Block Diagram of LVR, LVI

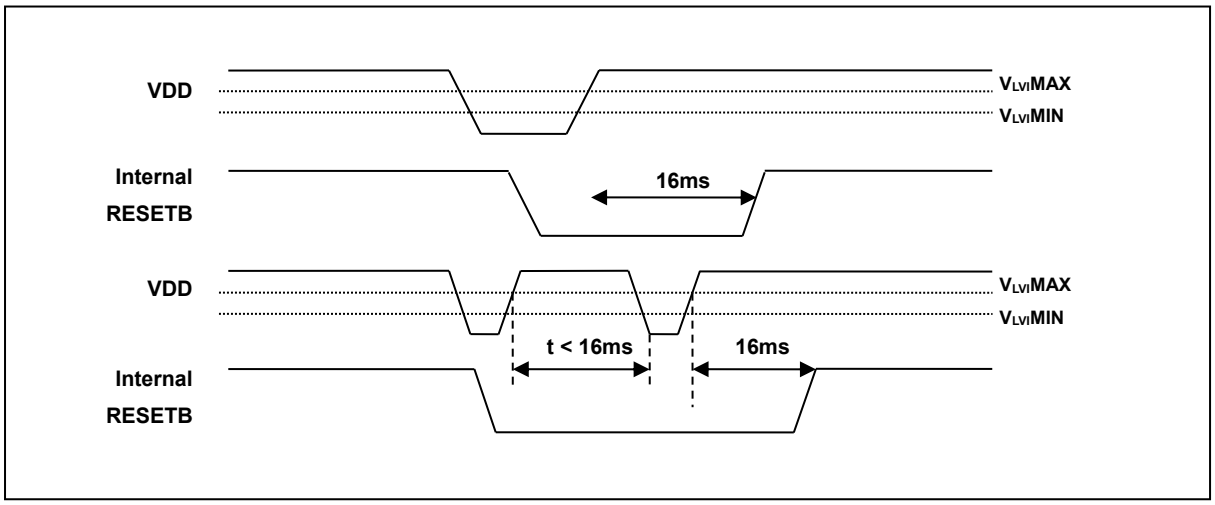


Figure 13.10 Internal Reset at the power fail situation

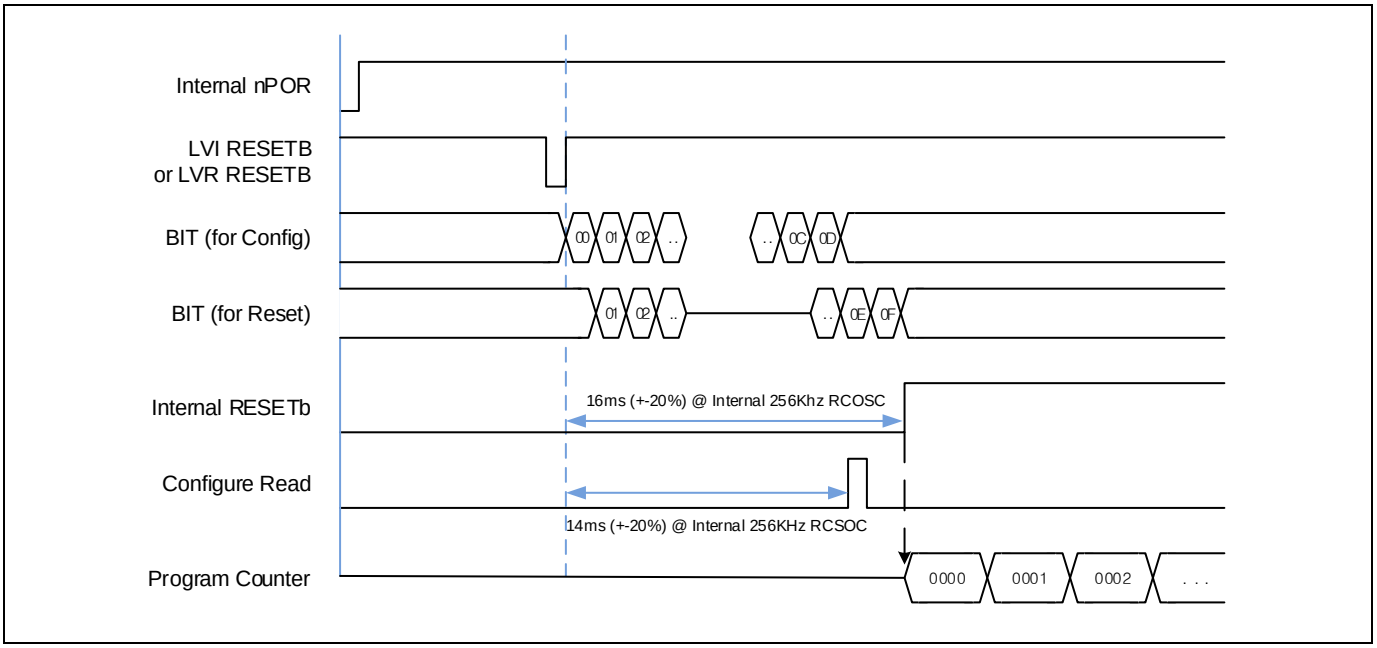


Figure 13.11 Configuration timing when Power-on

13.6.1 Register Map

Name	Address	Direction	Default	Description
RSFR	85H	R/W	88H	Reset Source Flag Register
LVIR	86H	R/W	09H	LVI Control Register

Table 13.3 Register Map

13.6.2 Reset Operation Register description

Reset Register consists of the Reset Source Flag Register (RSFR) and the LVI Control Register (LVIR).

13.6.3 Register description for Reset Operation

RSFR (Reset Source Flag Register) : 85H

7	6	5	4	3	2	1	0
PORF	CRCRF	WDTRF	OCDRF	LVRRF	LVIRF	EXTRF	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	-

Initial value : 88H

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

CRCRF CRC Reset flag bit. The bit is reset by writing '0' to this bit or by Power ON reset.
0 No detection
1 Detection

WDTRF Watch Dog Reset flag bit. The bit is reset by writing '0' to this bit or by Power ON reset.
0 No detection
1 Detection

OCDRF On-Chip Debug Reset flag bit. The bit is reset by writing '0' to this bit or by Power ON reset.
0 No detection
1 Detection

LVRRF Low Voltage Reset flag bit. The bit is reset by writing '0' to this bit or by Power ON reset.
0 No detection
1 Detection

LVIRF Low Voltage Indicator Reset flag bit. The bit is reset by writing '0' to this bit or by Power ON reset.
0 No detection
1 Detection

EXTRF External Reset flag bit. The bit is reset by writing '0' to this bit or by Power ON reset.
0 No detection
1 Detection

Note 1. By writing 'A5h' to AUTHORITY register, RSFR changing is possible.

Note 2. Only Power on reset resets the LVIR register.

LVIR (LVI Control Register) : 86H

7	6	5	4	3	2	1	0
-	-	LVI_BYPASS	LVR_BYPASS	LVIINTON	LVILS1	LVILS0	LVREN
-	-	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 09H

LVI_BYPASS Select LVI debounce or bypass
0 Debounce
1 Bypass

LVR_BYPASS Select LVR debounce or bypass
0 Debounce
1 Bypass

LVIINTON Select LVI reset or interrupt
0 reset
1 interrupt

LVILS[1:0] LVI level Voltage

LVILS1	LVILS0	Description
0	0	disable
0	1	2.4V
1	0	2.8V
1	1	3.5V

LVREN LVR stop mode operation
0 LVR disable
1 LVR enable

Note 1. By writing 'A5h' to AUTHORITY register, LVIR changing is possible.

14 On-chip Debug System

14.1 Overview

14.1.1 Description

On-chip debug system (OCD2) of A97C450 can be used for programming the non-volatile memories and on-chip debugging. Detail descriptions for programming via the OCD2 interface can be found in the following chapter.

Figure 14.1 shows a block diagram of the OCD2 interface and the On-chip Debug system.

14.1.2 Feature

- Two-wire external interface: 1-wire serial clock input, 1-wire bi-directional serial data bus
- Debugger Access to:
  - All Internal Peripheral Units
  - Internal data RAM
  - Program Counter
  - Flash and Data EEPROM Memories
- Extensive On-chip Debug Support for Break Conditions, Including
  - Break Instruction
  - Single Step Break
  - Program Memory Break Points on Single Address
  - Programming of Flash, EEPROM, Fuses and Lock Bits through the two-wire Interface
  - On-chip Debugging Supported by OCD dongle
- Operating frequency
- Supports the maximum frequency of the target MCU

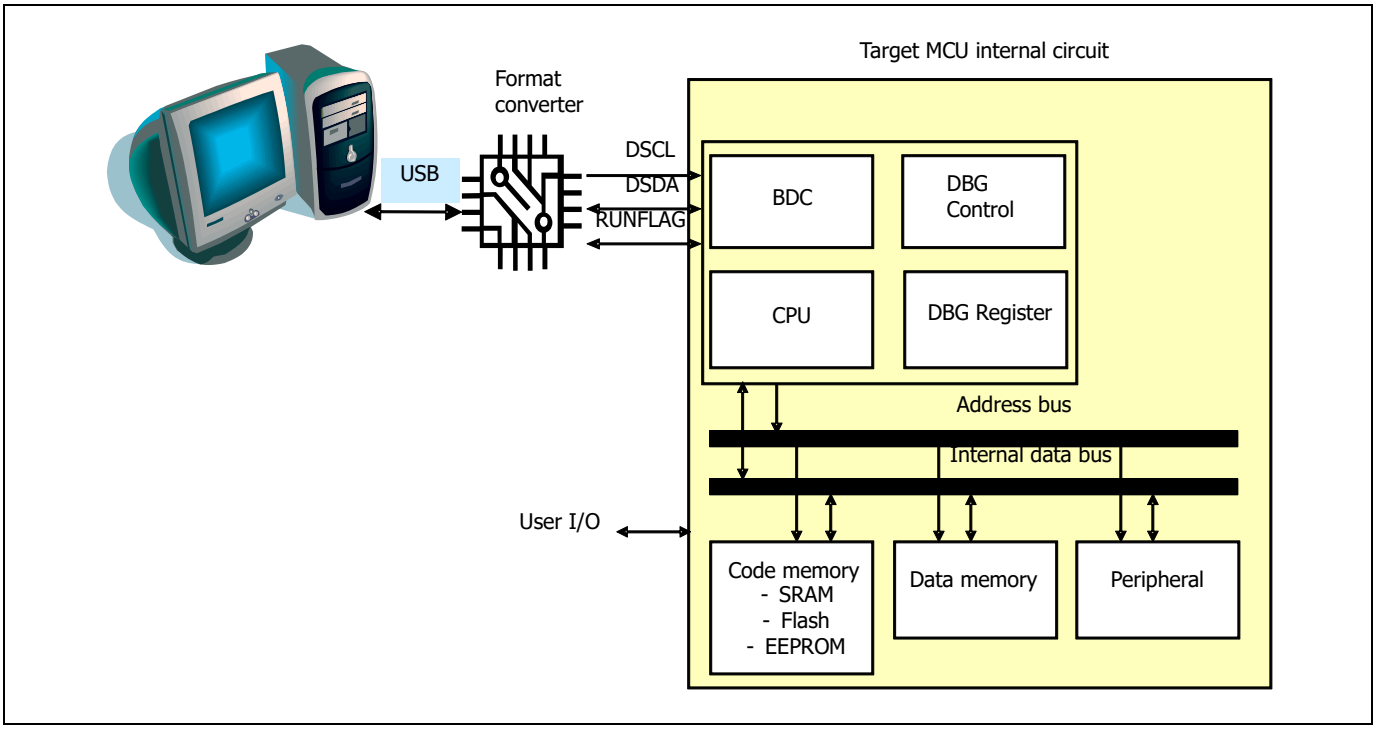


Figure 14.1 Block Diagram of On-Chip Debug System

14.2 Two-Pin External Interface

14.2.1 Basic Transmission Packet

- 10-bit packet transmission using two-pin interface.
- 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.

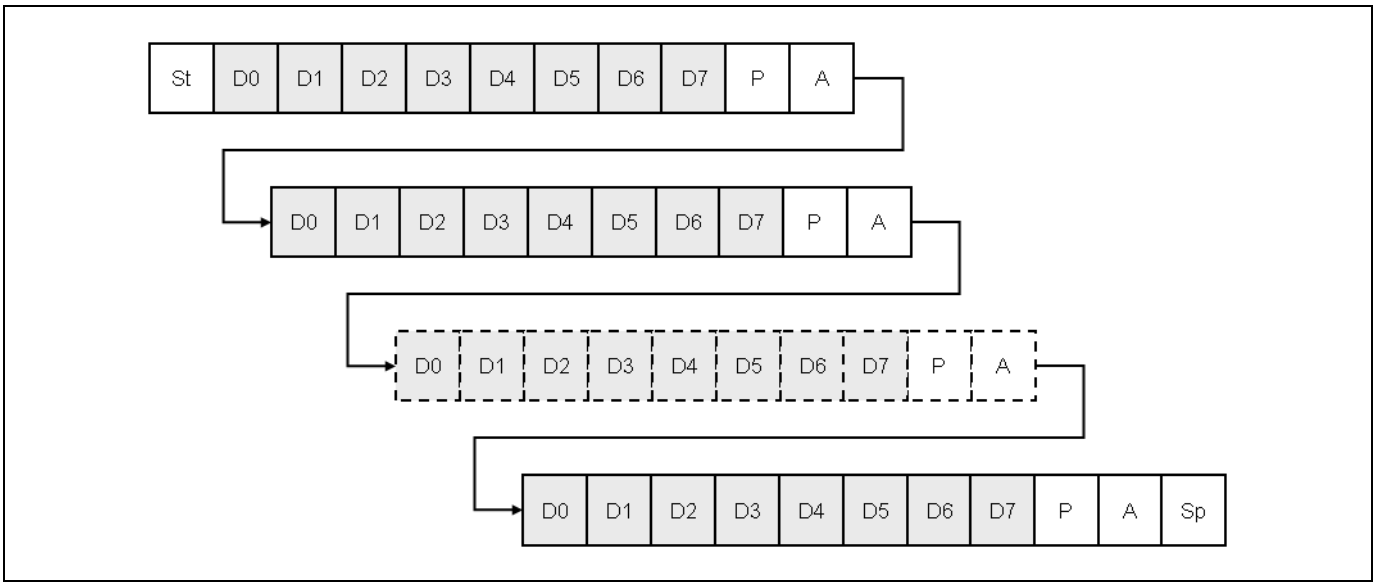


Figure 14.2 10-bit Transmission Packet

14.2.2 Packet Transmission Timing

14.2.2.1 Data Transfer

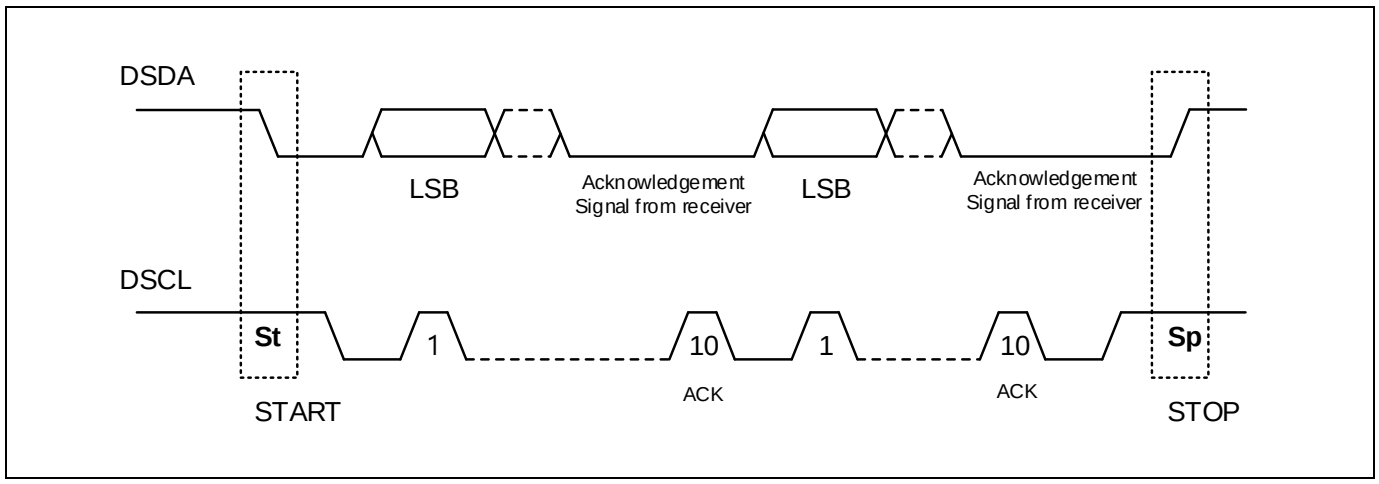


Figure 14.3 Data transfer on the twin bus

14.2.2.2 Bit Transfer

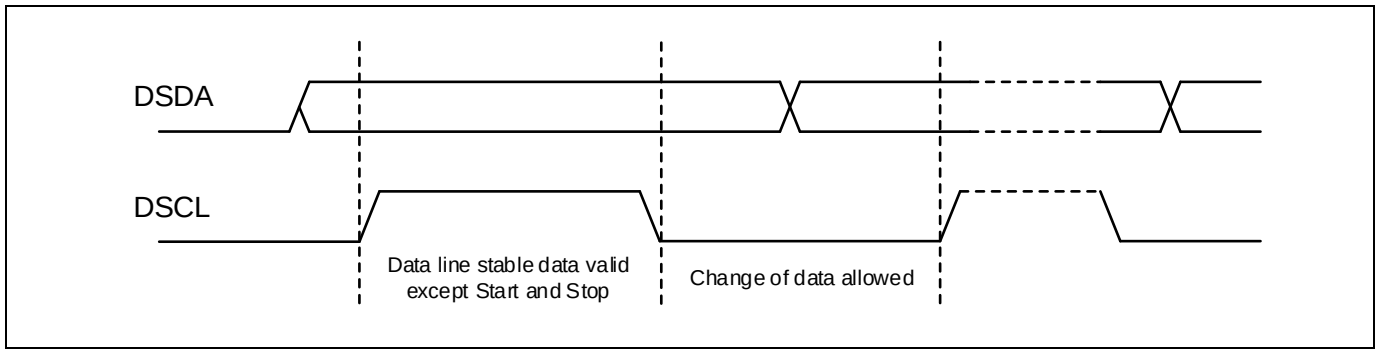


Figure 14.4 Bit transfer on the serial bus

14.2.2.3 Start and Stop Condition

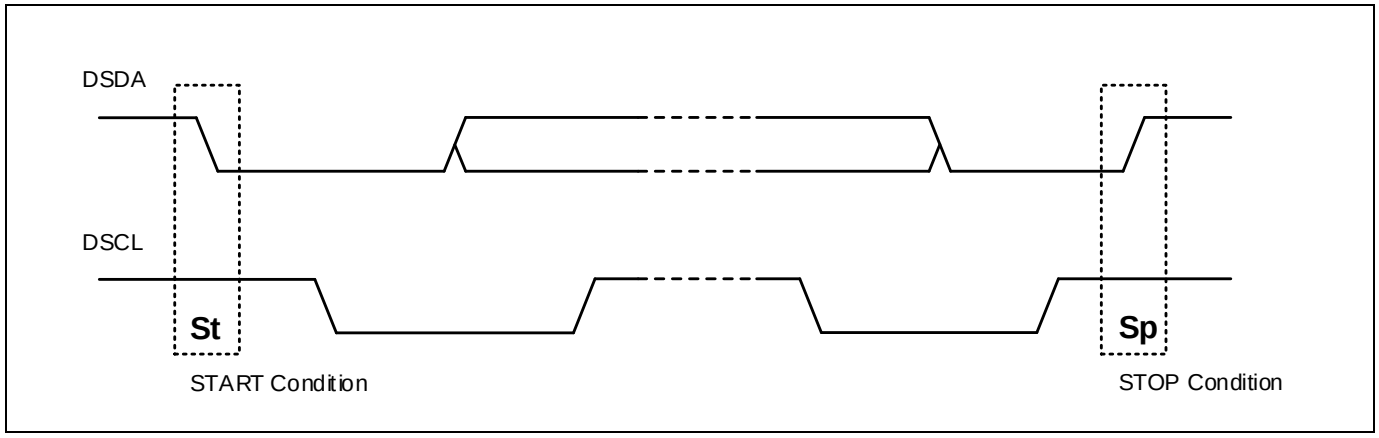


Figure 14.5 Start and stop condition

14.2.2.4 Acknowledge bit

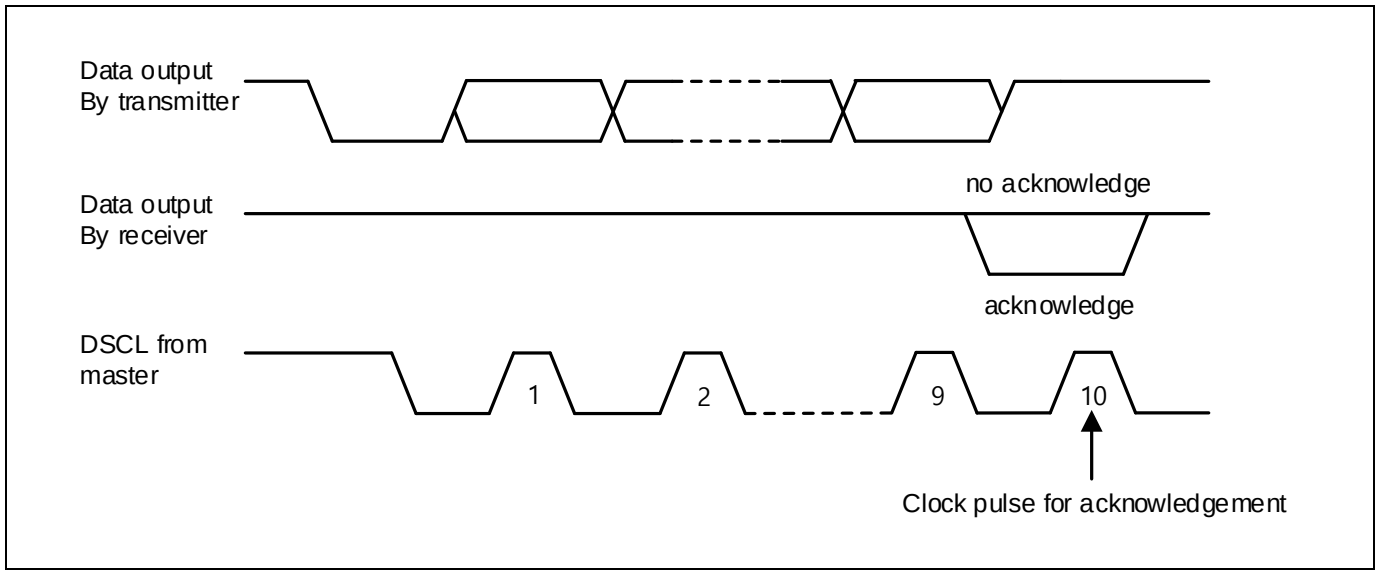


Figure 14.6 Acknowledge on the serial bus

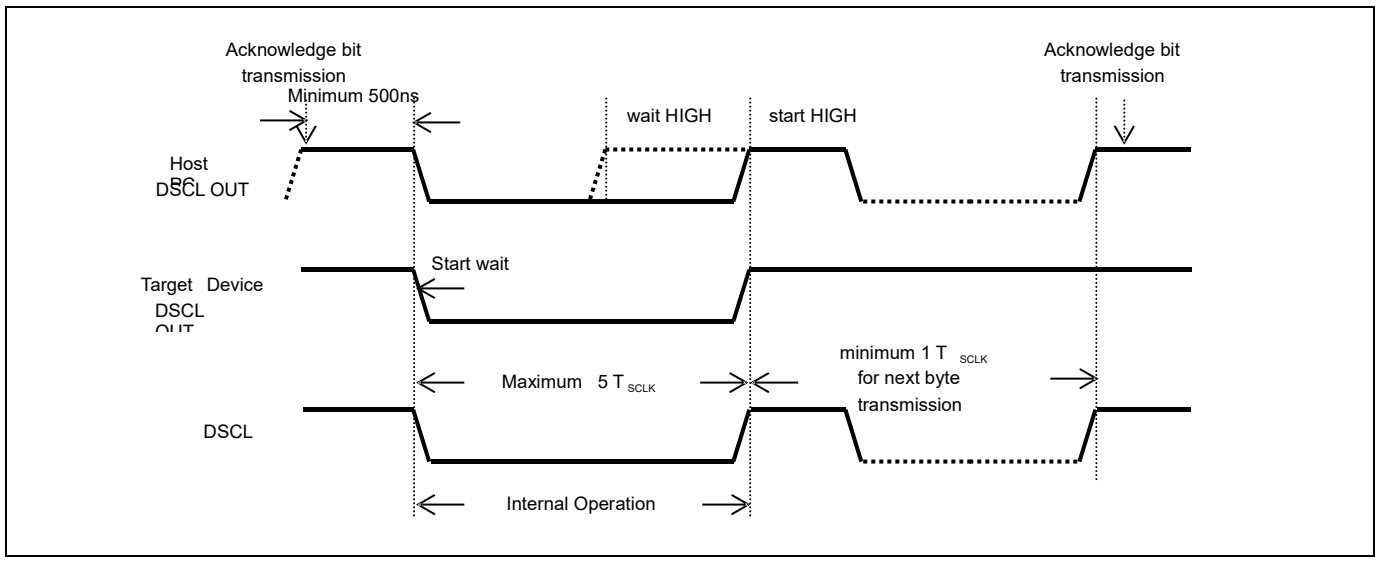


Figure 14.7 Clock synchronization during wait procedure

14.2.3 Connection of Transmission

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

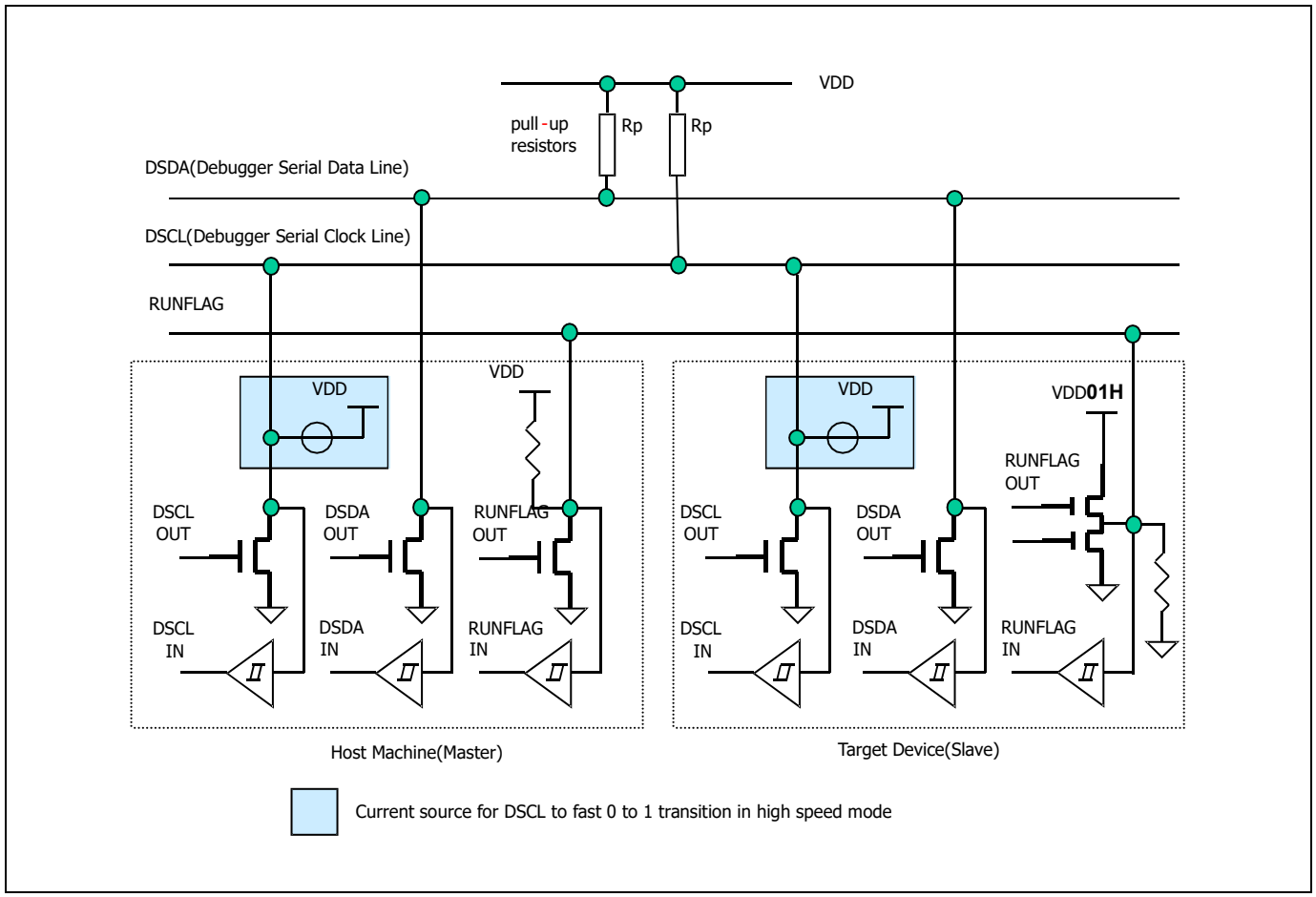


Figure 14.3 Connection of Transmission

15 Memory Programming

15.1 Overview

15.1.1 Description

A97C450 has flash memory to which a program can be written, erased, and overwritten while mounted on the board. Serial ISP modes and byte-parallel ROM writer mode are supported.

15.1.2 Features

- Flash Size : 64KB
- 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 and Data EEPROM are Mode Register (FEMR), Control Register (FECR), Status Register (FESR), Time Control Register (FETCR), Address Low Register (FEARL), Address Middle Register (FEARM), address High Register (FEARH), Data Register (FEDR), and OTP Page Selection Register (OTPADD). They are mapped to SFR area and can be accessed only in programming mode.

15.2.1 Register Map

Name	Address	Direction	Default	Description
FEMR	EAH	R/W	00H	Flash Mode Register
FECR	EBH	R/W	03H	Flash Control Register
FESR	ECH	R/W	80H	Flash Status Register
FETCR	EDH	R/W	00H	Flash Time Control Register
FEARL	F2H	R/W	00H	Flash Address Low Register
FEARM	F3H	R/W	00H	Flash Address Middle Register
FEARH	F4H	R/W	00H	Flash Address High Register
FEDR	F5H	R/W	00H	Flash Data Register
OTPADD	93H	R/W	00H	OTP Page Selection Register

Table 15.1 Memory Programming Register Map

15.2.2 Register description for Flash

FEMR (Flash Mode Register) : EAH

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

ESEL Select data EEPROM
- 0 Deselect data EEPROM
- 1 Select data EEPROM

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) : EBH

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

AEE — Enable data EEPROM bulk erase mode
- 0 Disable bulk erase mode of data EEPROM
- 1 Enable bulk erase mode of data EEPROM

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 and data EEPROM. It is cleared automatically after 1 clock
- 0 No operation
- 1 Start to program or erase of Flash and data EEPROM

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

nFERST — Reset Flash or data EEPROM control logic. It is cleared automatically after 1 clock
- 0 No operation
- 1 Reset Flash or data EEPROM 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) : ECH

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

Initial value : 80H

PEVBSY Operation status flag. It is cleared 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

PCRCRD CRC read Enable
- 0 CRC read disable(Checksum read for verify operation)
- 1 16-bit CRC read enable (from FEARM, FEARL)

MWAIT Enable CPU wait
- 0 Disable CPU wait
- 1 Enable CPU wait during flash erase or program.

ROMINT Flash interrupt request flag. Auto-cleared when program/erase/verify starts. Active in program/erase/verify completion
- 0 No interrupt request.
- 1 Interrupt request.

WMODE Write mode flag

EMODE Erase mode flag

VMODE Verify mode flag

FETCR (Flash Time Control Register) : EDH

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

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 one at each divided watchdog ring clock frequency (=WDTRC/2). 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 at 256 kHz clock: (255+1) * 2 * ((1/fWDTRC) * 2) = 4.0ms

In the case of 10% of error rate of counter source clock, program or erase time is 2.63~4.44ms

* Program/erase time calculation

For page write or erase, Tpe = (FETCR+1) * 2 * ((1/fWDTRC) * 2)

For bulk erase, Tbe = (FETCR+1) * 2 * ((1/fWDTRC) * 2)

1/fWDTRC = input system clock into EEPROM control block

Tpe = Typical program or erase time (2.5 ms)

FETCR = Value of FETCR register

※ Recommended program/erase time at fWDTRC 256kHz(FETCR = A0H)

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

Table 15.2 Program/Erase Time

FEARL (Flash address low Register) : F2H

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) : F3H

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) : F4H

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 results(FEARH, FEARM, FEARL) or 16-bit CRC results(FEARM, FEARL) when PCRCRD bit of FESR is set.

FEDR (Flash control Register) : F5H

7	6	5	4	3	2	1	0
FEDR7	FEDR6	FEDR5	FEDR4	FEDR3	FEDR2	FEDR1	FEDR0
W	W	W	W	W	W	W	W

Initial value : 00H

FEDR[7:0]	Flash data

Data register. In no program/erase/verify mode, READ/WRITE of FECR read or write data from FLASH to this register or from this register to Flash.

The sequence of writing data to this register is used for EEPROM program entry. The mode entrance sequence is to write 0xA5 and 0x5A to it in order.

OTPADD (OTP Page Selection Register) : 93H

7	6	5	4	3	2	1	0
OTPADD7	OTPADD6	OTPADD5	OTPADD4	OTPADD3	OTPADD2	OTPADD1	OTPADD0
W	W	W	W	W	W	W	W

Initial value : 00H

OTPADD7 OTP page 7 selection
0 Deselect
1 Select

OTPADD6 OTP page 6 selection
0 Deselect
1 Select

OTPADD5 OTP page 5 selection
0 Deselect
1 Select

OTPADD4 OTP page 4 selection
0 Deselect
1 Select

OTPADD3 OTP page 3 selection
0 Deselect
1 Select

OTPADD2 OTP page 2 selection
0 Deselect
1 Select

OTPADD1 OTP page 1 selection
0 Deselect
1 Select

OTPADD0 OTP page 0 selection
0 Deselect
1 Select

15.3 Memory map

15.3.1 Flash Memory Map

Program memory uses 64Kbyte of Flash memory. It is read by byte and written by byte or page. One page is 64 bytes.

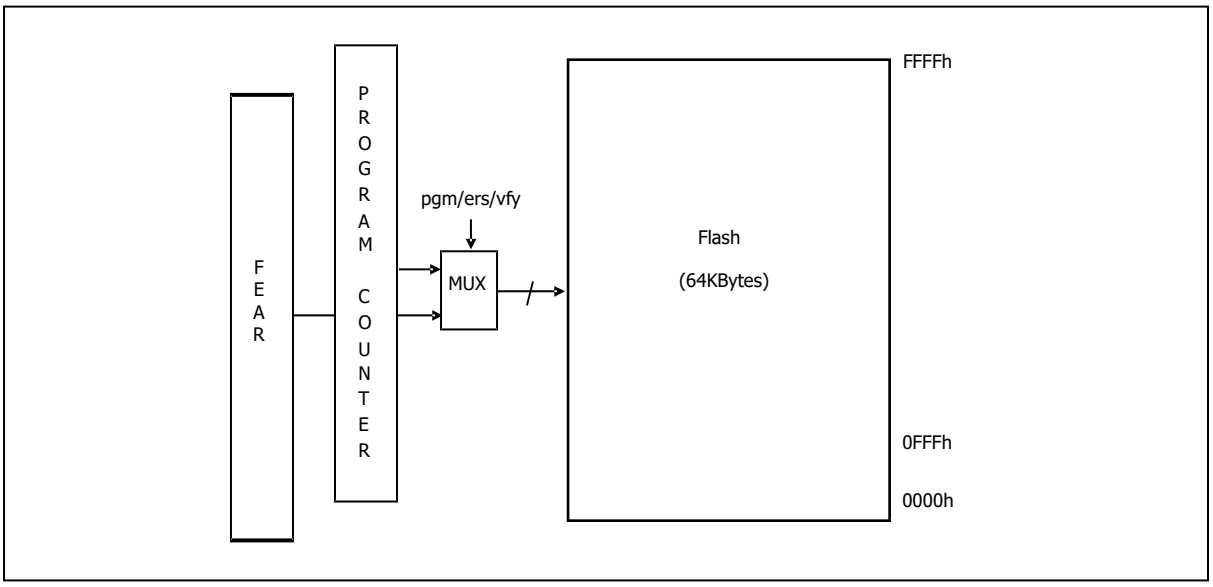


Figure 15.1 Flash Memory Map

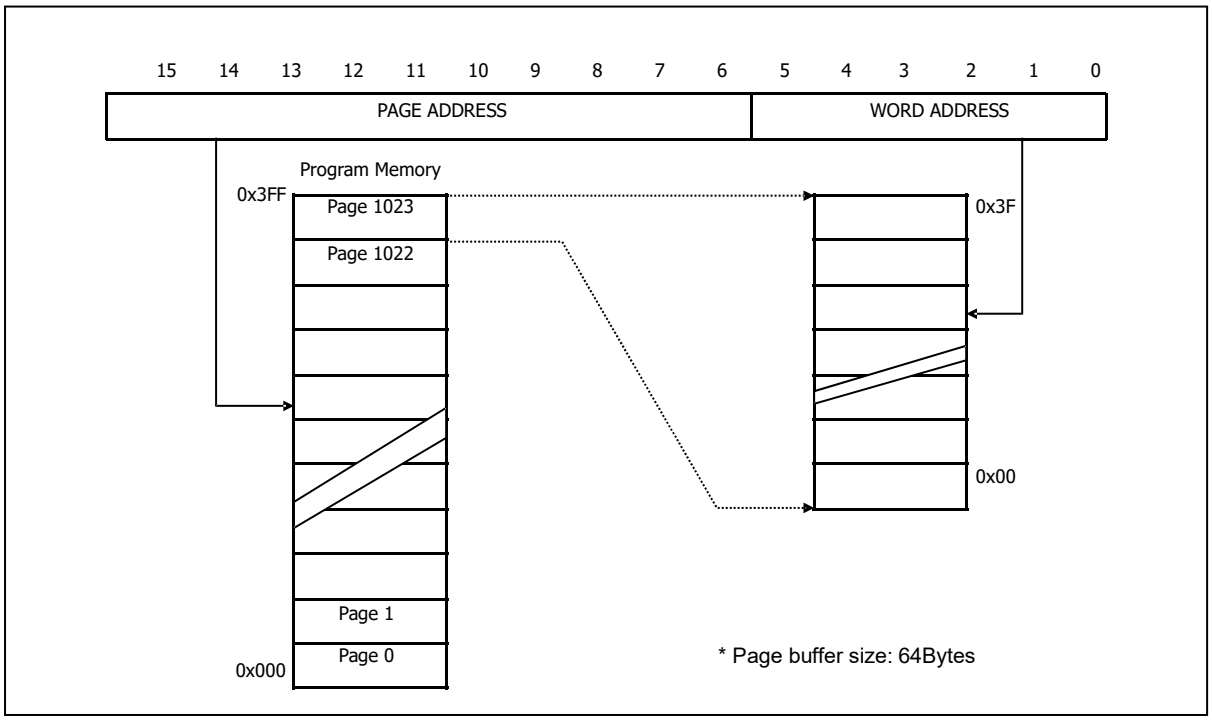


Figure 15.2 Address configuration of Flash memory

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

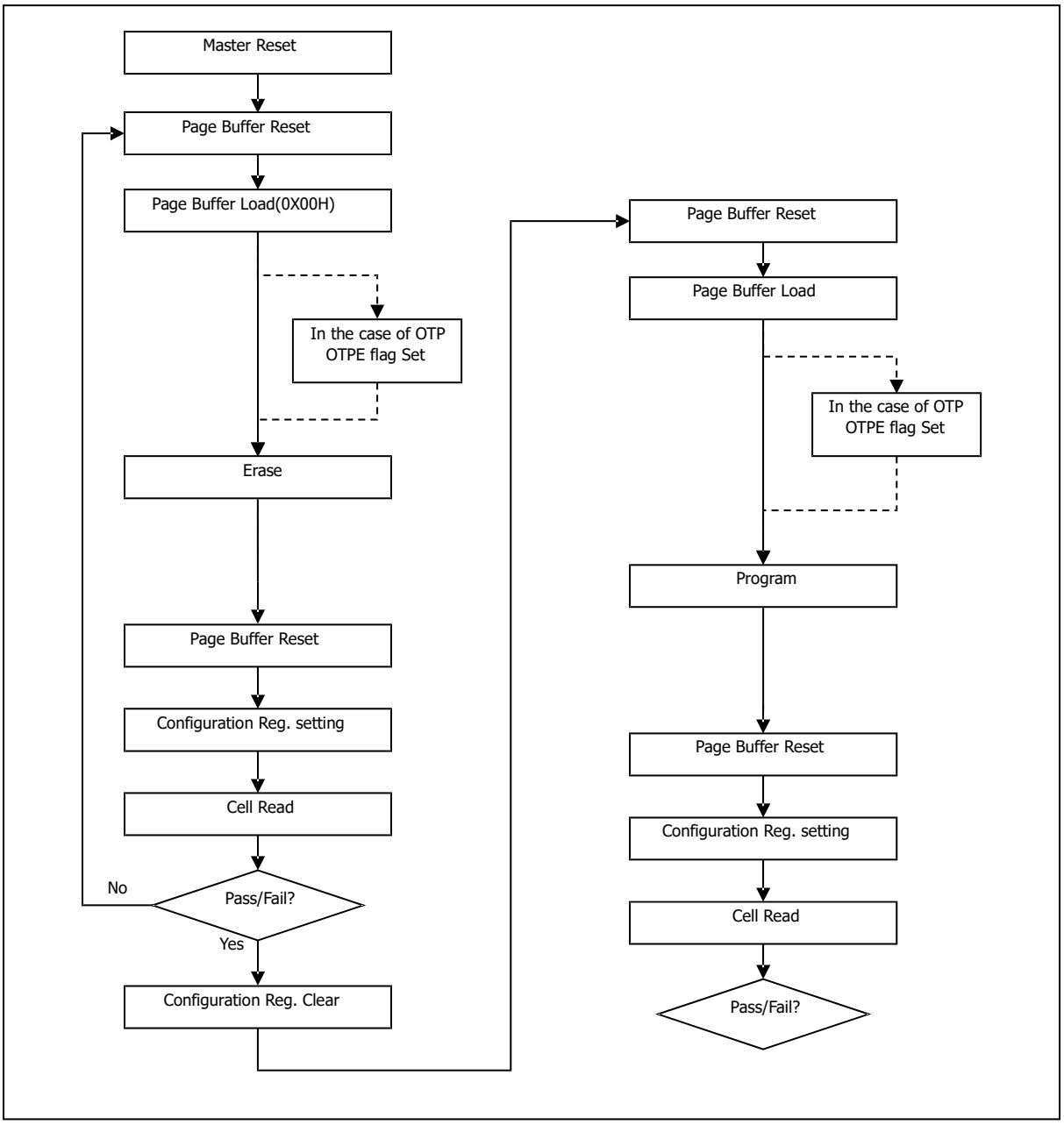


Figure 15.3 The sequence of page program and erase of Flash memory

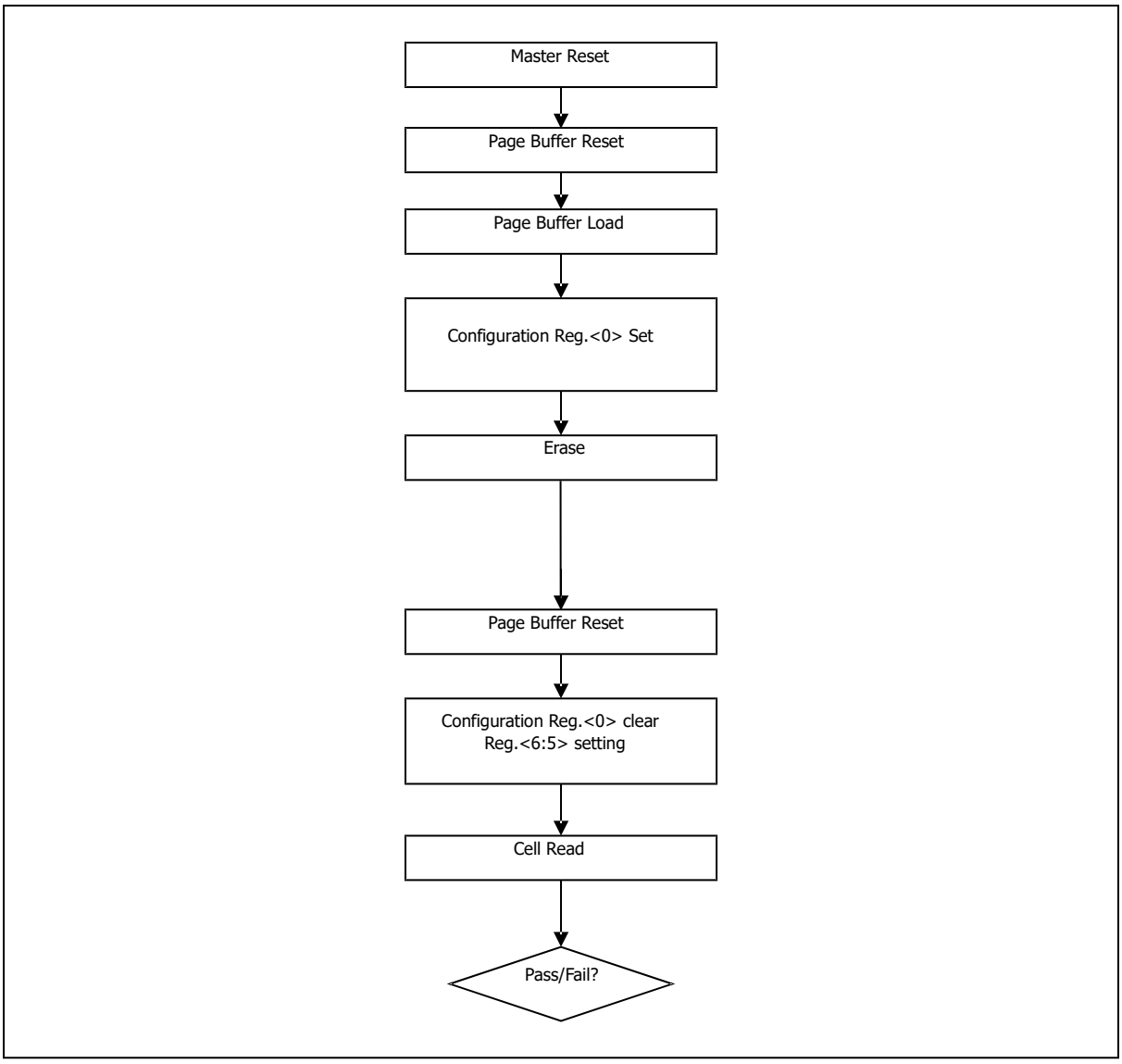


Figure 15.4 The sequence of bulk erase of Flash memory

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 sequence.[2]

(1) Write 0xAA to 0xF555.

(2) Write 0x55 to 0xFAAA.

(3) Write 0xA5 to 0xF555.

[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 Data EEPROM operation

Program and erase operation of Data EEPROM are executed by direct and indirect address mode.

Direct address mode uses external data area of 8051. Indirect address mode uses address register of SFR area..

15.4.2.1 Data EEPROM 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 Data EEPROM.

15.4.2.2 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 sequence.[2]

(1) EA = 0 (Grobal interrupt disable)

(2) Write 0xA5 to FEDR.

(3) Write 0x5A to FEDR.

(4) EA = 1 (Grobal interrupt Enable)

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

[2] Command sequence to activate data EEPROM write/erase mode. It is composed of sequentially writing to data register.(FEDR)

[3] **No other instructions should be entered in the middle of sequential operations**.

15.4.2.3 EEPROM write mode

Step 1. Enable program mode.

Step 2. Reset page buffer. FEMR: 0100_0001 FECR:0000_0010

Step 3. Select page buffer. FEMR:0100_1001

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

Step 5. Set write mode. FEMR:0110_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.

15.4.2.4 EEPROM page erase mode

Step 1. Enable program mode.

Step 2. Reset page buffer. FEMR: 0100_0001 FECR:0000_0010

Step 3. Select page buffer. FEMR:0100_1001

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

Step 5. Set erase mode. FEMR:0101_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.

15.4.2.5 EEPROM bulk erase mode

Step 1. Enable program mode.

Step 2. Reset page buffer. FEMR: 0100_0001 FECR:0000_0010

Step 3. Select page buffer. FEMR:0100_1001

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

Step 5. Set erase mode. FEMR:0101_0001.

Step 6. Set FETCR

Step 7. Start bulk erase. FECR:0100_1011

Step 8. Insert one NOP operation

Step 9. Read FESR until PEVBSY is 1.

15.4.2.6 Data EEPROM program verify mode

Step 1. Enable program mode.

Step 2. Set program verify mode. FEMR:0110_0011

Step 3. Read data from Flash.

15.4.2.7 Data EEPROM erase verify mode

Step 1. Enable program mode.

Step 2. Set erase verify mode. FEMR:0101_0011

Step 3. Read data from Flash.

15.4.2.8 Data EEPROM page buffer read

Step 1. Enable program mode.

Step 2. Select page buffer. FEMR:0100_1001

Step 3. Read data from Flash.

15.4.3 Summary of Flash and Data EEPROM Program/Erase Mode

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

Table 15.3 Operation Mode

15.5 Parallel Mode

15.5.1 Overview

Parallel program mode transfers address and data by byte. 3-byte address can be entered by one from the lease significant byte of address. If only LSB is changed, only one byte can be transferred. And if the second byte is changed, the first and second byte can be transferred. Upper 4-bit of the most significant byte selects memory to be accessed. Table 15.4 show memory type according to address. Address auto-increment is supported when read or write data without address

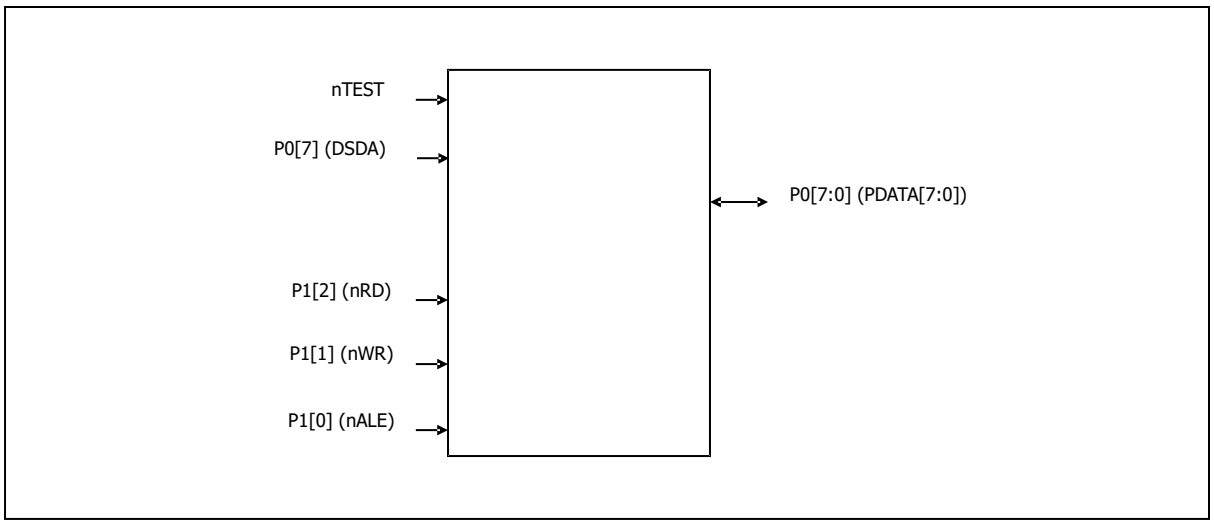


Figure 15.5 Pin diagram for parallel programming

ADDRH[7:4]				Memory Type
0	0	0	0	Program Memory
0	0	0	1	External Memory
0	0	1	0	SFR

Table 15.4 The selection of memory type by ADDRH[7:4]

15.5.2 Parallel Mode instruction format

Instruction	Signal	Instruction Sequence													
n-byte data read with 3-byte address	nALE	L		L		L		H		H		H		H	
	nWR	L	H	L	H	L	H	H	H	H	H	H	H	H	H
	nRD	H	H	H	H	H	H	L	H	L	H	L	H	L	H
	PDATA	ADDRL		ADDRM		ADDRH		DATA0		DATA1		---		DATAn	
n-byte data write with 3-byte address	nALE	L		L		L		H		H		H		H	
	nWR	L	H	L	H	L	H	L	H	L	H	L	H	L	H
	nRD	H	H	H	H	H	H	H	H	H	H	H	H	H	H
	PDATA	ADDRL		ADDRM		ADDRH		DATA0		DATA1		---		DATAn	
n-byte data read with 2-byte address	nALE	L		L		H		H		H		H		H	
	nWR	L	H	L	H	H	H	H	H	H	H	H	H	H	H
	nRD	H	H	H	H	L	H	L	H	L	H	L	H	L	H
	PDATA	ADDRL		ADDRM		DATA0		DATA1		DATA2		---		DATAn	
n-byte data write with 2-byte address	nALE	L		L		H		H		H		H		H	
	nWR	L	H	L	H	L	H	L	H	L	H	L	H	L	H
	nRD	H	H	H	H	H	H	H	H	H	H	H	H	H	H
	PDATA	ADDRL		ADDRM		DATA0		DATA1		DATA2		---		DATAn	
n-byte data read with 1-byte address	nALE	L		H		H		H		H		H		H	
	nWR	L	H	H	H	L	H	L	H	L	H	L	H	L	H
	nRD	H	H	L	H	H	H	H	H	H	H	H	H	H	H
	PDATA	ADDRL		DATA0		DATA1		DATA2		DATA3		---		DATAn	
n-byte data write with 1-byte address	nALE	L		H		H		H		H		H		H	
	nWR	L	H	L	H	L	H	L	H	L	H	L	H	L	H
	nRD	H	H	H	H	H	H	H	H	H	H	H	H	H	H
	PDATA	ADDRL		DATA0		DATA1		DATA2		DATA3		---		DATAn	

Table 15.5 Parallel Mode instruction format

15.5.3 Parallel Mode timing diagram

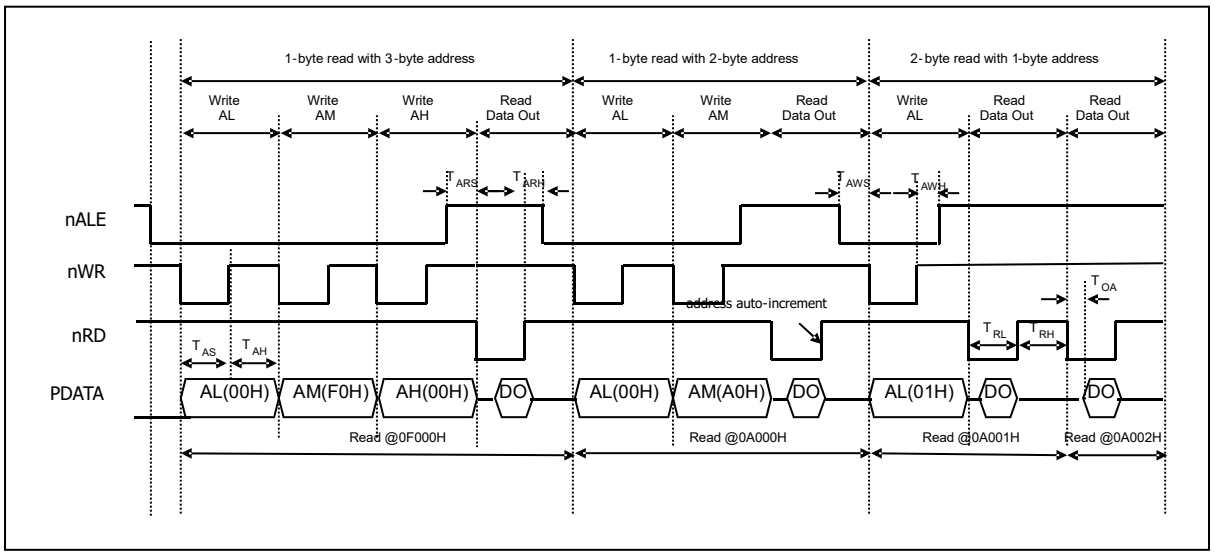


Figure 15.6 Parallel Byte Read Timing of Program Memory

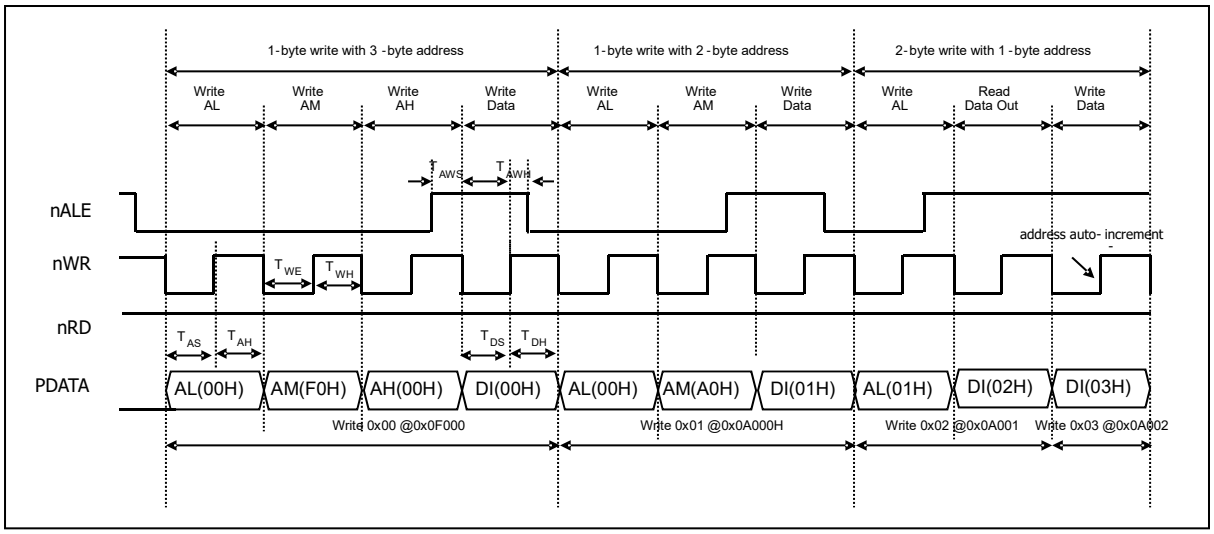


Figure 15.7 Parallel Byte Write Timing of Program Memory

15.6 Mode entrance method of ISP and byte-parallel mode

15.6.1 Mode entrance method for ISP

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

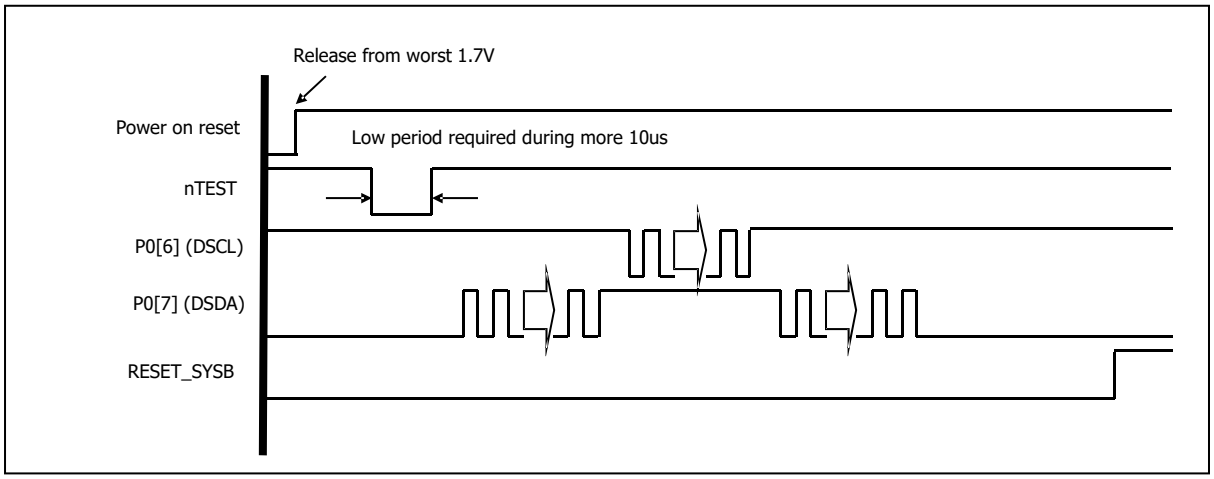


Figure 14.8 ISP mode

15.6.2 Mode entrance of Byte-parallel

TARGET MODE	P0[3:0]	P0[3:0]	P0[3:0]
Byte-Parallel Mode	4‘h5	4‘hA	4‘h5

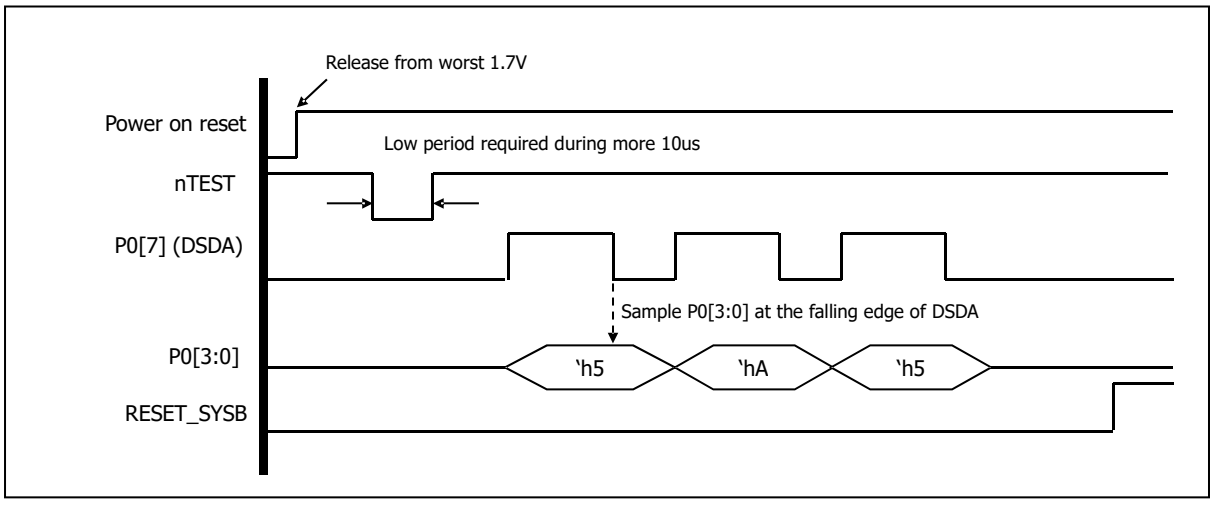


Figure 14.1 Byte-parallel mode

15.7 Security

A97C450 provides Lock bits which can be left un-programmed (“0”) or can be programmed (“1”) to obtain the additional features listed in Table 15.6. The Lock bit can only be erased to “0” with the bulk erase command and a value of more than 0x80 at FETCR.

LOCK MODE	USER MODE								ISP/PMODE							
	FLASH				OTP				FLASH				OTP			
LOCKF	R	W	PE	BE	R	W	PE	BE	R	W	PE	BE	R	W	PE	BE
0	O	O	O	X	X	X	X	X	O	O	O	O	O	O	O	O
1	O	O	O	X	X	X	X	X	X	X	X	O	O	X	X	O

Table 15.6 Security policy using lock-bits

- LOCKF: Lock bit of Flash memory
- R: Read
- W: Write
- PE: Page erase
- BE: Bulk Erase
- O: Operation is possible.
- X: Operation is impossible.

16 Configure option

16.1 Configure option Control Register

FUSE_CONF (Pseudo-Configure Data) : 000H (OPT)

7	6	5	4	3	2	1	0
-	RSTDIS	-	BSIZE1	BSIZE0	LOCKB	LOCKE	LOCKF
-	R/W	-	R/W	R/W	R/W	R/W	R/W

Initial value : 40H

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

BSIZE Select Specific Area for Write Protection

Note)When LOCKB is set, it's applied
- 00 0000H~01FFH, 0.5Kbytes (default)
- 01 0000H~03FFH, 1kbytes
- 10 0000H~07FFH, 2kbytes
- 11 0000H~0FFFH, 4kbytes

LOCKB Select Code Write Protection with Specific Area
- 0 Disable Code Write Protection
- 1 Enable Code Write Protection

LOCKE Select Data EEPROM Write Protection with Specific Area
- 0 Disable Data EEPROM Write Protection
- 1 Enable Data EEPROM Write Protection

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

In OCD debug mode, user can change FUSE_CONF bits value temporarily except LOCKF for debugging job.

FUSE_SWAPEN (Bootloader SWAP Enable) : 001H (OPT)

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

Initial value : 00H

Writing 5Ah to BOOT_SWAPEN enables bootloader swap operation

FUSE_XTAL (XTAL/Sub-XTAL Control) : 002H (OPT)

7	6	5	4	3	2	1	0
-	XTAL_FIL_SKIP	FUSE_XTALEN[1:0]		ISEL[1:0]		SUB_NC_BYPS	FUSE_SUBENA
-	R/W	R/W	R/W	R/W	R/W-	R/W	R/W

Initial value : 00H

XTAL_FIL_SKIP	Main-OSC noise canceller bypass selection	
	0	Use noise canceller
	1	Bypass noise canceller
FUSE_XTALEN[1:0]	Enable main-OSC and select whether to use P53, P54 as GPIO or main-OSC	
	00	Disable main-OSC. P54, P53 is used as GPIO
	01	Enable main-OSC. P54 is used as XOUT and P53 is used as XIN
	10	Enable main-OSC. P54 is used as GPIO and P53 is used as XIN
ISEL[1:0]	Sub osc Current option. This value be written by OCD or Write device according to FUSE_SUBENA bit. FUSE_SUBENA = 0 ⇒ ISEL[1:0] = 00b FUSE_SUBENA = 1 ⇒ ISEL[1:0] = 11b	
SUB_NC_BYPS	Sub-OSC noise canceller bypass selection	
	0	Use noise canceller
	1	Bypass noise canceller
FUSE_SUBENA	Enable sub-OSC and use P51, P50 as SXI and SXO	
	0	Disable sub-OSC. P51, P50 is used as GPIO
	1	Enable sub-OSC. P51, P50 is used as SXI and SXO

FUSE_BTSEL (Bootloader select) : 043H (OPT)

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

Initial value : 00H

This option select the bootloader when boot loader swap is enabled.

Boot load Select

0xA5 = Select Page 1, Others = Select Page 0

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 or 4 machine cycles to execute as listed in the following table. 1 machine cycle comprises 2 system clock cycles.

ARITHMETIC				
Mnemonic	**Description**	**Bytes**	**Cycles**	**Hex code**
ADD A, Rn	Add register to A	1	1	28-2F
ADD A, dir	Add direct byte to A	2	1	25
ADD A, @Ri	Add indirect memory to A	1	1	26-27
ADD A, #data	Add immediate to A	2	1	24
ADDC A, Rn	Add register to A with carry	1	1	38-3F
ADDC A, dir	Add direct byte to A with carry	2	1	35
ADDC A, @Ri	Add indirect memory to A with carry	1	1	36-37
ADDC A, #data	Add immediate to A with carry	2	1	34
SUBB A, Rn	Subtract register from A with borrow	1	1	98-9F
SUBB A, dir	Subtract direct byte from A with borrow	2	1	95
SUBB A, @Ri	Subtract indirect memory from A with borrow	1	1	96-97
SUBB A, #data	Subtract immediate from A with borrow	2	1	94
INC A	Increment A	1	1	04
INC Rn	Increment register	1	1	08-0F
INC dir	Increment direct byte	2	1	05
INC @Ri	Increment indirect memory	1	1	06-07
DEC A	Decrement A	1	1	14
DEC Rn	Decrement register	1	1	18-1F
DEC dir	Decrement direct byte	2	1	15
DEC @Ri	Decrement indirect memory	1	1	16-17
INC DPTR	Increment data pointer	1	2	A3
MUL AB	Multiply A by B	1	4	A4
DIV AB	Divide A by B	1	4	84
DA A	Decimal Adjust A	1	1	D4

LOGICAL				
Mnemonic	**Description**	**Bytes**	**Cycles**	**Hex code**
ANL A, Rn	AND register to A	1	1	58-5F
ANL A, dir	AND direct byte to A	2	1	55
ANL A, @Ri	AND indirect memory to A	1	1	56-57
ANL A, #data	AND immediate to A	2	1	54
ANL dir, A	AND A to direct byte	2	1	52
ANL dir, #data	AND immediate to direct byte	3	2	53
ORL A, Rn	OR register to A	1	1	48-4F
ORL A, dir	OR direct byte to A	2	1	45
ORL A, @Ri	OR indirect memory to A	1	1	46-47
ORL A, #data	OR immediate to A	2	1	44
ORL dir, A	OR A to direct byte	2	1	42
ORL dir, #data	OR immediate to direct byte	3	2	43
XRL A, Rn	Exclusive-OR register to A	1	1	68-6F
XRL A, dir	Exclusive-OR direct byte to A	2	1	65
XRL A, @Ri	Exclusive-OR indirect memory to A	1	1	66-67
XRL A, #data	Exclusive-OR immediate to A	2	1	64
XRL dir, A	Exclusive-OR A to direct byte	2	1	62
XRL dir, #data	Exclusive-OR immediate to direct byte	3	2	63
CLR A	Clear A	1	1	E4
CPL A	Complement A	1	1	F4
SWAP A	Swap Nibbles of A	1	1	C4
RL A	Rotate A left	1	1	23
RLC A	Rotate A left through carry	1	1	33
RR A	Rotate A right	1	1	03
RRC A	Rotate A right through carry	1	1	13

DATA TRANSFER				
Mnemonic	**Description**	**Bytes**	**Cycles**	**Hex code**
MOV A, Rn	Move register to A	1	1	E8-EF
MOV A, dir	Move direct byte to A	2	1	E5
MOV A, @Ri	Move indirect memory to A	1	1	E6-E7
MOV A, #data	Move immediate to A	2	1	74
MOV Rn, A	Move A to register	1	1	F8-FF
MOV Rn, dir	Move direct byte to register	2	2	A8-AF
MOV Rn, #data	Move immediate to register	2	1	78-7F
MOV dir, A	Move A to direct byte	2	1	F5
MOV dir, Rn	Move register to direct byte	2	2	88-8F
MOV dir, dir	Move direct byte to direct byte	3	2	85
MOV dir, @Ri	Move indirect memory to direct byte	2	2	86-87
MOV dir, #data	Move immediate to direct byte	3	2	75
MOV @Ri, A	Move A to indirect memory	1	1	F6-F7
MOV @Ri, dir	Move direct byte to indirect memory	2	2	A6-A7
MOV @Ri, #data	Move immediate to indirect memory	2	1	76-77
MOV DPTR, #data	Move immediate to data pointer	3	2	90
MOVC A, @A+DPTR	Move code byte relative DPTR to A	1	2	93
MOVC A, @A+PC	Move code byte relative PC to A	1	2	83
MOVX A, @Ri	Move external data(A8) to A	1	2	E2-E3
MOVX A, @DPTR	Move external data(A16) to A	1	2	E0
MOVX @Ri, A	Move A to external data(A8)	1	2	F2-F3
MOVX @DPTR, A	Move A to external data(A16)	1	2	F0
PUSH dir	Push direct byte onto stack (except PUSH SP)[NOTE1]	2	2	C0
POP dir	Pop direct byte from stack	2	2	D0
XCH A, Rn	Exchange A and register	1	1	C8-CF
XCH A, dir	Exchange A and direct byte	2	1	C5
XCH A, @Ri	Exchange A and indirect memory	1	1	C6-C7
XCHD A, @Ri	Exchange A and indirect memory nibble	1	1	D6-D7

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

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

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

ADDITIONAL INSTRUCTIONS (selected through EO[7:4])				
Mnemonic	**Description**	**Bytes**	**Cycles**	**Hex code**
MOVC @(DPTR++), A	M8051W/M8051EW-specific instruction supporting software download into program memory	1	2	A5
TRAP	Software break command	1	1	A5

In the above table, an entry such as E8-EF indicates a continuous block of hex opcodes used for 8 different registers, the register numbers of which are defined by the lowest three bits of the corresponding code. Non-continuous blocks of codes, shown as 11→F1 (for example), are used for absolute jumps and calls, with the top 3 bits of the code being used to store the top three bits of the destination address.

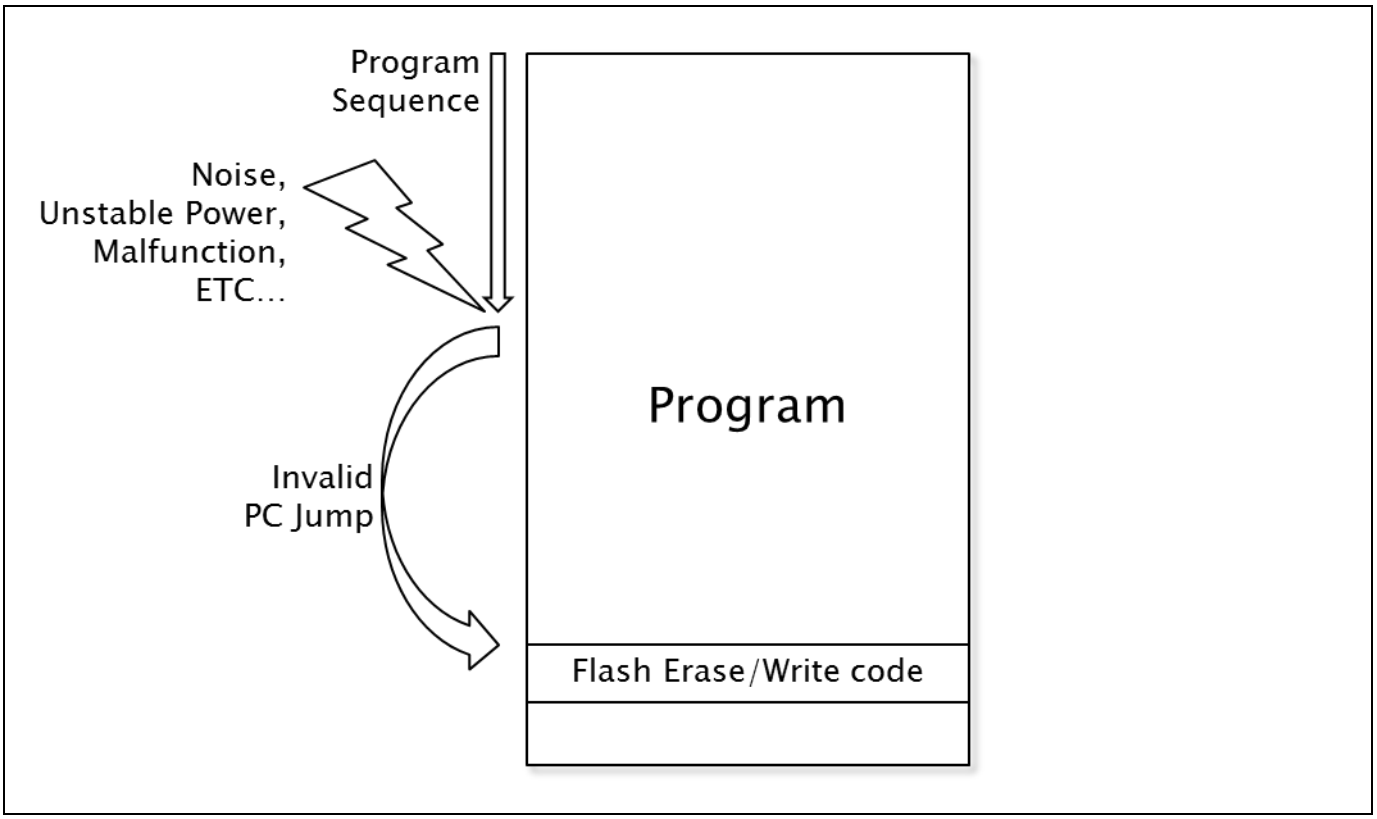
The CJNE instructions use the abbreviation #d for immediate data; other instructions use #data.

NOTE1) If the operand of the PUSH instruction is SP, the operation is slightly different from that of the 8051. The use of this instruction only occurs when using the assembly language, and its use is very limited. Refer to "ERRATA_95_96_97 CPU(M8051) Incompatible Instruction" for more details.

17.2 Flash Protection for Invalid Erase/Write

➢ Overview

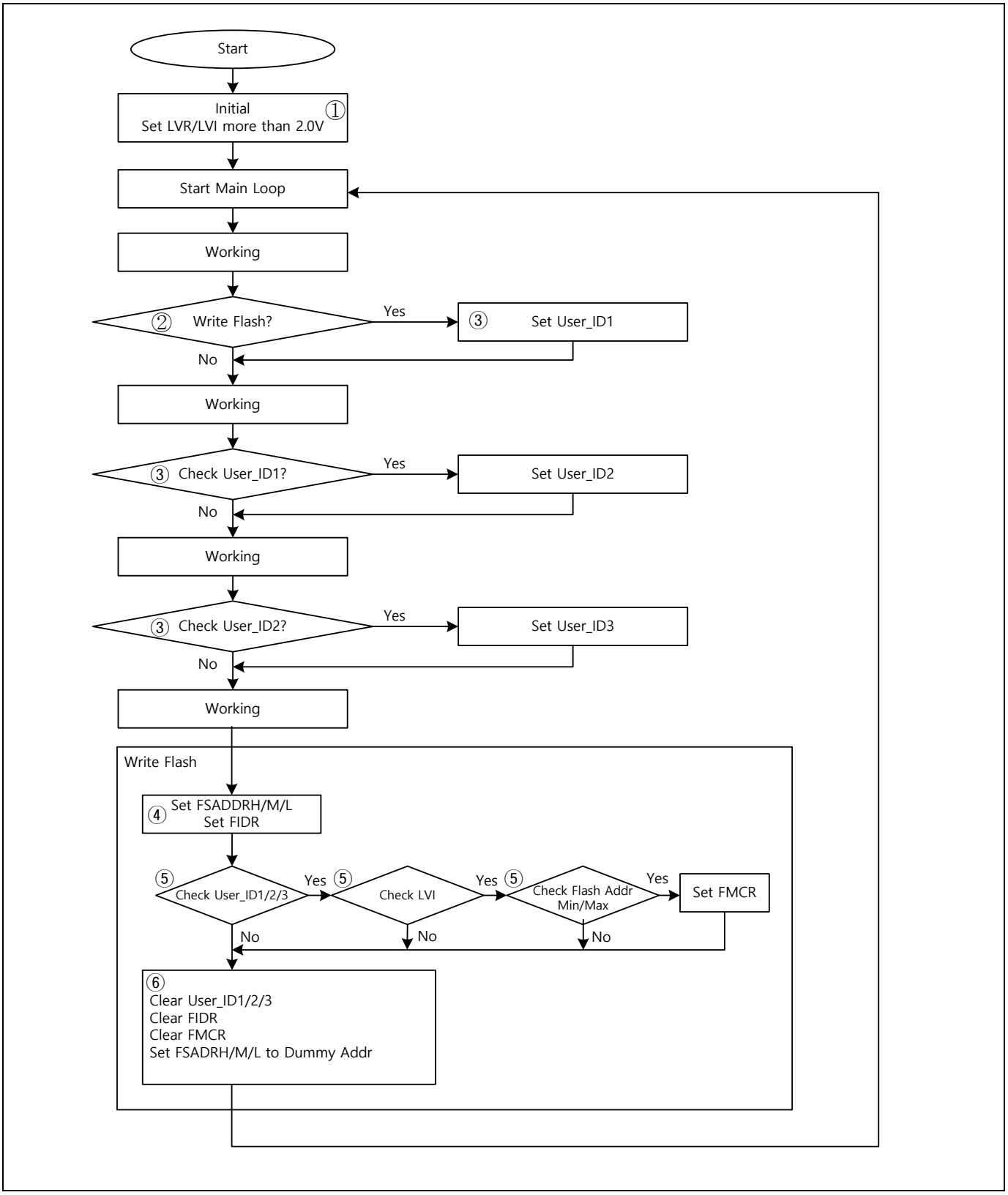
This is example to prevent changing code or data in flash by abnormal operation(noise, unstable power, malfunction, etc…).



➢ How to protect the flash

- Divide into decision and execution to Erase/Write in flash.
  - Check the program sequence from decision to execution in order of precedence about Erase/Write.
  - Setting the flags in program and check the flags in main loop at the end
  - When the Flash Erase/Write is executed, check the flags. If not matched, do not execute.
- Check the range of Flash Sector Address
  - If the flash sector address is outside of specific area, do not execute.
- Use the Dummy Address
  - Set the flash sector address to dummy address in usually run time.
  - Change the flash sector address to real area range shortly before Erase/Write.
  - Even if invalid Erase/Write occurred, it will be Erase/Write in dummy address in flash.
- Use the LVR/LVI
  - Unstable or low powers give an adverse effect on MCU. So use the LVR/LVI

➢ Flowchart


Start
Initial
Set LVR/LVI more than 2.0V ①
Start Main Loop
Working
② Write Flash?
Yes
③ Set User_ID1
No
Working
③ Check User_ID1?
Yes
Set User_ID2
No
Working
③ Check User_ID2?
Yes
Set User_ID3
No
Working
Write Flash
④ Set FSADDRH/M/L
Set FIDR
⑤ Check User_ID1/2/3
Yes
⑤ Check LVI
Yes
⑤ Check Flash Addr Min/Max
Yes
Set FMCR
No
No
No
⑥
Clear User_ID1/2/3
Clear FIDR
Clear FMCR
Set FSADRH/M/L to Dummy Addr

➢ Descript of Flowchart

- ① Initialization
  - Set the LVR/LVI
    Check the power by LVR/LVI and do not execute under unstable or low power.
  - Initialize User_ID1/2/3
  - Set Flash Sector Address High/Middle/Low to Dummy address
    Dummy address is set to unused area range in flash.
- ② Decide to Write
  - When the Erase/Write are determined, set flag. Do not directly Erase/Write in flash.
  - Make the user data.
- ③ Check and Set User_ID1/2/3
  - In the middle of source, insert code which can check and set the flags.
  - By setting the User_ID 1/2/3 sequentially and identify the flow of the program.
- ④ Set Flash Sector Address
  - Set address to real area range shortly before Erase/Write in flash.
    Set to Dummy address after Erase/Write
    Even if invalid work occurred, it will be Erase/Write in Dummy address in flash.
- ⑤ Check Flags
  - If every flag(User_ID1/2/3, LVI, Flash Address Min/Max) was set, than do Erase/Write.
  - If the Flash Sector Address is outside of Min/Max, do not execute
  - Address Min/Max is set to unused area.
- ⑥ Initialize Flags
  - Initialize User_ID1/2/3
  - Set Flash Sector Address to Dummy Address
- Sample Source
  - Refer to the ABOV homepage.
  - It is created based on the A97C450.
  - Each product should be modified according to the Page Buffer Size and Flash Size

➢ Etc

- Protection by Configure option
  - Set flash protection by MCU Write Tool(OCD, PGM+, etc…)
    Specific Area :
    4KBytes (Address 0000H – 0FFFH)
    2KBytes (Address 0000H – 07FFH)
    1KBytes (Address 0000H – 03FFH)
    512 Bytes (Address 0000H – 01FFH)
  - The range of protection may be different each product.

Table of contents

Revision History **2**

1 Overview **3**

1.1. Description 3
1.2 Ordering information 3
1.3 Features 4
1.5 MTP programming 10
1.5.1 Overview 10
1.5.2 On-Board programming 10
1.5.2.1 Circuit Design Guide 10

2 Block diagram **12**

3 Pin assignment **13**

4 Package Diagram **14**

5 Pin Description **15**

6 Port Structures **17**

6.1 General Purpose I/O Port with Analog Input 17

7 Electrical Characteristics **18**

7.1 Absolute Maximum Ratings 18
7.2 Recommended Operating Conditions 18
7.3 A/D Converter Characteristics 19
7.4 Analog Comparator Characteristics 19
7.5 Power-On Reset Characteristics 19
7.6 Low Voltage Reset and Low Voltage Indicator Characteristics 20
7.7 Internal RC Oscillator Characteristics 20
7.8 Internal Watch-Dog Timer RC Oscillator Characteristics 20
7.9 DC Characteristics 21
7.10 AC Characteristics 22
7.11 SPI Characteristics 23
7.12 USART Characteristics 24
7.13 I2C Characteristics 25
7.14 Data Retention Voltage in Stop Mode 26
7.15 Internal Flash ROM Characteristics 27
7.16 Internal EEPROM Characteristics 27
7.17 Input/Output Capacitance 27
7.18 Main Clock Oscillator Characteristics 28
7.19 Sub Clock Oscillator Characteristics 29
7.20 Operating Voltage Range 30
7.21 Recommended Circuit and Layout 31
7.22 Recommended Circuit and Layout with SMPS Power 31
7.23 Typical Characteristics 32

8 Memory **34**

8.1 Program Memory 34
8.2 Data Memory 36
8.3 External Data Memory 38
8.4 Extended Stack Pointer(XSP) 38
8.5 Memory Extension Stack 39
8.6 Bootloader Swap 41
8.7 SFR Map 42
8.7.1 SFR Map Summary 42
8.7.2 SFR Map 44

9 I/O Ports 48
9.1 I/O Ports 48
9.2 Port Register 48
9.2.1 Data Register (Px) 48
9.2.2 Direction Register (PxIO) 48
9.2.3 Pull-up Resistor Selection Register (PxPU) 48
9.2.4 Open-drain Selection Register (PxOD) 48
9.2.5 De-bounce Enable Register (PxDB) 48
9.2.6 Port Selection Register 1 (PSR1) 48
9.2.7 Port Selection Register 2 (PSR2) 49
9.2.8 Port Debounce Length Selection Register (IODBCNT) 49
9.2.9 Register Map 50
9.3 P0 Port 51
9.3.1 P0 Port Description 51
9.3.2 Register description for P0 51
9.4 P1 Port 53
9.4.1 P1 Port Description 53
9.4.2 Register description for P1 53
9.5 P2 Port 55
9.5.1 P2 Port Description 55
9.5.2 Register description for P2 55
9.6 P3 Port 57
9.6.1 P3 Port Description 57
9.6.2 Register description for P3 57
9.7 P4 Port 59
9.7.1 P4 Port Description 59
9.7.2 Register description for P4 59
9.8 P5 Port 61
9.8.1 P5 Port Description 61
9.8.2 Register description for P5 61
10 Interrupt Controller 64
10.1 Overview 64
10.2 External Interrupt 65
10.3 Block Diagram 66
10.4 Interrupt Vector Table 67
10.5 Interrupt Sequence 68
10.6 Effective Timing after Controlling Interrupt Bit 69
10.7 Multi Interrupt 70
10.8 Interrupt Enable Accept Timing 71
10.9 Interrupt Service Routine Address 71
10.10 Saving/Restore General-Purpose Registers 71
10.11 Interrupt Timing 72
10.12 Interrupt Register Overview 73
10.12.1 Interrupt Enable Register (IE, IE1, IE2, IE3, IE4) 73
10.12.2 10.12.2 Interrupt Priority Register (IP0H/L, IP1H/L, IP2H/L, IP3H/L, IP4H/L) 73
10.12.3 External Interrupt Flag Register (EIFLAG) 73
10.12.4 External Interrupt Edge Register (EIEDGE) 73
10.12.5 External Interrupt Polarity Register (EIPOLA) 73
10.12.6 External Interrupt Both Edge Enable Register (EIBOTH) 73
10.12.7 External Interrupt Enable Register (EIENAB) 73
10.12.8 Register Map 74
10.12.9 Interrupt Register Description 74
10.12.10 Register Description for Interrupt 75
11 Peripheral Hardware 82
11.1 Stack Pointer Management Unit 82

11.1.1 Overview 82
11.1.2 Register Map 83
11.1.3 Register Description for SPMU 83
11.2 Clock Generator 87
11.2.1 Overview 87
11.2.2 Block Diagram 87
11.2.3 CPU clock and Peripherals Clock 88
11.2.4 Register Map 89
11.2.5 Clock Generator Register Description 89
11.2.6 Register Description for Clock Generator 90
11.3 Basic Interval Timer 91
11.3.1 Overview 91
11.3.2 Block Diagram 91
11.3.3 Register Map 91
11.3.4 Basic Interval Timer Register Description 92
11.3.5 Register Description for Basic Interval Timer 92
11.4 Watch Dog Timer 93
11.4.1 Overview 93
11.4.2 WDT Interrupt Timing Waveform 93
11.4.3 Block Diagram 94
11.4.4 Register Map 94
11.4.5 Watch Dog Timer Register Description 94
11.4.6 Register Description for Watch Dog Timer 95
11.5 Watch Timer 96
11.5.1 Overview 96
11.5.2 Block Diagram 96
11.5.3 Register Map 97
11.5.4 Watch Timer Register Description 97
11.5.5 Register Description for Watch Timer 97
11.6 Timer 0/1 99
11.6.1 Overview 99
11.6.2 8-Bit Timer/Counter Mode 100
11.6.3 8-Bit PWM Mode 101
11.6.4 8-Bit Capture Mode 103
11.6.5 Block Diagram 105
11.6.6 Register Map 105
11.6.7 Timer/Counter 0 Register Description 105
11.6.8 Register Description for Timer/Counter 0/1 106
11.7 Timer 2/3 110
11.7.1 Overview 110
11.7.2 16-bit Timer/Counter Mode 110
11.7.3 16-bit Capture Mode 112
11.7.4 16-bit PPG Mode 114
11.7.5 Block Diagram 116
11.7.6 Register Map 116
11.7.7 Timer/Counter 2/3 Register Description 117
11.7.8 Register Description for Timer/Counter 2/3 117
11.8 Timer 4/5/6 120
11.8.1 Overview 120
11.8.2 16-bit Timer/Counter Mode 121
11.8.3 16-bit Capture Mode 123
11.8.4 16-bit PPG Mode 125
11.8.5 Block Diagram 127
11.8.6 Register Map 127
11.8.7 Timer/Counter 4/5/6 Register Description 128
11.8.8 Register Description for Timer/Counter 4/5/6 128
11.9 Buzzer Driver 131
11.9.1 Overview 131

11.9.2 Block Diagram 131
11.9.3 Register Map 132
11.9.4 Buzzer Driver Register Description 132
11.9.5 Register Description for Buzzer Driver 132
11.10 USART 133
11.10.1 Overview 133
11.10.2 Block Diagram 134
11.10.3 Clock Generation 135
11.10.4 External Clock (XCK) 136
11.10.5 Synchronous mode Operation 136
11.10.6 Data format 137
11.10.7 Parity bit 137
11.10.8 USART Transmitter 138
11.10.8.1 Sending Tx data 138
11.10.8.2 Transmitter flag and interrupt 138
11.10.8.3 Parity Generator 138
11.10.8.4 Disabling Transmitter 139
11.10.9 USART Receiver 139
11.10.9.1 Receiving Rx data 139
11.10.9.2 Receiver flag and interrupt 139
11.10.9.3 Parity Checker 140
11.10.9.4 Disabling Receiver 140
11.10.9.5 Asynchronous Data Reception 140
11.10.10 SPI Mode 142
11.10.10.1 SPI Clock formats and timing 142
11.10.11 Register Map 144
11.10.12 USART Register Description 144
11.10.13 Register Description for USART 145
11.10.14 Baud Rate setting (example) 151
11.10.15 0% Error Baud Rate 152
11.11 SPI 153
11.11.1 Overview 153
11.11.2 Block Diagram 153
11.11.3 Data Transmit / Receive Operation 154
11.11.4 SS pin function 154
11.11.5 Timing Waveform 155
11.11.6 Register Map 155
11.11.7 SPI Register description 156
11.11.8 Register Description for SPI 156
11.12 I2C 158
11.12.1 Overview 158
11.12.2 Block Diagram 158
11.12.3 I2C bit Transfer 159
11.12.4 Start / Repeated Start / Stop 159
11.12.5 Data Transfer 160
11.12.6 Acknowledge 160
11.12.7 Synchronization / Arbitration 161
11.12.8 Operation 162
11.12.8.1 Master Transmitter 162
11.12.8.2 Master Receiver 163
11.12.8.3 Slave Transmitter 165
11.12.8.4 Slave Receiver 165
11.12.9 Register Map 167
11.12.10 I2C Register Description 167
11.12.11 Register Description for I2C 168
11.13 12-bit A/D Converter 172
11.13.1 Overview 172
11.13.2 Block Diagram 173
11.13.3 ADC Operation 174
11.13.4 Register Map 175

11.13.5 ADC Register Description 175
11.13.6 Register Description for ADC 176
11.14 HDMI™- CEC Controller 179
11.14.1 Introduction 179
11.14.2 HDMI™-CEC register map 179
11.14.3 HDMI-CEC Registers description 180
11.15 IR Controller 187
11.15.1 Introduction 187
11.15.2 IR Controller Register map 187
11.15.3 IR Controller Registers description 187
11.16 Comparator 192
11.16.1 Overview 192
11.16.2 Block Diagram 192
11.16.3 Register Map 192
11.16.4 Comparator Register description 193
11.17 RTC (Real-time Clock) 196
11.17.1 Overview 196
11.17.2 Block Diagram 196
11.17.3 RTC Operation 197
11.17.4 Shifting to Power-Down mode after starting operation 198
11.17.5 Reading/writing RTC 199
11.17.6 Setting alarm of RTC 200
11.17.7 Error Correction of RTC 201
11.17.8 Register Map 203
11.17.9 RTC Register description 203
11.18 BISC 211
11.18.1 Overview 211
11.18.2 Register Map 212
11.18.3 BISC Register description 212
11.19 CRC 213
11.19.1 Overview 213
11.19.2 Block Diagram 213
11.19.3 Register Map 214
11.19.4 CRC Register description 214
11.19.5 CRC Polynomial 216

12 Power Down Operation 217

12.1 Overview 217
12.2 Peripheral Operation in IDLE/STOP Mode 217
12.3 IDLE, SLEEP1 mode 218
12.4 STOP mode (STOP1, STOP2, SLEEP2) 218
12.5 Release Operation of Power-Down Mode 220
12.5.1 Register Map 221
12.5.2 Power Down Operation Register description 221
12.5.3 Register description for Power Down Operation 221

13 RESET 222

13.1 Overview 222
13.2 Reset source 222
13.3 Block Diagram 222
13.4 Power ON RESET 223
13.5 External RESETB Input 226
13.6 Low Voltage Reset and Low Voltage Indicator Processor 227
13.6.1 Register Map 228
13.6.2 Reset Operation Register description 228
13.6.3 Register description for Reset Operation 229

14 On-chip Debug System 231

14.1 Overview 231
14.1.1 Description 231
14.1.2 Feature 231
14.2 Two-Pin External Interface 232
14.2.1 Basic Transmission Packet 232
14.2.2 Packet Transmission Timing 233
14.2.2.1 Data Transfer 233
14.2.3 Connection of Transmission 235

15 Memory Programming 236

15.1 Overview 236
15.1.1 Description 236
15.1.2 Features 236
15.2 Flash Control and status register 236
15.2.1 Register Map 236
15.2.2 Register description for Flash 237
15.3 Memory map 242
15.3.1 Flash Memory Map 242
15.4 Serial In-System Program Mode 243
15.4.1 Flash operation 243
15.4.1.1 Flash Read 244
15.4.1.2 Enable program mode 245
15.4.1.3 Flash write mode 245
15.4.1.4 Flash page erase mode 245
15.4.1.5 Flash bulk erase mode 246
15.4.1.6 Flash OTP area read mode 246
15.4.1.7 Flash OTP area write mode 246
15.4.1.8 Flash OTP area erase mode 247
15.4.1.9 Flash program verify mode 247
15.4.1.10 OTP program verify mode 247
15.4.1.11 Flash erase verify mode 247
15.4.1.12 Flash page buffer read 247
15.4.2 Data EEPROM operation 247
15.4.2.1 Data EEPROM Read 248
15.4.2.2 Enable program mode 248
15.4.2.3 EEPROM write mode 248
15.4.2.4 EEPROM page erase mode 249
15.4.2.5 EEPROM bulk erase mode 249
15.4.2.6 Data EEPROM program verify mode 249
15.4.2.7 Data EEPROM erase verify mode 249
15.4.2.8 Data EEPROM page buffer read 249
15.4.3 Summary of Flash and Data EEPROM Program/Erase Mode 250
15.5 Parallel Mode 251
15.5.1 Overview 251
15.5.2 Parallel Mode instruction format 252
15.5.3 Parallel Mode timing diagram 253
15.6 Mode entrance method of ISP and byte-parallel mode 254
15.6.1 Mode entrance method for ISP 254
15.6.2 Mode entrance of Byte-parallel 254
15.7 Security 255

16 Configure option 256

16.1 Configure option Control Register 256

17 APPENDIX 258

17.1 Instruction Table 258
17.2 Flash Protection for Invalid Erase/Write 262

Table of contents 265