ABOV SEMICONDUCTOR CO., Ltd.

8-BIT SINGLE-CHIP
MICROCONTROLLERS

A96T336

USER MANUAL

FAE team

REVISION HISTORY

VERSION	COMMENT	DATE
0.1	First Release	2015-11-25
0.2	SFR & XSFR Map updated	2016-04-05
0.3	Self-Touch Wakeup inserted	2016-05-02
0.4	Update PSR information	2016-05-18
0.5	Update Dimming PAD	2016-05-20
0.6	Delete Touch ADC information	2016-08-10
0.7	Correct RC-OSC in 7.1, 7.6 Characteristics	2016-08-16
0.8	Update FECON information	2016-10-01
0.9	T336 information modified. Package modified. XSFR Map updated (2EE0~2EEF)	2016-11-03

Published by FAE team

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

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

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

Table of Contents

List of Figures

A96T336

CMOS SINGLE-CHIP 8-BIT MICROCONTROLLER WITH 16CH TOUCH SWITCH

1. Overview

1.1 Description

The A96T336 is an advanced CMOS 8-bit microcontroller with 16K bytes of FLASH. This is powerful microcontroller which provides a highly flexible and cost effective solution to many embedded control applications. This provides the following features : 16CH self-capacitive Touch switch, 16K bytes of FLASH, 256 bytes of SRAM, 1792 bytes of XRAM, 16-bit timer/counter, watchdog timer, I^2C, USART, on-chip POR, BOD, LVI and 16-bit PWM output, on-chip oscillator and clock circuitry. The A96T336 also supports power saving modes to reduce power consumption.

Device name	ROM size	SRAM size	Package
A96T336LU	16Kbytes FLASH	I:256 bytes X:1792 bytes	24 QFN
A96T336LD			24 SOP
A96T336FD			20 SOP
A96T336AU			16 QFN
A96T336AD			16 SOP
A96T336EW			12 WLCSP
A96T336HD			8 SOP

1.2 Features

- **CPU**
  - 8 Bit CISC Core(8051 Compatible,2 clock per cycle)
- **16K Bytes On-chip FLASH**
  - Endurance : 10,000 times
  - Retention : 10 years
- **256 Bytes SRAM**
- **1792 Bytes XRAM**
- **General Purpose I/O**
  - 22 Ports (P0[7:0], P1[5:0], P2[7:0]) : 24 Pin Package
  - 31 step current dimming Ports (P0[3:1])
- **16-Ch SelfCapacitive Touch Switch**
  - Fast Initial Self-Calibration.
  - Fine Tuning for the Best Sensibility by 256-step.
  - Key Detection Mode : Single / Multi-Mode
  - 16-bit Sensing Resolutions
  - Support Self-Wakeup at Power down (CS0~CS15)
- **16-Ch 10-bit AD Converter**
- **Basic Interval Timer**
- **Timer/ Counter**
  - 16Bit×2Ch
- **10Ch 16-bit PWM (using Timer0, 1)**
- **Watch Dog Timer**
- **32-bit Free Run Counter**
- **I^2C with 1.8V Interface**
- **USART**
- **Interrupt Sources**
  - External (2)
  - TOUCH (1)
  - Free Run Counter (1)
  - I^2C (1)
  - USART (2)
  - Timer (2)
  - LVI (1)
  - WDT (1)
  - BIT (1)
  - ROM (1)
- **On-Chip RC-Oscillator**
  - 16MHz OSC (±3%@-40~+85℃)
- **On-Chip WDT-Oscillator**
  - 256kHz OSC (±10%@-40~+85℃)
- **Power On Reset & Brown-Out Detector**
  - 1.4V (POR)
  - 1.6V (BOD)
- **Programmable Low-Voltage Indicator**
  - 3-Level (2.5V / 3.6V / 4.2V)
- **Minimum Instruction Execution Time**
  - 200ns (@10MHz, NOP Instruction)
- **Power down mode**
  - IDLE, STOP1, STOP2 mode
- **Operating Frequency**
  - 2, 4, 8, 16MHz (internal RC oscillator)
- **Operating Voltage**
  - 2.7V ~ 5.5V (@ 16MHz)
- **Operating Temperature : -40 ~ +85℃**
- **Package Type**
  - 24 QFN, SOP
  - 20 SOP
  - 16 QFN, SOP
  - 12 WLCSP
  - 8 SOP

1.3 Ordering Information

Table 1-1 Ordering Information of A96T336

Device name	ROM size	SRAM size	Package
A96T336LU	16Kbytes FLASH	I:256 bytes X:1792 bytes	24 QFN
A96T336LD			24 SOP
A96T336FD			20 SOP
A96T336AU			16 QFN
A96T336AD			16 SOP
A96T336EW			12 WLCSP
A96T336HD			8 SOP

1.4 Development Tools

1.4.1 Compiler

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

The A96T336 core is Mentor 8051. Anyway, device ROM size is smaller than 64KB.Developer can use all kinds of third party's standard 8051 compiler.

1.4.2 OCD Emulator and Debugger

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

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

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

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

Connection:

- P10 (A96T336 DSDA pin)
- P11 (A96T336 DSCL pin)

OCD connector diagram: Connect OCD and user system

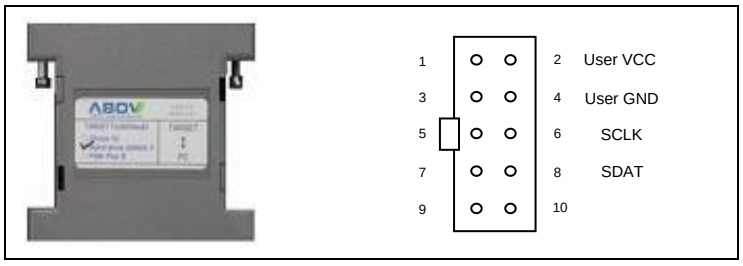


Figure 1-1 OCD Debugger and Pin Description

1.4.3 Programmer

Single programmer:

PGMplus USB: It programs MCU device directly.

Figure 1-2 Single Programmer

OCD emulator: It can write code in MCU device too.

Because of, OCD debugging supports ISP (In System Programming).

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

Gang programmer:

It programs 8 MCU devices at once.

So, it is mainly used in mass production line.

Gang programmer is standalone type, it means it does not require host PC.

Figure 1-3 Gang Programmer

2. Block Diagram

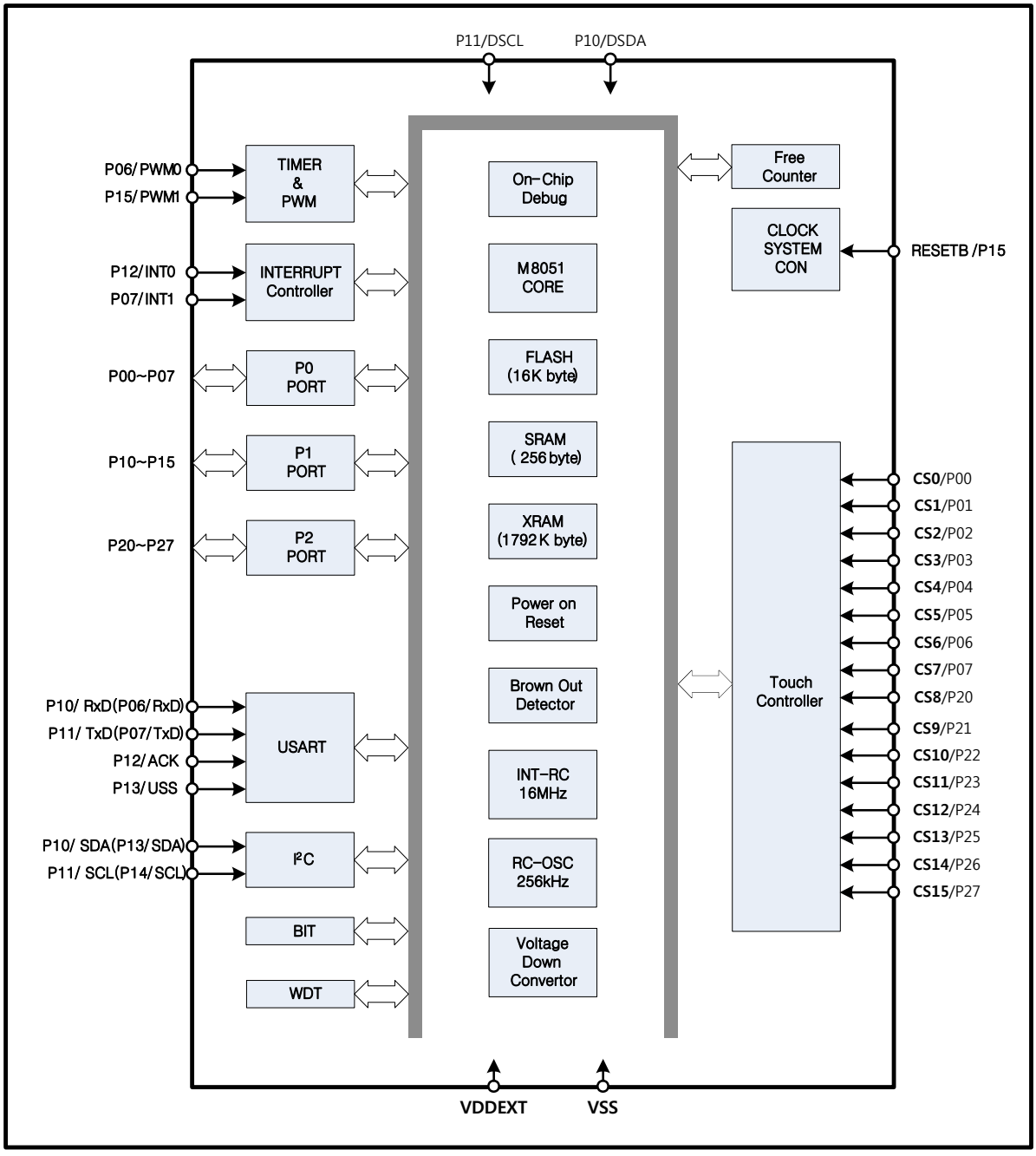


Figure 2-1 A96T336 Block Diagram

3. Pin Assignment

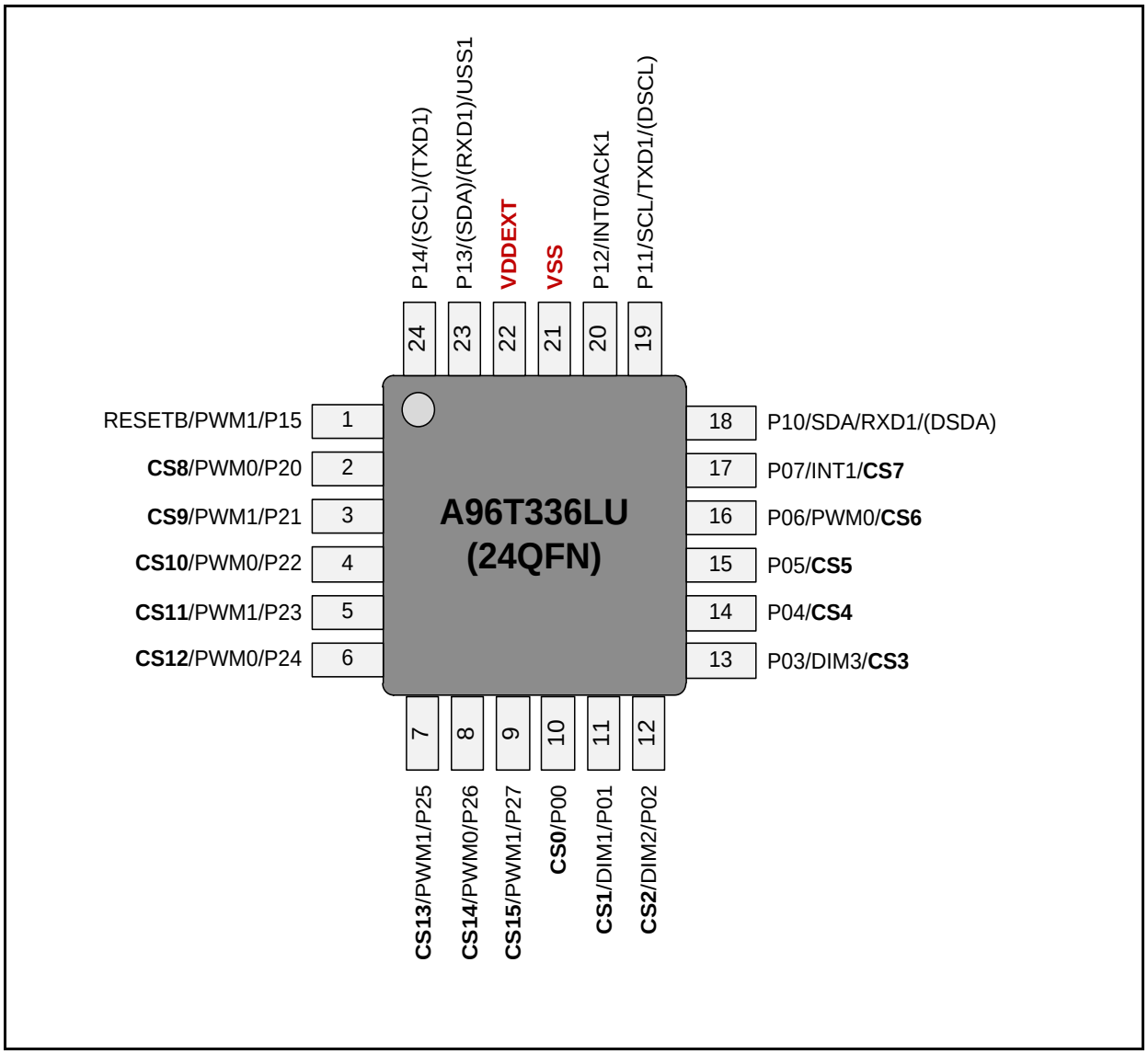


Figure 3-1 24-QFN Pin Assignment

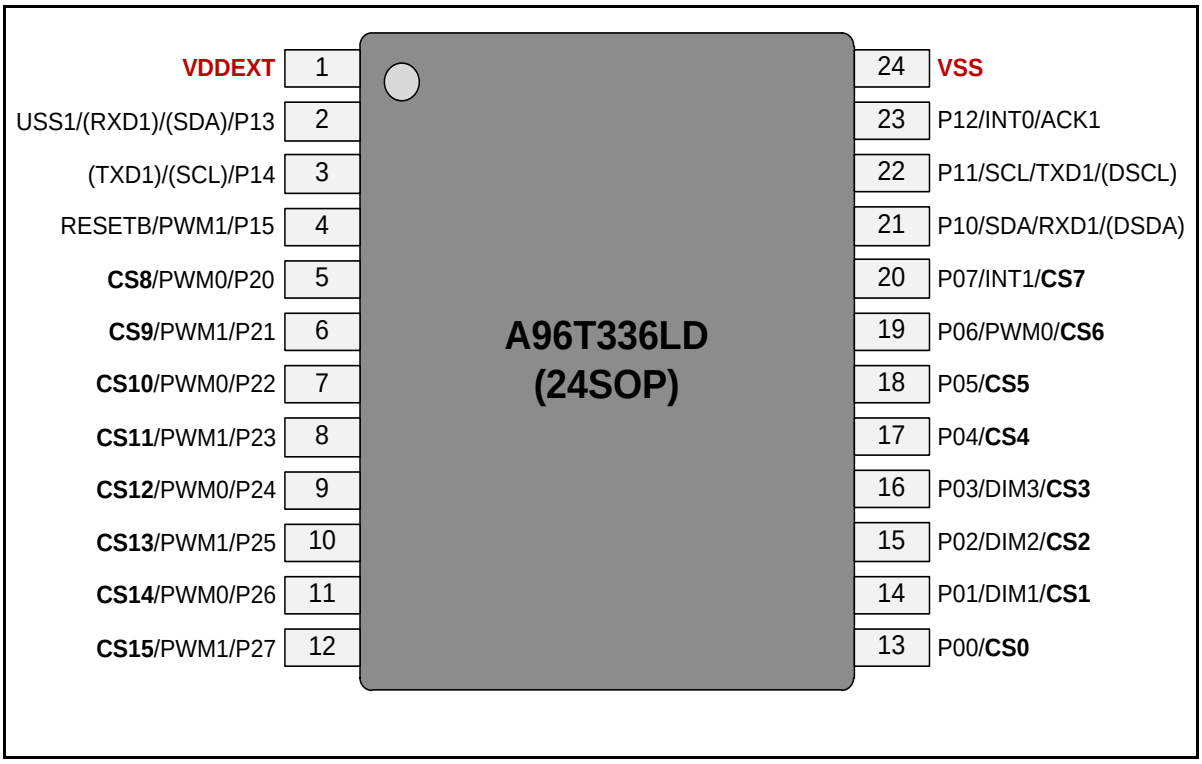


Figure 3-2 24-SOP Pin Assignment

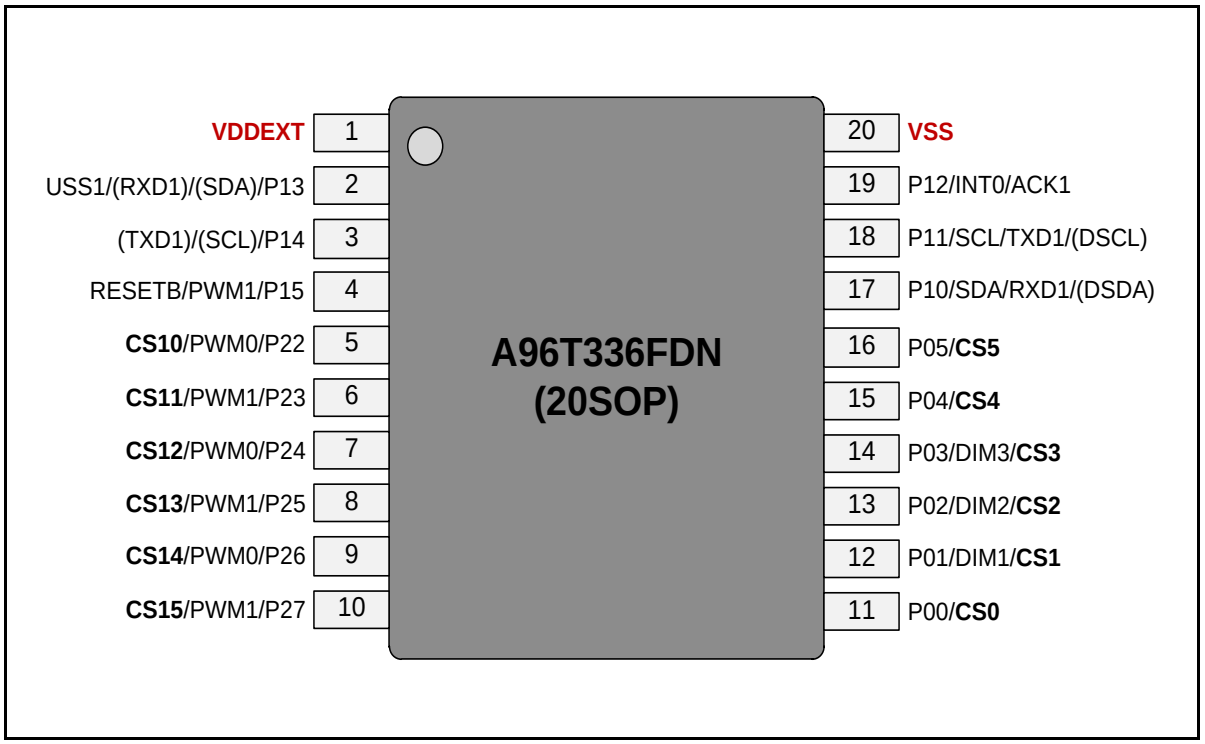


Figure 3-3 20 SOP Pin Assignment

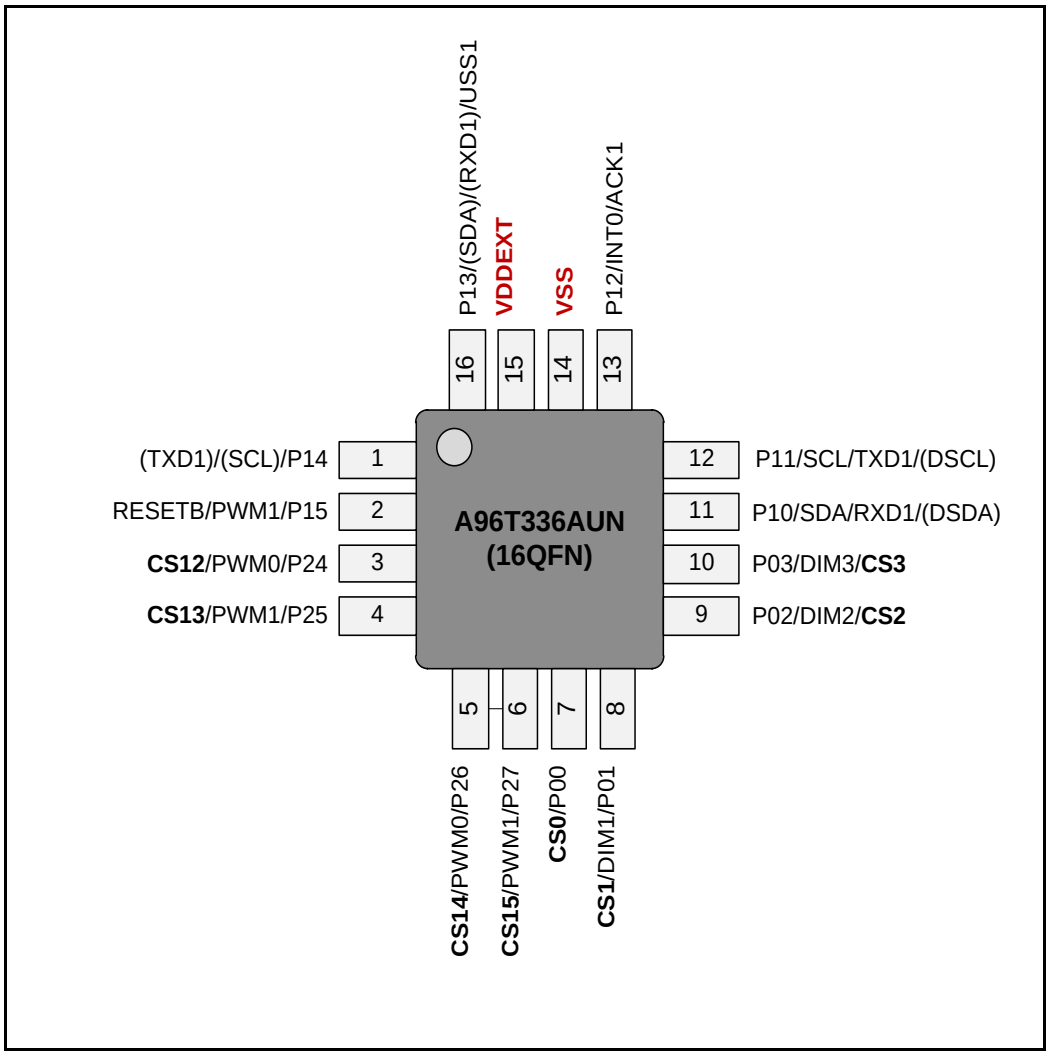


Figure 3-4 16 QFN Pin Assignment

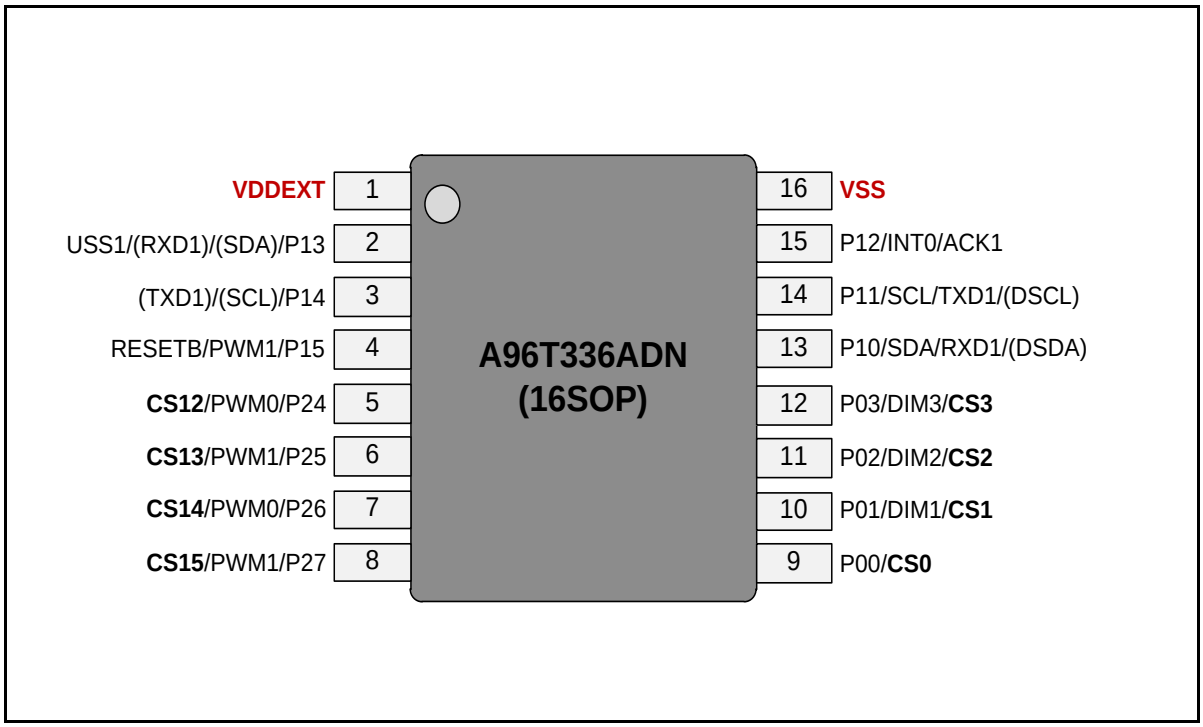


Figure 3-5 16 SOP Pin Assignment

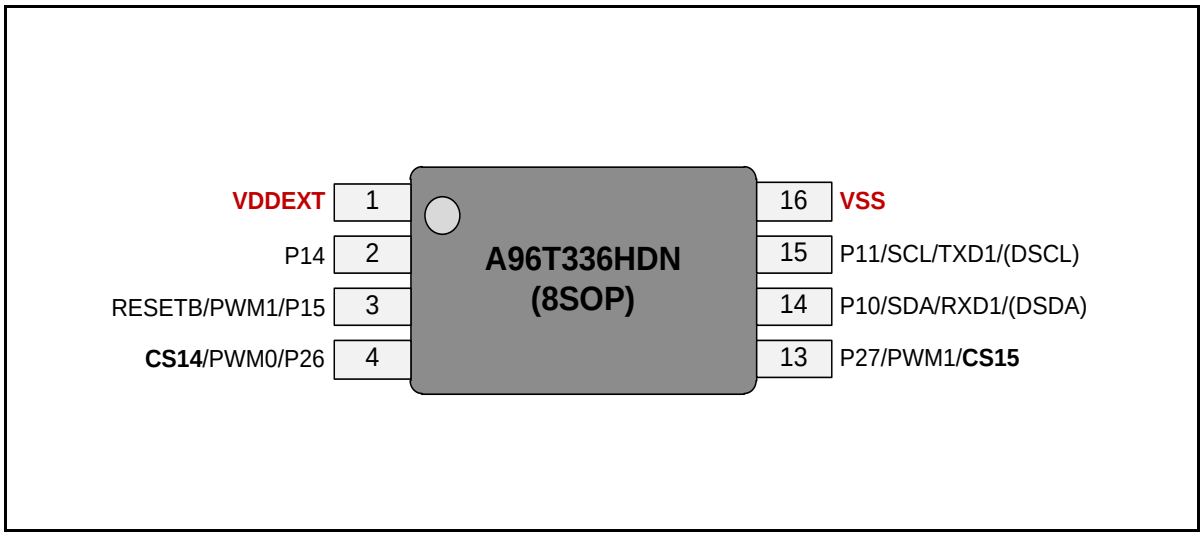
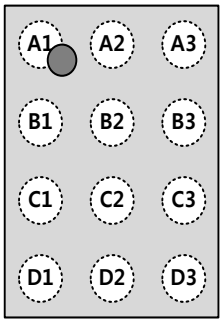


Figure 3-6 8 SOP Pin Assignment

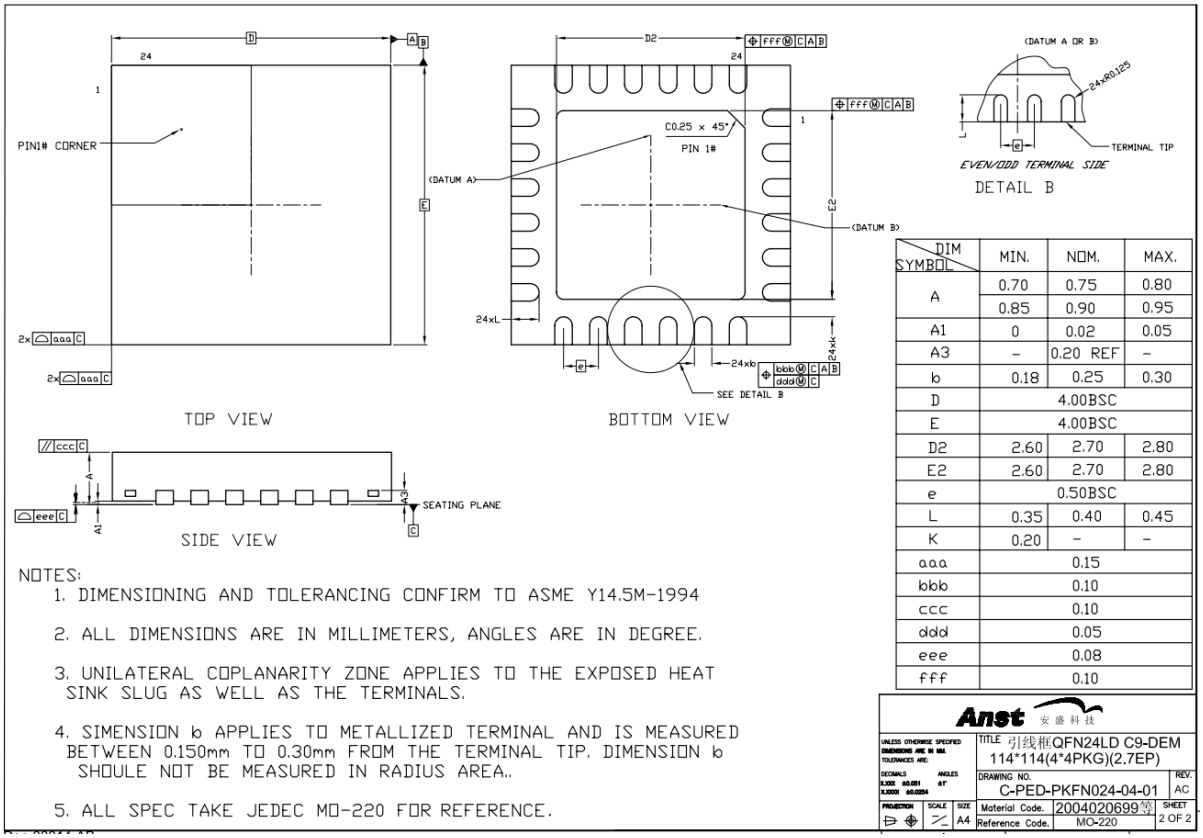


Top View

	1	2	3
A	P12/INT0/ACK1	VSS	VDDEXT
B	P11/SCL/TXD1/(DSCL)	P03/DIM3/CS3	P13/(SDA)/(RXD1)/USS1
C	P10/SDA/RXD1/(DSDA)	P02/DIM2/CS2	P14/(SCL)/(TXD1)
D	P01/DIM1/CS1	P27/PWM1/CS15	P26/PWM0/CS14

Figure 3-7 12 WLCSP Pin Assignment

4. Package Diagram



DIM / SYMBOL	MIN.	NOM.	MAX.
A	0.70	0.75	0.80
	0.85	0.90	0.95
A1	0	0.02	0.05
A3	–	0.20 REF	–
b	0.18	0.25	0.30
D	4.00BSC		
E	4.00BSC		
D2	2.60	2.70	2.80
E2	2.60	2.70	2.80
e	0.50BSC		
L	0.35	0.40	0.45
K	0.20	–	–
aaa	0.15		
bbb	0.10		
ccc	0.10		
ddd	0.05		
eee	0.08		
fff	0.10		

Anst 安盛科技

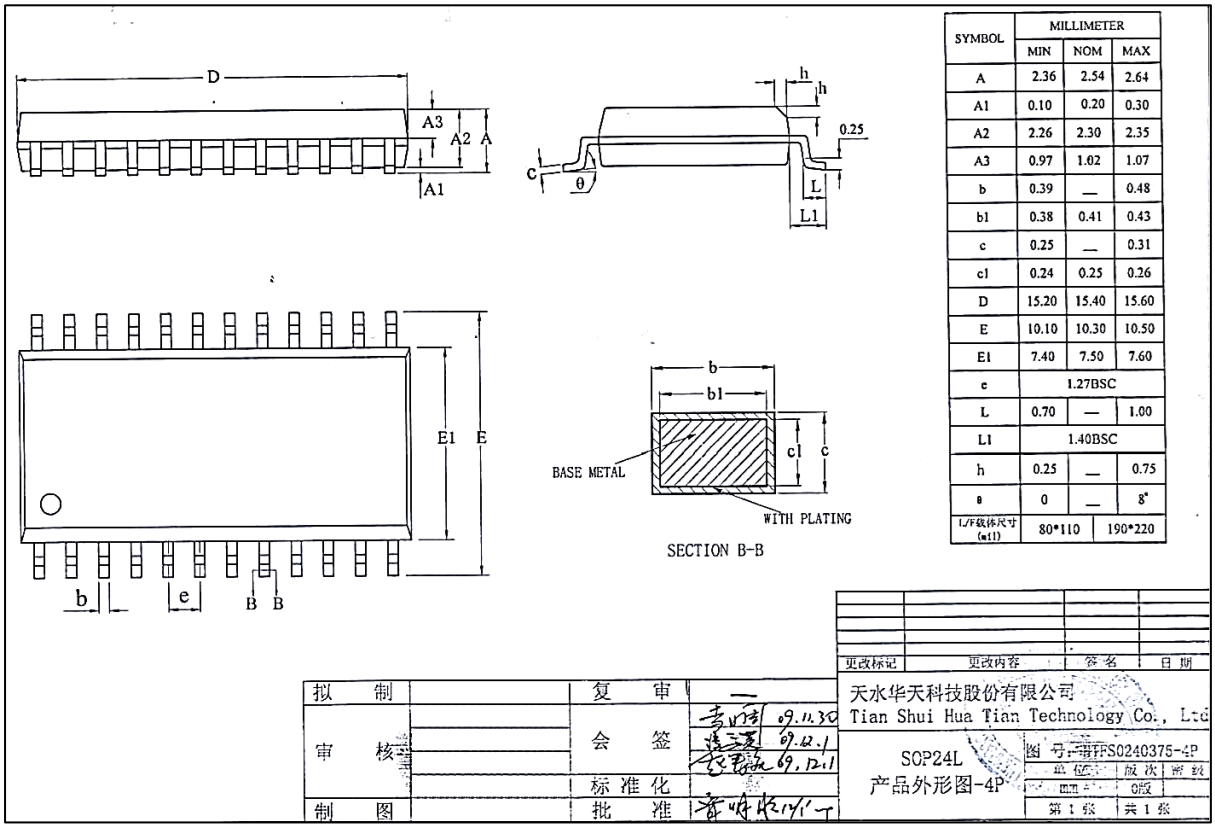
TITLE 引线框QFN24LD C9-DEM 114*114(4*4PKG)(2.7EP)

DRAWING NO. C-PED-PKFN024-04-01 REV. AC

Material Code. 2004020699等 SHEET 2 OF 2

Reference Code. MO-220

Figure 4-1 24-pin QFN Package



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	2.36	2.54	2.64
A1	0.10	0.20	0.30
A2	2.26	2.30	2.35
A3	0.97	1.02	1.07
b	0.39	—	0.48
b1	0.38	0.41	0.43
c	0.25	—	0.31
c1	0.24	0.25	0.26
D	15.20	15.40	15.60
E	10.10	10.30	10.50
E1	7.40	7.50	7.60
e	1.27BSC		
L	0.70	—	1.00
L1	1.40BSC		
h	0.25	—	0.75
θ	0	—	8°
上/下载体尺寸 (mil)	80*110		190*220

更改标记	更改内容	签名	日期

拟制		复审	—
审核		会签	09.11.30 / 09.12.1 / 09.12.1
制图		标准化	
		批准	

天水华天科技股份有限公司
Tian Shui Hua Tian Technology Co., Ltd

SOP24L
产品外形图-4P

图号: TFS0240375-4P
单位: mm | 版次: 0版 | 密级:
第 1 张 | 共 1 张

Figure 4-2 24-pin SOP Package

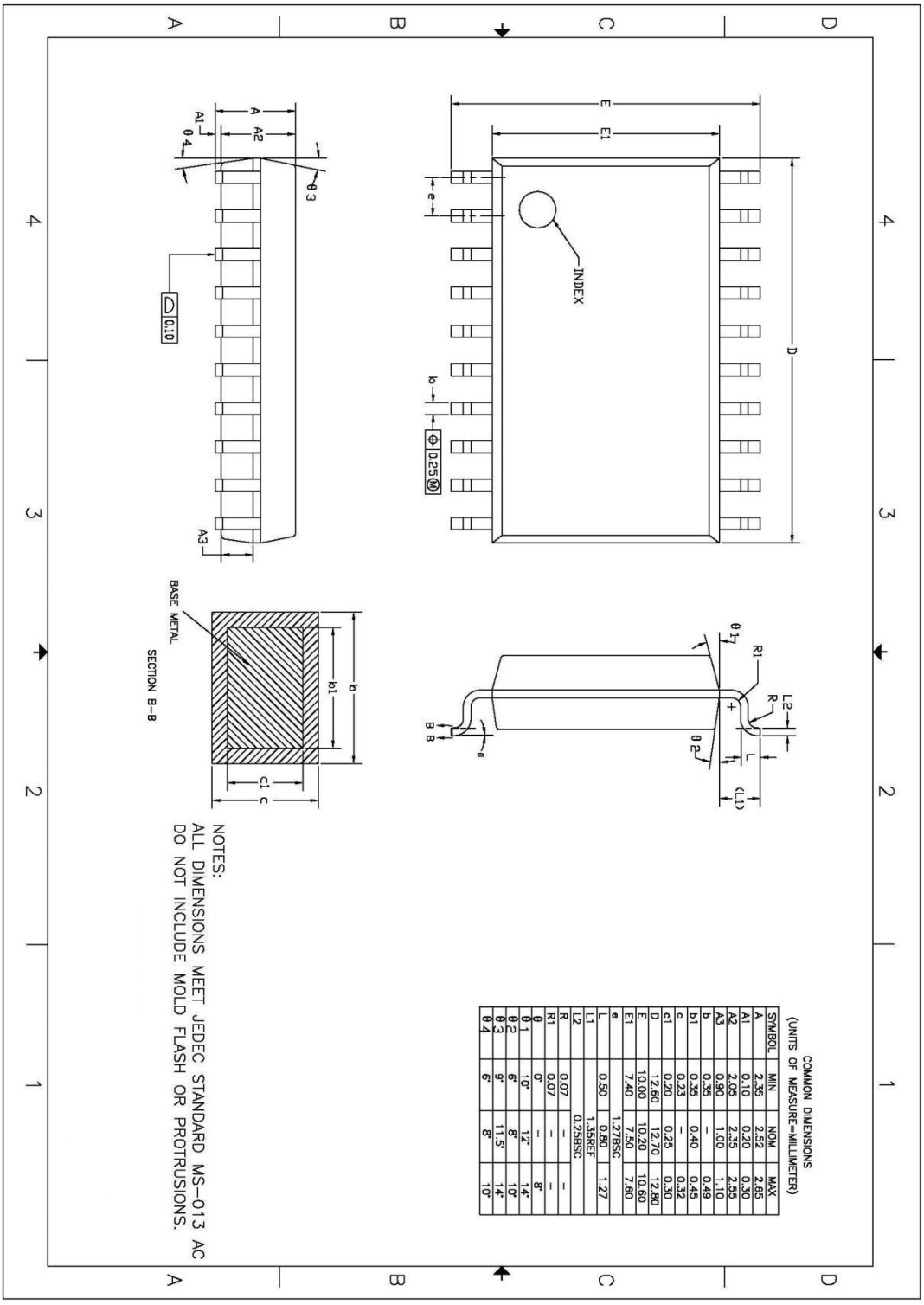
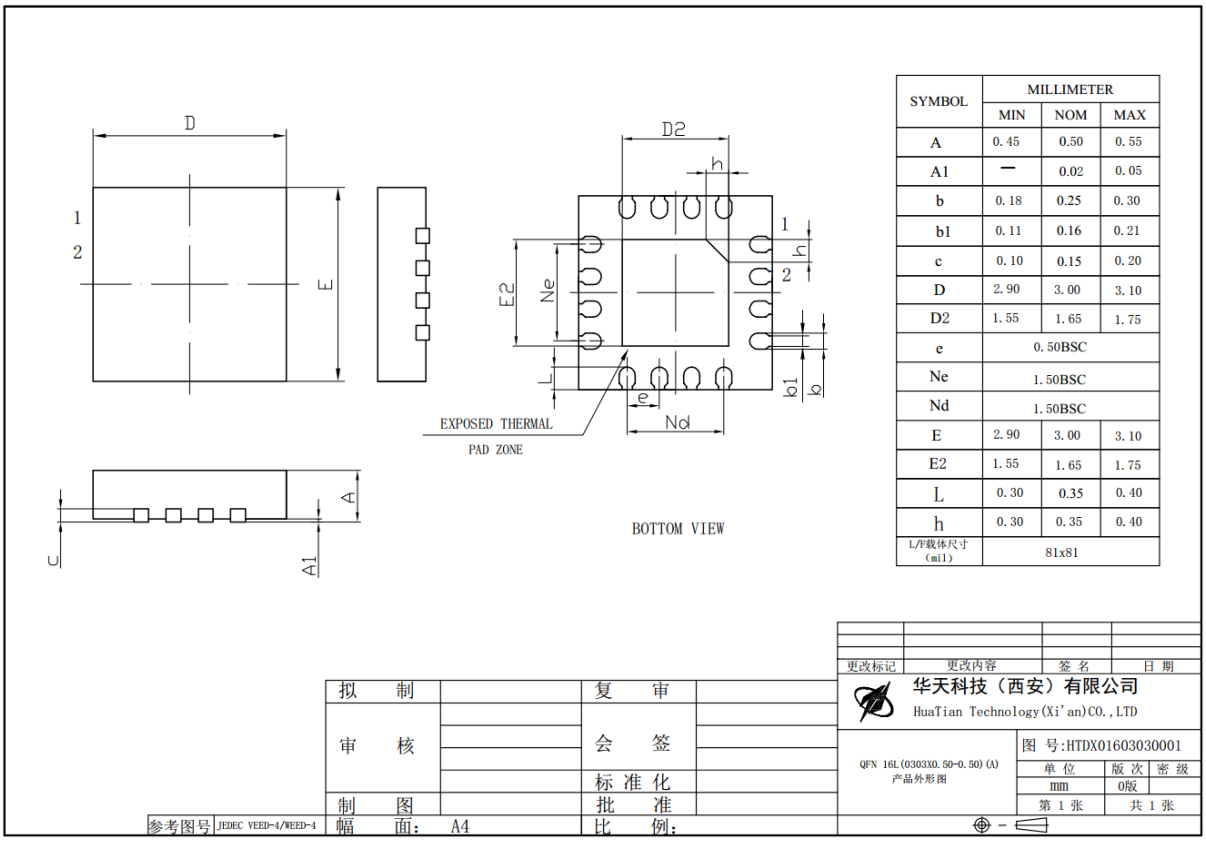


Figure 4-3 20-pin SOP Package



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	0.45	0.50	0.55
A1	—	0.02	0.05
b	0.18	0.25	0.30
b1	0.11	0.16	0.21
c	0.10	0.15	0.20
D	2.90	3.00	3.10
D2	1.55	1.65	1.75
e	0.50BSC		
Ne	1.50BSC		
Nd	1.50BSC		
E	2.90	3.00	3.10
E2	1.55	1.65	1.75
L	0.30	0.35	0.40
h	0.30	0.35	0.40
L/F载体尺寸 (mil)	81x81		

更改标记	更改内容	签 名	日 期

拟 制		复 审	
审 核		会 签	
		标准化	
制 图		批 准	
参考图号 JEDEC VEED-4/WEED-4	幅 面： A4	比 例：	

华天科技（西安）有限公司
HuaTian Technology(Xi'an)CO.,LTD

QFN 16L(0303X0.50-0.50)(A)
产品外形图

图 号:HTDX01603030001		
单 位	版 次	密 级
mm	0版	
第 1 张	共 1 张	

Figure 4-4 16-pin QFN Package

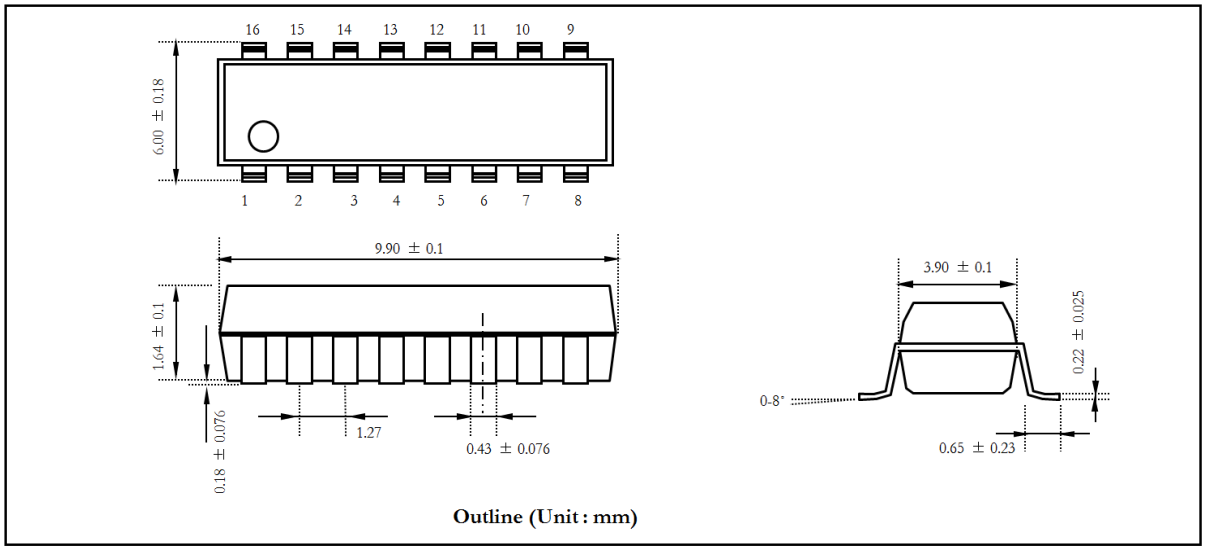


Figure 4-5 16-pin SOP(150MIL) Package

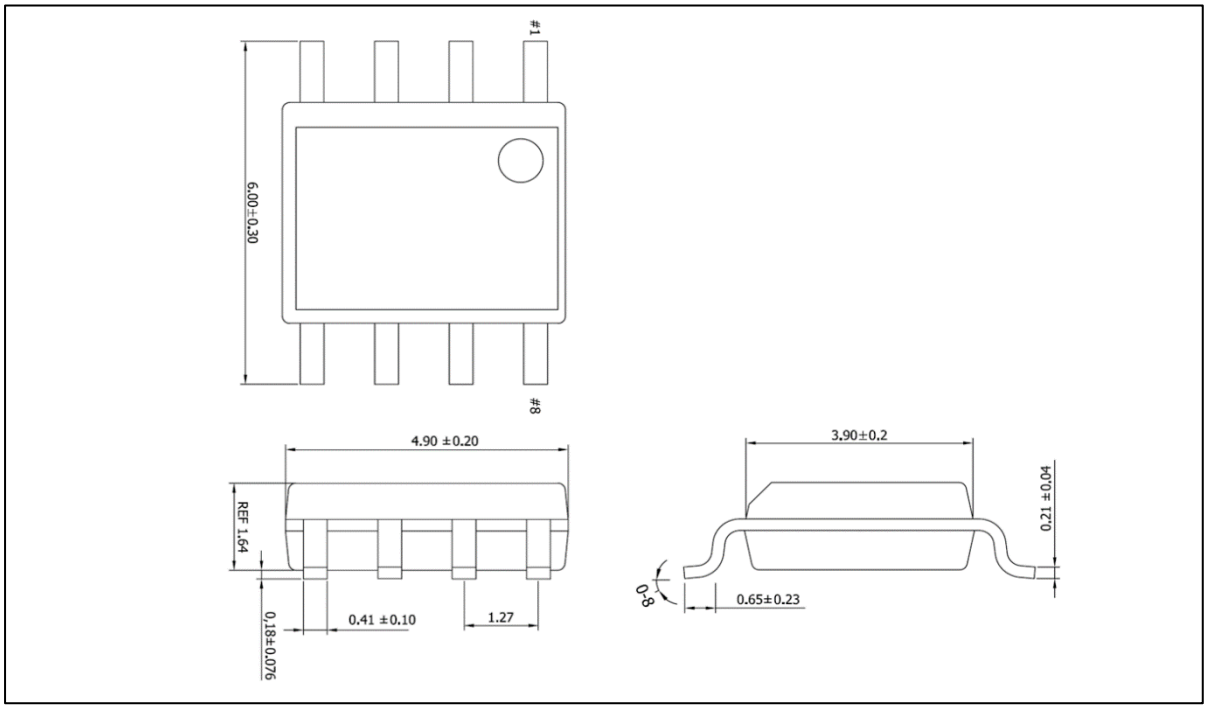


Figure 4-6 8-pin SOP(150MIL) Package

5. Pin Description

Table 5-1 Normal Pin description

PIN Name		I/O	Function	@RESET	Shared with
16PIN	24PIN				
P00	P00	I/O	Port P0 8-Bit I/O Port Can be set in input or output mode in 1-bit units Internal pull-up register can be used via software when this port is used as input port Open Drain enable register can be used via software when this port is used as output port 3-pin LED Dimming Pins : P0[3:1]	Input	AN0/CS0
P01	P01				DIM1/AN1/CS1
P02	P02				DIM2/AN2/CS2
P03	P03				DIM3/AN3/CS3
-	P04				AN4/CS4
-	P05				AN5/CS5
-	P06				PWM0/AN6/CS6
-	P07				INT1/AN7/CS7
P10	P10	I/O	Port P1 6-Bit I/O Port Can be set in input or output mode in 1-bit units Internal pull-up register can be used via software when this port is used as input port Open Drain enable register can be used via software when this port is used as output port	Input	SDA/RXD1/(DSDA)
P11	P11				SCL/TXD1/(DSCL)
P12	P12				INT0/ACK
P13	P13				USS1/(RXD1)/(SDA)
P14	P14				(TXD1)/(SCL)
P15	P15				RESETB/PWM1
-	P20	I/O	Port P2 8-Bit I/O Port Can be set in input or output mode in 1-bit units Internal pull-up register can be used via software when this port is used as input port Open Drain enable register can be used via software when this port is used as output port	Input	PWM0/AN8/CS8
-	P21				PWM1/AN9/CS9
-	P22				PWM0/AN10/CS10
-	P23				PWM1/AN11/CS11
P24	P24				PWM0/AN12/CS12
P25	P25				PWM1/AN13/CS13
P26	P26				PWM0/AN14/CS14
P27	P27				PWM1/AN15/CS15

6. Port Structures

6.1 General Purpose I/O Port

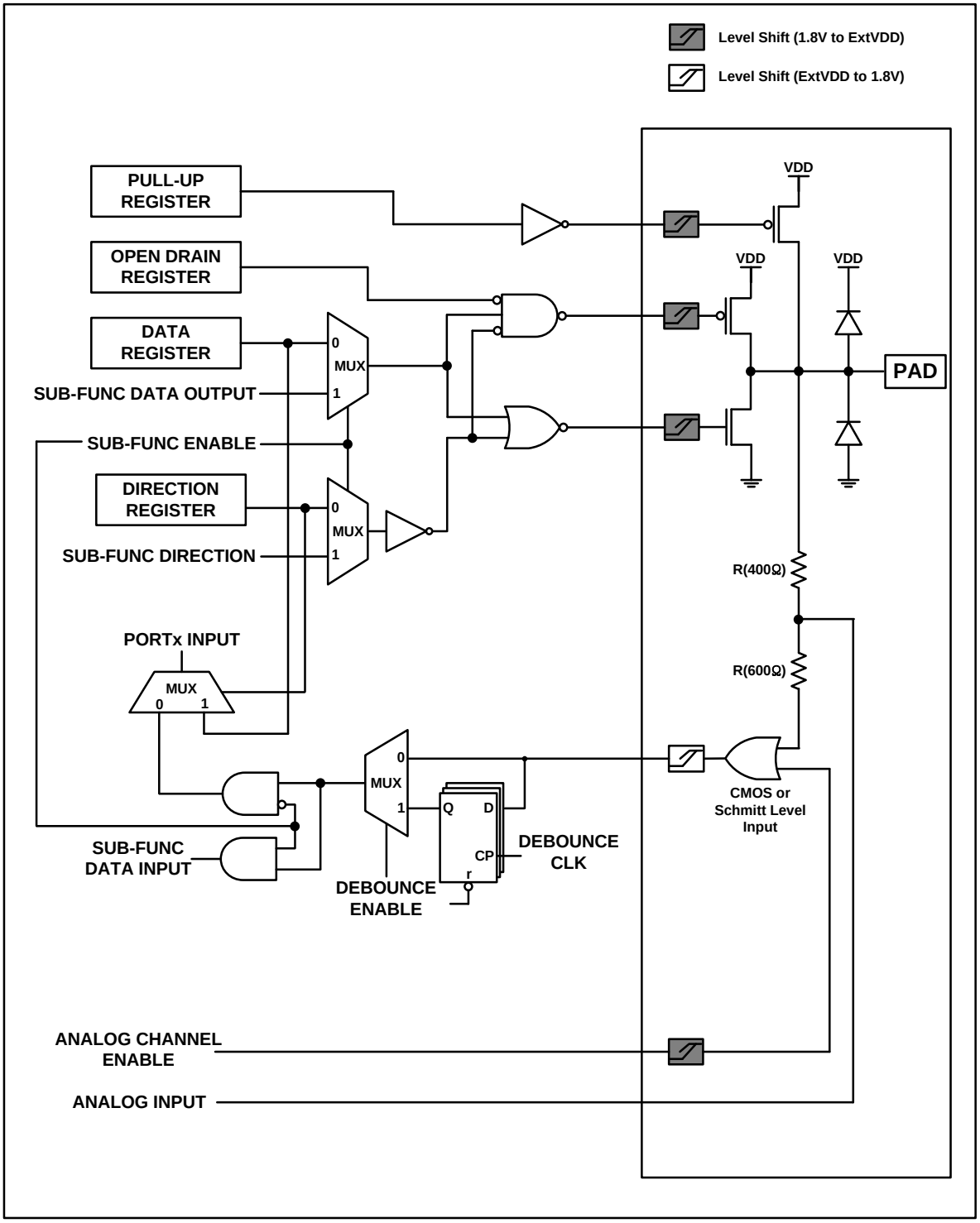


Figure 6-1 General Purpose I/O Port

6.2 General Purpose I/O Port with 1.8V Interface (P10, P11)

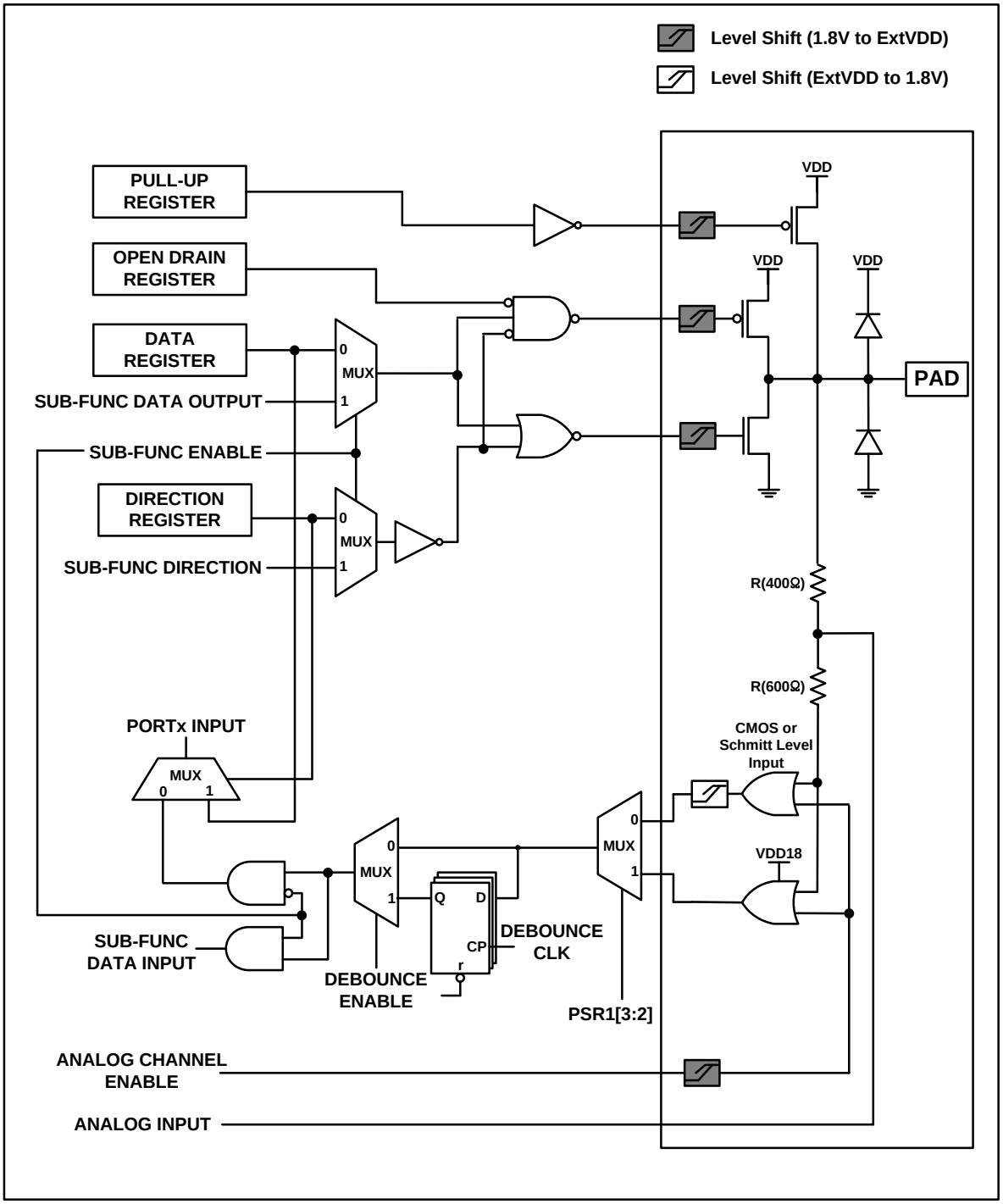


Figure 6-2 General Purpose I/O Port with 1.8V Interface (SDA/P10, SCL/P11)

6.3 Open-Drain I/O Port with 1.8V Interface (P12, P13, P14)

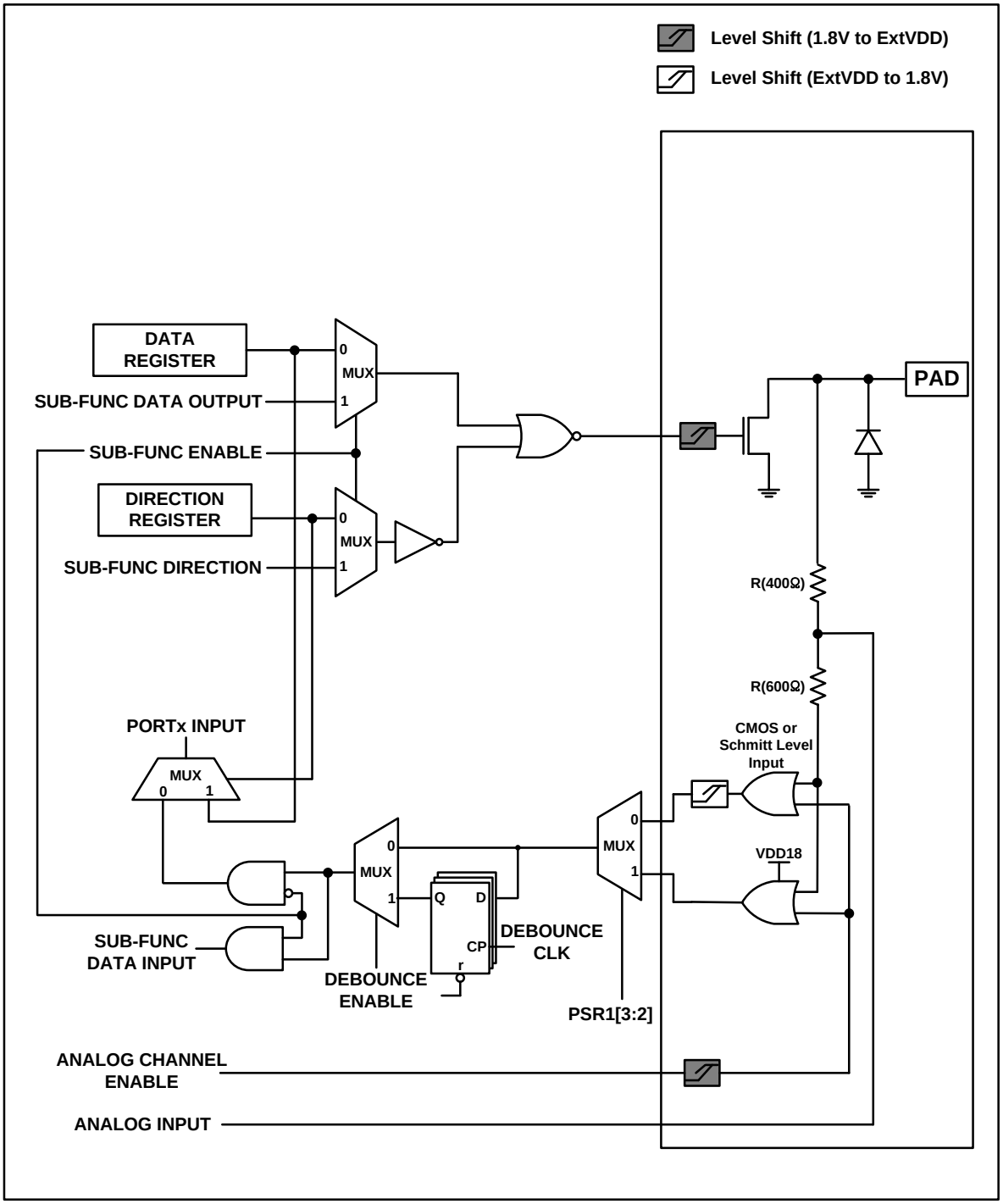


Figure 6-3 Open-Drain I/O Port with 1.8V Interface (INT0/P12, SDA/P13, SCL/P14)

7. Electrical Characteristics

Absolute Maximum Ratings

Table 7-1 Absolute Maximum Ratings

Parameter	Symbol	Rating	Unit
Supply Voltage	VDD	-0.3~+6.5	V
	VSS	-0.3~+0.3	V
Normal Voltage Pin	VI	-0.3~VDD+0.3	V
	VO	-0.3~VDD+0.3	V
	IOH	10	mA
	∑IOH	80	mA
	IOL	20	mA
	∑IOL	160	mA
Total Power Dissipation	PT	600	mW
Storage Temperature	TSTG	-65 ~ +150	℃

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

7.1 Recommended Operating Conditions

Table 7-2 Recommended Operation Conditions

Parameter	Symbol	Condition	MIN	TYP	MAX	Unit
Supply Voltage	VDD	Internal RC-OSC 16MHz	2.7	-	5.5	V
Operating Temperature	TOPR	VDD=2.7~5.5V	-40	-	85	℃
Operating Frequency	FIRC	Internal RC-OSC	-	16	-	MHz
	FWDT	Internal WDT Ring-OSC	-	256	-	kHz

7.2 Voltage Dropout Converter Characteristics

Table 7-3 Voltage Dropout Converter Characteristics

Parameter	Symbol	Condition	MIN	TYP	MAX	Unit
Operating Voltage		-	2.7	-	5.5	V
Operating Temperature		-	-40	-	+85	℃
Regulation Voltage		-	-	1.9	-	V
Drop-out Voltage		-	-	0.02	-	V
Current Drivability		RUN/IDLE	-	20	-	mA
		SUB-ACTIVE	-	1	-	mA
		STOP1	-	50	-	uA
		STOP2	-	10	-	uA
Operating Current	IDD1	RUN/IDLE	-	0.5	-	mA
	IDD2	SUB-ACTIVE	-	0.1	-	mA
	SIDD1	STOP1	-	5	-	uA
	SIDD2	STOP2	-	0.1	-	uA
Drivability Transition Time	TRAN1	SUB to RUN	-	1	-	us
	TRAN2	STOP to RUN	-	200	-	us

Note) STOP1: WDTRC OSC run, STOP2: WDTRC OSC stop.

7.3 Power-On Reset Characteristics

Table 7-4 Power-On Reset Characteristics

Parameter	Symbol	Condition	MIN	TYP	MAX	Unit
Operating Voltage		-	VSS	-	5.5	V
Operating Temperature		-	-40	-	+85	℃
RESET Release Level	POR	-	1.3	1.4	1.5	V
Operating Current	IDD	-	-	1	-	uA
VDD Rise Rate	VRR		0.05		5	V/ms

7.4 Brown Out Detector & Low Voltage Indicator Characteristics

Table 7-5 Brown Out Detector & Low Voltage Indicator Characteristics

Parameter	Symbol	Condition	MIN	TYP	MAX	Unit
Operating Voltage		-	VSS	-	5.5	V
Operating Temperature		-	-40	-	+85	℃
Detection Level	4.2V	LVI	3.8	-	4.6	V
	3.6V	LVI	3.2	-	3.8	V
	2.5V	LVI	2.25	-	2.75	V
	1.6V	BOD	1.5	-	1.8	V
Hysteresis		-	-	40	-	mV
Operating Current	IDD	-	-	30	-	uA

7.5 Internal RC Oscillator Characteristics

Table 7-6 Internal RC Oscillator Characteristics

Parameter	Symbol	Condition	MIN	TYP	MAX	Unit
Operating Voltage		-	2.7	-	5.5	V
Operating Temperature		-	-40	25	+85	℃
Frequency		-	15.52	16	16.48	MHz
Stabilization Time		-	-	1	-	ms
Operating Current	IDD	-	-	400	-	uA

7.6 Ring-Oscillator Characteristics

Table 7-7 Ring-Oscillator Characteristics

Parameter	Symbol	Condition	MIN	TYP	MAX	Unit
Operating Voltage		-	2.7	-	5.5	V
Operating Temperature		-	-40	-	+85	℃
Frequency		-	230.4	256	281.6	kHz
Stabilization Time		-	-	1	-	ms
Operating Current	IDD	-	-	2	-	uA

7.1 Touch Switch Characteristics

Table 7-8 Touch Switch Characteristics

Parameter	Symbol	Condition	MIN	TYP	MAX	Unit
Operating Voltage	V_{DD}	-	2.7	-	5.5	V
	V_{DDA}	-	2.7	-	5.5	V
VDC Voltage	V_{CCL}	From MCU	-	1.9	-	V
SNR (Signal-to-Noise Ratio)	SNR		-	26	-	dB
Self-Calibration Time	T_{CAL}	-	-	10	-	ms
Scan Speed	T_{SCAN}		-	10	-	ms
Supply Current	I_{VDD}		-	0.5	-	mA
Operation Temperature	T_{OPER}	-	-40	-	+85	℃

7.2 A/D Converter Characteristics

Table 7-9 A/D Converter Characteristics

Parameter	Symbol	Pin/Condition	MIN	TYP	MAX	Unit
AV_{DD} Input Current	I_{AVDD}	VDD	-	-	100	uA
AV_{DD} Input Voltage	AV_{DD}	VDD	2.7	-	5.5	V
Analog Input Voltage Range	V_{AIN}	AN0~AN15	V_{SS}-0.3	-	V_{SS}+0.3	V

Resolution	N_R	-	-	10	-	Bit
Analog Input Capacitance	C_{AIN}	AN0~AN15	-	-	30	pF
Integral Non Linearity Error	N_{INL}	-	-	±1	±3	LSB
Differential Non linearity Error	N_{DNL}		-	±1	±1.5	
Zero Offset Error	N_{ZOE}		-	±1	±1.5	
Full Scale Error	N_{TOE}		-	±1	±3	
Conversion Time	T_{CONV}	FXIN=2MHz	-	-	10	uS

7.3 DC Characteristics

Table 7-10 DC Characteristics

(VDD =2.7~5.5V, VSS =0V, f_{XIN}=16.0MHz, TA=+25℃)

Parameter	Symbol	Condition	MIN	TYP	MAX	Unit
Input Low Voltage	VIL0	ALL I/O	-0.5	-	0.3*VDD	V
	VIL1	P10, P11, P12 ~ P14 are input 1.8V level	-0.5	-	0.3*1.8V	
Input High Voltage	VIH0	ALL I/O	0.7*VDD	-	VDD	V
	VIH1	P10, P11, P12 ~ P14 are input 1.8V level	0.7*1.8V	-	VDD	
Output Low Voltage	VOL1	ALL I/O (IOL=20mA, VDD=4.5V)	-	-	1	V
	VOL2	Dimming PAD (IOL=11.5mA, VDD=3.3V)	-	0.176	-	V
		Dimming PAD (IOL=18.5mA, VDD=3.3V)	-	0.3	-	V
		Dimming PAD (IOL=37mA, VDD=3.3V)	-	0.75	-	V
Output High Voltage	VOH1	ALL I/O (IOH=-10mA, VDD=5V)	3.5	-	-	V
Input High Leakage Current	IIH	ALL PAD	-	-	1	uA
Input Low Leakage Current	IIL	ALL PAD	-1	-	-	uA
Pull-Up Resister	RPU1	ALL PAD @5V	30	-	70	kΩ
	RPU2	ALL PAD @3V	70	-	150	kΩ
Power Supply Current	IDD1	Run Mode, 16MHz @5V	-	3	5	mA
	IDD2	Sleep Mode, 16MHz @5V	-	2	3	mA
	IDD3	STOP1 Mode, WDT Active @5V (BOD disable)	-	5	-	uA
	IDD4	STOP2 Mode, WDT Disable @5V (BOD disable)	-	1	-	uA

Note) 1. STOP1: WDT only running, STOP2: All function disable.

2. Dimming PAD: P01, P02, and P03.

7.4 AC Characteristics

Table 7-11 AC Characteristics (VDD=5.0V±10%, VSS=0V, TA=-40~+85℃)

Parameter	Symbol	PIN	MIN	TYP	MAX	Unit
Operating Frequency	fMCP	-	2	-	16	MHz
System Clock Cycle Time	tSYS	-	62.5	-	500	ns
Oscillation Stabilization Time (16MHz)	tMST1	-	-	-	10	ms
External Interrupt Input Width	tIW	INT0~INTx	2	-	-	tSYS
External Interrupt Transition Time	tFI,tRI	INT0~INTx			20	ns
nRESET Input Pulse “L” Width	tRST	nRESET	8	-	-	tSYS

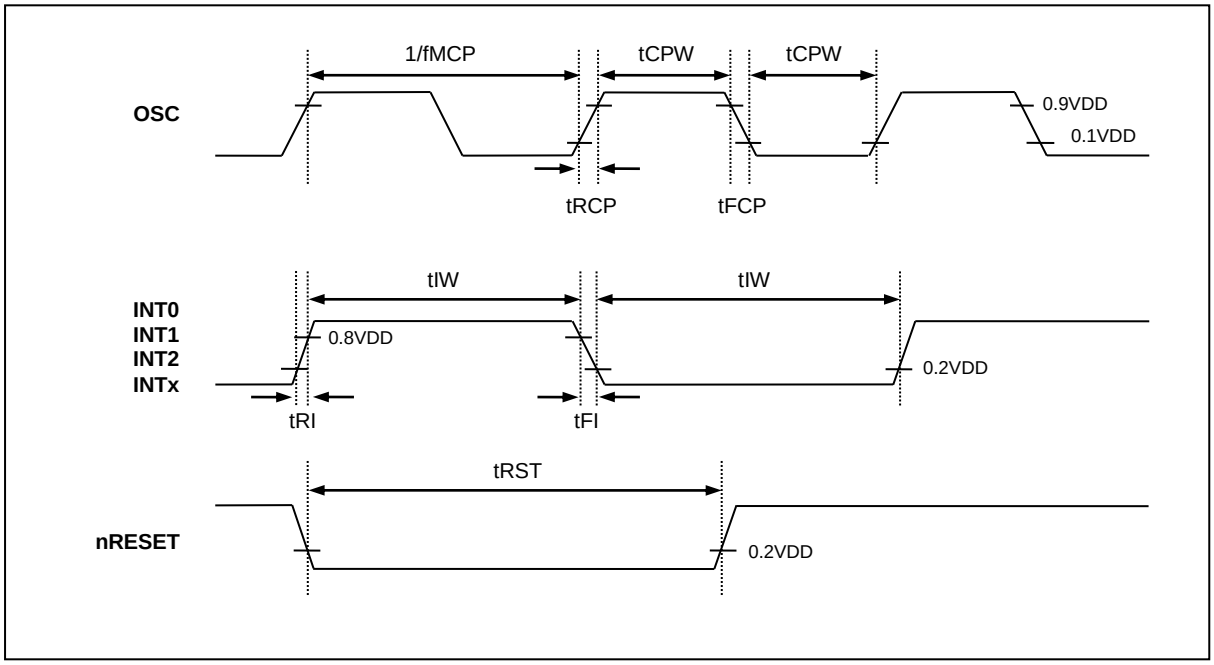


Figure 7-1 AC Timing

7.5 Typical Characteristics

These graphs and tables provided in this section are for design guidance only and are not tested or guaranteed. In some graphs or tables the data presented are outside specified operating range (e.g. outside specified VDD range). This is for information only 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.

8. Memory

The A96T336 addresses two separate address memory stores: Program memory and Data memory. The logical separation of Program and Data memory allows Data memory to be assessed by 8-bit addresses, which can be more quickly stored and manipulated by 8-bit CPU. Nevertheless, 16-bit Data memory addresses can also be generated through the DPTR register.

Program memory can only be read, not written to. There can be up to 64K bytes of Program memory in a bank. In the A96T336 FLASH version of these devices the 16K bytes of Program memory are provided on-chip. Data memory can be read and written to up to 256 bytes internal memory (DATA) including the stack area.

8.1 Program Memory

A 16-bit program counter is capable of addressing up to 64K bytes for one bank of memory space, but this device has 16K bytes program memory space.

Figure 8-1 shows a map of the lower part of the program memory. After reset, the CPU begins execution from location 0000H. Each interrupt is assigned a fixed location in program memory. The interrupt causes the CPU to jump to that location, where it commences execution of the service routine. External interrupt 0, for example, is assigned to location 0003H. If external interrupt 0 is going to be used, its service routine must begin at location 0003H. If the interrupt is not going to be used, its service location is available as general purpose program memory. If an interrupt service routine is short enough (as is often the case in control applications), it can reside entirely within that 8 byte interval. Longer service routines can use a jump instruction to skip over subsequent interrupt locations, if other interrupts are in use.

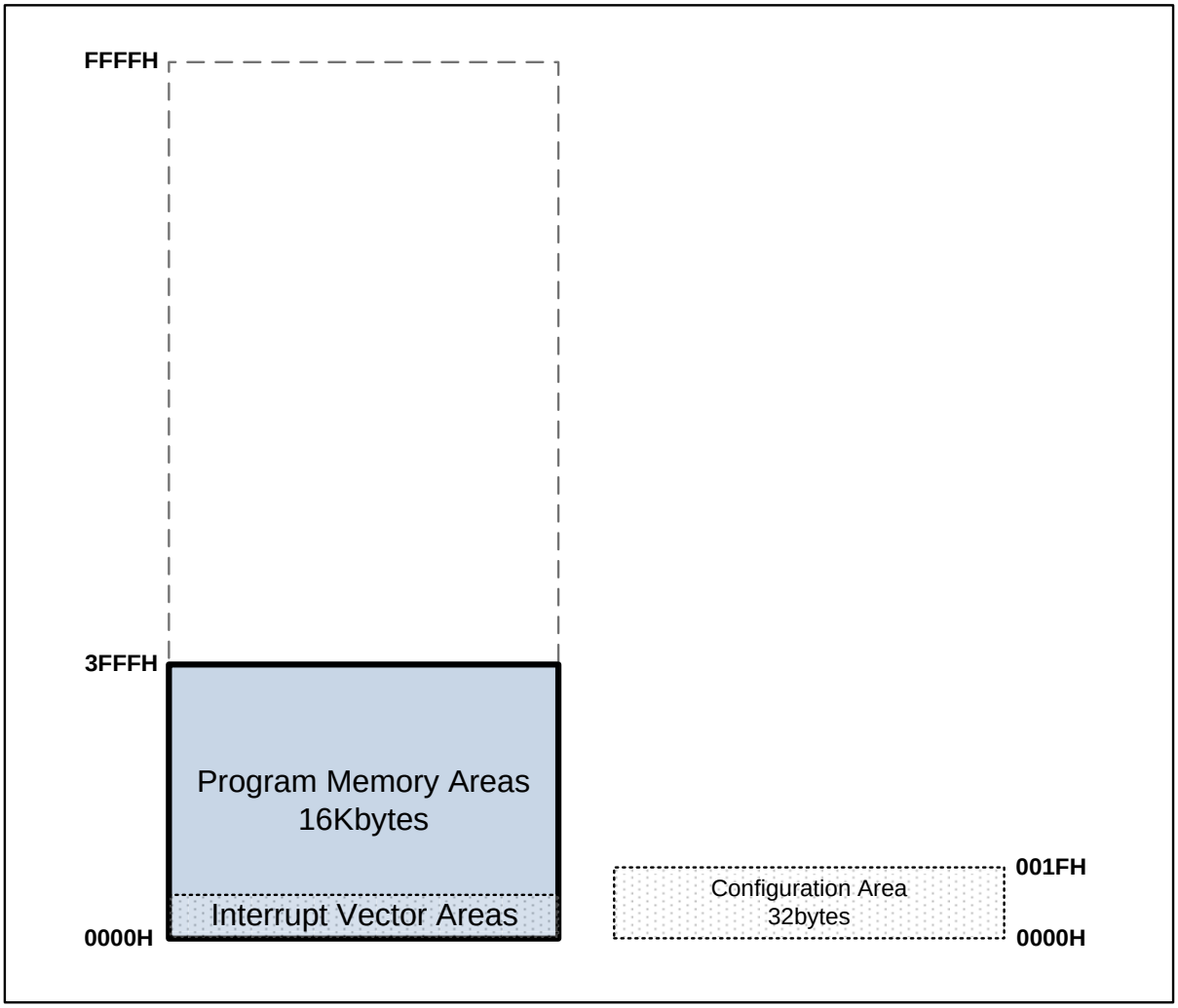


Figure 8-1 Program Memory

- User Function Mode: 16KBytes Included Interrupt Vector Region
- Non-volatile and reprogramming memory: Flash memory based on EEPROM cell

8.2 Data Memory

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

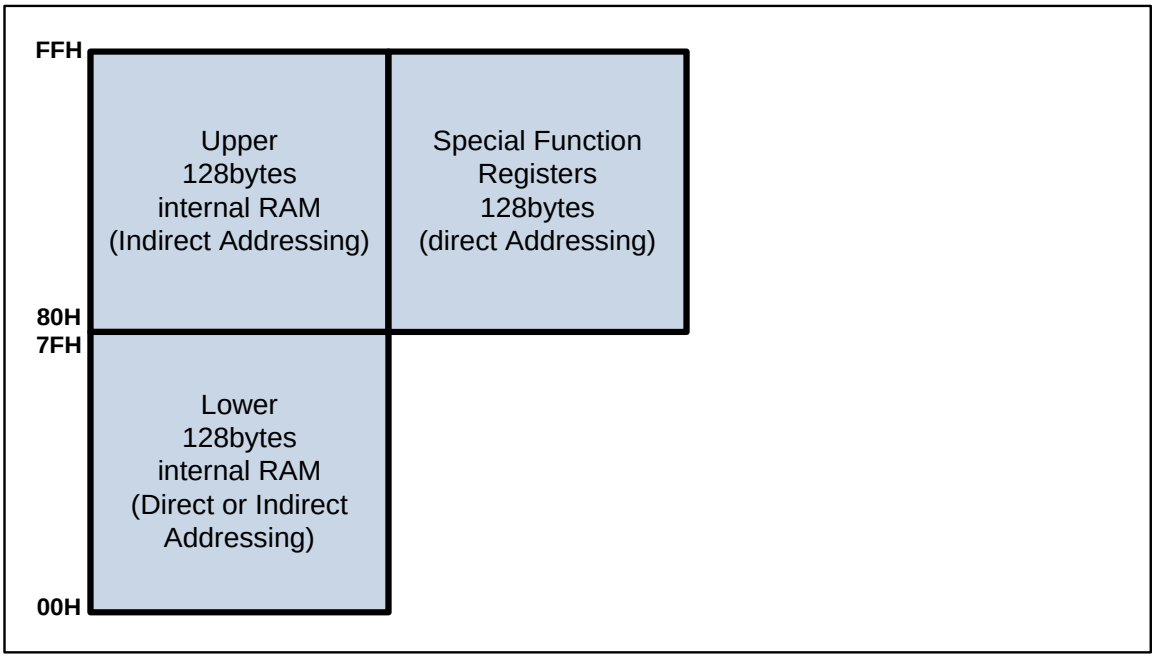


Figure 8-2 Data Memory Map

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

Internal Data memory addresses are always one byte wide, which implies an address space of only 256 bytes. However, the addressing modes for internal RAM can in fact accommodate 384 bytes, 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 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 used of code space, since register instructions are shorter than instructions that use direct addressing.

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

All of the bytes in the lower 128 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 user RAM and stack pointer.

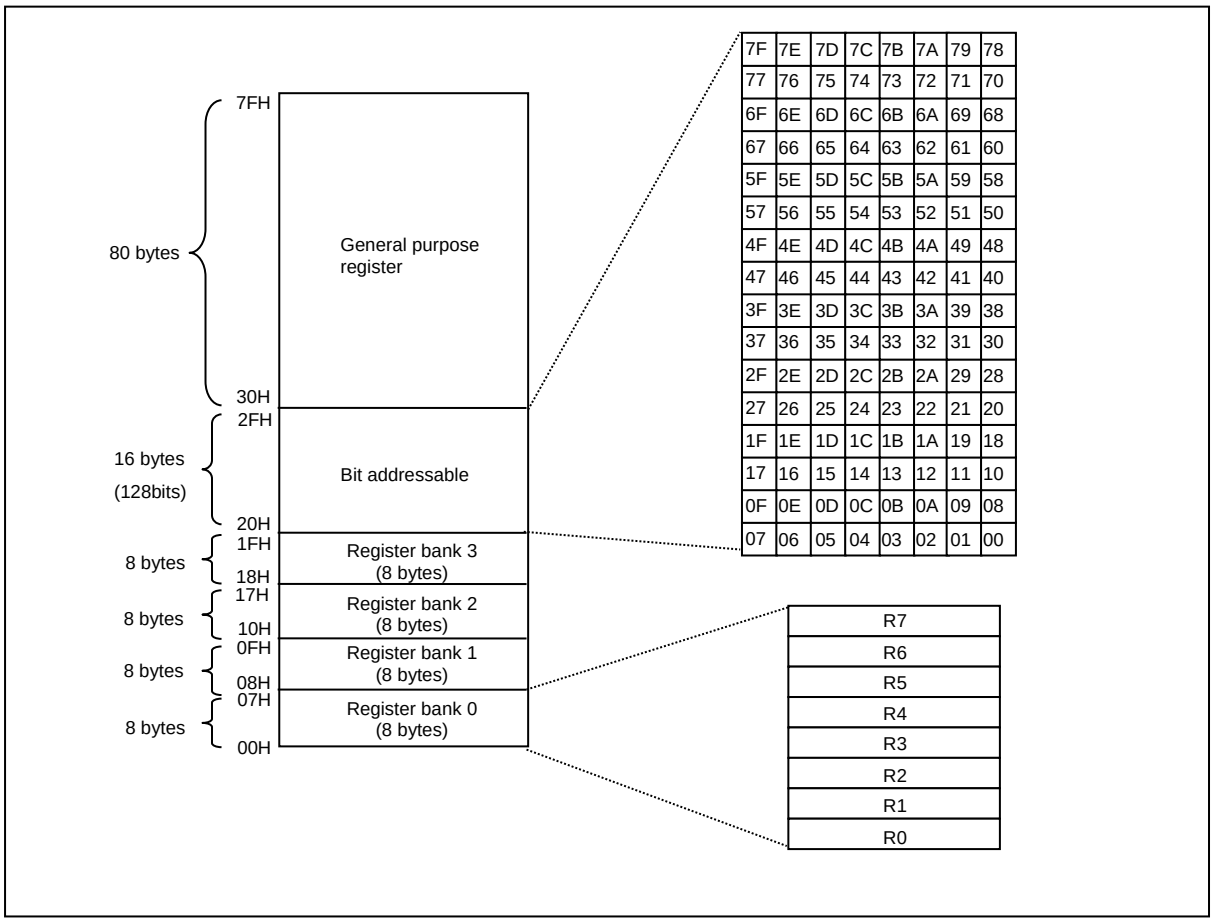


Figure 8-3 Lower 128 bytes RAM

8.3 XSFR

A96T336 has 1792BytesXSRAM. This area has no relation with RAM/FLASH. It can read and write through SFR with 8-bit unit.

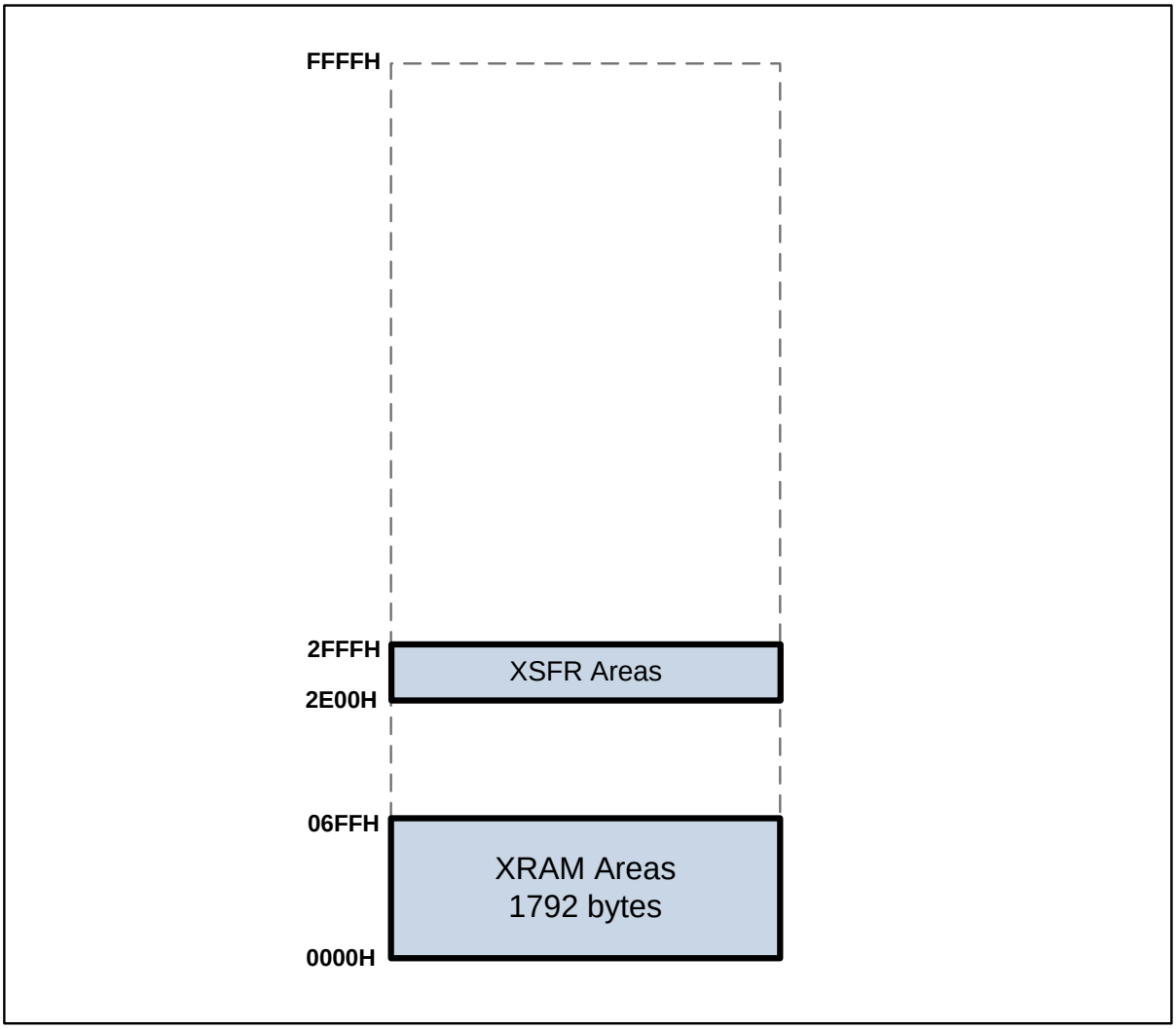


Figure 8-4 XDATA Memory Area

8.4 SFR Map

8.4.1 SFR Map Summary

Table 8-1 SFR Map Summary

	0H/8H[1]	1H/9H	2H/AH	3H/BH	4H/CH	5H/DH	6H/EH	7H/FH
2F60H	FCHH	FCHL	FCLH	FCLL	FCMR	-	-	-
2F58	-	-	-	-	-	-	TEST_B	TEST_A
2F50	FUSE_CONF	FUSE_CAL0	FUSE_CAL1	FUSE_RING	FUSE_BGR3	FUSE_FLAG	FUSE_TOUC	FUSE_PKGx
2EE8	TDRVP	TINTP	TRSTN	TDRVN	TINTN	-	-	-
2EE0	-	-	-	SCI	SCC	SVREF	-	TRSTP
2EC8			TS_INTEG_CNT	TS_FREQ_NUM	TS_FREQ_DEL	TS_CLK_CFG	TRIM_OSC	TRIM_A_OSC
2EC0	TS_CON	-	TS_MODE	TS_SUM_CNT	TS_CH_SELH	TS_CH_SELL	TS_WAKE_DEL	TS_SLP_CR
2E58	SCO12_H	SCO12_L	SCO13_H	SCO13_L	SCO14_H	SCO14_L	SCO15_H	SCO15_L
2E50	SCO8_H	SCO8_L	SCO9_H	SCO9_L	SCO10_H	SCO10_L	SCO11_H	SCO11_L
2E48	SCO4_H	SCO4_L	SCO5_H	SCO5_L	SCO6_H	SCO6_L	SCO7_H	SCO7_L
2E40	SCO0_H	SCO0_L	SCO1_H	SCO1_L	SCO2_H	SCO2_L	SCO3_H	SCO3_L
2E18	SUM_CH12H	SUM_CH12L	SUM_CH13H	SUM_CH13L	SUM_CH14H	SUM_CH14L	SUM_CH15H	SUM_CH15L
2E10	SUM_CH8H	SUM_CH8L	SUM_CH9H	SUM_CH9L	SUM_CH10H	SUM_CH10L	SUM_CH11H	SUM_CH11L
2E08	SUM_CH4H	SUM_CH4L	SUM_CH5H	SUM_CH5L	SUM_CH6H	SUM_CH6L	SUM_CH7H	SUM_CH7L
2E00	SUM_CH0H	SUM_CH0L	SUM_CH1H	SUM_CH1L	SUM_CH2H	SUM_CH2L	SUM_CH3H	SUM_CH3L
F8H	IP1	-	UCTRL1	UCTRL2	UCTRL3	USTAT	UBAUD	UDATA
F0H	B	-	FEARL	FEARM	FEARH	-	-	-
E8H	-	-	FEMR	FECR	FESR	FETCR	FEARL1	FEARM1
E0H	ACC	-	-	-	-	-	-	-
D8H	-	-	I2CMR	I2CSR	I2CSCLLR	I2CSCLHR	I2CSDAHR	I2CDR
D0H	PSW	-	DIMM1	DIMM2	DIMM3	TMISR	I2CSAR1	I2CSAR
C8H	-	-	-	-	-	-	-	-
C0H	-	-	-	-	-	-	-	-
B8H	IP	-	T1CR	T1CR1	PWM1DRL CDR1L / T1L	PWM1DRH CDR1H / T1H	PWM1PRL T1DRL	PWM1PRH T1DRH
B0H	-	-	T0CR	T0CR1	PWM0DRL CDR0L / T0L	PWM0DRH CDR0H / T0H	PWM0PRL T0DRL	PWM0PRH T0DRH
A8H	IE	IE1	IE2	PSR0	PSR1	PSR2	DIMME	PSRPWM
A0H	-	-	EO	EIENAB	EIFLAG	EIEDGE	EIPOLA	EIBOTH
98H	-	P2IO	P2PU	P2OD	P2DB	-	-	-
90H	P2	P1IO	P1PU	P1OD	P1DB	P0PU	P0OD	P0DB
88H	P1	P0IO	SCCR	BCCR	BITR	WDTMR	WDTR /WDTCR	RSFR
80H	P0	SP	DPL	DPH	DPL1	DPH1	BODR	PCON

Note: 1) These registers are bit-addressable

8.4.2 Compiler Compatible SFR

ACC (Accumulator) : E0H

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

Initial value : 00H

ACC	Accumulator

B (B Register) : F0H

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

Initial value : 00H

B	B Register

SP (Stack Pointer) : 81H

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

Initial value : 07H

SP	Stack Pointer

DPL (Data Pointer Low Byte) : 82H

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

Initial value : 00H

DPL	Data Pointer Low Byte

DPH (Data Pointer High Byte) : 83H

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

Initial value : 00H

DPH	Data Pointer High Byte

DPL1 (Data Pointer Low Byte) : 84H

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

Initial value : 00H

DPL Data Pointer Low Byte

DPH1 (Data Pointer High Byte) : 85H

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

Initial value : 00H

DPH Data Pointer High 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.0
R	R	R	R/W	R	R	R	R/W

Initial value : 00H

TRAP_EN Select the instruction

0 Select MOVC @(DPTR++), A

1 Select Software TRAP instruction

DPSEL Select Banked Data Point Register

0 DPTR0

1 DPTR1

9. I/O Ports

9.1 I/O Ports

The A96T336 has 22 I/O ports (P0 ~ P2). Each port can be easily configured by software as I/O pin, internal pull up and open drain pin to meet various system configurations and design requirements.

9.2 Port Register

9.2.1 Data Register (P0~P2)

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 (P0IO~P2IO)

Each I/O pin can independently use as an input or an output through the PxIO register. Bits cleared in this read/write register will select the corresponding pin in Px to become an input, setting a bit sets the pin to output. All bits are cleared by a system reset.

9.2.3 Pull-up Resistor Selection Register (P0PU~P2PU)

The on-chip pull-up resistor can be connected to them in 1-bit units 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 (P0OD~P2OD)

There is internally open-drain selection register (PxOD) in P0 ~ P2. The open-drain selection register controls the open-drain enable/disable of each port. Ports become push-pull by a system reset. You should connect an internal resistor or an external resistor in open-drain output mode.

9.2.5 De-bounce Enable Register (P0DB~P2DB)

P0 ~ P2 support de-bounce function. De-bounce time of each ports has 16us

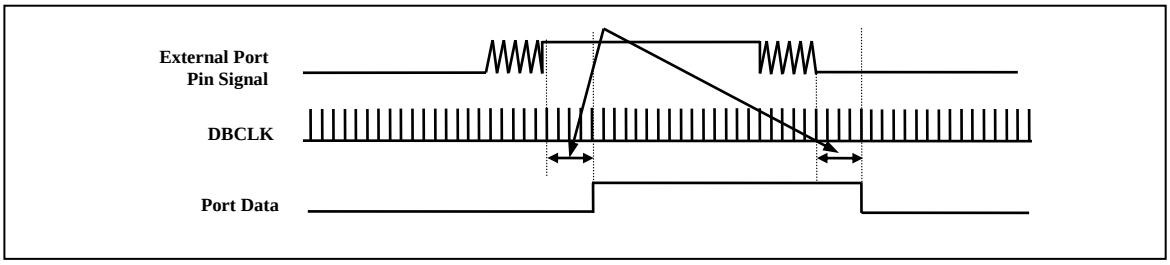


Figure 9-1 Debounce Function

9.2.6 Port Selection Register PSR0,1,2, PSRPWM

9.2.6.1 Port Selection Register (PSR0,2 and PSR1[5])

PSRx registers prevent the input leakage current when ports are connected to analog inputs. If the bit of PSRx 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 '1'.

The bit of PSR0 is 0x00, Enable P0 function (Digital input mode)
The bit of PSR0 is 0xFF, Disable P0 function (Digital input of the corresponding port is always "1")

The bit of PSR1[5] is 0x0, Enable P1[5] function (Digital input mode)
The bit of PSR1[5] is 0x1, Disable P1[5] function (Digital input of the corresponding port is always "1")

The bit of PSR2 is 0x00, Enable P2 function (Digital input mode)
The bit of PSR2 is 0xFF, Disable P2 function (Digital input of the corresponding port is always "1")

9.2.6.2 Special Port Selection Register (PSR1[4:0])

The bit of PSR1[0] is '0', I^2C SCL/SDA use P11/P10
The bit of PSR1[0] is '1', I^2C SCL/SDA use P14/P13

The bit of PSR1[1] is '0', USART RXD/TXD use P10/P11
The bit of PSR1[1] is '1', USART RXD/TXD use P13/P14

The bit of PSR1[2] is '0', VDD/VSS input level use P12
The bit of PSR1[2] is '1', 1.8V/VSS input level use P12

The bit of PSR1[3] is '0', VDD/VSS input level use P14/P13
The bit of PSR1[3] is '1', 1.8V/VSS input level use P14/P13

The bit of PSR1[4] is '0', VDD/VSS input level use P11/P10
The bit of PSR1[4] is '1', 1.8V/VSS input level use P11/P10

9.2.6.3 PWM Port Selection Register (PSRPWM)

The bit of PSRPWM[2:0] is 0x00 , PWM0 Out P06 (Default)
The bit of PSRPWM[2:0] is 0x01 , PWM0 Out P20
The bit of PSRPWM[2:0] is 0x02 , PWM0 Out P22
The bit of PSRPWM[2:0] is 0x03 , PWM0 Out P24
The bit of PSRPWM[2:0] is 0x04 , PWM0 Out P26

The bit of PSRPWM[5:3] is 0x00 , PWM1 Out P15 (Default)
The bit of PSRPWM[5:3] is 0x01 , PWM1 Out P21

The bit of PSRPWM[5:3] is 0x02 , PWM1 Out P23
The bit of PSRPWM[5:3] is 0x03 , PWM1 Out P25
The bit of PSRPWM[5:3] is 0x04 , PWM1 Out P27

9.2.7 Register Map

Table 9-1 Register Map

Name	Address	Dir	Default	Description
P0	80H	R/W	00H	P0 Data Register
P0IO	89H	R/W	00H	P0 Direction Register
P0PU	95H	R/W	00H	P0 Pull-up Resistor Selection Register
P0OD	96H	R/W	00H	P0 Open-drain Selection Register
P0DB	97H	R/W	00H	P0 Debounce Enable Register
P1	88H	R/W	00H	P1 Data Register
P1IO	91H	R/W	00H	P1 Direction Register
P1PU	92H	R/W	00H	P1 Pull-up Resistor Selection Register
P1OD	93H	R/W	00H	P1 Open-drain Selection Register
P1DB	94H	R/W	00H	P1 Debounce Enable Register
P2	90H	R/W	00H	P2 Data Register
P2IO	99H	R/W	00H	P2 Direction Register
P2PU	9AH	R/W	00H	P2 Pull-up Resistor Selection Register
P2OD	9BH	R/W	00H	P2 Open-drain Selection Register
P2DB	9CH	R/W	00H	P2 Debounce Enable Register
PSR0	ABH	R/W	00H	P0 Port Selection Register
PSR1	ACH	R/W	00H	P1 Port Selection Register
PSR2	ADH	R/W	00H	P2 Port Selection Register
PSRPWM	AFH	R/W	00H	PWM Port Selection Register
DIMME	AEH	R/W	00H	Dimming Enable Register
DIMM1	D2H	R/W	00H	Dimming Value Select Register
DIMM2	D3H	R/W	00H	Dimming Value Select Register
DIMM3	D4H	R/W	00H	Dimming Value Select Register

9.3 P0, P1, P2 Port

9.3.1 Px Port Description

Px is 8-bit I/O port. Px control registers consist of Data register (Px), direction register (PxIO), debounce enable register (PxDB), pull-up register selection register (PxPU), open-drain selection register (PxOD).

9.3.2 Register Description for Px

P0, P1,P2(Px Data Register) : 80H, 88H, 90H

7	6	5	4	3	2	1	0
Px7	Px6	Px5	Px4	Px3	Px2	Px1	Px0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

Px[7:0] I/O Data

P0IO, P1IO, P2IO(Px Direction Register) : 89H, 91H, 99H

7	6	5	4	3	2	1	0
Px7IO	Px6IO	Px5IO	Px4IO	Px3IO	Px2IO	Px1IO	Px0IO
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

PxIO[7:0] Px data I/O direction.

0 Input

1 Output

P0PU, P2PU (Px Pull-up Resistor Selection Register) : 95H, 9AH

7	6	5	4	3	2	1	0
Px7PU	Px6PU	Px5PU	Px4PU	Px3PU	Px2PU	Px1PU	Px0PU
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

PxPU[7:0] Configure pull-up resistor of Px port

0 Disable

1 Enable

P1PU (P1 Pull-up Resistor Selection Register) : 92H

7	6	5	4	3	2	1	0
P17PU	P16PU	P15PU	-	-	-	P11PU	P10PU
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
(P1[4:2] : Only Pull-up disable)

1 Enable

P0OD, P1OD, P2OD (Px Open-drain Selection Register) : 96H, 9BH

7	6	5	4	3	2	1	0
Px7OD	Px6OD	Px5OD	Px4OD	Px3OD	Px2OD	Px1OD	Px0OD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

PxOD[7:0] Configure open-drain of Px port

0 Disable

1 Enable

P1OD (P1 Open-drain Selection Register) : 93H

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

Initial value : 00H

P1OD[7:0] Configure open-drain of Px port

0 Disable

1 Enable
(P1[4:2] : Only Open-drain output)

P0DB, P1DB, P2DB (PxDebounce Enable Register) :97H, 94H, 9CH

7	6	5	4	3	2	1	0
Px7DB	Px6DB	Px5DB	Px4DB	Px3DB	Px2DB	Px1DB	Px0DB
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

PxDB[7:0] Configure debounce of Px port

0 Disable

1 Enable

PSR0,2 and PSR1[5] registers prevent the input leakage current when ports are connected to analog inputs (ADC input) or outputs (TOUCH scan wave output). If the bit of PSR0,2 and PSR1[5] 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 '1'.

PSR0 (P0 Port Selection Register) : ABH

7	6	5	4	3	2	1	0
PSR07	PSR06	PSR05	PSR04	PSR03	PSR02	PSR01	PSR00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

PSR0[7:0] Digital Input Mode

0 Enable P0 Function (Digital Input Mode, Default)

1 Disable P0 Function

PSR2 (P2 Port Selection Register) : ADH

7	6	5	4	3	2	1	0
PSR27	PSR26	PSR25	PSR24	PSR23	PSR22	PSR21	PSR20

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

PSR2[7:0] Digital Input Mode

0 Enable P2 Function (Digital Input Mode, Default)

1 Disable P2 Function

PSR1 (P1 Port Selection Register) : ACH

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

Initial value : 00H

PSR1[5] Digital Input Mode

0 Enable P1[5] Function (Digital Input Mode, Default)

1 Disable P1[5] Function

PSR1[4] VIH Level Selector for P10/P11 Input

0 VDD Level (default)

1 1.8V Level

PSR1[3] VIH Level Selector for P13/P14 Input

0 VDD Level (default)

1 1.8V Level

PSR1[2] VIH Level Selector for P12 Input

0 VDD Level (default)

1 1.8V Level

PSR1[1] USART ports selection register

0 P10, P11 for USART (default)

1 P13, P14 for USART

PSR1[0] I^2C ports selection register

0 P10, P11 for I^2C (default)

1 P13, P14 for I^2C

PSRPWM (Port Selection Register for PWM) : AFH

7	6	5	4	3	2	1	0
-	-	PWPSR5	PSRPWM4	PSRPWM3	PSRPWM2	PSRPWM1	PSRPWM0
-	-	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

PSRPWM[2:0] PWM0 Output Port Selection

0x00 PWM0 Output to P06 (Default)

0x01 PWM0 Output to P20

0x02 PWM0 Output to P22

0x03 PWM0 Output to P24

0x04 PWM0 Output to P26

PSRPWM[5:3] PWM1 Output Port Selection

0x00 PWM1 Output to P15 (Default)

0x01 PWM1 Output to P21

0x02 PWM1 Output to P23

0x03 PWM1 Output to P25

0x04 PWM1 Output to P27

DIMME (Dimming Enable Register) : AEH

7	6	5	4	3	2	1	0
-	-	-	-	-	DIMME2	DIMME1	DIMME0
-	-	-	-	-	R/W	R/W	R/W

Initial value : 00H

DIMME[2] Dimming enable P03 port.

0 Port P03 select without Dimming. (Default)

1 Dimming select by DIMM3 register value when P03 is output state.

DIMME[1] Dimming enable P02 port.

0 Port P02 select without Dimming. (Default)

1 Dimming select by DIMM2 register value when P02 is output state.

DIMME[0] Dimming enable P01 port.

0 Port P01 select without Dimming. (Default)

1 Dimming select by DIMM1 register value when P01 is output state.

DIMM1 (Dimming Value Select Register) : D2H

7	6	5	4	3	2	1	0
-	-	-	DIMM14	DIMM13	DIMM12	DIMM11	DIMM10
-	-	-	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

DIMM1[4:0] Select N-TR Total Current value of Dimming port P01.

DIMM14	DIMM13	DIMM12	DIMM11	DIMM10	
0	0	0	0	0	N/A (default)
0	0	0	0	1	1/31 x I_{total}
0	0	0	1	0	2/31 x I_{total}
:	:	:	:	:	
1	1	1	1	0	30/31 x I_{total}
1	1	1	1	1	31/31 x I_{total}

DIMM2 (Dimming Value Select Register) : D3H

7	6	5	4	3	2	1	0
-	-	-	DIMM24	DIMM23	DIMM22	DIMM21	DIMM20
-	-	-	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

DIMM2[4:0] Select N-TR Total Current value of Dimming port P02.

DIMM24	DIMM23	DIMM22	DIMM21	DIMM20	
0	0	0	0	0	N/A (default)
0	0	0	0	1	1/31 x I_{total}

0	0	0	1	0	2/31 x I_{total}
:	:	:	:	:	
1	1	1	1	0	30/31 x I_{total}
1	1	1	1	1	31/31 x I_{total}

DIMM3 (Dimming Value Select Register) : D4H

7	6	5	4	3	2	1	0
-	-	-	DIMM34	DIMM33	DIMM32	DIMM31	DIMM30
-	-	-	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

DIMM3[4:0] Select N-TR Total Current value of Dimming port P03.

DIMM34	DIMM33	DIMM32	DIMM31	DIMM30	
0	0	0	0	0	N/A (default)
0	0	0	0	1	1/31 x I_{total}
0	0	0	1	0	2/31 x I_{total}
:	:	:	:	:	
1	1	1	1	0	30/31 x I_{total}
1	1	1	1	1	31/31 x I_{total}

Note) I_{total} : Total current of drive N-TR

Table 9-2 When I_{total} = 18.5mA, LED drive current table.

	Control	Current (mA)
1	00001	0.596774194
2	00010	1.193548387
3	00011	1.790322581
4	00100	2.387096774
5	00101	2.983870968
6	00110	3.580645161
7	00111	4.177419355
8	01000	4.774193548
9	01001	5.370967742
10	01010	5.967741935
11	01011	6.564516129
12	01100	7.161290323
13	01101	7.758064516
14	01110	8.35483871
15	01111	8.951612903
16	10000	9.548387097
17	10001	10.14516129
18	10010	10.74193548
19	10011	11.33870968
20	10100	11.93548387
21	10101	12.53225806
22	10110	13.12903226
23	10111	13.72580645
24	11000	14.32258065
25	11001	14.91935484
26	11010	15.51612903

27	11011	16.11290323
28	11100	16.70967742
29	11101	17.30645161
30	11110	17.90322581
31	11111	18.5

10. Interrupt Controller

10.1 Overview

The A96T336 supports up to 18 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 interrupt controller has following features:

- receive the request from 18 interrupt source
- 6 group priority
- 4 priority levels
- Multi Interrupt possibility
- If the requests of different priority levels are received simultaneously, the request of higher priority level is serviced
- Each interrupt source can control by EA bit and each IEx bit
- Interrupt latency: 5~8 machine cycles in single interrupt system

The maskable interrupts are enabled through five pair of interrupt enable registers (IE, IE1, IE2). Bits of IE, IE1, IE2 register each individually enable/disable a particular 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 A96T336 supports a 4-level priority scheme. Each maskable interrupt is individually assigned to one of four priority levels by writing to IP or IP1.

Priority sets two bit which is to IP and IP1 register about group. If two requests of different priority levels are received simultaneously, the request of higher priority level is serviced. If the request of same or lower priority level is received, that request is not serviced.

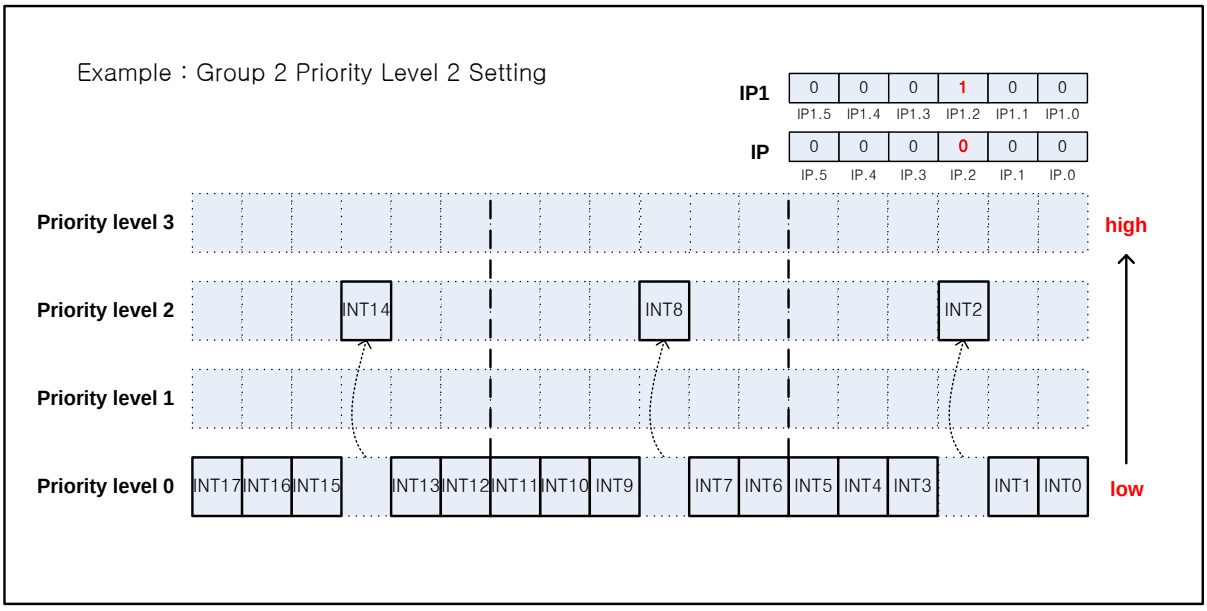


Figure 10-1 Interrupt Group Priority Level

10.2 Block Diagram

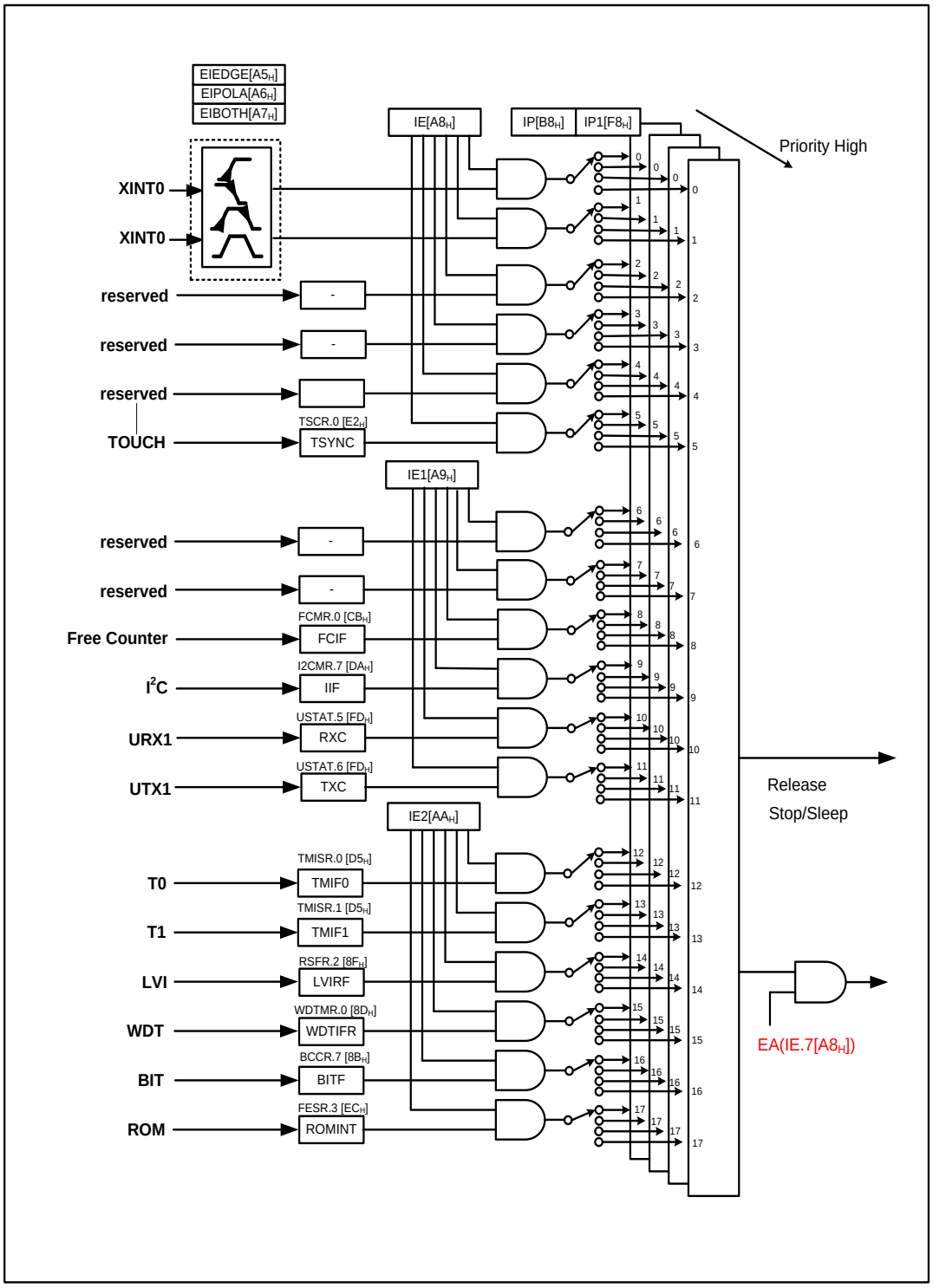


Figure 10-2 Block Diagram of Interrupt

10.3 Interrupt Vector Table

The interrupt controller supports 32 interrupt sources as shown in the Table 10-1below. When interrupt becomes service, long call instruction (LCALL) is executed in the vector address. Interrupt request 32 has a decided priority order.

Table 10-1 Interrupt Vector Address Table

Interrupt Source	Symbol	Interrupt Enable Bit	Priority	Mask	Vector Address
Hardware Reset	RESETB	0	0	Non-Maskable	0000H
External Interrupt	INT0	IE0.0	1	Maskable	0003H
External Interrupt	INT1	IE0.1	2	Maskable	000BH
-	INT2	IE0.2	3	Maskable	0013H
-	INT3	IE0.3	4	Maskable	001BH
-	INT4	IE0.4	5	Maskable	0023H
Touch Raw Data Sync	INT5	IE0.5	6	Maskable	002BH
-	INT6	IE1.0	7	Maskable	0033H
-	INT7	IE1.1	8	Maskable	003BH
Free Run Count	INT8	IE1.2	9	Maskable	0043H
I^2C	INT9	IE1.3	10	Maskable	004BH
INT_URX	INT10	IE1.4	11	Maskable	0053H
INT_UTX	INT11	IE1.5	12	Maskable	005BH
T0	INT12	IE2.0	13	Maskable	0063H
T1	INT13	IE2.1	14	Maskable	006BH
LVI	INT14	IE2.2	15	Maskable	0073H
WDT	INT15	IE2.3	16	Maskable	007BH
BIT	INT16	IE2.4	17	Maskable	0083H
ROM	INT17	IE2.5	18	Maskable	008BH

For mask-able interrupt execution, first EA bit must set ‘1’ and specific interrupt source must set ‘1’ by writing a ‘1’ to associated bit in the IEx. If interrupt request is received, specific interrupt request flag set ‘1’. And it remains ‘1’ until CPU accepts interrupt. After that, interrupt request flag will be cleared automatically.

10.4 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 stack. For the interrupt service routine, the interrupt controller gives the address of LJMP instruction to CPU. After finishing the current instruction, at the next instruction to go interrupt service routine needs 5~8 machine cycle and the interrupt service task is terminated upon execution of an interrupt return instruction [RETI]. After generating interrupt, to go to interrupt service routine, the following process is progressed

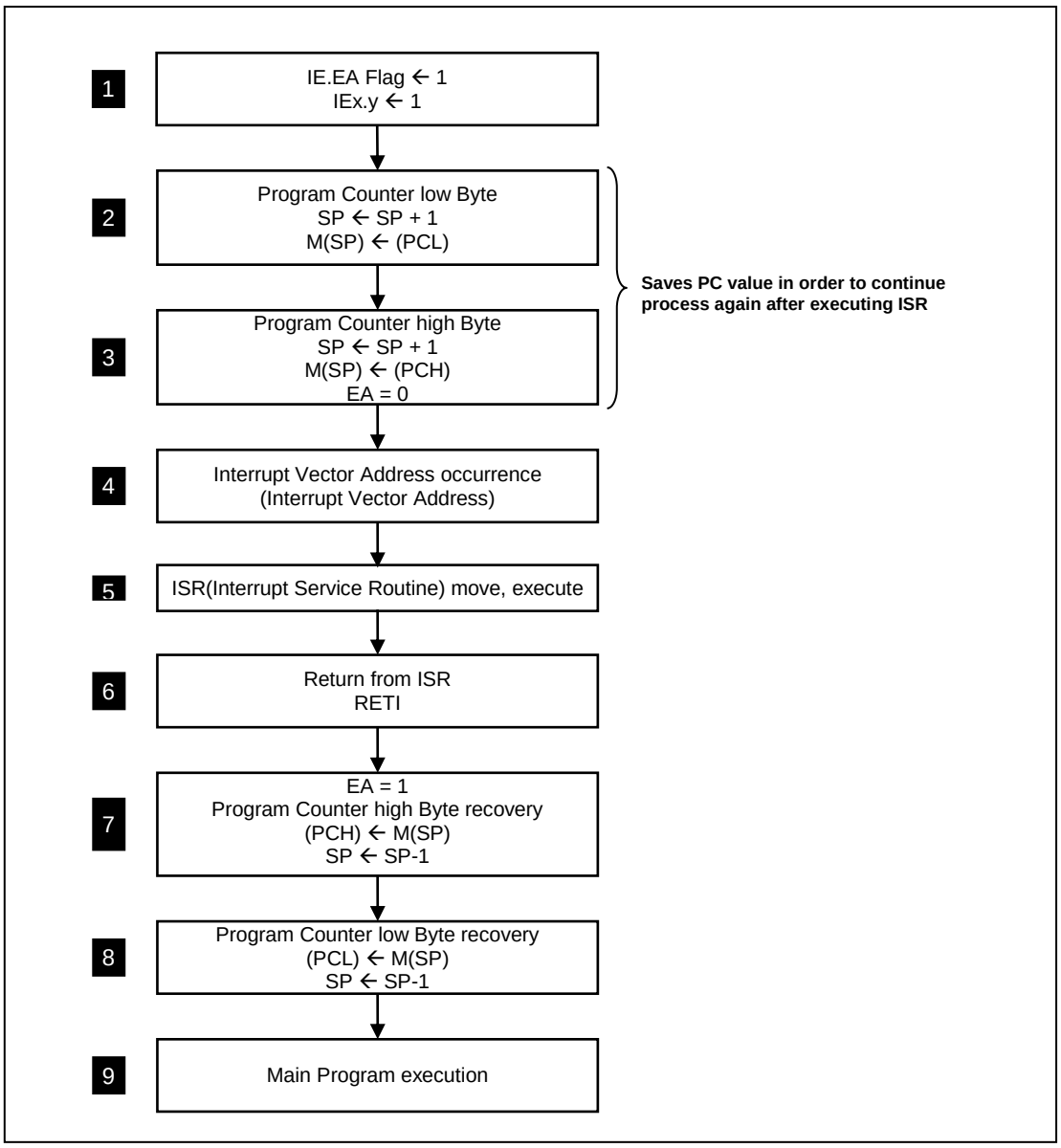


Figure 10-3 Interrupt Sequence Flow

10.5 Effective Timing after Controlling Interrupt bit

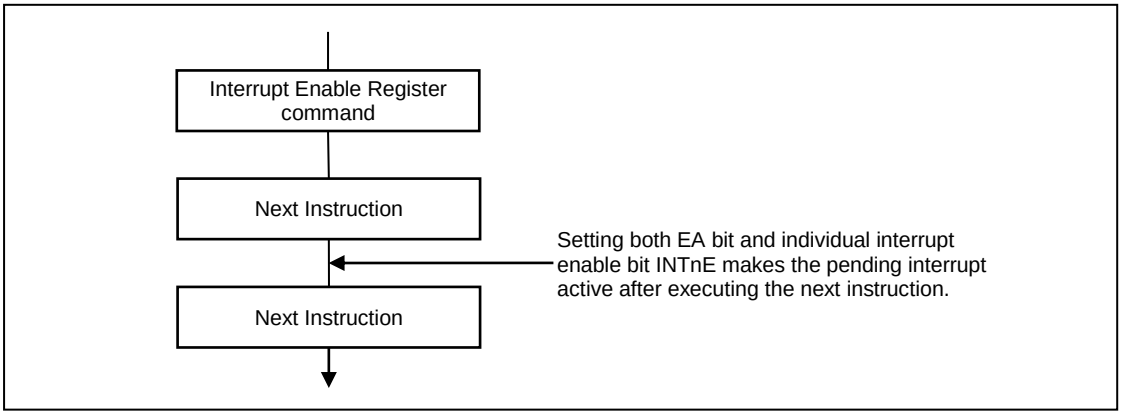


Figure 10-4 Interrupt Enable Register Effective Timing

10.6 Multi Interrupt

If two requests of different priority levels are received simultaneously, the request of higher priority level is serviced. If requests of the interrupt are received at the same time simultaneously, an interrupt polling sequence determines by hardware which request is serviced. However, multiple processing through software for special features is possible.

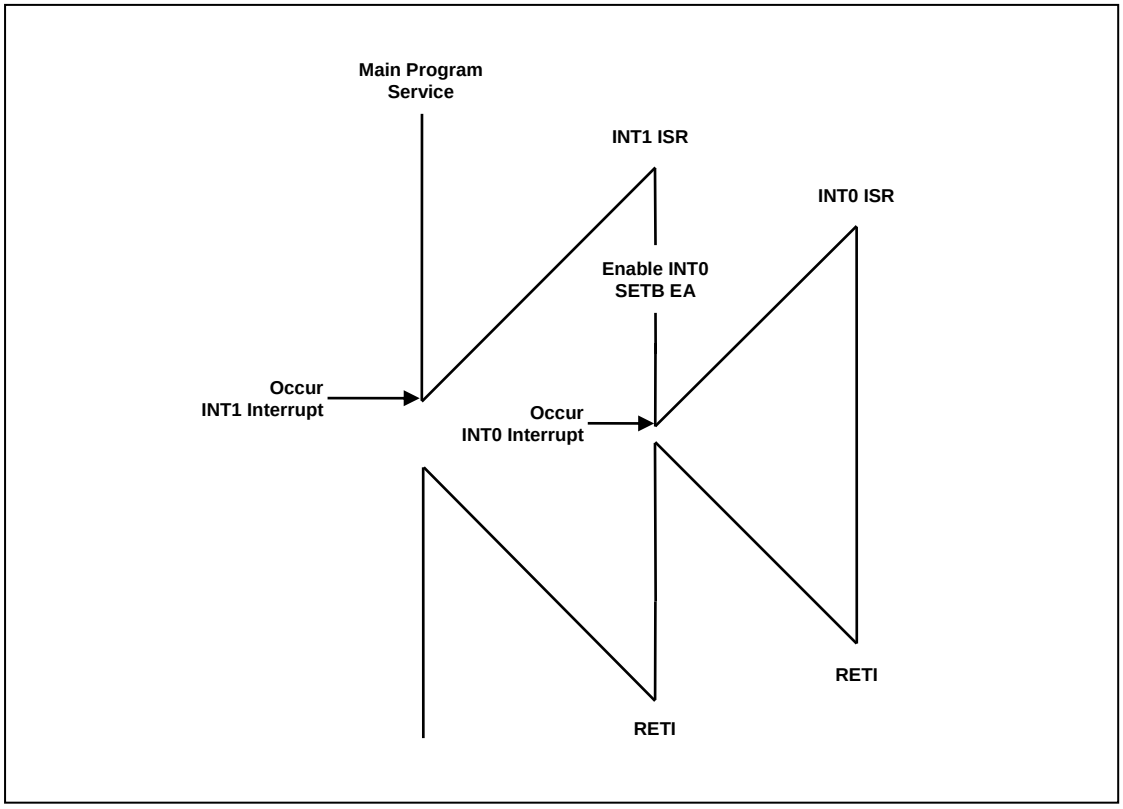


Figure 10-5 Execution of Multi Interrupt

Following example is shown to service INT0 routine during INT1 routine in Figure 10-5. In this

example, INT0 interrupt priority is higher than INT1 interrupt priority. If some interrupt is lower than INT1 priority, it can't service its interrupt routine.

Example) Software Multi Interrupt:

```
INT1:   MOV     IE, #81H      ; Enable INT0 only
        MOV     IE1, #00H     ; Disable others
        SETB    EA            ; Enable global interrupt (necessary for multi interrupt)
        :
```

10.7 Interrupt Enable Accept Timing

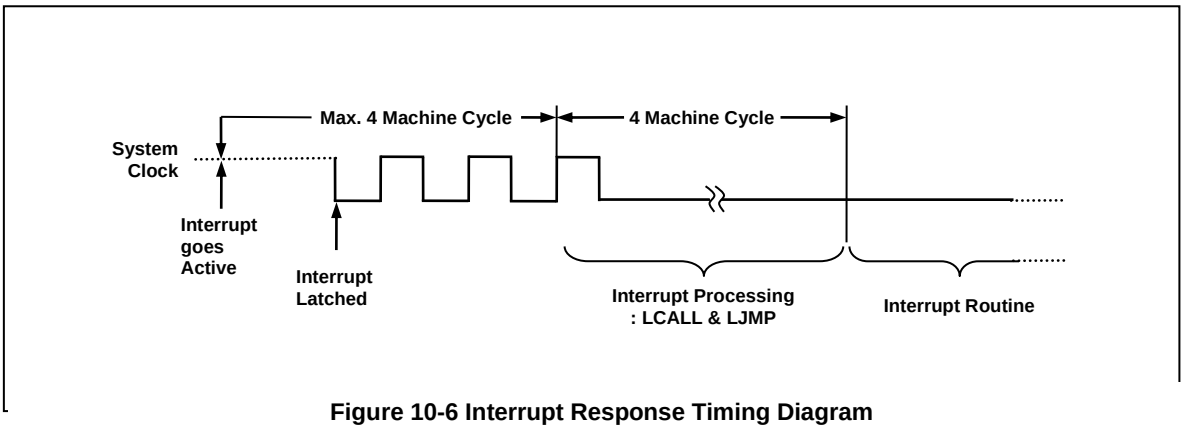


Figure 10-6 Interrupt Response Timing Diagram

10.8 Interrupt Service Routine Address

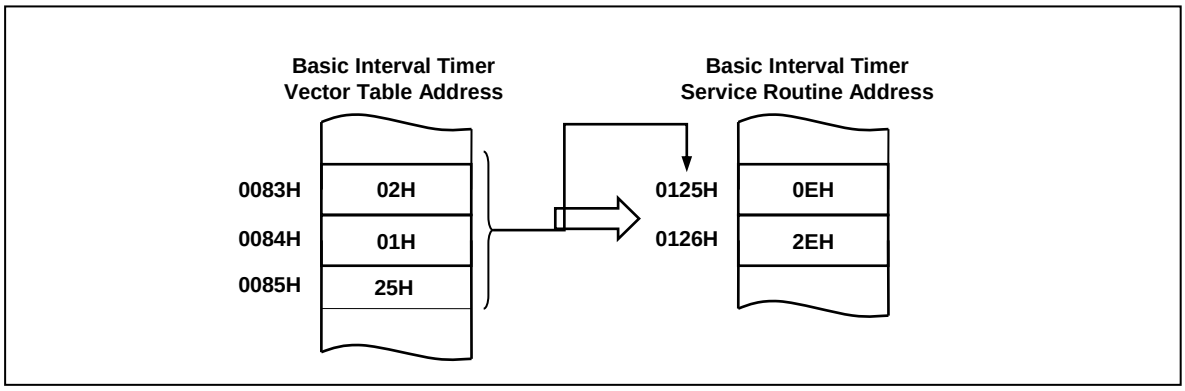


Figure 10-7 Correspondence between Vector Table Address and the Entry Address of ISR

10.9 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
```

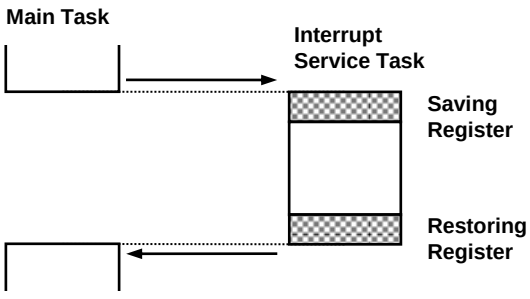


Figure 10-8 Saving/Restore Process Diagram & Sample Source

10.10 Interrupt Timing

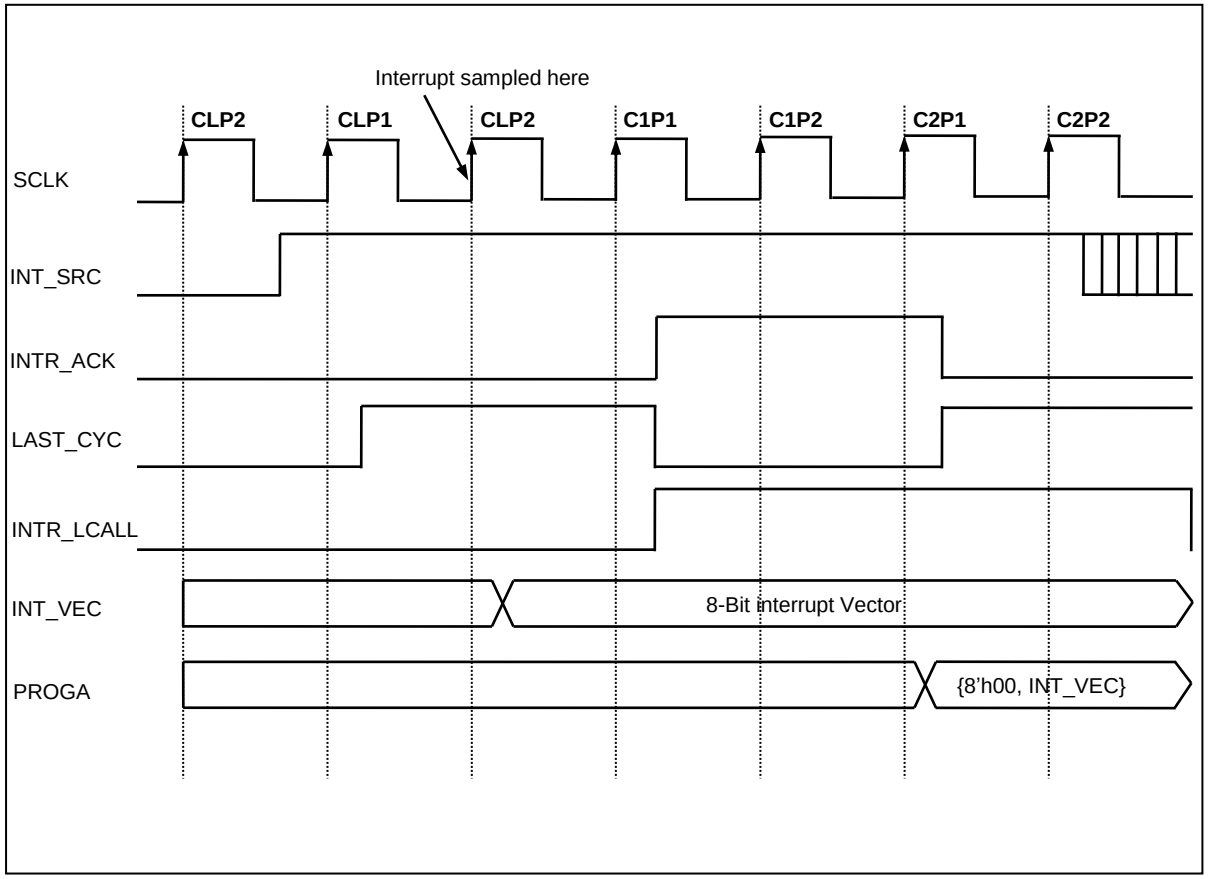


Figure 10-9 Timing Chart of Interrupt Acceptance and Interrupt Return Instruction

Interrupt source sampled at last cycle of the command. When sampling interrupt source, it is decided to low 8-bit of interrupt vector. M8051W core makes interrupt acknowledge at first cycle of command, executes long call to jump interrupt routine as INT_VEC.

Note) command cycle CxPx: L=Last cycle, 1=1st cycle or 1st phase, 2=2nd cycle or 2nd phase

10.11 Interrupt Register Overview

10.11.1 Interrupt Enable Register (IE, IE1, IE2)

Interrupt enable register consists of Global interrupt control bit (EA) and peripheral interrupt control bits. Totally 32 peripheral are able to control interrupt.

10.11.2 Interrupt Priority Register (IP, IP1)

The 32 interrupt divides 8 groups which have each 4 interrupt sources. A group can decide 4 levels interrupt priority using interrupt priority register. Level 3 is the high priority, while level 0 is the low priority. Initially, IP, IP1 reset value is '0'. At that initialization, low interrupt number has a higher priority than high interrupt number. If decided the priority, low interrupt number has a higher priority than high interrupt number in that group.

10.11.3 Register Map

Table 10-2 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
IP	B8H	R/W	00H	Interrupt Priority Register
IP1	F8H	R/W	00H	Interrupt Priority Register 1
EIENAB	A3H	R/W	00H	Interrupt Enable Register
EIFLAG	A4H	R/W	00H	Interrupt Flag Register
EIEDGE	A5H	R/W	00H	Interrupt Edge Register
EIPOLA	A6H	R/W	00H	Interrupt Polarity Register
EIBOTH	A7H	R/W	00H	Interrupt Both Edge Register

10.12 Interrupt Register Description

The Interrupt Register is used for controlling interrupt functions. Also it has pin change interrupt control registers. The interrupt register consists of Interrupt Enable Register (IE), Interrupt Enable Register 1 (IE1), Interrupt Enable Register 2 (IE2), Interrupt Priority Register (IP), Interrupt Priority Register 2 (IP2), P0 Pin Change Interrupt Register (PCI0), and P1 Pin Change Interrupt Enable Register (PCI1). The pin change interrupt on P0 and P1 ports receive the bot edge (posedge and negedge) interrupt request.

10.12.1 Register Description for Interrupt

IE (Interrupt Enable Register) : A8H

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

Initial value : 00H

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

INT5E Enable or disable Touch Raw data Sync Interrupt
- 0 Disable
- 1 Enable

INT4E Reserved
- 0 Disable
- 1 Enable

INT3E Reserved
- 0 Disable
- 1 Enable

INT2E Reserved
- 0 Disable
- 1 Enable

INT1E Enable or disable External Interrupt 1

0 Disable
1 Enable

INT0E Enable or disable External Interrupt 0
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 UART TX Interrupt
0 Disable
1 Enable

INT10E Enable or disable UART RX Interrupt
0 Disable
1 Enable

INT9E Enable or disable I^2C Interrupt
0 Disable
1 Enable

INT8E Enable or disable Free Run Counter
0 Disable
1 Enable

INT7E Reserved
0 Disable
1 Enable

INT6E Reserved
0 Disable
1 Enable

IE2 (Interrupt Enable Register 2) : AAH

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

Initial value : 00H

INT17E Enable or disable ROM Interrupt (not for user)
0 Disable
1 Enable

INT16E Enable or disable BIT Interrupt
0 Disable
1 Enable

INT15E Enable or disable Watch Dog Timer Interrupt
0 Disable
1 Enable

INT14E Reserved
0 Disable

	1	Enable
INT13E	Enable or disable Timer Interrupt	
	0	Disable
	1	Enable
INT12E	Enable or disable Timer 0 Interrupt	
	0	Disable
	1	Enable

IP (Interrupt Priority Register) : B8H

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

Initial value : 00H

IP1 (Interrupt Priority Register 1) : F8H

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

Initial value : 00H

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

IP1x	IPx	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
-	-	-	-	-	-	ENAB1	ENAB0
-	-	-	-	-	-	R/W	R/W

Initial value: 00H

ENAB1	Enable or Disable External Interrupt 1	
	0	Disable External Interrupt 1(default)
	1	Enable External Interrupt 1
ENAB0	Enable or Disable External Interrupt 0	
	0	Disable External Interrupt 0(default)
	1	Enable External Interrupt 0

EIFLAG (External Interrupt Flag Register) : A4H

7	6	5	4	3	2	1	0
-	-	-	-	-	-	FLAG1	FLAG0
-	-	-	-	-	-	R/W	R/W

Initial value: 00H

If External Interrupt is occurred, the flag becomes ‘1’. The flag can be cleared by writing a ‘0’ to bit. It is also cleared automatically

after interrupt service routine is served.

FLAG1 When External Interrupt 1 is occurred this bit is set.

0 External Interrupt 1 is not occurred

1 External Interrupt 1 is occurred

FLAG0 When External Interrupt 0 is occurred this bit is set.

0 External Interrupt 0 is not occurred

1 External Interrupt 0 is occurred

EIEDGE (External Interrupt Edge Register) : A5H

7	6	5	4	3	2	1	0
-	-	–	-	-	-	EDGE1	EDGE0
-	-	-	-	-	-	R/W	R/W

Initial value : 00H

EDGE1 Determines the type of External interrupt 1, edge or level sensitive.

0 Level (default)

1 Edge

EDGE0 Determines the type of External interrupt 0, edge or level sensitive.

0 Level (default)

1 Edge

EIPOLA (External Interrupt Polarity Register) : A6H

7	6	5	4	3	2	1	0
-	-	-	-	-	-	POLA1	POLA0
-	-	-	-	-	-	R/W	R/W

Initial value: 00H

According to EIEDGE, this register acts differently. If EIEDGE is level type, external interrupt polarity have level value. If EIEDGE is edge type, external interrupt polarity have edge value.

POLA1 Determine the polarity of External Interrupt 1

0 When High level or rising edge, Interrupt occur(default)

1 When Low level or falling edge, Interrupt occur

POLA0 Determine the polarity of External Interrupt 0

0 When High level or rising edge, Interrupt occur(default)

1 When Low level or falling edge, Interrupt occur

EIBOTH (External Interrupt Both Edge Enable Register) : A7H

7	6	5	4	3	2	1	0
-	-	-	-	-	-	BOTH1	BOTH0
-	-	-	-	-	-	R/W	R/W

Initial value: 00H

If BOTHx is written to '1', the corresponding external pin interrupt is enabled by both edges (no level).

And EIEDGE and EIPOLA register value are ignored.

BOTH1 Determine the type of External Interrupt 1

0 Both edge detection Disable (default)

1 Both edge detection Enable

BOTH0 Determine the type of External Interrupt 0

0	Both edge detection Disable (default)
1	Both edge detection Enable

11. Peripheral Hardware

11.1 Clock Generator

11.1.1 Overview

As shown in Figure 11-1, the clock generator produces the basic clock pulses which provide the system clock to be supplied to the CPU and the peripheral hardware. It contains main-frequency clock oscillator. The default system clock is INT-RC Oscillator and the default division rate is one. In order to stabilize system internally, use 256kHz RING oscillator for BIT, WDT and ports de-bounce.

- Calibrated Internal RC Oscillator (16 MHz)
  - . INT-RC OSC/1 (16 MHz)
  - . INT-RC OSC/2 (8 MHz)
  - . INT-RC OSC/4 (4 MHz)
  - . INT-RC OSC/8 (2 MHz)

11.1.2 Block Diagram

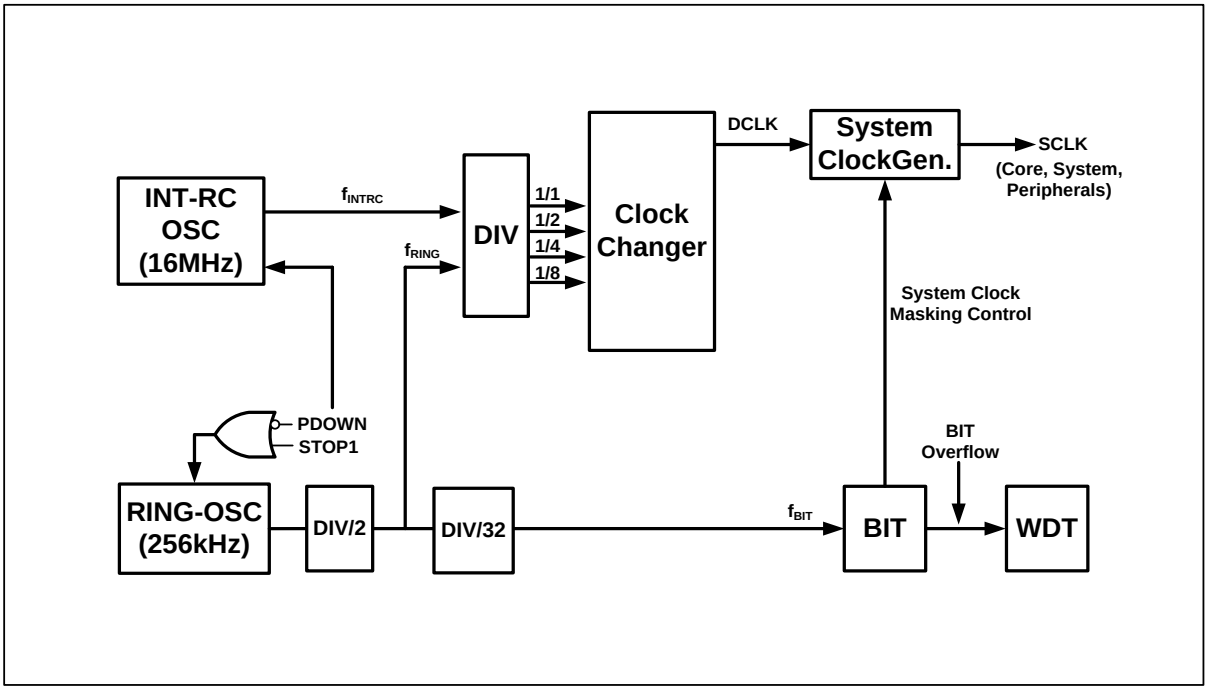


Figure 11-1 Clock Generator Block Diagram

11.1.3 Register Map

Table 11-1 Register Map

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

11.1.4 Clock Generator Register Description

The Clock Generation Register uses clock control for system operation. The clock generation consists of System and Clock register.

11.1.5 Register Description for Clock Generator

SCCR (System and Clock Control Register) : 8AH

7	6	5	4	3	2	1	0
STOP1	DIV1	DIV0	CBYS	ISTOP	REDUCEC	REDUCET	CS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :00H

STOP1 Control the STOP Mode

Note) When PCON=0x03, This bit is applied. When PCON=0x01, This bit is not applied.

0 STOP2 Mode (at PCON=0x03) (default)

1 STOP1 Mode (at PCON=0x03)

DIV[1:0] When using f_{INTRC} as system clock, determine division rate. Note) when using f_{INTRC} as system clock, only division rate come into effect.

Note) To change by software, CBYS set to '1'

DIV1	DIV0	description
0	0	$f_{INTRC/1}$ (16MHz, default)
0	1	$f_{INTRC/2}$ (8MHz)
1	0	$f_{INTRC/4}$ (4MHz)
1	1	$f_{INTRC/8}$ (2MHz)

CBYS Control the scheme of clock change. If this bit set to '0', clock change is controlled by hardware. But if this set to '1', clock change is controlled by software. Ex) when setting CS, if CBYS bit set to '0', it is not changed right now, CPU goes to STOP mode and then when wake-up, it applies to clock change.

Note) when clear this bit, keep other bits in SCCR.

0 Clock changed by hardware during stop mode (default)

1 Clock changed by software

ISTOP Control the operation of INT-RC Oscillation

Note) when CBYS='1', It is applied

0 RC-Oscillation enable (default)

1 RC-Oscillation disable

REDUCEC When STOP1 Mode is set, This bit is applied for reduce current

0 Normal current (default)

1 Reduce current

REDUCET When TS_MODE[2] is set (TOUCH wakeup mode), This bit is

applied for reduce current

0	Normal current (default)
1	Reduce current

CS Determine System Clock

Note) by CBYS bit, reflection point is decided

CS	Description
0	f_{INTRC} INTRC (16 MHz, default)
1	f_{RING} (128 kHz)

11.2 BIT

11.2.1 Overview

The 8-bit Basic Interval Timer that is free-run and can't stop. Block diagram is shown in Figure 11-2. In addition, the Basic Interval Timer generates the time base for watchdog timer counting. It also provides a Basic interval timer interrupt (BITF).

The 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 clock function, time interrupt occurrence

11.2.2 Block Diagram

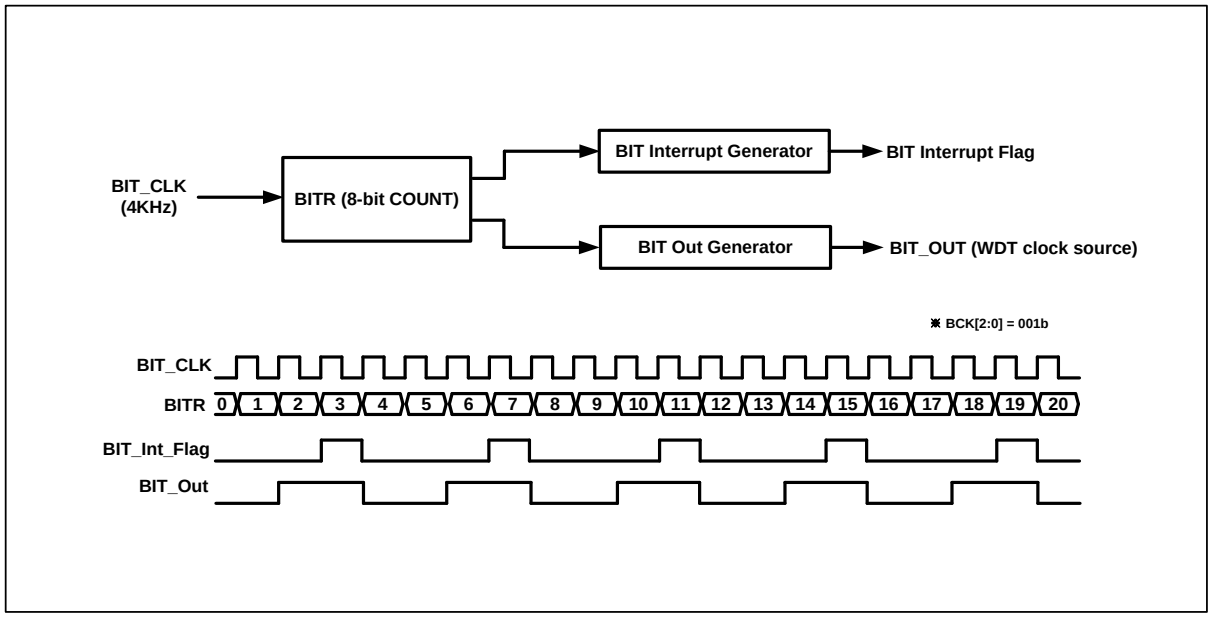


Figure 11-2 BIT Block Diagram

11.2.3 Register Map

Table 11-2 Register Map

Name	Address	Dir	Default	Description
BCCR	8BH	R/W	05H	BIT Clock Control Register
BITR	8CH	R	00H	Basic Interval Timer Register

11.2.4 Bit 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.

11.2.5 Register Description for Bit Interval Timer

BCCR (BIT Clock Control Register) : 8BH

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

Initial value : 05H

BITF When BIT Interrupt occurs, this bit becomes '1'. For clearing bit, write '0' to this bit.

0 no generation

1 generation

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 4KHz)

BCK2	BCK1	BCK0	
0	0	0	0.5msec (BIT Clock * 2)
0	0	1	1msec
0	1	0	2msec
0	1	1	4msec
1	0	0	8msec
1	0	1	16msec (default)
1	1	0	32msec
1	1	1	64msec

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

11.3 WDT

11.3.1 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)

11.3.2 Block Diagram

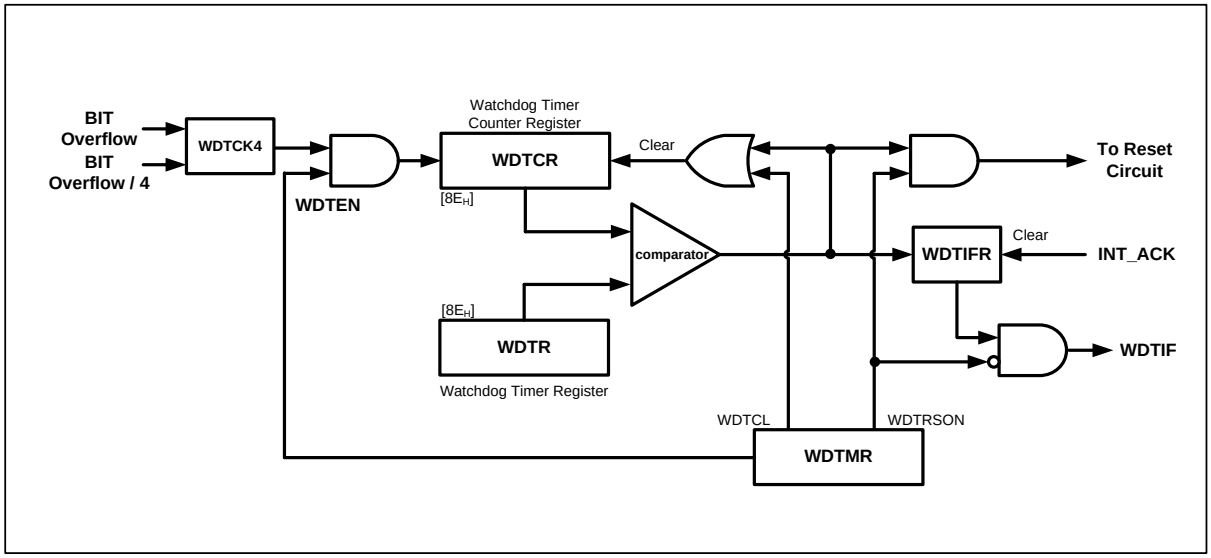


Figure 11-3 WDT Block Diagram

11.3.3 Register Map

Table 11-3 Register Map

Name	Address	Dir	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

11.3.4 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).

11.3.5 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	WDTCK4	-	-	-	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)

WDTCK4	Control WDT clock source
	0 BIT overflow output is clock source (default)
	1 BIT overflow output / 4 is clock source
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

11.3.6 WDT Interrupt Timing Waveform

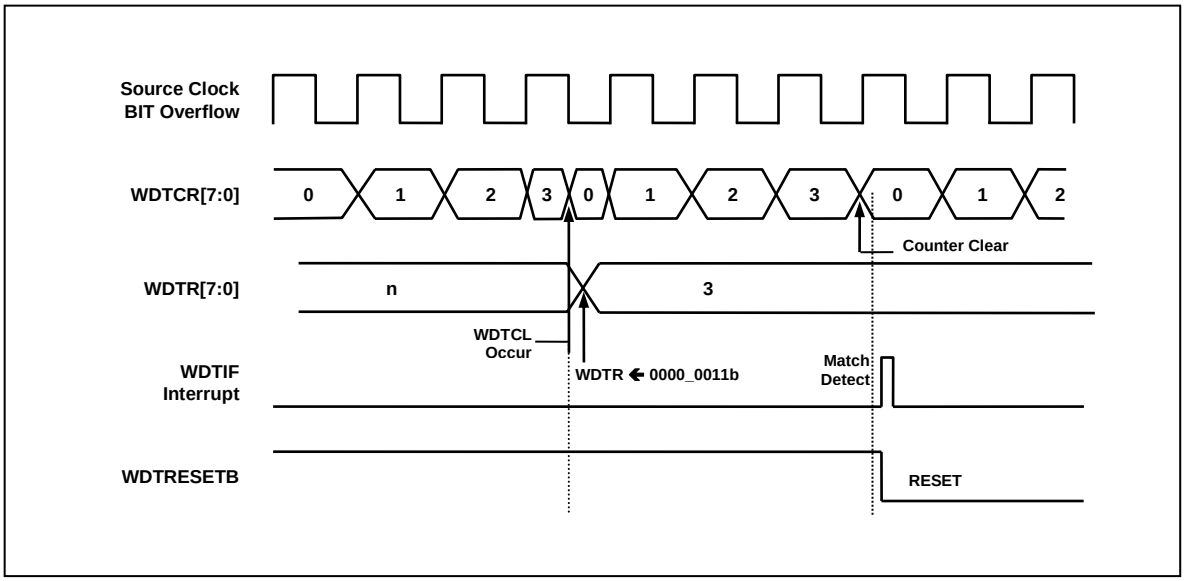


Figure 11-4 WDT Interrupt Timing Waveform

11.4 Free Run Counter

11.4.1 Overview

The clock source of the free run counter is BIT overflow output. The interval of free run counter interrupt is decided by BIT overflow period and 32 bits count value of the four registers. The counter value is the 32 bits value (unsigned long type) and stored to the four registers to easily read. The four registers are (MSB) FCHH, FCHL, FCLH, & FCLL (LSB). If the free run counter starts counting, counter value will be increased with 1 from 0x00000000 to 0xFFFFFFFF. If the counter value is 0xFFFFFFFF, it will be reset to 0x0000000 and continue the counting eternally. If the count period is about 1ms, the full counting time will be about 49 days.

When the count value becomes 0xFFFFFFFF, the interrupt request flag is generated (Option). If the free run counter is not needed, reset the FCEN bit of the Free Run Counter Enable Register (FCMR) to be disable the free run counter.

11.4.2 Block Diagram

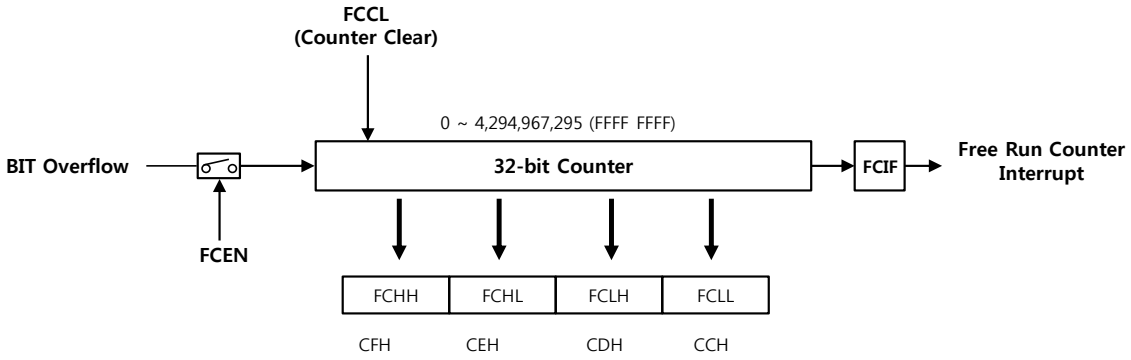


Figure 11-5 Free Run Counter Block Diagram

11.4.3 Register Map

Table 11-4 Register Map

이름	어드레스	방향	Initial value	설명
FCMR	2F64H	R/W	00H	Free Run Counter Enable Register
FCLL	2F63H	R	00H	Free Run Counter Low L
FCLH	2F62H	R	00H	Free Run Counter Low H
FCHL	2F61H	R	00H	Free Run Counter High L
FCHH	2F60H	R	00H	Free Run Counter High H

11.4.4 Free Run Counter Register Description

Free Run Counter consists of Free Run Counter Enable Register (FCMR), Free Run Counter

Register (FCHH, FCHL, FCLH, FCLL).

11.4.5 Register Description for Free Run Counter

FCMR (Free Run Counter Enable Register) : 2F64H

7	6	5	4	3	2	1	0
FCEN	FCCL	-	-	-	-	-	FCIF
W/R	W						W/R

Initial value : 00H

FCEN Free Run Counter Operation Control
- 0 Free Run Counter Disable (Default)
- 1 Free Run Counter Enable

FCCL Free Run Counter Clear

FCIF Free Run Counter Interrupt Flag
- 0 No Free Run Counter Interrupt
- 1 Free Run Counter interrupt occurred, write"0" to clear interrupt flag

FCLL (Free Run Counter Low L) : 2F63H

7	6	5	4	3	2	1	0
FCLL7	FCLL6	FCLL5	FCLL4	FCLL3	FCLL2	FCLL1	FCLL0
R	R	R	R	R	R	R	R

Initial value : 00H

FCLL[7:0] Free Run Counter Low L

FCLH (Free Run Counter Low H) : 2F62H

7	6	5	4	3	2	1	0
FCLH7	FCLH6	FCLH5	FCLH4	FCLH3	FCLH2	FCLH1	FCLH0
R	R	R	R	R	R	R	R

Initial value : 00H

FCLH[7:0] Free Run Counter Low H

FCHL (Free Run Counter High L) : 2F61H

7	6	5	4	3	2	1	0
FCHL7	FCHL6	FCHL5	FCHL4	FCHL3	FCHL2	FCHL1	FCHL0
R	R	R	R	R	R	R	R

Initial value : 00H

FCHL[7:0] Free Run Counter High L

FCHH (Free Run Counter High H) : 2F60H

7	6	5	4	3	2	1	0
FCHH7	FCHH6	FCHH5	FCHH4	FCHH3	FCHH2	FCHH1	FCHH0
R	R	R	R	R	R	R	R

Initial value : 00H

FCHH[7:0] Free Run Counter High H

11.5 Timer/PWM

11.5.1 16-bit Timer/Event Counter 0, 1

11.5.1.1 Overview

The 16-bit timerx consists of Multiplexer, Timer Data Register High/Low, Timer Register High/Low, Timer Mode Control Register, PWM Duty High/Low, PWM Period High/Low Register It is able to use internal 16-bit timer/counter without a port output function.

The 16-bit timer x is able to use the divided clock of the main clock selected from prescaler output.

11.5.1.2 16-Bit Timer/Counter Mode

In the 16-bit Timer/Counter Mode, If the TxH + TxL value and the TxDRH + TxDRL value are matched, Tx/PWMx port outputs. The output is 50:50 of duty square wave, the frequency is following

$$f_{COMP} = \frac{\text{Timer Clock Frequency}}{2 \times \text{Prescaler Value} \times (TxDR + 1)}$$

$f_{COMP}$is timer output frequency and TxDR is the 16 bits value of TxDRH and TxDRL.
To export the compare output as Tx/PWMx, the Tx_PE bit in the TxCR1 register must set to '1'.
The 16-bit Timer/Counter Mode is selected by control registers as shown in Figure 11-6.

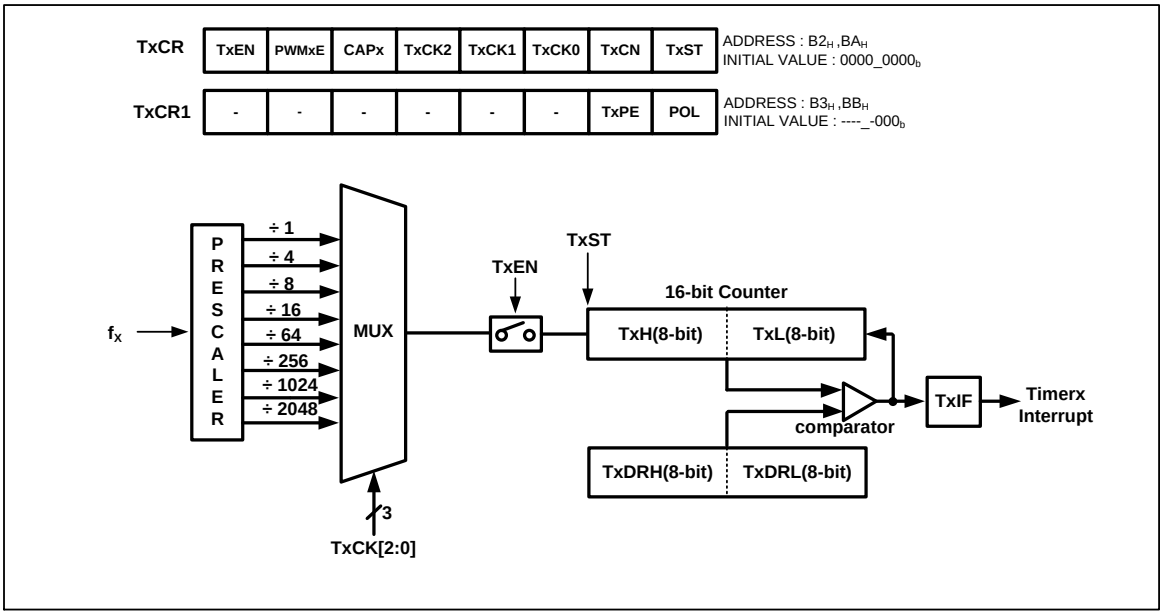


Figure 11-6 Timer x 16-bit Mode Block Diagram

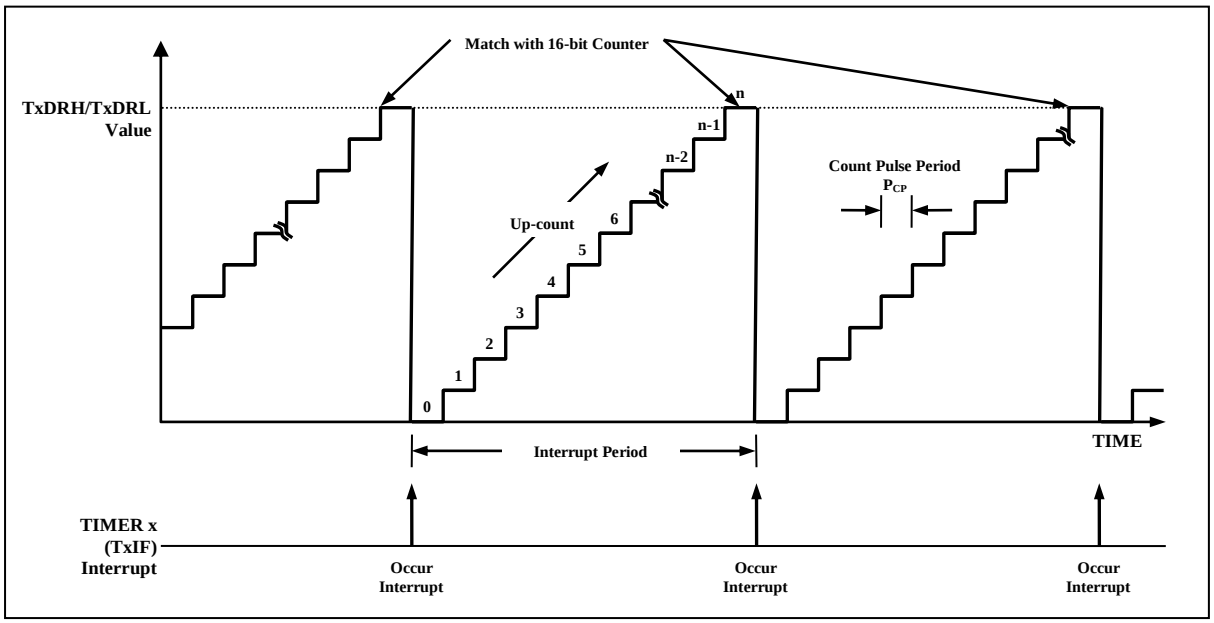


Figure 11-7 Interrupt of Timer/Counter Mode

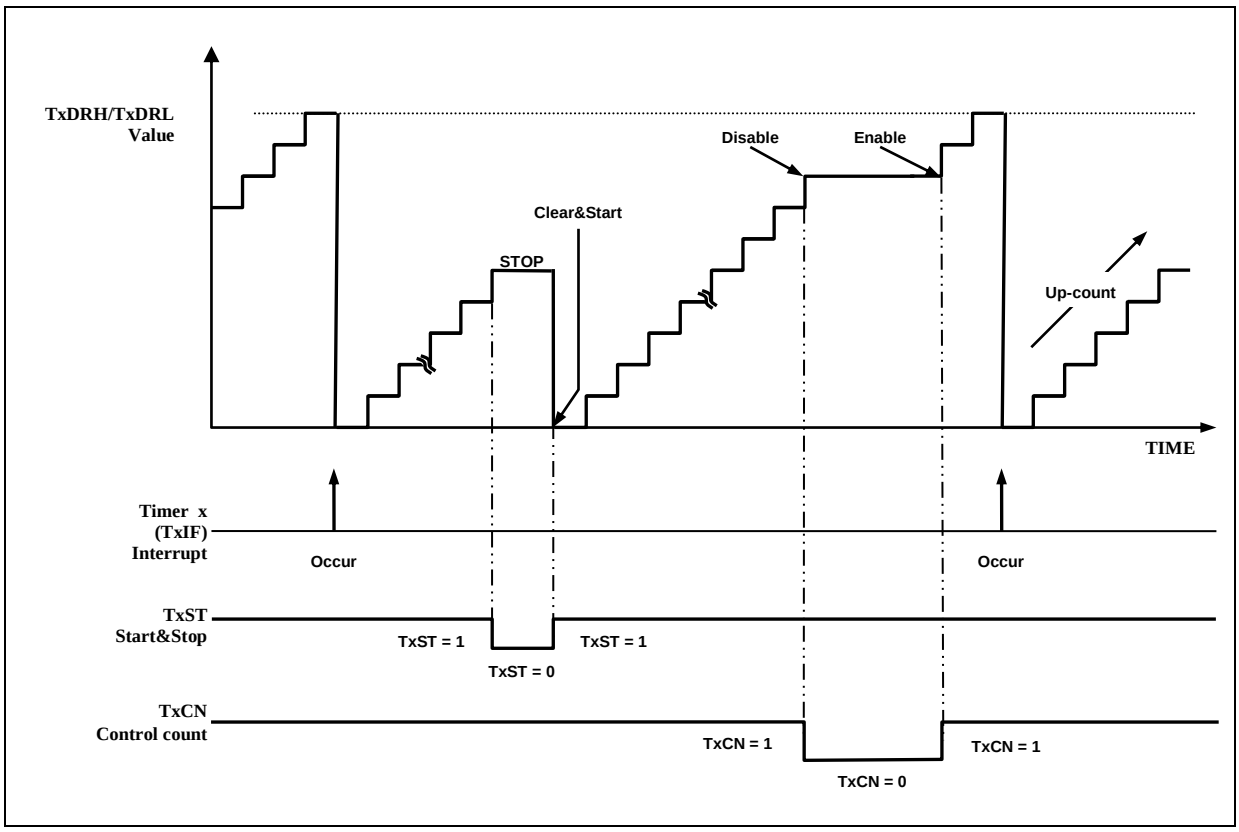


Figure 11-8 Operation Example of Timer/Event Counter Mode

11.5.1.3 16-Bit Capture Mode

The timer x capture mode is set by CAPx as '1' in TxCR register. The clock is same source as Output Compare mode. The interrupt occurs at TxH, TxL and TxDRH, TxDRL matching time. The capture result is loaded into CDRxH, CDRxL. The TxH, TxL value is automatically cleared(0000_H) by hardware and restarts counter.

This timer interrupt in capture mode is very useful when the pulse width of captured signal is wider than the maximum period of timer. As the EIEDGE and EIPOLA register setting, the external interrupt INTx function is chosen.

The CDRxH, PWMxDRH and TxH are in same address. In the capture mode, reading operation is read the CDRxH, not TxH because path is opened to the CDRxH. PWMxDRH will be changed in writing operation. The PWMxDRL, TxL, CDRxL has the same function.

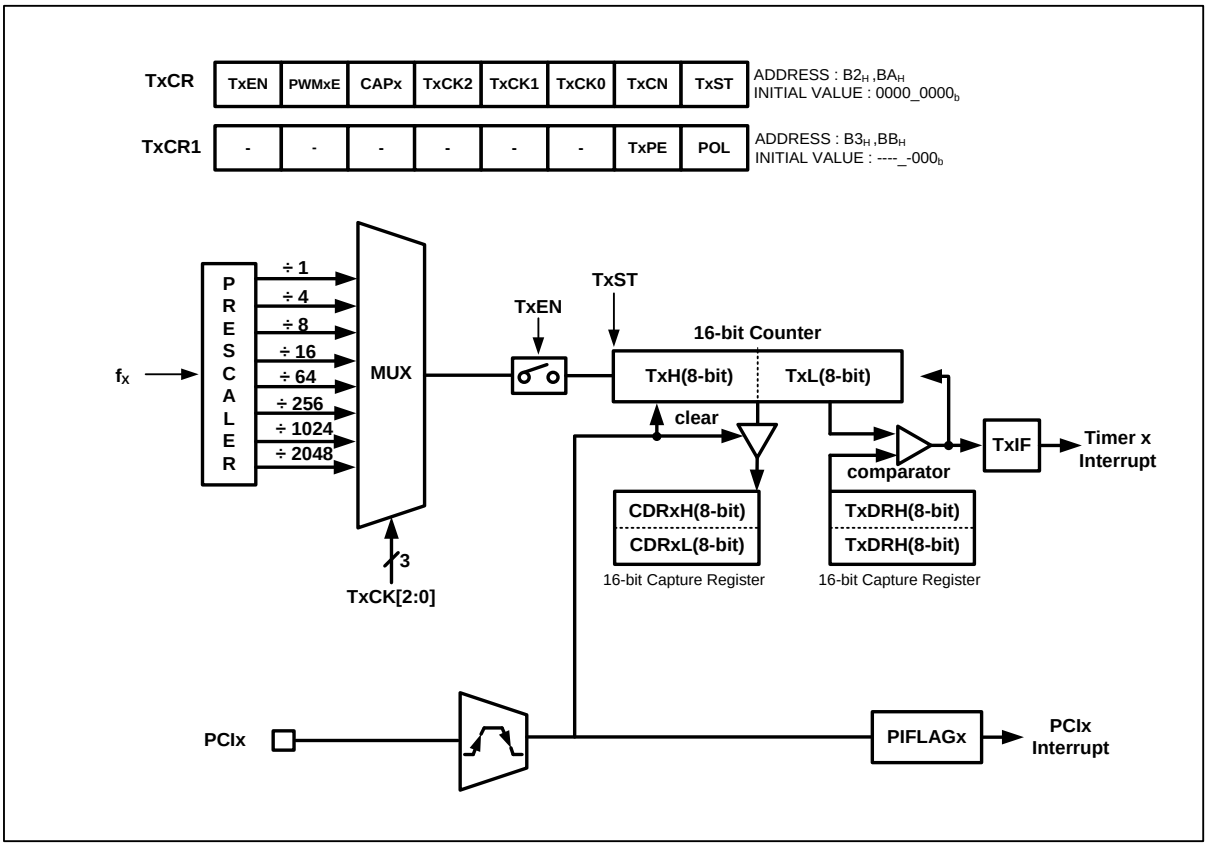


Figure 11-9 Timer x 16bit Capture Mode

.

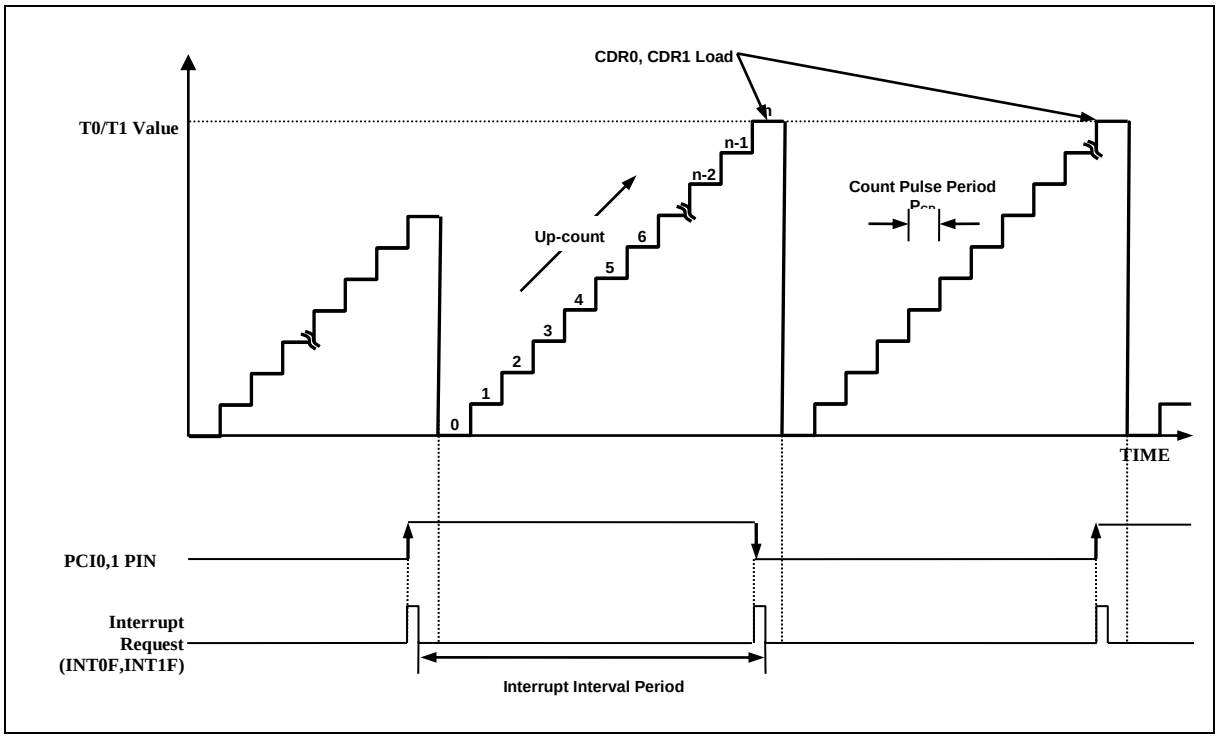


Figure 11-10 Input Capture Mode Operation of Timer 0, 1

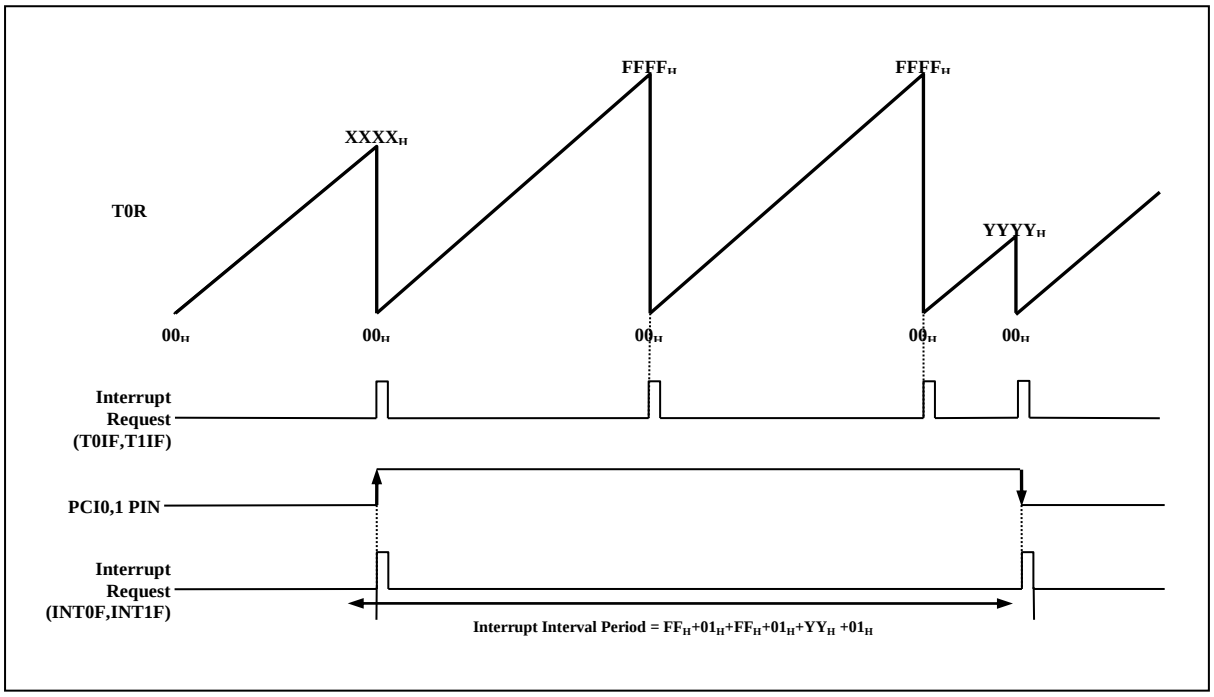


Figure 11-11 Express Timer Overflow in Capture Mode

11.5.1.4 PWM Mode

The timer x has a PWM (pulse Width Modulation) function. In PWM mode, the Tx/PWMx output pin outputs up to 16-bit resolution PWM output. This pin should be configured as a PWM output by set TX_PE to '1'. The PWM output mode is determined by the PWMxPRH, PWMxPRL, PWMxDRH and PWMxDRL. And you should configure PWMxE bit to "1" in TxCR register. PWM Period and Duty same output shown in Figure 11-14 Example of PWM at 8MHz

PWM Period = [PWMxPRH, PWMxPRL] X Timer x Clock Period

PWM Duty = [PWMxDRH, PWMxDRL] X Timer x Clock Period

Table 11-5 PWM Frequency vs. Resolution at 8 MHz

Resolution	Frequency		
	TxCK[2:0]=000 (125ns)	TxCK[2:0]=001(500ns)	TxCK[2:0]=010(1us)
16-bit	122.070Hz	30.469Hz	15.259Hz
15-bit	244.141Hz	60.938Hz	30.518Hz
10-bit	7.8125KHz	1.95KHz	976.563Hz
9-bit	15.625KHz	3.9KHz	1.953KHz
8-bit	31.25KHz	7.8KHz	3.906KHz

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

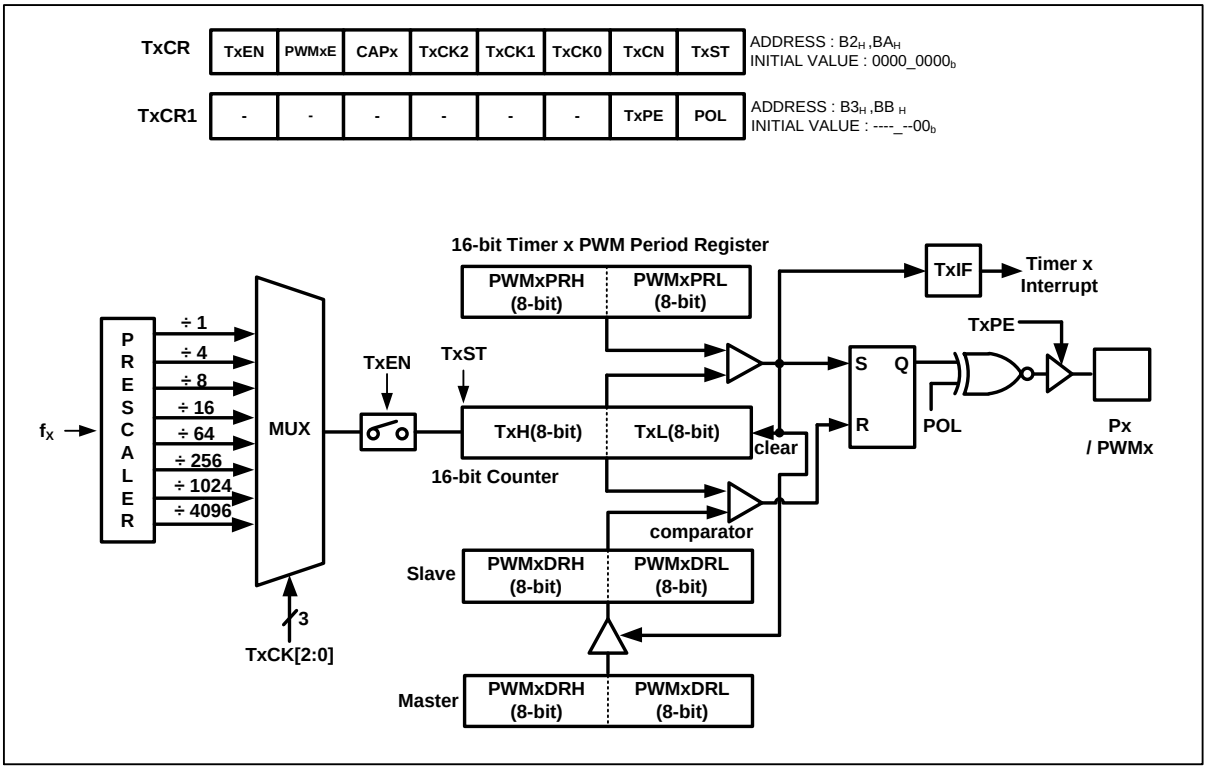


Figure 11-12 PWM Mode

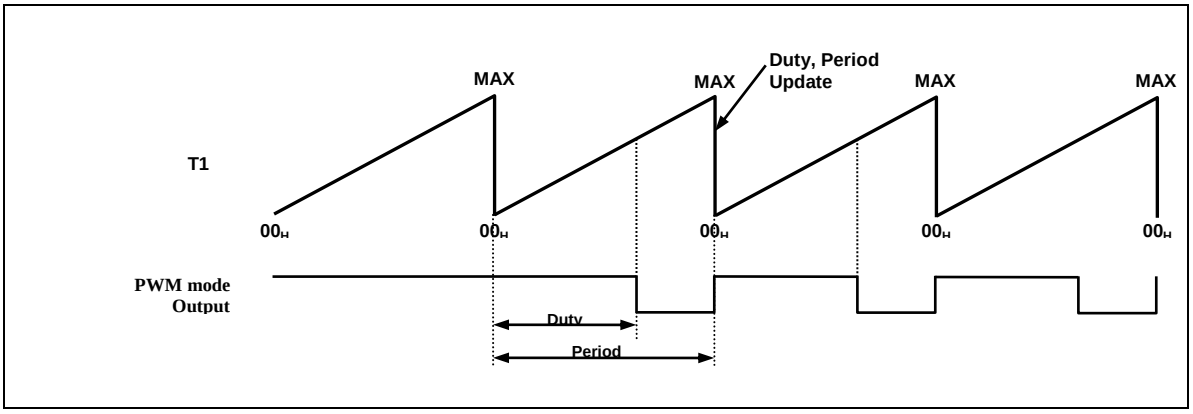


Figure 11-13 Example of PWM Output Waveform

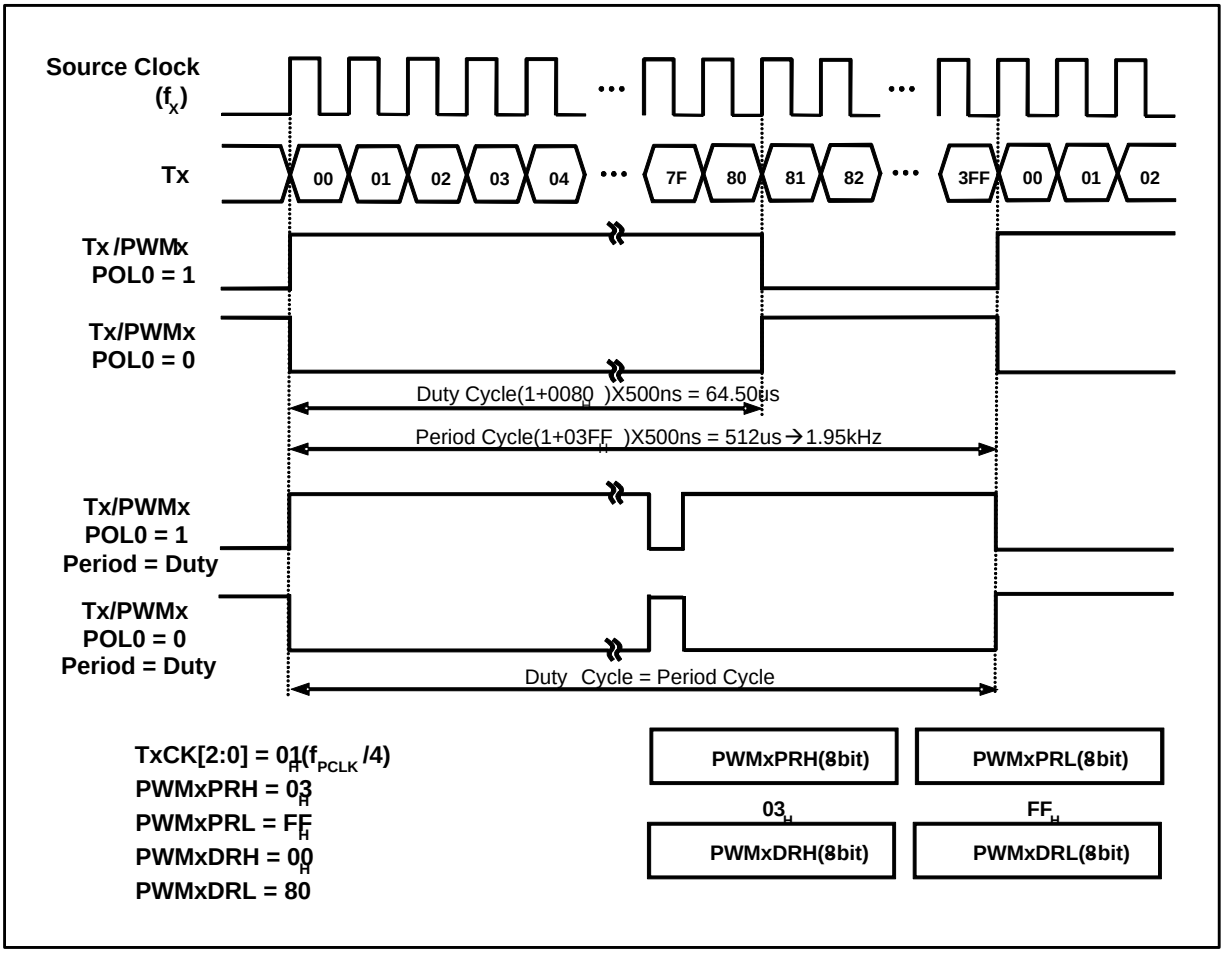


Figure 11-14 Example of PWM at 8MHz

11.5.1.5 Register Map

Table 11-6 Register Map

Name	Address	Dir	Default	Description
T0CR	$B2_H$	R/W	00_H	Timer 0 Mode Control Register
T0CR1	$B3_H$	R/W	00_H	Timer 0 Mode Control Register 1
T0L	$B4_H$	R	00_H	Timer 0 Low Register
PWM0DRL	$B4_H$	R/W	00_H	PWM 0 Duty Register Low
CDR0L	$B4_H$	R	00_H	Timer 0 Capture Data Register Low
T0H	$B5_H$	R	00_H	Timer 0 Register High
PWM0DRH	$B5_H$	R/W	00_H	PWM 0 Duty Register High
CDR0H	$B5_H$	R	00_H	Timer 0 Capture Data Register High
T0DRL	$B6_H$	W	FF_H	Timer 0 Data Register Low
PWM0PRL	$B6_H$	W	FF_H	PWM 0 Period Register Low
T0DRH	$B7_H$	W	FF_H	Timer 0 Data Register High
PWM0PRH	$B7_H$	W	FF_H	PWM 0 Period Register High
T1CR	BA_H	R/W	00_H	Timer 1 Mode Control Register
T1CR1	BB_H	R/W	00_H	Timer 1 Mode Control Register 1
T1L	BC_H	R	00_H	Timer 1 Register Low
PWM1DRL	BC_H	R/W	00_H	PWM 1 Duty Register Low
CDR1L	BC_H	R	00_H	Timer 1 Capture Data Register Low
T1H	BD_H	R	00_H	Timer 1 Register High
PWM1DRH	BD_H	R/W	00_H	PWM 1 Duty Register High
CDR1H	BD_H	R	00_H	Timer 1 Capture Data Register High
T1DRL	BE_H	W	FF_H	Timer 1 Data Register Low
PWM1PRL	BE_H	W	FF_H	PWM 1 Period Register Low
T1DRH	BF_H	W	FF_H	Timer 1 Data Register High
PWM1PRH	BF_H	W	FF_H	PWM 1 Period Register High

11.5.1.6 Timer/Counter x Register Description

The Timer 0~1 Register consists of Timer 0~1 Mode Control Register (T0CR), (T1CR), Timer 0~1 Mode Control Register 1 (T0CR1), (T1CR1),Timer 0~1 Low Register (T0L), (T1L),Timer 0~1 Data Register Low (T0DRL), (T1DRL),Timer 0~1 High Register (T0H), (T1H) Timer 0~1 Data Register High (T0DRH), (T1DRH), Timer 0~1 Capture Data Low Register (CDR0L), (CDR1L), Timer 0~1 Capture Data High Register (CDR0H), (CDR1H), PWM0~1 Low Duty Register (PWM0DRL), (PWM1DRL), PWM0~1 High Duty Register (PWM0DRH), (PWM1DRH), PWM0~1 Low Period Register (PWM0PRL), (PWM1PRL), PWM0~1 High Period Register (PWM0PRH), (PWM1PRH).

11.5.1.7 Register Description for Timer/Counter x

T0CR, T1CR (Timer 0~1 Mode Control Register): B2H, BAH

7	6	5	4	3	2	1	0
TxEN	PWMxE	CAPx	TxCK2	TxCK1	TxCK0	TxCN	TxST
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00_H

TxEN Control Timer X
- 0 0
- 1 Timer X enable

PWMxE Control PWM X enable
- 0 PWM X disable
- 1 PWM X enable

CAPx Control Timer X capture mode.
- 0 Timer mode
- 1 Capture mode

TxCK[2:0] Select clock source of Timer X. Fx is the frequency of main system

TxCK2	TxCK1	TxCK0	description
0	0	0	f_X
0	0	1	f_X /4
0	1	0	f_X /8
0	1	1	f_X /16
1	0	0	f_X /64
1	0	1	f_X /256
1	1	0	f_X /1024
1	1	1	f_X /2048

TxCN Control Timer X Count pause/continue.
- 0 Temporary count stop
- 1 Continue count

TxST Control Timer x start/stop
- 0 Counter stop
- 1 Clear counter and start

Note) set TxST bit after write to Tx, PWMx, CDRx registers.

T0CR1, T1CR1 (Timer 0~1Mode Control Register 1) : B3H, , BBH

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

Initial value : 00_H

T0_PE Control Timer X Output port
- 0 Timer X Output disable
- 1 Timer X Output enable

POL Configure PWM polarity
- 0 Negative (Duty Match: Clear)
- 1 Positive (Duty Match: Set)

T0L, T1L (Timer 0~1Register Low, Read Case) : B4H, BCH

7	6	5	4	3	2	1	0
TxL7	TxL6	TxL5	TxL4	TxL3	TxL2	TxL1	TxL0
R	R	R	R	R	R	R	R

Initial value : 00_H

TxL[7:0] TxL Counter Period Low data.

CDR0L, CDR1L (Capture 0~1 Data Register Low, Read Case) : B4H, BCH

7	6	5	4	3	2	1	0
CDRxL07	CDRxL06	CDRxL05	CDRxL04	CDRxL03	CDRxL02	CDRxL01	CDRxL00
R	R	R	R	R	R	R	R

Initial value : 00_H

CDRxL[7:0] Tx Capture Low data.

PWM0DRL, PWM1DRL (PWM0~1Duty Register Low, Write Case) : B4H, BCH

7	6	5	4	3	2	1	0
PWMxLD7	PWMxLD6	PWMxLD5	PWMxLD4	PWMxLD3	PWMxLD2	PWMxLD1	PWMxLD0
W	W	W	W	W	W	W	W

Initial value : 00_H

PWMxLD[7:0] Tx PWM Duty Low data
Note) Writing is effective only when PWMxE = 1 and T0ST = 0

T0H, T1H (Timer 0~1 Register High, Read Case) : B5H, BDH

7	6	5	4	3	2	1	0
TxH7	TxH6	TxH5	TxH4	TxH3	TxH2	TxH1	TxH0
R	R	R	R	R	R	R	R

Initial value : 00_H

TxH[7:0] TxH Counter Period High data.

CDR0H, CDR1H (Capture 0~1Data High Register, Read Case) : B5H, BDH

7	6	5	4	3	2	1	0
CDRxH07	CDRxH06	CDRxH05	CDRxH04	CDRxH03	CDRxH02	CDRxH01	CDRxH00
R	R	R	R	R	R	R	R

Initial value : 00_H

CDRxH[7:0] Tx Capture High data

PWM0DRH, PWM1DRH (PWM0~1 Duty Register High, Write Case) : B5H, BDH

7	6	5	4	3	2	1	0
PWMxHD7	PWMxHD6	PWMxHD5	PWMxHD4	PWMxHD3	PWMxHD2	PWMxHD1	PWMxHD0
W	W	W	W	W	W	W	W

Initial value : 00_H

PWMxHD[7:0] Tx PWM Duty High data
Note) Writing is effective only when PWMxE = 1 and T0ST = 0

T0DRL, T1DRL (Timer 0~1 Data Register Low, Write Case) : B6H, BEH

7	6	5	4	3	2	1	0
TxLD7	TxLD6	TxLD5	TxLD4	TxLD3	TxLD2	TxLD1	TxLD0
W	W	W	W	W	W	W	W

Initial value : FF_H

TxLD[7:0] TxL Compare Low data

Note) Be sure to clear PWMxE before loading this register.

PWM0PRL, PWM1PRL (PWM 0~1Period Register Low, Write Case) : B6H

7	6	5	4	3	2	1	0
PWM0LP7	PWM0LP6	PWM0LP5	PWM0LP4	PWM0LP3	PWM0LP2	PWM0LP1	PWM0LP0
W	W	W	W	W	W	W	W

Initial value : FF_H

PWM0LP[7:0] T0 PWM Duty Low data
Note) Writing is effective only when PWM0E = 1 and T0ST = 0

T0DRH, T1DRH (Timer 0~1 Data Register High, Write Case) : B7H, BFH

7	6	5	4	3	2	1	0
TxHD7	TxHD6	TxHD5	TxHD4	TxHD3	TxHD2	TxHD1	TxHD0
W	W	W	W	W	W	W	W

Initial value : FF_H

TxHD[7:0] TxH Compare High data

Note) Be sure to clear PWMxE before loading this register.

PWM0PRH, PWM1PRH (PWM 0~1 Period Register High, Write Case) : B7H, BFH

7	6	5	4	3	2	1	0
PWMxHP7	PWMxHP6	PWMxHP5	PWMxHP4	PWMxHP3	PWMxHP2	PWMxHP1	PWMxHP0
R/W	W	W	W	W	W	W	W

Initial value : FF_H

PWMxHP[7:0] Tx PWM Duty High data
Note) Writing is effective only when PWMxE = 1 and T0ST = 0

11.5.2 Timer Interrupt Status Register (TMISR)

11.5.2.1 Register Description for TMISR

TMISR (Timer Interrupt Status Register) : D5H

7	6	5	4	3	2	1	0
-	-	-	-	-	-	TMIF1	TMIF0
-	-	-	-	-	-	R/W	R/W

Initial value : 00H

TMIF1 Timer 1 Interrupt Flag
- 0 No Timer 1 interrupt
- 1 Timer 1 interrupt occurred, write “1” to clear interrupt flag

TMIF0 Timer 0 Interrupt Flag
- 0 No Timer 0 interrupt
- 1 Timer 0 interrupt occurred, write “1” to clear interrupt flag

Note) The Timer Interrupt Status Register contains interrupt information of each timers. Even if user disabled timer interrupt at IE2, user could check timer interrupt condition from this register.

11.6 USART

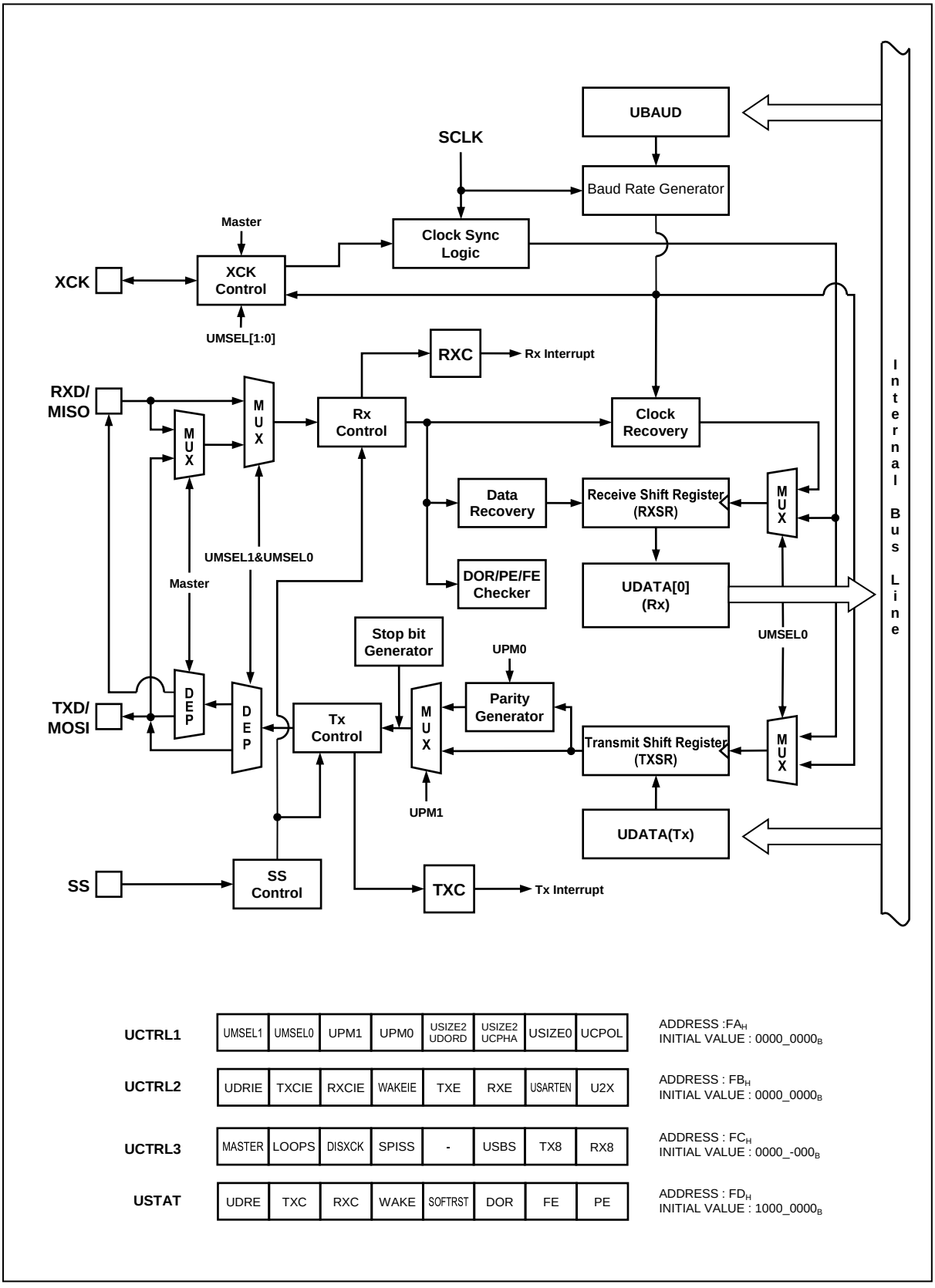
11.6.1 Overview

The Universal Synchronous and Asynchronous serial Receiver and Transmitter (USART) is a highly flexible serial communication device. The main features are listed below.

- Full Duplex Operation (Independent Serial Receive and Transmit Registers)
- Asynchronous or Synchronous Operation
- Master or Slave Clocked Synchronous and SPI Operation
- Supports all four SPI Modes of Operation (Mode 0, 1, 2, 3)
- LSB First or MSB First Data Transfer @SPI mode
- High Resolution Baud Rate Generator
- Supports Serial Frames with 5,6,7,8, or 9 Data Bits and 1 or 2 Stop Bits
- Odd or Even Parity Generation and Parity Check Supported by Hardware
- Data OverRun Detection
- Framing Error Detection
- Digital Low Pass Filter
- Three Separate Interrupts on TX Complete, TX Data Register Empty and RX Complete
- Double Speed Asynchronous Communication Mode

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

11.6.2 Block Diagram



UCTRL1	UMSEL1	UMSEL0	UPM1	UPM0	USIZE2 UDORD	USIZE2 UCPHA	USIZE0	UCPOL	ADDRESS :FA_H INITIAL VALUE : 0000_0000_B
UCTRL2	UDRIE	TXCIE	RXCIE	WAKEIE	TXE	RXE	USARTEN	U2X	ADDRESS : FB_H INITIAL VALUE : 0000_0000_B
UCTRL3	MASTER	LOOPS	DISXCK	SPISS	-	USBS	TX8	RX8	ADDRESS : FC_H INITIAL VALUE : 0000_-000_B
USTAT	UDRE	TXC	RXC	WAKE	SOFTRST	DOR	FE	PE	ADDRESS : FD_H INITIAL VALUE : 1000_0000_B

Figure 11-15 USART Block Diagram

11.6.3 Clock Generation

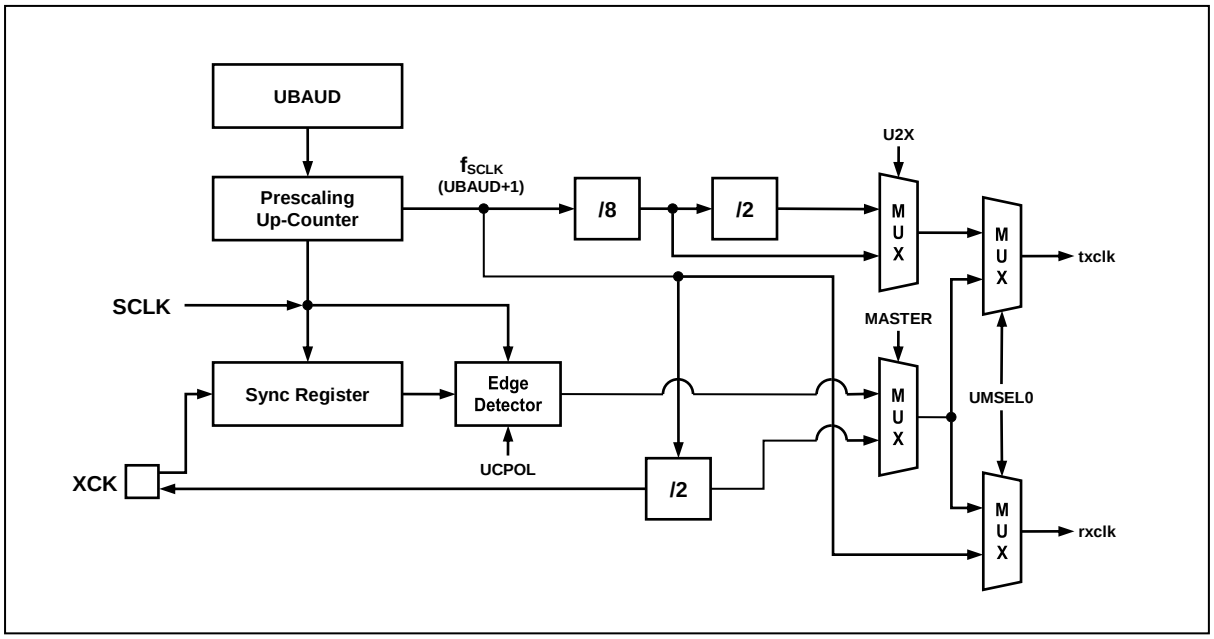


Figure 11-16 Clock Generation Block Diagram

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

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

Table 11-7 Equations for Calculating Baud Rate Register Setting

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	$\text{Baud Rate} = \frac{\text{fSCLK}}{2(\text{UBAUDx}+1)}$

11.6.4 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 therefore 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).

11.6.5 Synchronous Mode Operation

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

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

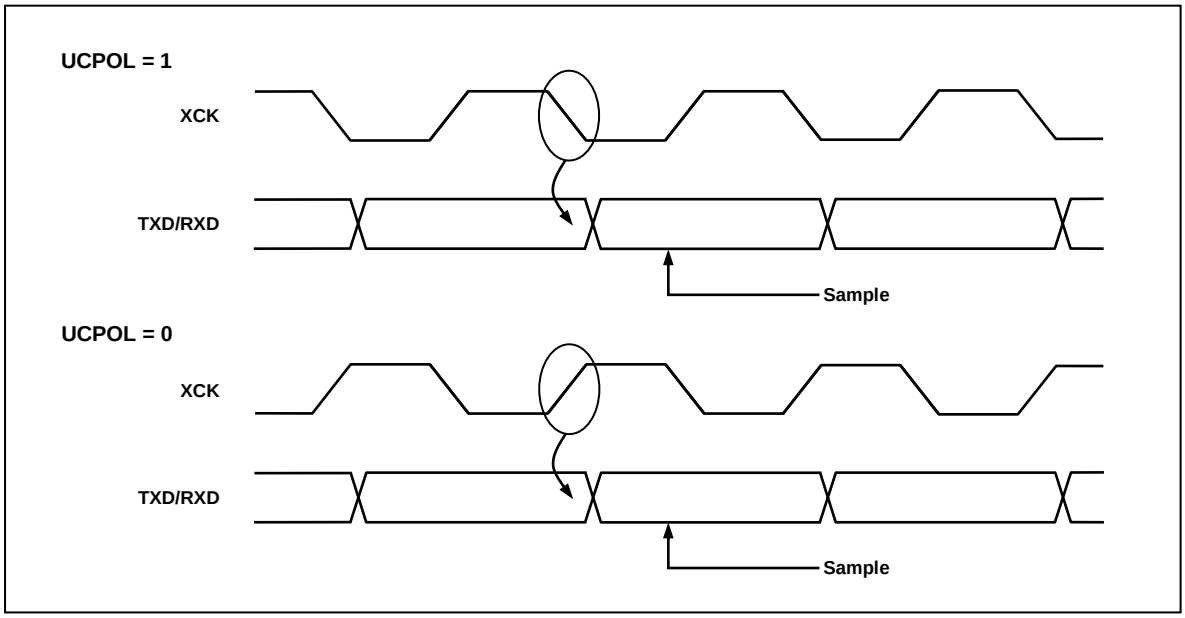


Figure 11-17 Synchronous Mode XCKn Timing

11.6.6 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.

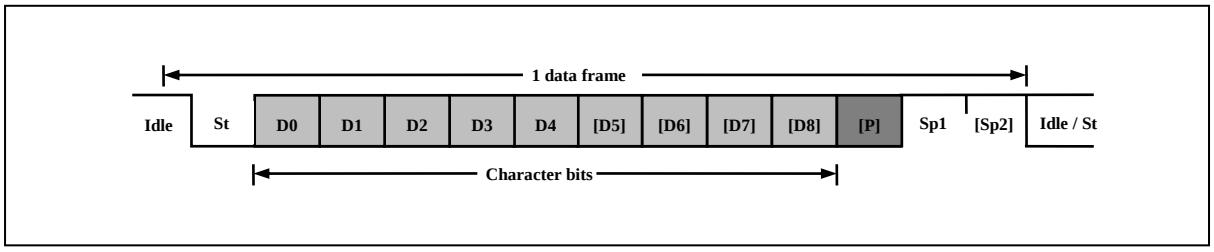


Figure 11-18 Frame Format

1 data frame consists of the following bits

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

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

11.6.7 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 the MSB and first stop bit of a serial frame.

$P_{even} = D_{n-1} \wedge \ldots \wedge D_3 \wedge D_2 \wedge D_1 \wedge D_0 \wedge 0$
$P_{odd} = D_{n-1} \wedge \ldots \wedge D_3 \wedge D_2 \wedge D_1 \wedge D_0 \wedge 1$
P_{even} : Parity bit using even parity
P_{odd} : Parity bit using odd parity
D_n : Data bit n of the character

11.6.8 USART Transmitter

The USART Transmitter is enabled by setting the TXE bit in UCTRL2 register. When the Transmitter is enabled, the normal port operation of the TXD pin is overridden by the serial output pin of USART. The baud-rate, operation mode and frame format must be setup once before doing any transmissions. If synchronous or spi operation is used, the clock on the XCK pin will be overridden and used as transmission clock. If USART operates in spi mode, SS pin is used as SS input pin in slave mode or can be configured as SS output pin in master mode. This can be done by setting SPISS bit in UCTRL3 register.

11.6.8.1 Sending Tx data

A data transmission is initiated by loading the transmit buffer (UDATA register I/O location) with the data to be transmitted. The data written in transmit buffer is moved to the shift register when the shift register is ready to send a new frame. The shift register is loaded with the new data if it is in idle state or immediately after the last stop bit of the previous frame is transmitted. When the shift register is loaded with new data, it will transfer one complete frame at the settings of control registers. If the 9-bit characters are used in asynchronous or synchronous operation mode (USIZE[2:0]=7), the ninth bit must be written to the TX8 bit in UCTRL3 register before loading transmit buffer (UDATA register).

11.6.8.2 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 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 data to be transmitted that has not yet been moved into the shift register. And also this flag can be cleared by writing ‘0’ to this bit position. Writing ‘1’ to this bit position is prevented.

When the Data Register Empty Interrupt Enable (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.

11.6.8.3 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 MSB and the first stop bit of the sending frame.

11.6.8.4 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.

11.6.9 USART Receiver

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

11.6.9.1 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 2^{nd} stop bit in the frame, the 2^{nd} stop bit is ignored by the Receiver. That is, receiving the first stop bit means that a complete serial frame is present in the receiver shift register and contents of the shift register are to be moved into the receive buffer. The receive buffer is read by reading the UDATA register.

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

11.6.9.2 Receiver flag and interrupt

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

The Receive Complete (RXC) flag indicates whether there are unread data present in the receive buffer. This flag is set when there are unread data in the receive buffer and cleared when the receive buffer is empty. If the Receiver is disabled (RXE=0), the receiver buffer is flushed and the RXC flag is cleared.

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

The USART Receiver has three error flags which are Frame Error (FE), Data OverRun (DOR) and Parity Error (PE). These error flags can be read from the USTAT register. As data received are stored in the 2-level receive buffer, these error flags are also stored in the same position of receive buffer. So, before reading received data from UDATAregister, read the USTAT register first which contains error flags.

The Frame Error (FE) flag indicates the state of the first stop bit. The FE flag is zero when the stop bit was correctly detected as one, and the FE flag is one when the stop bit was incorrect, ie detected as zero. 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 zero.

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

11.6.9.3 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.

11.6.9.4 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.

11.6.9.5 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 and low pass filters the incoming bits, 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.

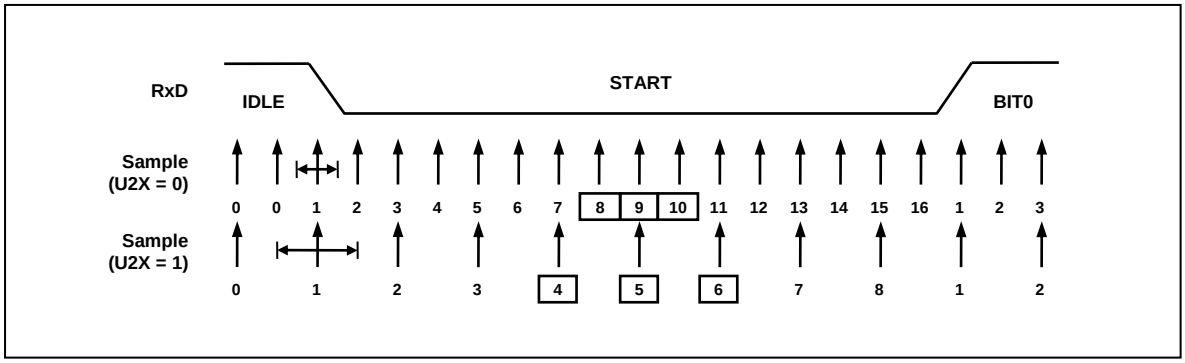


Figure 11-19 Start Bit Sampling

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.

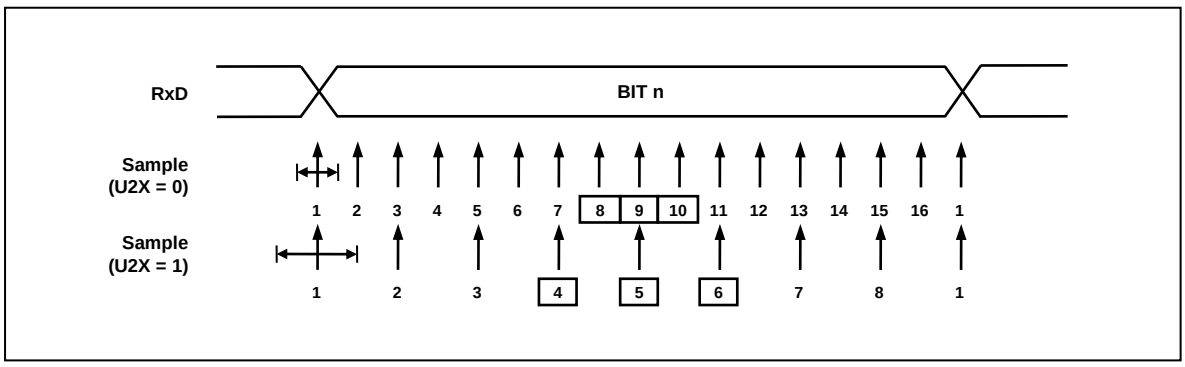


Figure 11-20 Sampling of Data and Parity Bit

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 goes idle state and monitors the RXD line to check a valid high to low transition is detected (start bit detection).

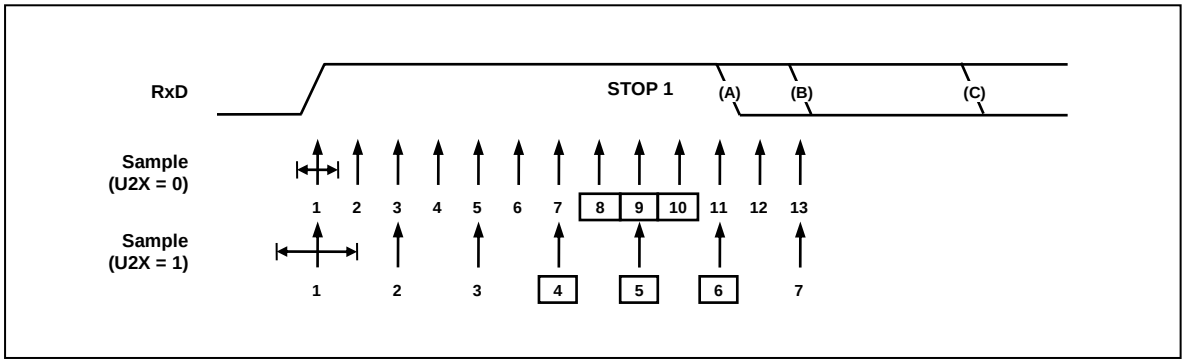


Figure 11-21 Stop Bit Sampling and Next Start Bit Sampling

11.6.10 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.

11.6.10.1 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 chooses between two different clock phase relationships between the clock and data. Note that UCPHA and UCPOL bits in UCTRL1 register have different meanings according to the UMSEL[1:0] bits which decides the operating mode of USART.

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

Table 11-8 CPOL Functionality

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)

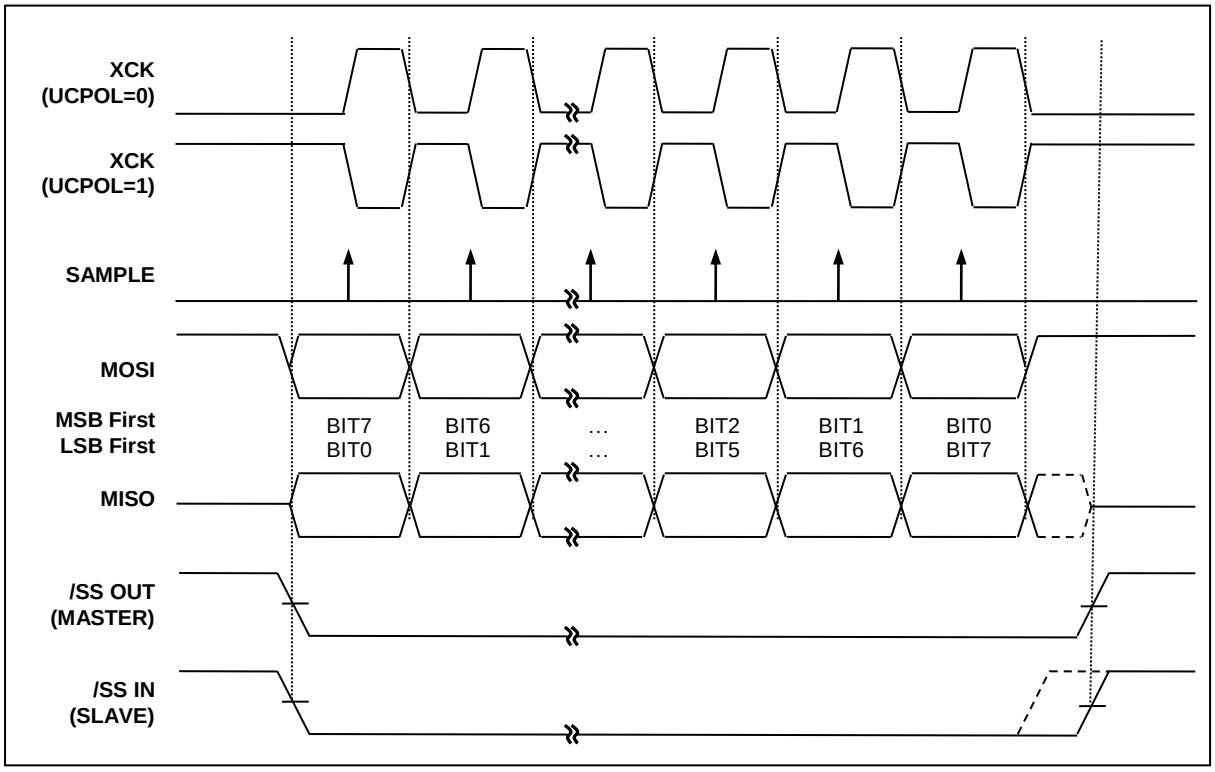


Figure 11-22 SPI Clock Formats when UCPHA=0

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.

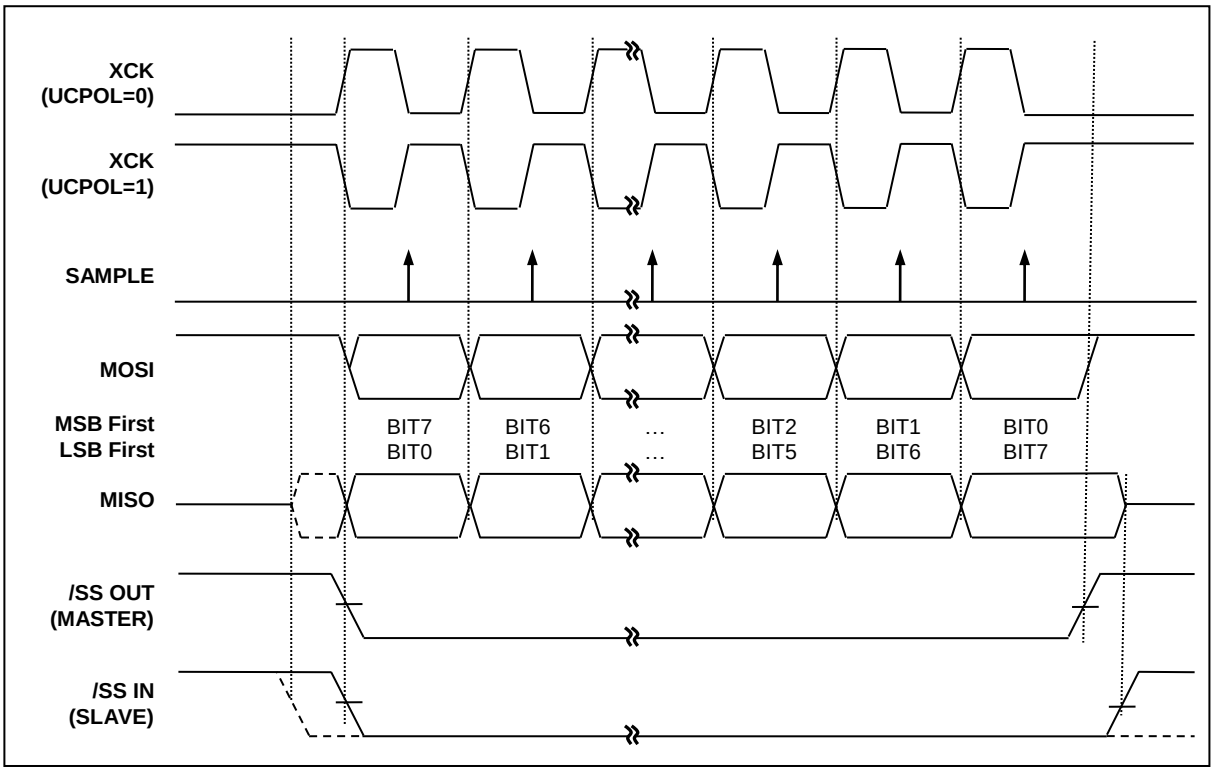


Figure 11-23 SPI Clock Formats when UCPHA=1

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.

11.6.11 Register Map

Table 11-9 Register Map

Name	Address	Dir	Default	Description
UCTRL1	FAH	R/W	00H	USART Control 1 Register
UCTRL2	FBH	R/W	00H	USART Control 2 Register
UCTRL3	FCH	R/W	00H	USART Control 3 Register
USTAT	FDH	R	80H	USART Status Register
UBAUD	FEH	R/W	FFH	USART Baud Rate Generation Register
UDATA	FFH	R/W	FFH	USART Data Register

11.6.12 USART Register Description

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

11.6.13 Register Description for USART

UCTRL1 (USART Control 1 Register) : 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.

USIZE2	USIZE1	USIZE0	Data length
0	0	0	5 bit
0	0	1	6 bit
0	1	0	7 bit
0	1	1	8 bit
1	0	0	Reserved
1	0	1	Reserved

1	1	0	Reserved
1	1	1	9 bit

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

0 LSB First

1 MSB First

UCPOL Selects polarity of XCK in synchronous or spi mode

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

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

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

UCPOL	UCPHA	Leading Edge	Trailing Edge
0	0	Sample (Rising)	Setup (Falling)
0	1	Setup (Rising)	Sample (Falling)
1	0	Sample (Falling)	Setup (Rising)
1	1	Setup (Falling)	Sample (Rising)

UCTRL2 (USART Control 2 Register) : 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

UCTRL3 (USART Control 3 Register) : 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'

USTAT (USART Status Register) : FDH

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

Initial value : 80_H

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

0 Transmit buffer is not empty.

1 Transmit buffer is empty.

TXC This flag is set when the entire frame in the transmit shift register has been shifted out and there is no new data currently present in the transmit buffer. This flag is automatically cleared when the interrupt service routine of a TXC interrupt is executed. It is also cleared by writing '0' to this bit position. This flag can generate a TXC interrupt.

0 Transmission is ongoing.

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

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

0 There is no data unread in the receive buffer

1 There are more than 1 data in the receive buffer

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

0 No WAKE interrupt is generated.

1 WAKE interrupt is generated.

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

0 No operation

1 Reset USART

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

0 No Data OverRun

1 Data OverRun detected

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

0 No Frame Error

1 Frame Error detected

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

0 No Parity Error

1 Parity Error detected

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

UBAUD (USART Baud-Rate Generation Register) : 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.

UDATA (USART Data Register) : 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.

11.6.14 Baud Rate Setting (example)

Table 11-10 Examples of UBAUD Settings for Commonly Used Oscillator Frequencies

(a) fOSC = Multiples of 1MHz

Baud Rate	fOSC =1.00MHz				fOSC =2.00MHz				fOSC =4.00MHz			
	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%	51	0.2%	103	0.2%	103	0.2%	207	0.2%
4800	12	0.2%	25	0.2%	25	0.2%	51	0.2%	51	0.2%	103	0.2%
9600	6	-7.0%	12	0.2%	12	0.2%	25	0.2%	25	0.2%	51	0.2%
14.4K	3	8.5%	8	-3.5%	8	-3.5%	16	2.1%	16	2.1%	34	-0.8%
19.2K	2	8.5%	6	-7.0%	6	-7.0%	12	0.2%	12	0.2%	25	0.2%
28.8K	1	8.5%	3	8.5%	3	8.5%	8	-3.5%	8	-3.5%	16	2.1%
38.4K	1	-18.6%	2	8.5%	2	8.5%	6	-7.0%	6	-7.0%	12	0.2%
57.6K	-	-	1	8.5%	1	8.5%	3	8.5%	3	8.5%	8	-3.5%
76.8K	-	-	1	-18.6%	1	-18.6%	2	8.5%	2	8.5%	6	-7.0%
115.2K	-	-	-	-	-	-	1	8.5%	1	8.5%	3	8.5%
230.4K	-	-	-	-	-	-	-	-	-	-	1	8.5%
250K	-	-	-	-	-	-	-	-	-	-	1	0.0%
0.5M	-	-	-	-	-	-	-	-	-	-	-	-

Baud Rate	fOSC =8.00MHz				fOSC =16.00MHz			
	U2X=0		U2X=1		U2X=0		U2X=1	
	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR
2400	207	0.2%	-	-	-	-	-	-
4800	103	0.2%	207	0.2%	207	0.2%	-	-
9600	51	0.2%	103	0.2%	103	0.2%	207	0.2%
14.4K	34	-0.8%	68	0.6%	68	0.6%	138	-0.1%
19.2K	25	0.2%	51	0.2%	51	0.2%	103	0.2%
28.8K	16	2.1%	34	-0.8%	34	-0.8%	68	0.6%
38.4K	12	0.2%	25	0.2%	25	0.2%	51	0.2%
57.6K	8	-3.5%	16	2.1%	16	2.1%	34	-0.8%
76.8K	6	-7.0%	12	0.2%	12	0.2%	25	0.2%
115.2K	3	8.5%	8	-3.5%	8	-3.5%	16	2.1%
230.4K	1	8.5%	3	8.5%	3	8.5%	8	-3.5%
250K	1	0.0%	3	0.0%	3	0.0%	7	0.0%
0.5M	-	-	1	0.0%	1	0.0%	3	0.0%

(b) fOSC = Multiples of 1.8342MHz

Baud Rate	fOSC =1.8432MHz (x1)				fOSC =3.6864MHz (x2)				fOSC =7.3728MHz (x4)			
	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	47	0.0%	95	0.0%	95	0.0%	191	0.0%	191	0.0%	-	-
4800	23	0.0%	47	0.0%	47	0.0%	95	0.0%	95	0.0%	191	0.0%
9600	11	0.0%	23	0.0%	23	0.0%	47	0.0%	47	0.0%	95	0.0%
14.4K	7	0.0%	15	0.0%	15	0.0%	31	0.0%	31	0.0%	63	0.0%
19.2K	5	0.0%	11	0.0%	11	0.0%	23	0.0%	23	0.0%	47	0.0%
28.8K	3	0.0%	7	0.0%	7	0.0%	15	0.0%	15	0.0%	31	0.0%
38.4K	2	0.0%	5	0.0%	5	0.0%	11	0.0%	11	0.0%	23	0.0%
57.6K	1	-25.0%	3	0.0%	3	0.0%	7	0.0%	7	0.0%	15	0.0%
76.8K	1	0.0%	2	0.0%	2	0.0%	5	0.0%	5	0.0%	11	0.0%
115.2K	-	-	1	0.0%	1	0.0%	3	0.0%	3	0.0%	7	0.0%
230.4K	-	-	-	-	-	-	1	0.0%	1	0.0%	3	0.0%
250K	-	-	-	-	-	-	1	-7.8%	1	-7.8%	3	-7.8%
0.5M	-	-	-	-	-	-	-	-	-	-	1	-7.8%

Baud Rate	fOSC =11.0592MHz (x6)				fOSC =14.7456MHz (x8)							
	U2X=0		U2X=1		U2X=0		U2X=1					
	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR				
2400	-	-	-	-	-	-	-	-				
4800	143	0.0%	-	-	191	0.0%	-	-				
9600	71	0.0%	143	0.0%	95	0.0%	191	0.0%				
14.4K	47	0.0%	95	0.0%	63	0.0%	127	0.0%				
19.2K	35	0.0%	71	0.0%	47	0.0%	95	0.0%				
28.8K	23	0.0%	47	0.0%	31	0.0%	63	0.0%				
38.4K	17	0.0%	35	0.0%	23	0.0%	47	0.0%				
57.6K	11	0.0%	23	0.0%	15	0.0%	31	0.0%				
76.8K	8	0.0%	17	0.0%	11	0.0%	23	0.0%				
115.2K	5	0.0%	11	0.0%	7	0.0%	15	0.0%				
230.4K	2	0.0%	5	0.0%	3	0.0%	7	0.0%				
250K	2	-7.8%	5	-7.8%	3	-7.8%	6	5.3%				
0.5M	-	-	2	-7.8%	1	-7.8%	3	-7.8%				

11.7 I²C

11.7.1 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 400 KHz data transfer speed
- 7 bit address
- Support 2 slave addresses
- Both master and slave operation
- Bus busy detection

11.7.2 Block Diagram

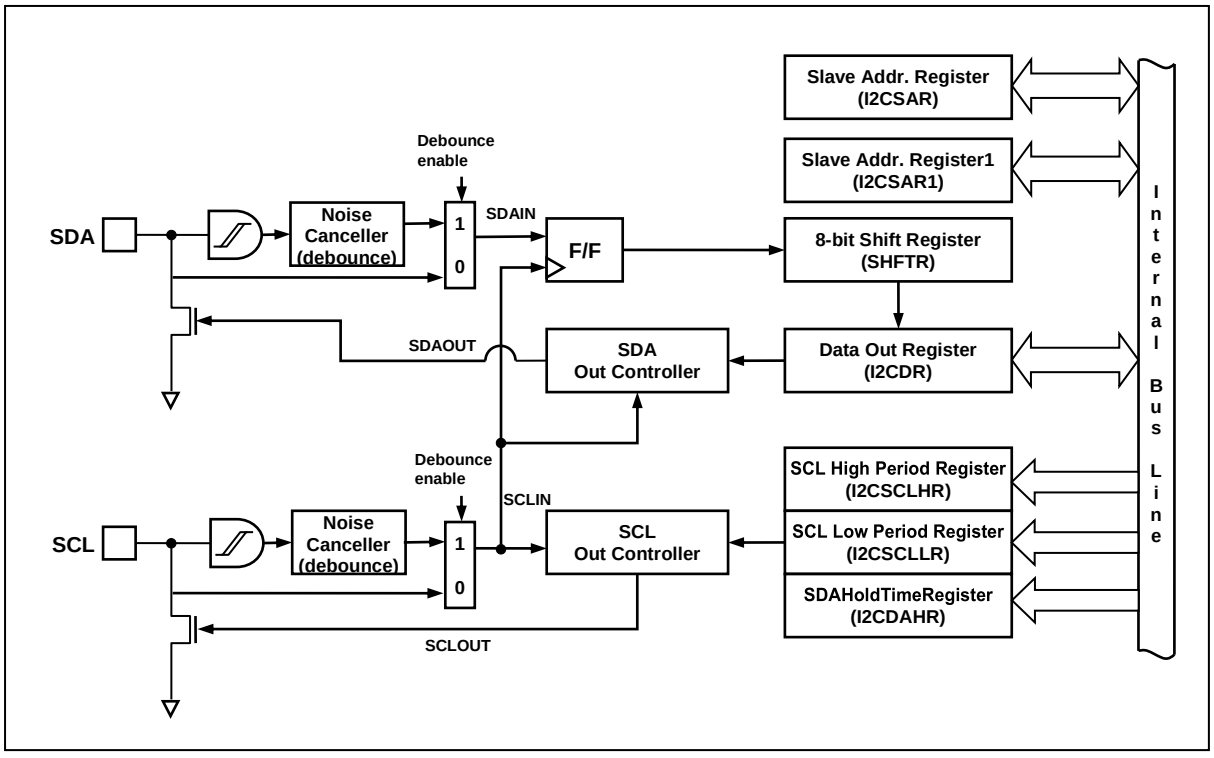


Figure 11-24 I²C Block Diagram

11.7.3 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.

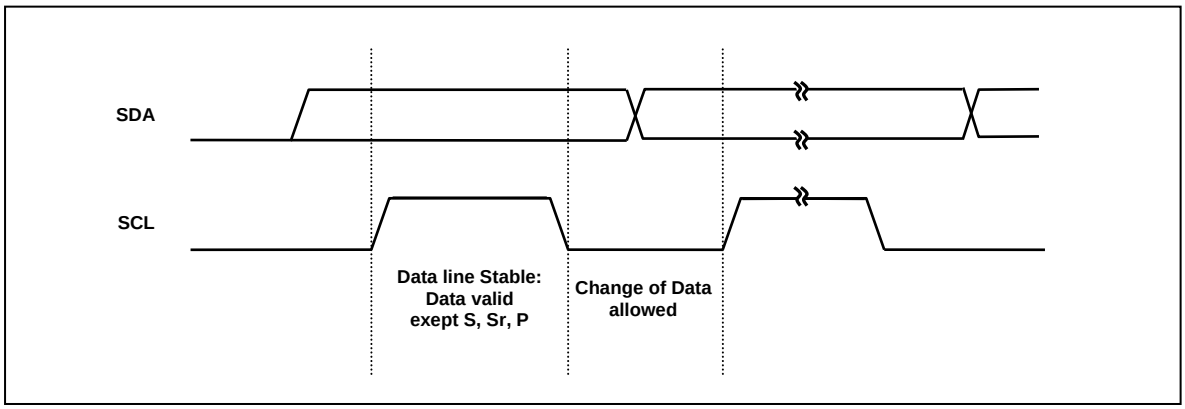


Figure 11-25 Bit Transfer on the I²C-Bus

11.7.4 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.

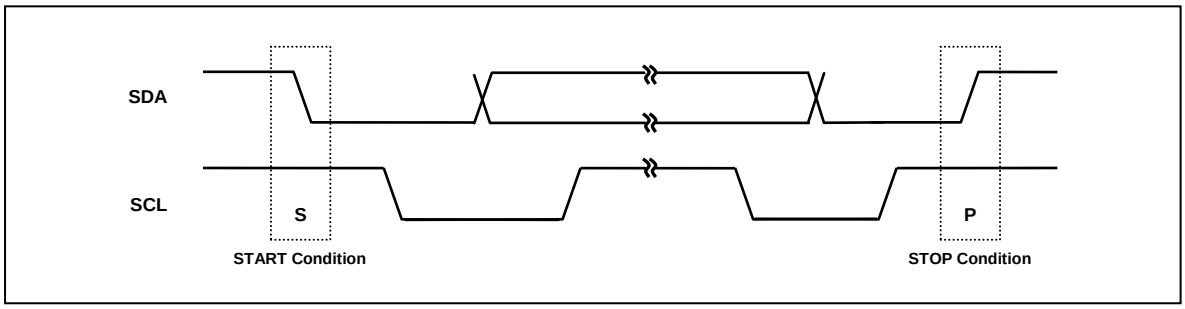


Figure 11-26 START and STOP Condition

11.7.5 Data Transfer

Every byte put on the SDA line must be 8-bits long. The number of bytes that can be transmitted per transfer is unlimited. Each byte has to be followed by an acknowledge bit. Data is transferred with the most significant bit (MSB) first. If a slave can't receive or transmit another complete byte of data until it has performed some other function, it can hold the clock line 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.

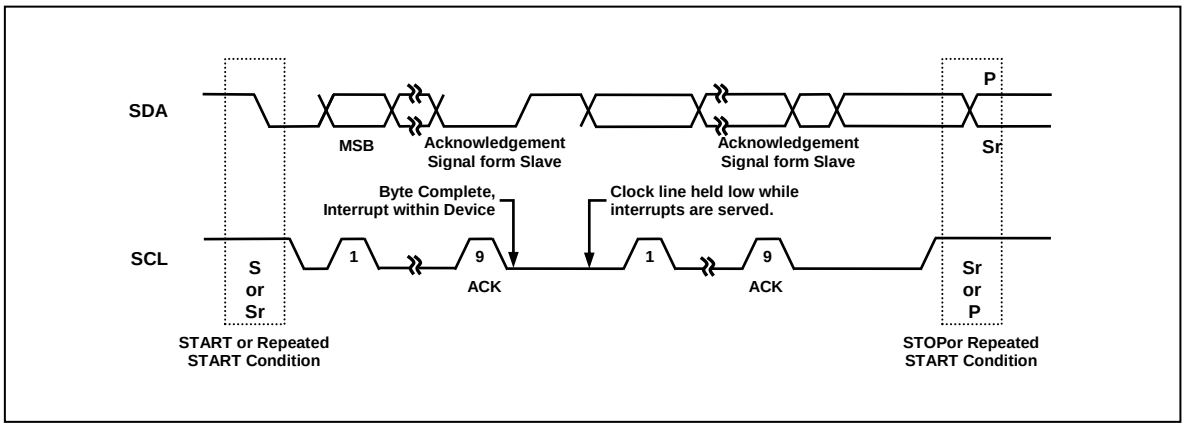


Figure 11-27 Data Transfer on the I²C-Bus

11.7.6 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.

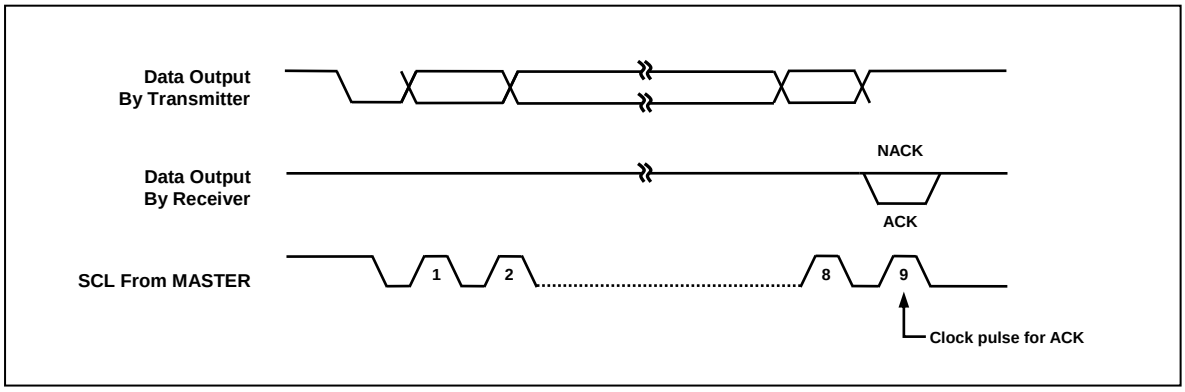


Figure 11-28 Acknowledge on the I²C-Bus

11.7.7 Synchronization / Arbitration

Clock synchronization is performed 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 I^2C bus. Its first stage is comparison of the address bits.

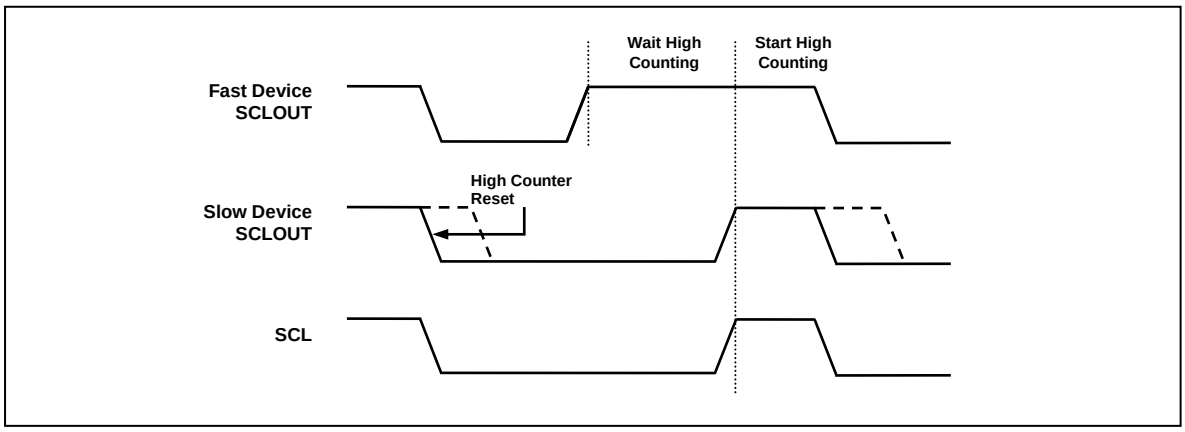


Figure 11-29 Clock Synchronization during Arbitration Procedure

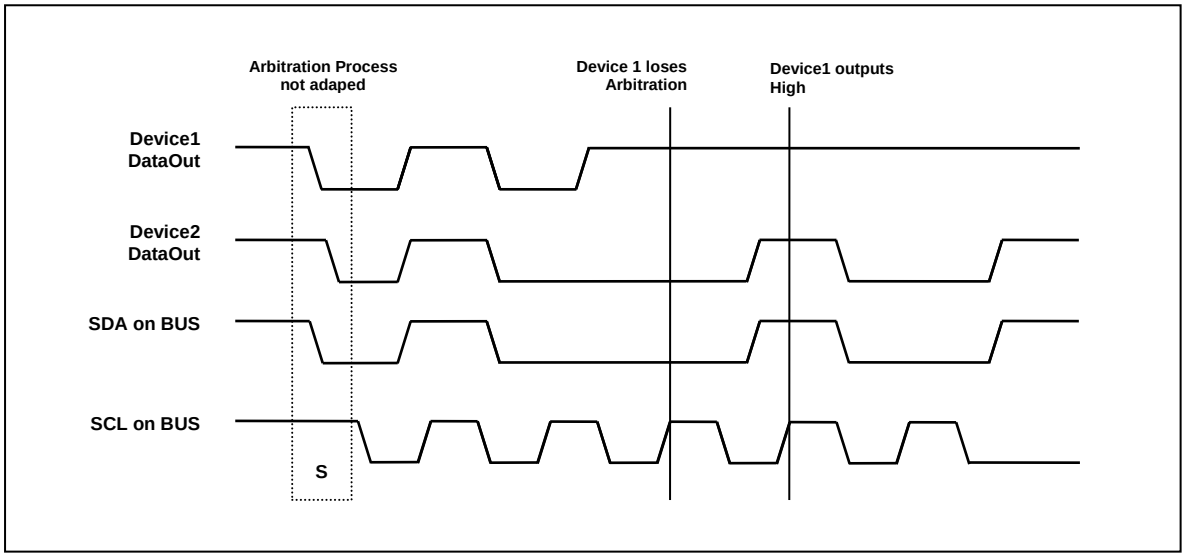


Figure 11-30 Arbitration Procedure of Two Masters

11.7.8 I2C Block operation

The I2C block as peripheral design is independently operating with main CPU operation. The operation of I2C block does a byte unit of I2C frame. After finishing a byte operation (transmit/receive data and clock) on I2C bus system, I2C block generate I2C interrupt for next byte operation. The I2C Interrupt service manage I2C block with the SFR registers, data load/read register (I2CDR) from/to I2C bus system, block control register (I2CMR), the state register (I2CSR) contained operation result.

An operation unit of I2C H/W block generates/ receive 9 SCL clock that are for 8 bits data and an ACK. I2C block send / receive ACK signal at 9th clock of SCL according to I2C specification.

The I2C application software initialize I2C block condition depended on clock system, I2C devices condition after system power on.

An application S/W prepares I2C bus communication resource on RAM buffers. If it is to set the start flag in I2CMR register. I2C block start to generate start signal and send a Slave address to slave device. All steps of I2C communication service except start signal and slave address is done by H/W block and I2C Interrupt service. Therefore main application software can reduce time resource while I2C Data write/read operation.

I2C block design supports both functions of master/ Slave on the same block. In case of Masker device it generate SCL clock to slave device and the case of slave mode receive SCL clock from master device.

I2C block decide SDA data direction with the data direction bit (R/W) of device address in both cases of master and slave mode(TMODE bit 0-> Receive, 1-> Transmit)

Note) When an I^2C interrupt is generated by I2C block, IIF flag in I2CMR register is set and it is cleared by writing any value to I2CSR. When I^2C interrupt occurs, the SCL line is hold LOW for reading/writing I2CDR register and control I2CMR until writing any value to I2CSR. When the IIF flag is set, the I2CSR contains a value for the state of the I2C bus. According to the value in I2CSR, software can decide what to do next.

I^2C can operate in 4 modes by configuring master/slave, transmitter/receiver.

I2C block initialization process

After power ON, it is necessary to have to initialize I2C block for that I2C Block provide I2C Slave device service

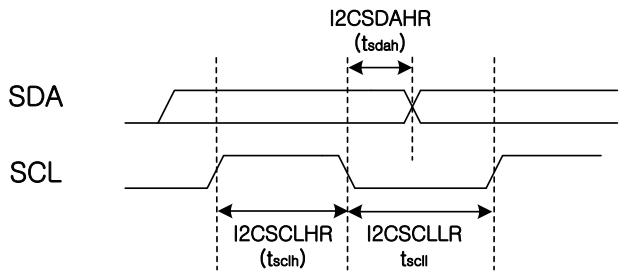
① I2C block will start operation (operation clock active) by setting IICEN bit on I2CMR register.

I2CMR = IICEN; // I2C block enable

② Reset I2C block by setting RESET bit on I2CMR register.

I2CMR = RESET; // Reset I2C block by S/W

③ Depended on I2C devices, it shall define I2C SCL max clock and write the value of SCL Low /high time and SDA hold time on I2CSCLLR, I2CSCLHR, I2CSDAHR as following diagram



The timing values are calculated as the follow formula

tscll =tsysclk (4xI2CSCLLR +1) → SCL clock low time

tsclh =tsysclk (4xI2CSCLHR +3) → SCL clock High time

tsdah =tsysclk (I2CSDAHR +1) → SDA data hold time after falling edge of SCL

* t_{sysclk} = system clock timing

Ex) In case of I2C clock (100KHz) and system clock(4MHz), each of t_{scll}, t_{sclh}times is 5us and tsdah is 2.5 us.

I2CSCLLR = 5; I2CSCLHR = 4; I2CSDAHR = 9;

④ It is to decide I2C Slave device address and write the address to I2CSAR

I2CSAR = SELF_ADDRESS;

⑤ Finally be ready to get I2C data from I2C bus system as slave device by setting I2C interrupt enable, I2C block enable, ACK enable bits on I2CMR register

I2CMR = IICEN+INTEN+ACKEN; // I2C interrupt enable

I2C interrupt Service

I2C Interrupt service will use for next management action and data load/read from I2C block after I2C H/W block operation (as I2C Master/ Slave device). Because I2C block acts I2C data receiving/writing as a byte unit, I2C block make I2C interrupt for next action of I2C block. While the interrupt happen, I2C block serve the state of I2C bus condition and operation result to I2CSR register. Interrupt service look both registers of I2CMR and I2CSR and do next steps (Save a data from I2CDR, load a data to I2CDR, make STOP condition or Re-start so on).

I2C Interrupt occur at after the following cases

1) As I2C Master Device

- sending a byte on I2CDR register after setting Start bit. (GCALL interrupt)
- sending a byte on I2CDR register after write to I2CSR. (TEND interrupt)
- receiving a byte on I2CDR after write to I2CSR (TEND interrupt)
- Occurring an arbitration loss (MLOST interrupt)
- detecting Stop condition (STOP interrupt)

2) As I2C Slave device

- getting start condition and same device address from a Master (SSEL interrupt)
- sending a byte on I2CDR register after write to I2CSR. (TEND interrupt)
- receiving a byte on I2CDR after write to I2CSR (TEND interrupt)
- detecting Stop condition (STOP interrupt)

Depended on above results I2C service provide services to read/write data from/to I2CDR, generate STOP condition, make next I2C Block action by writing a data to I2CSR register.

Bus arbitration of I2C block processes from I2C bus start condition to last data of I2C data frame. If getting an arbitration loss (MLOST interrupt), I2C interrupt service make I2C block Reset for bus free.

Master transmitter

Main software is to have write/read data to/from slave I2C device. The software has to be ready to get number of data with internal RAM or sending data on internal RAM according to I2C bus protocol type of Slave device. It writes Salve Address to I2CDR register in I2C Block and then if it set START bit on I2CMR register I2C block send slave address with SCL clock to slave device. I2C Block takes master mode (MASTER bit -> 1) and take the read/write state (TMODE bit, read(0), write(1)) according to the data direction bit (R/W) of device address.

The following is examples software for the case of master mode

Master write

```
I2CMR  = IICEN+INTEN;            // set I2C block( enable IIC block, I2C interrupt)
```

```
I2CDR  =  Slave Address + Write mode;  // load target Salve Address
I2CMR  |= SRT;                    // generate start condition and send slave address
```

I2C Interrupt Service

```
If(Master Mode) and (TMODE)
   If( ACK and GCALL or ACK and TEND )
    If ( Not End of Data )
    I2CDR  =  NEXT DATA;       // load target Salve Address
      I2CSR = 0xFF;              // Byte transmit start
   ELSE
    I2CMR = IICEN+INTEN+STP;      // STOP generation
    ELSE
     Initialize I2C block              // if have ACK error, any error
  End of I2C interrupt service
```

Master Read **(without sub address of Slave device)**

```
I2CMR  =  IICEN+INTEN;        // start generate
I2CDR  =  Slave Address + Read mode;           // load target Salve Address
I2CMR  |= SRT;               // generate start condition
```

I2C Interrupt Service

```
If (Master mode) and ( TMODE)
   If( ACK and GCALL )
  I2CMR |= ACKEN                // After receive data, generate ACK
   I2CSR = 0xFF;                 // Byte transmit start
  ELSE
  if ACK and TEND )
    If ( Not End of Data )
        If(LAST Data)
          I2CMR &= ~ACKEN        // After receive data, generate ACK
     I2C_buffer = I2CDR           // read
     I2CSR = 0xFF;              // Byte transmit start
   ELSE
  If( ~ACK and TEND)
   I2CMR = IICEN+INTEN+STP;    // STOP generation
    I2CSR = 0xFF;              // Byte transmit start
   ELSE
    Initialize I2C block              // if have ACK error, any error
End of I2C interrupt service
```

Slave Receiver

I2C Block that is under IIC enable and INTEN enable on I2CMR is monitoring I2C bus lines for being a start condition and self-address with I2CSAD. To have both signals of start signal and getting self-address, I2C block generate I2C interrupt with the status bits (SSEL, BUSY RXACK, SLAVE mode ..) after sending ACK signal. At the time **I2C block control SCL line to low state** for ready to get/handle next i2c data. If I2C block by I2C interrupt service is ready for next step, it is to release the SCL line to high state for getting next SCL clock from the master. I2C Block decide bus direction (data receive/transmission) by data direction (R/W) bit in Slave address from master. The state of bus direction is on TMOD bit on I2CSR register. If the master generate Stop condition I2C block receive STOP condition and generate I2C interrupt. I2C interrupt service write any data to I2CSR and finish Slave operation. I2C interrupt service and state register condition is diagrammed in Figure xxxx.

I2C Interrupt service

```
I2C Slave service
        if(Getting SSEL and send ACK)    // received Self-address form master
           if(TMODE)                     // data direction (R/W)
            I2CDR=I2C_TXData             // Transmission mode, Load data
        else
  I2C_RXData =I2CDR
        else
        if (Get STOP condition)
        else
           if (TMODE)                    // data direction (R/W)
                I2CDR= I2C_TXData        // Transmission mode, Load data
        else
                I2C_RXData =I2CDR        // Save received Data
           I2CSR=0xff;
```

11.7.9 Register Map

Name	Address	Dir	Default	Description
I2CMR	DAH	R/W	00H	I²C Mode Control Register
I2CSR	DBH	R	00H	I²C Status Register
I2CSCLLR	DCH	R/W	3FH	SCL Low Period Register
I2CSCLHR	DDH	R/W	3FH	SCL High Period Register
I2CSDAHR	DEH	R/W	01H	SDA Hold Time Register
I2CDR	DFH	R/W	FFH	I²C Data Register
I2CSAR	D7H	R/W	00H	I²C Slave Address Register
I2CSAR1	D6H	R/W	00H	I²C Slave Address Register 1

11.7.10 I²C Register Description

I²C Registers are composed of I²C Mode Control Register (I2CMR), I²C Status Register (I2CSR), SCL Low Period Register (I2CSCLLR), SCL High Period Register (I2CSCLHR), SDA Hold Time Register (I2CSDAHR), I²C Data Register (I2CDR), and I²C Slave Address Register (I2CSAR).

11.7.11 Register Description for I²C

I2CMR (I²C Mode Control Register) : DAH

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

I2CSR (I²C Status Register) : DBH

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

Initial value : 00H

GCALL This bit has different meaning depending on whether I²C is master or slave. Note 1)

When I²C is a master, this bit represents whether it received AACK (Address ACK) from slave.

When I²C is a slave, this bit is used to indicate general call.

0 No AACK is received (Master mode)

1 AACK is received (Master mode)

0 Received address is not general call address (Slave mode)

1 General call address is detected (Slave mode)

TEND This bit is set when 1-Byte of data is transferred completely. Note 1)

0 1 byte of data is not completely transferred

1 1 byte of data is completely transferred

STOP This bit is set when STOP condition is detected. Note 1)

0 No STOP condition is detected

1 STOP condition is detected

SSEL This bit is set when I²C is addressed by other master. Note 1)

0 I²C is not selected as slave

1 I²C is addressed by other master and acts as a slave

MLOST This bit represents the result of bus arbitration in master mode. Note 1)

0 I²C maintains bus mastership

1 I²C has lost bus mastership during arbitration process

BUSY This bit reflects bus status.

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

1 I²C bus is busy

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

0 I²C is a receiver

1 I²C is a transmitter

RXACK This bit shows the state of ACK signal.

0 No ACK is received

1 ACK is generated at ninth SCL period

Note 1) These bits can be source of interrupt.

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

I2CSCLLR (SCL Low Period Register) : DCH

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 I^{2}C operates in master mode. The base clock is SCLK, the system clock, and the period is calculated by the formula :t_{SCLK}× (4 × SCLL + 1) where t_{SCLK} is the period of SCLK.

I2CSCLHR (SCL High Period Register) : DDH

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 I^{2}C operates in master mode. The base clock is SCLK, the system clock, and the period is calculated by the formula :t_{SCLK}× (4 × SCLH + 3) where t_{SCLK} is the period of SCLK.

So, the operating frequency of I^{2}C in master mode (fI2C) is calculated by the following equation.

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

I2CSDAHR (SDA Hold Time Register) : DEH

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}× 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}× (SDAH + 1). So, to insure normal operation in slave mode, the value t_{SCLK}× (SDAH + 1) must be smaller than the period of SCL.

I2CDR (I^{2}C Data Register) : DFH

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 I^{2}C is configured as a transmitter, load this register with data to be transmitted. When I^{2}C is a receiver, the received data is stored into this register.

I2CSAR (I^{2}C Slave Address Register) : D7H

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

Initial value : 00H

SLA[7:1] These bits configure the slave address of this I^{2}C module when I^{2}C operates in slave mode.

GCALLEN This bit decides whether I^{2}C allows general call address or not when I^{2}C operates in slave mode.

0 Ignore general call address

1 Allow general call address

I2CSAR1 (I^{2}C Slave Address Register 1) : D6H

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

Initial value : 00H

SLA[7:1] These bits configure the slave address of this I^{2}C module when I^{2}C operates in slave mode.

GCALLEN This bit decides whether I^{2}C allows general call address or not when I^{2}C operates in slave mode.

0 Ignore general call address

1 Allow general call address

11.8 16-Channel Touch Switch

11.8.1 Features

. Self-Capacitive Touch Key Sensor.
. Total 16-channel Touch Key Support.
. 16-bits Sensing Resolutions.
. Fast Initial Self Calibration.
. Key Detection Mode : Single/Multi-Mode.
. Clock Frequency during Sensing Operation : 16MHz.
. The Improvement of the SNR by Bias-Calibration in Analog Sensing Block.
. VDD Operating Voltage : 2.7V ~ 5.5V.
. Current Consumption : T.B.D.
. Current Consumption@STOPmode :< 1uA.
. Operation Temperature : -40℃ ~ 85℃.

11.8.2 Register Map

Name	Address	Dir	Default	Description
SUM_CH0H	2E00H	R	00H	Touch Sensor Ch0 Sum High-Byte Register
SUM_CH0L	2E01H	R	00H	Touch Sensor Ch0 Sum Low-Byte Register
…				
SUM_CH15H	2E1EH	R	00H	Touch Sensor Ch15 Sum High-Byte Register
SUM_CH15L	2E1FH	R	00H	Touch Sensor Ch15 Sum Low-Byte Register
SCO0_H	2E40H	R/W	00H	Touch Sensor Offset Capacitor Selection High-Byte Register for Ch0
SCO0_L	2E41H	R/W	00H	Touch Sensor Offset Capacitor Selection Low-Byte Register for Ch0
…				
SCO15_H	2E5EH	R/W	00H	Touch Sensor Offset Capacitor Selection High-Byte Register for Ch15
SCO15_L	2E5FH	R/W	00H	Touch Sensor Offset Capacitor Selection Low-Byte Register for Ch15
TS_CON	2EC0H	R/W	00H	Touch Sensor Control Register
TS_MODE	2EC2H	R/W	10H	Touch Sensor Mode Register
TS_SUM_CNT	2EC3H	R/W	01H	Touch Sensor Sum Repeat Count Register
TS_CH_SELH	2EC4H	R/W	00H	Touch Sensor Channel Selection High-Byte Register
TS_CH_SELL	2EC5H	R/W	00H	Touch Sensor Channel Selection Low-Byte Register
TS_WAKE_DEL	2EC6H	R/W	00H	Touch Sensor Wakeup Delta Register
TS_SLP_CR	2EC7H	R/W	01H	Touch Sensor Low Pass Filter Control Register
TS_INTEG_CNT	2ECAH	R/W	32H	Touch Sensor Sensing Integration Count Register
TS_FREQ_NUM	2ECBH	R/W	FFH	Touch Sensor Frequency Number Register
TS_FREQ_DEL	2ECCB	R/W	00H	Touch Sensor Frequency Delta Register
TS_CLK_CFG	2ECDH	R/W	10H	Touch Sensor Clock Configuration Register
TRIM_OSC	2ECEH	R/W	B8H	Touch Sensor RING OSC. Trimming Selection Register
SCI	2EE3H	R/W	34H	Touch Sensor Input Capacitor Selection Register
SCC	2EE4H	R/W	04H	Touch Sensor Conversion Capacitor Selection Register
SVREF	2EE5H	R/W	03H	Touch Sensor VREF Selection Register
TRSTP	2EE7H	R/W	03H	Touch Sensor Reset Time of Positive Sensing Register
TDRVP	2EE8H	R/W	03H	Touch Sensor Driving Time of Positive Sensing Register
TINTP	2EE9H	R/W	14H	Touch Sensor Integration Time of Positive Sensing Register
TRSTN	2EEAH	R/W	03H	Touch Sensor Reset Time of Negative Sensing Register
TDRVN	2EEBH	R/W	03H	Touch Sensor Driving Time of Negative Sensing Register
TINTN	2EECH	R/W	14H	Touch Sensor Integration Time of Negative Sensing Register

11.8.3 Register Description for Touch Sensing

SUM_CH0_H (Touch Sensor Ch0 Sum High-Byte Register) : 2E00H

7	6	5	4	3	2	1	0
SUM_CH0_H							
R	R	R	R	R	R	R	R

Initial value : 00H

SUM_CH0_L (Touch Sensor Ch0 Sum Low-Byte Register) : 2E01H

7	6	5	4	3	2	1	0
SUM_CH0_L							
R	R	R	R	R	R	R	R

Initial value : 00H

SUM_CH1_H (Touch Sensor Ch1 Sum High-Byte Register) : 2E02H
SUM_CH1_L (Touch Sensor Ch1 Sum Low-Byte Register) : 2E03H
SUM_CH2_H (Touch Sensor Ch2 Sum High-Byte Register) : 2E04H
SUM_CH2_L (Touch Sensor Ch2 Sum Low-Byte Register) : 2E05H
SUM_CH3_H (Touch Sensor Ch3 Sum High-Byte Register) : 2E06H
SUM_CH3_L (Touch Sensor Ch3 Sum Low-Byte Register) : 2E07H
SUM_CH4_H (Touch Sensor Ch4 Sum High-Byte Register) : 2E08H
SUM_CH4_L (Touch Sensor Ch4 Sum Low-Byte Register) : 2E09H
SUM_CH5_H (Touch Sensor Ch5 Sum High-Byte Register) : 2E0AH
SUM_CH5_L (Touch Sensor Ch5 Sum Low-Byte Register) : 2E0BH
SUM_CH6_H (Touch Sensor Ch6 Sum High-Byte Register) : 2E0CH
SUM_CH6_L (Touch Sensor Ch6 Sum Low-Byte Register) : 2E0DH
SUM_CH7_H (Touch Sensor Ch7 Sum High-Byte Register) : 2E0EH
SUM_CH7_L (Touch Sensor Ch7 Sum Low-Byte Register) : 2E0FH
SUM_CH8_H (Touch Sensor Ch8 Sum High-Byte Register) : 2E10H
SUM_CH8_L (Touch Sensor Ch8 Sum Low-Byte Register) : 2E11H
SUM_CH9_H (Touch Sensor Ch9 Sum High-Byte Register) : 2E12H
SUM_CH9_L (Touch Sensor Ch9 Sum Low-Byte Register) : 2E13H
SUM_CH10_H (Touch Sensor Ch10 Sum High-Byte Register) : 2E14H
SUM_CH10_L (Touch Sensor Ch10 Sum Low-Byte Register) : 2E15H
SUM_CH11_H (Touch Sensor Ch11 Sum High-Byte Register) : 2E16H
SUM_CH11_L (Touch Sensor Ch11 Sum Low-Byte Register) : 2E17H
SUM_CH12_H (Touch Sensor Ch12 Sum High-Byte Register) : 2E18H
SUM_CH12_L (Touch Sensor Ch12 Sum Low-Byte Register) : 2E19H
SUM_CH13_H (Touch Sensor Ch13 Sum High-Byte Register) : 2E1AH
SUM_CH13_L (Touch Sensor Ch13 Sum Low-Byte Register) : 2E1BH
SUM_CH14_H (Touch Sensor Ch14 Sum High-Byte Register) : 2E1CH
SUM_CH14_L (Touch Sensor Ch14 Sum Low-Byte Register) : 2E1DH
SUM_CH15_H (Touch Sensor Ch15 Sum High-Byte Register) : 2E1EH
SUM_CH15_L (Touch Sensor Ch15 Sum Low-Byte Register) : 2E1FH
SCO0_H (Touch Sensor Offset Capacitor Selection High-Byte Register for CH0) : 2E40H

7	6	5	4	3	2	1	0
SCO0_H							
							RW

Initial value : 00H

SCO0_L (Touch Sensor Offset Capacitor Selection Low-Byte Register for CH0) : 2E41H

7	6	5	4	3	2	1	0
SCO0_L							
RW	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

SCO1_H (Touch Sensor Offset Capacitor Selection High-Byte Register for CH1) : 2E42H
SCO1_L (Touch Sensor Offset Capacitor Selection Low-Byte Register for CH1) : 2E43H
SCO2_H (Touch Sensor Offset Capacitor Selection High-Byte Register for CH2) : 2E44H
SCO2_L (Touch Sensor Offset Capacitor Selection Low-Byte Register for CH2) : 2E45H
SCO3_H (Touch Sensor Offset Capacitor Selection High-Byte Register for CH3) : 2E46H
SCO3_L (Touch Sensor Offset Capacitor Selection Low-Byte Register for CH3) : 2E47H
SCO4_H (Touch Sensor Offset Capacitor Selection High-Byte Register for CH4) : 2E48H
SCO4_L (Touch Sensor Offset Capacitor Selection Low-Byte Register for CH4) : 2E49H
SCO5_H (Touch Sensor Offset Capacitor Selection High-Byte Register for CH5) : 2E4AH
SCO5_L (Touch Sensor Offset Capacitor Selection Low-Byte Register for CH5) : 2E4BH
SCO6_H (Touch Sensor Offset Capacitor Selection High-Byte Register for CH6) : 2E4CH
SCO6_L (Touch Sensor Offset Capacitor Selection Low-Byte Register for CH6) : 2E4DH
SCO7_H (Touch Sensor Offset Capacitor Selection High-Byte Register for CH7) : 2E4EH
SCO7_L (Touch Sensor Offset Capacitor Selection Low-Byte Register for CH7) : 2E4FH
SCO8_H (Touch Sensor Offset Capacitor Selection High-Byte Register for CH8) : 2E50H
SCO8_L (Touch Sensor Offset Capacitor Selection Low-Byte Register for CH8) : 2E51H
SCO9_H (Touch Sensor Offset Capacitor Selection High-Byte Register for CH9) : 2E52H
SCO9_L (Touch Sensor Offset Capacitor Selection Low-Byte Register for CH9) : 2E53H
SCO10_H (Touch Sensor Offset Capacitor Selection High-Byte Register for CH10) : 2E54H
SCO10_L (Touch Sensor Offset Capacitor Selection Low-Byte Register for CH10) : 2E55H
SCO11_H (Touch Sensor Offset Capacitor Selection High-Byte Register for CH11) : 2E56H
SCO11_L (Touch Sensor Offset Capacitor Selection Low-Byte Register for CH11) : 2E57H
SCO12_H (Touch Sensor Offset Capacitor Selection High-Byte Register for CH12) : 2E58H
SCO12_L (Touch Sensor Offset Capacitor Selection Low-Byte Register for CH12) : 2E59H
SCO13_H (Touch Sensor Offset Capacitor Selection High-Byte Register for CH13) : 2E5AH
SCO13_L (Touch Sensor Offset Capacitor Selection Low-Byte Register for CH13) : 2E5BH
SCO14_H (Touch Sensor Offset Capacitor Selection High-Byte Register for CH14) : 2E5CH
SCO14_L (Touch Sensor Offset Capacitor Selection Low-Byte Register for CH14) : 2E5DH
SCO15_H (Touch Sensor Offset Capacitor Selection High-Byte Register for CH15) : 2E5EH
SCO15_L (Touch Sensor Offset Capacitor Selection Low-Byte Register for CH15) : 2E5FH

TS_CON (Touch Sensor Control Register) : 2EC0H

7	6	5	4	3	2	1	0
-	-	-	OSC_EN	BGR_EN	TS_FI	TS_CFG	TS_RUN
-	-	-	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

OSC_EN Oscillator Enable

0 Oscillator Disable (Default)

1 Oscillator Enable

BGR_EN Band Gap Reference Enable

0 BGR Disable (Default)

1 BGR Enable

TS_IF Touch Sensor Interrupt Flag

0 No new sensing results

1 In normal mode, this flag indicates that the new sensing results are generated. In WDT mode, this flag indicates that the current sensing values are different from the previous sensing values by ±TS_WAKE_DEL. For next sensing, this flag must be cleared.

TS_CFG Touch Sensor Register Configuration Enable

0 Touch Sensor Register Configuration Disable (Default)

1 All registers except TS_CON, TS_MODE, and TS_CLK can be access only when this flag is set.

TS_RUN Touch Sensor Enable

0 Touch Sensor Disable (Default)

1 Touch Sensor Enable

TS_MODE (Touch Sensor Mode Register) : 2EC2H

7	6	5	4	3	2	1	0
SREF	SC_GAIN	SAP1	SAP0	-	WDT	PORT1	PORT0
-	-	R/W	R/W	-	R/W	R/W	R/W

Initial value : 10H

SREF External Reference Offset Enable

0 Disable

1 Enable

SC_GAIN Gain Calibration Capacitor Enable

0 Gain Calibration Capacitor Disable (Default)

1 Gain Calibration Capacitor Enable

SAP[1:0] Touch Sensor mode or ADC mode Selection

01 Touch Sensor mode Select (Default)

10 ADC mode Select

WDT Watch-dog Timer mode Selection

0 Normal WDT mode (Default)

1 Only Touch Sensor will wake up in WDT. MCU is still in stop mode.

PORT[1:0] Port Configuration During Inactive Status

00 Input Floating

01 Output Low

10 Output High

TS_SUM_CNT (Touch Sensor Sum Repeat Count Register) : 2EC3H

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

Initial value : 01H

TS_CH_SEL_H (Touch Sensor Channel Selection High-Byte Register) : 2EC4H

7	6	5	4	3	2	1	0
CH15_SEL	CH14_SEL	CH13_SEL	CH12_SEL	CH11_SEL	CH10_SEL	CH9_SEL	CH8_SEL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

CH15_SEL Channel 15 Enable
- 0 Disable (Default)
- 1 Enable Touch Key by P27

CH14_SEL Channel 14 Enable
- 0 Disable (Default)
- 1 Enable Touch Key by P26

CH13_SEL Channel 13 Enable
- 0 Disable (Default)
- 1 Enable Touch Key by P25

CH12_SEL Channel 12 Enable
- 0 Disable (Default)
- 1 Enable Touch Key by P24

CH11_SEL Channel 11 Enable
- 0 Disable (Default)
- 1 Enable Touch Key by P23

CH10_SEL Channel 10 Enable
- 0 Disable (Default)
- 1 Enable Touch Key by P22

CH9_SEL Channel 9 Enable
- 0 Disable (Default)
- 1 Enable Touch Key by P21

CH8_SEL Channel 8 Enable
- 0 Disable (Default)
- 1 Enable Touch Key by P20

TS_CH_SEL_L (Touch Sensor Channel Selection Low-Byte Register) : 2EC5H

7	6	5	4	3	2	1	0
CH7_SEL	CH6_SEL	CH5_SEL	CH4_SEL	CH3_SEL	CH2_SEL	CH1_SEL	CH0_SEL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

CH7_SEL Channel 7 Enable
- 0 Disable (Default)
- 1 Enable Touch Key by P07

CH6_SEL Channel 6 Enable
- 0 Disable (Default)
- 1 Enable Touch Key by P06

CH5_SEL Channel 5 Enable
- 0 Disable (Default)
- 1 Enable Touch Key by P05

CH4_SEL Channel 4 Enable
- 0 Disable (Default)
- 1 Enable Touch Key by P04

CH3_SEL Channel 3 Enable
- 0 Disable (Default)
- 1 Enable Touch Key by P03

CH2_SEL Channel 2 Enable
- 0 Disable (Default)
- 1 Enable Touch Key by P02

CH1_SEL Channel 1 Enable
- 0 Disable (Default)
- 1 Enable Touch Key by P01

CH0_SEL Channel 0 Enable
- 0 Disable (Default)
- 1 Enable Touch Key by P00

TS_WAKE_DEL (Touch Sensor Wakeup Delta Register) : 2EC6H

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

Initial value : 00H

If the sensing value “A” is different from the sensing value “B” by ±TS_WAKE_DEL, touch sensor interrupt will be occurred, and then the MCU will be wakeup. If touch sensor interrupt is not generated, MCU is still in stop mode. When the sensing value “A” means the sensing value saved before entering in WDT mode, and the sensing value “B” means the sensing value saved in WDT mode temporarily.

TS_SLP_CR (Touch Sensor Low Pass Filter Control Register) : 2EC7H

7	6	5	4	3	2	1	0
-	SLP_C2	SLP_C1	SLP_C0	SLP_R3	SLP_R2	SLP_R1	SLP_R0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 01H

SLP_C[2:0] Capacitor Trimming for Input Low Pass Filter

000 0pF
001 4pF
010 8pF
011 12pF
100 16pF
101 20pF
110 24pF
111 28pF

SLP_R[3:0] Resistor Trimming for Input Low Pass Filter

0000 Open
0001 0K
0010 5K
0100 10K
1000 20K
1110 2.8K
0110 3.3K
1010 4.0K
1100 6.7K

TS_INTEG_CNT (Touch Sensor Sensing Integration Count Register) : 2ECAH

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

Initial value : 32H

TS_FREQ_NUM (Touch Sensor Frequency Number Register) : 2ECBH

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

Initial value : FFH

This register indicates the number of steps for frequency delta.

TS_FREQ_DEL (Touch Sensor Frequency Delta Register) : 2ECCH

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

Initial value : 00H

This register indicates the frequency differences in every sensing integration.

TS_CLK_CFG (Touch Sensor Clock Configuration Register) : 2ECDH

7	6	5	4	3	2	1	0
-	-	-	-	TSCLKOE	TSCLKDIV2	TSCLKDIV1	TSCLKDIV0
-	-	-	-	R/W	R/W	R/W	R/W

Initial value : 10H

TSCLKOE Divided Touch Sensor Clock Output Enable

0 Clock Output Disable (Default)

1 Clock Output Enable

TSCLKDIV[2:0] Touch Sensor Clock Divider (Refer to TRIM_OSC)

000 OSC_{ts} / 1 (20MHz, Default)

001 OSC_{ts} / 2

010 OSC_{ts} / 4

011 OSC_{ts} / 8

100 OSC_{ts} / 16

101 OSC_{ts} / 32

110 OSC_{ts} / 64

111 OSC_{ts} / 128

TRIM_OSC (Touch Sensor RING OSC. Trimming Selection Register) : 2ECEH

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

Initial value : B8H

TRIM_OSC[7:0] Touch Sensor RING OSC. Trimming Selection

00H 44MHz (maximum)

…

B8H 20MHz (default)

…

FFH 4MHz (minimum)

SCI (Touch Sensor Input Capacitor Selection Register) : 2EE3H

7	6	5	4	3	2	1	0
IBIAS_TRIM3	IBIAS_TRIM2	IBIAS_TRIM1	IBIAS_TRIM0		SCI2	SCI1	SCI0
R/W	R/W	R/W	R/W		R/W	R/W	R/W

Initial value : 34H

IBIAS_TRIM[3:0]	Touch Sensor Bias Current Source	
	0000	0.22uA
	0001	0.43uA
	0010	0.64uA
	0011	0.85uA
	0111	1.70uA
	1011	2.55uA
	1111	3.40uA
SCI[2:0]	Touch Sensor Input Capacitor	
	000	0.9pF
	001	1.8pF
	010	2.7pF
	011	3.6pF
	100	4.5pF
	101	5.4pF
	110	6.3pF
	111	7.2pF

SCC (Touch Sensor Conversion Capacitor Selection Register) : 2EE4H

7	6	5	4	3	2	1	0
					SCC2	SCC1	SCC0
					R/W	R/W	R/W

Initial value : 04H

SCC[2:0]	Touch Sensor Input Capacitor	
	000	2.4pF
	001	4.8pF
	010	7.2pF
	011	9.6pF
	100	12pF
	101	14.4pF
	110	16.8pF
	111	19.2pF

SVREF(Touch Sensor VREF Resistor Selection Register) : 2EE5H

7	6	5	4	3	2	1	0
				SVREF3	SVREF2	SVREF1	SVREF0
				R/W	R/W	R/W	R/W

Initial value : 04H

SVREF[3:0] Touch Sensor Conversion Capacitor

0000	Open
0001	2.5Kohm
0010	5Kohm
0100	10Kohm
1000	20Kohm
1111	1.3Kohm

TRSTP(Touch Sensor Reset Time of Positive Sensing Register) : 2EE7H

7	6	5	4	3	2	1	0
TRSTP7	TRSTP6	TRSTP5	TRSTP4	TRSTP3	TRSTP2	TRSTP1	TRSTP0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 03H

This register indicates the reset time of the touch sensor capacitance, so it depends on the load capacitance of the sensing channel. The more load capacitance of touch channel, the larger time of the reset time. If you have enough time, it is better to give generous.

TDRVP(Touch Sensor Driving Time of Positive Sensing Register) : 2EE8H

7	6	5	4	3	2	1	0
TDRVP7	TDRVP6	TDRVP5	TDRVP4	TDRVP3	TDRVP2	TDRVP1	TDRVP0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 03H

This register indicates the driving time of the touch sensor capacitor, so it depends on the load capacitance of the sensing channel. The more load capacitance of touch channel, the larger time of the driving time. And it also depends on the SVREF(2EE5) value. So, you must give the suitable driving time according to the load capacitor and SVREF register value.

TINTP(Touch Sensor Integration Time of Positive Sensing Register) : 2EE9H

7	6	5	4	3	2	1	0
TINTP7	TINTP6	TINTP5	TINTP4	TINTP3	TINTP2	TINTP1	TINTP0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 14H

This register indicates the integration time of the touch sensor AMP, so it depends on the SCI(2EE3H) and SCC(2EE4H) registers. The larger register value, the larger time of the integration time. If you have enough time, it is better to give generous.

TRSTN(Touch Sensor Reset Time of Negative Sensing Register) : 2EEAH

7	6	5	4	3	2	1	0
TRSTN7	TRSTN6	TRSTN5	TRSTN4	TRSTN3	TRSTN2	TRSTN1	TRSTN0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 03H

This register indicates the reset time of the touch sensor capacitance, so it depends on the load capacitance of the sensing channel. The more load capacitance of touch channel, the larger time of the reset time. If you have enough time, it is better to give generous.

TDRVN(Touch Sensor Driving Time of Negative Sensing Register) : 2EEBH

7	6	5	4	3	2	1	0
TDRVN7	TDRVN6	TDRVN5	TDRVN4	TDRN3	TDRVN2	TDRVN1	TDRVN0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 03H

This register indicates the driving time of the touch sensor capacitor, so it depends on the load capacitance of the sensing channel. The more load capacitance of touch channel, the larger time of the driving time. And it also depends on the SVREF(2EE5) value. So, you must give the suitable driving time according to the load capacitor and SVREF register value.

TINTN(Touch Sensor Integration Time of Negative Sensing Register) : 2EECH

7	6	5	4	3	2	1	0
TINTN7	TINTN6	TINTN5	TINTN4	TINTN3	TINTN2	TINTN1	TINTN0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 14H

This register indicates the integration time of the touch sensor AMP, so it depends on the SCI(2EE3) and SCC(2EE4) registers. The larger register value, the larger time of the integration time. If you have enough time, it is better to give generous.

11.8.4 User Programming Procedure

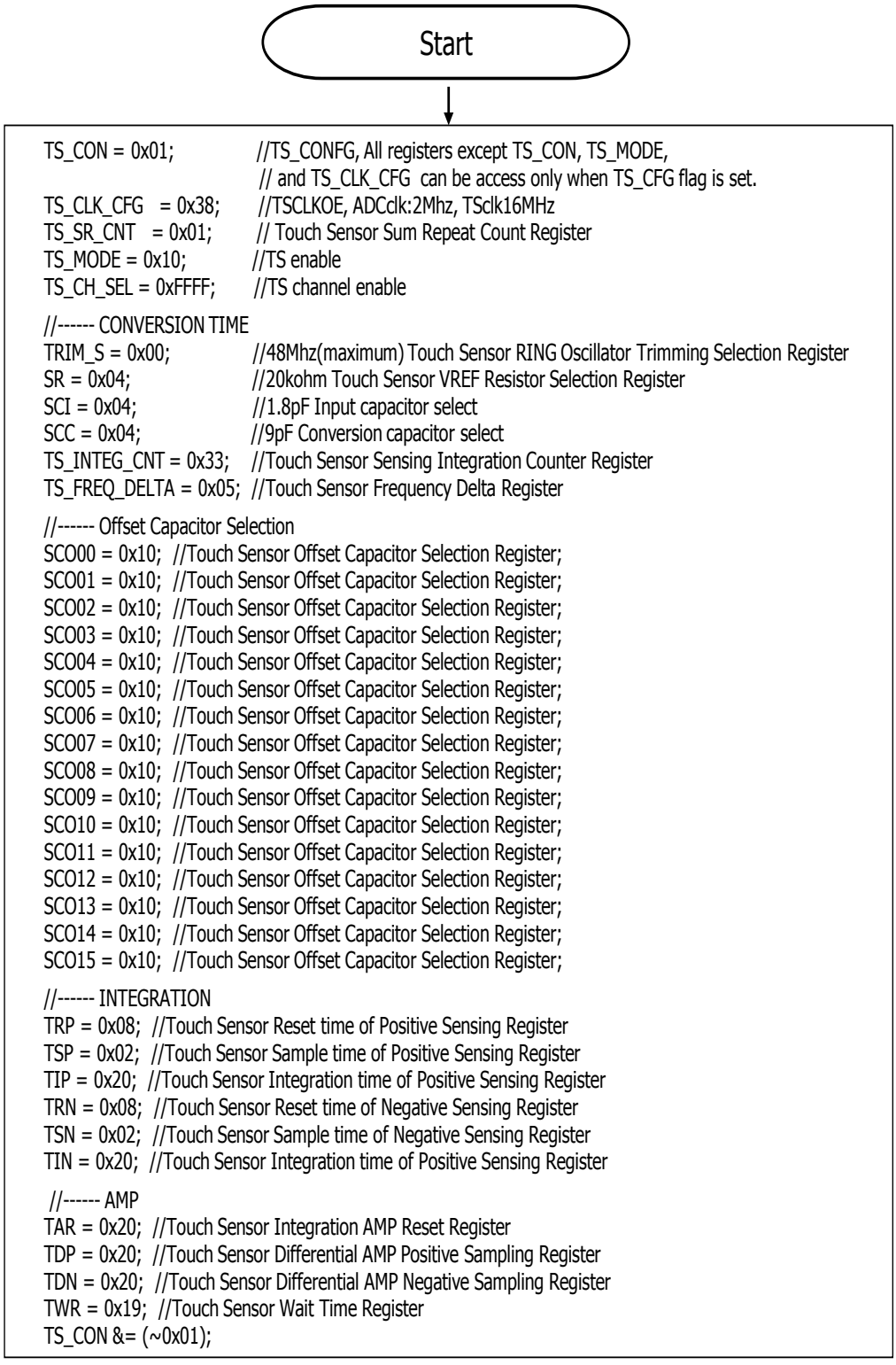


Figure 11-31 User Programming Procedure

12. Power Down Operation

12.1 Overview

The A96T336 has three power-down modes to minimize the power consumption of the device. In power down mode, power consumption is reduced considerably. The device provides three kinds of power saving functions, IDLE, STOP1 and STOP2 mode. In three modes, program is stopped.

12.2 Peripheral Operation in IDLE/STOP Mode

Table 12-1 Peripheral Operation during Power Down Mode.

Peripheral	IDLE Mode	STOP1 Mode	STOP2 Mode
CPU	ALL CPU Operation are Disable	ALL CPU Operation are Disable	ALL CPU Operation are Disable
RAM	Retain	Retain	Retain
Basic Interval Timer	Operates Continuously	Operates Continuously	Stop
Watch Dog Timer	Operates Continuously	Operates Continuously	Stop
Timer	Operates Continuously	Halted (Only when the Event Counter Mode is Enable, Timer operates Normally)	Halted (Only when the Event Counter Mode is Enable, Timer operates Normally)
I^2C	Operates Continuously	Stop	Stop
Internal OSC (16MHz)	Oscillation	Stop	Stop
Internal RCOSC (256kHz)	Oscillation	Oscillation	Stop
I/O Port	Retain	Retain	Retain
Control Register	Retain	Retain	Retain
Address Data Bus	Retain	Retain	Retain
Release Method	By RESET, all Interrupts	By RESET,External Interrupt, I^2C (slave mode), WDT, BIT, USART	By RESET, External Interrupt, I^2C (slave mode), USART

12.3 IDLE mode

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

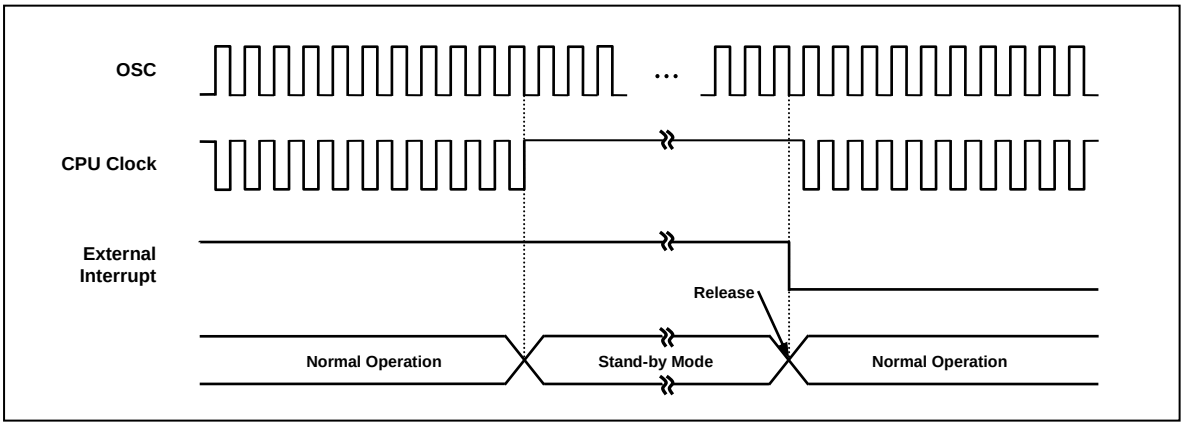


Figure 12-1 IDLE Mode Release Timing by External Interrupt

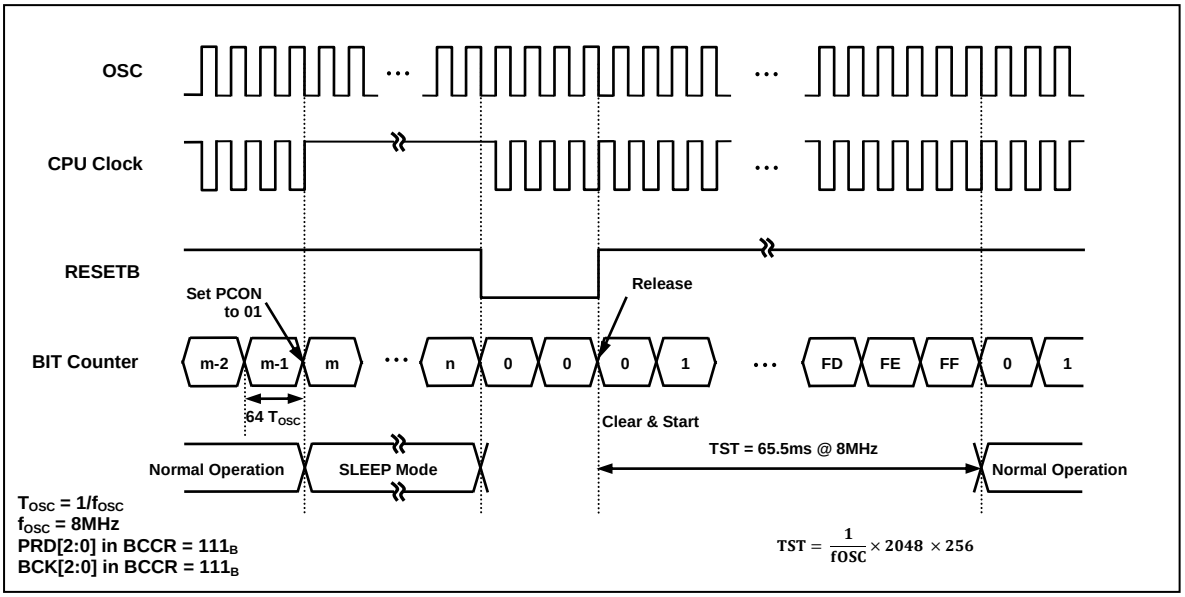


Figure 12-2 IDLE Mode Release Timing by RESETB

(Ex) MOV PCON, #0000_0001b ; setting of IDLE mode : set the bit of STOP and IDLE Control register (PCON)

12.4 STOP mode

The power control register is set to '03h' to enter the STOP Mode. In the stop mode, the main oscillator, system clock and peripheral clock is stopped, but watch timer continue to operate. With the clock frozen, all functions are stooped, but the on-chip RAM and control registers are held.

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

When exit from STOP mode, enough oscillation stabilization time is required to normal operation. Figure 12-3 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 20msec). This guarantees that oscillator has started and stabilized.

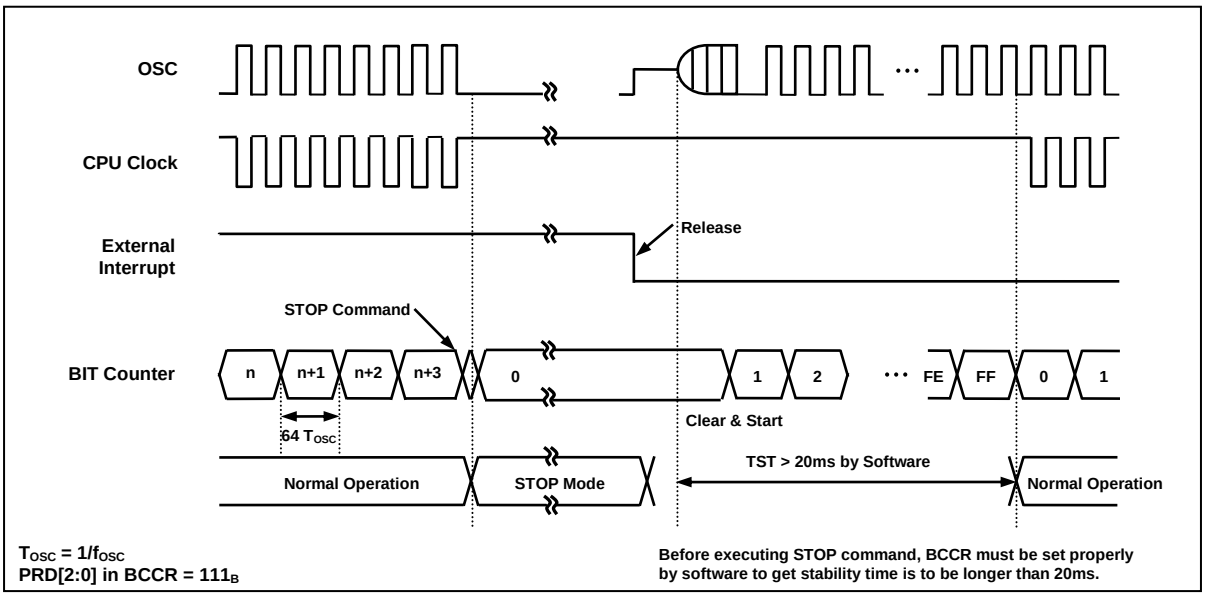


Figure 12-3 STOP Mode Release Timing by External Interrupt

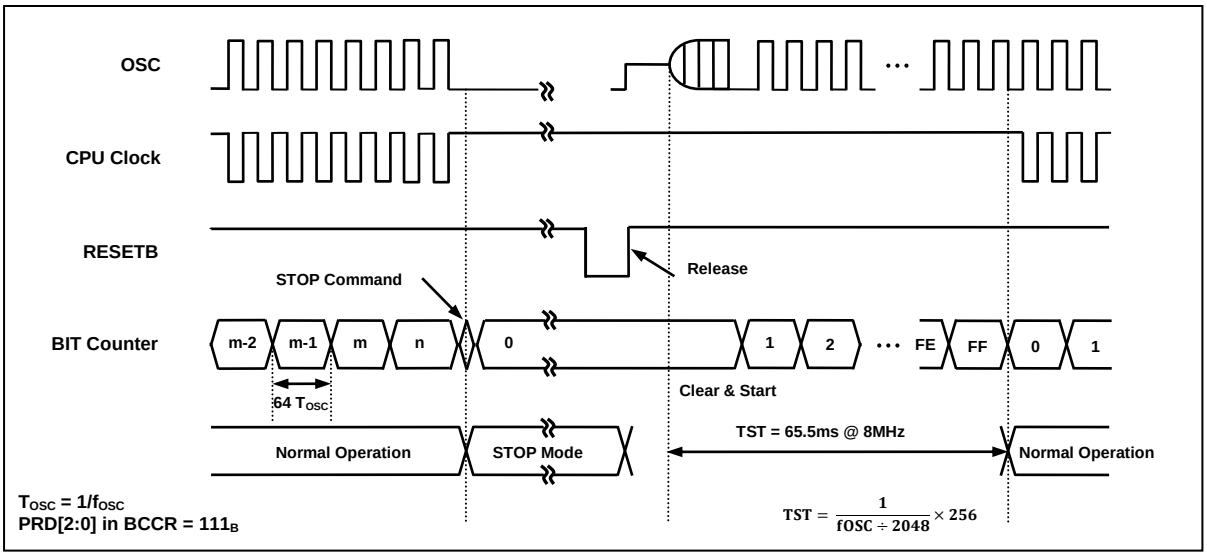


Figure 12-4 STOP Mode Release Timing by RESETB

12.5 Release Operation of STOP1, 2 Mode

After STOP1, 2 mode is released, the operation begins according to content of related interrupt register just before STOP1, 2 mode start (Figure 12-5). Interrupt Enable Flag of All (EA) of IE should be set to `1`. Released by only interrupt which each interrupt enable flag = `1`, and jump to the relevant interrupt service routine.

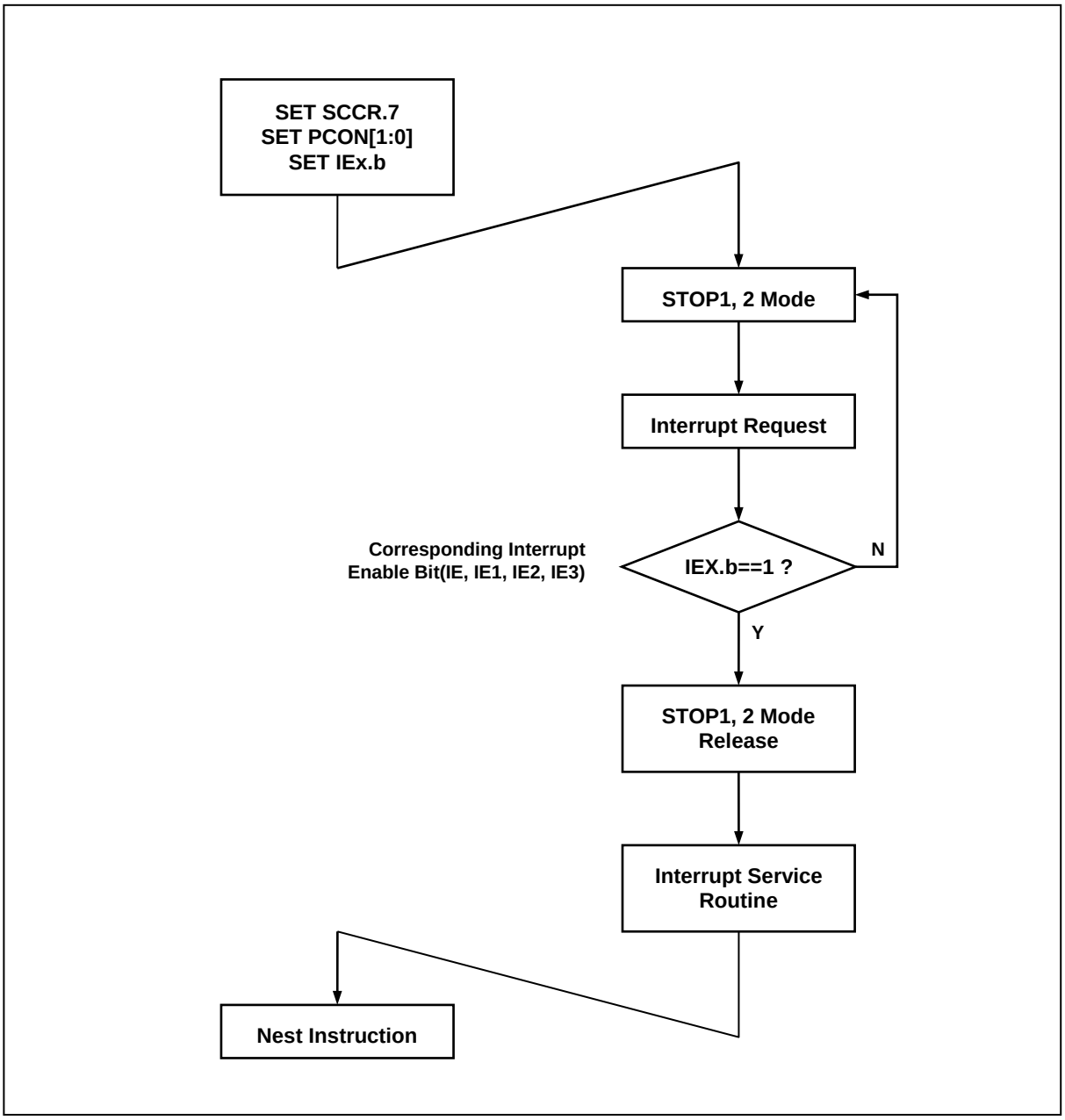


Figure 12-5 STOP1, 2 Mode Release Flow

12.5.1 Register Map

Table 12-2 Register Map

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

12.5.2 Power Down Operation Register Description

The Power Down Operation Register consists of the Power Control Register (PCON).

12.5.3 Register Description for Power Down Operation

PCON (Power Control Register) : 87H

7	6	5	4	3	2	1	0
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

IDLE Mode

01H IDLE mode enable

STOP1, 2 Mode

03H STOP1, 2 mode enable

Note)

1. To enter IDLE mode, PCON must be set to ‘01H’.
2. To STOP1,2 mode, PCON must be set to ‘03H’.
   (In STOP1,2 mode, PCON register is cleared automatically by interrupt or reset)
3. When PCON is set to ‘03H’, if SCCR[7] is set to ‘1’, it enters the STOP1 mode. if SCCR[7] is cleared to ‘0’, it enters the STOP2 mode
4. The different thing in STOP 1,2 is only clock operation of internal 256kHz-OSC during STOP mode operating.

13. RESET

13.1 Overview

The A96T336 has reset by external RESETB pin. The following is the hardware setting value.

Table 13-1 Reset state

On Chip Hardware	Initial Value
Program Counter (PC)	0000h
Accumulator	00h
Stack Pointer (SP)	07h
Peripheral Clock	On
Control Register	Peripheral Registers refer
Brown-Out Detector	Enable

13.2 Reset Source

The A96T336 has five types of reset generation procedures. The following is the reset sources.

- External RESETB
- Power ON RESET (POR)
- WDT Overflow Reset (In the case of WDTEN = `1`)
- BOD Reset (In the case of BODEN = `1 `)
- LVI Reset (In the case of LVI_S_I[1:0] ≠ `0 `)
- OCD Reset

13.3 Block Diagram

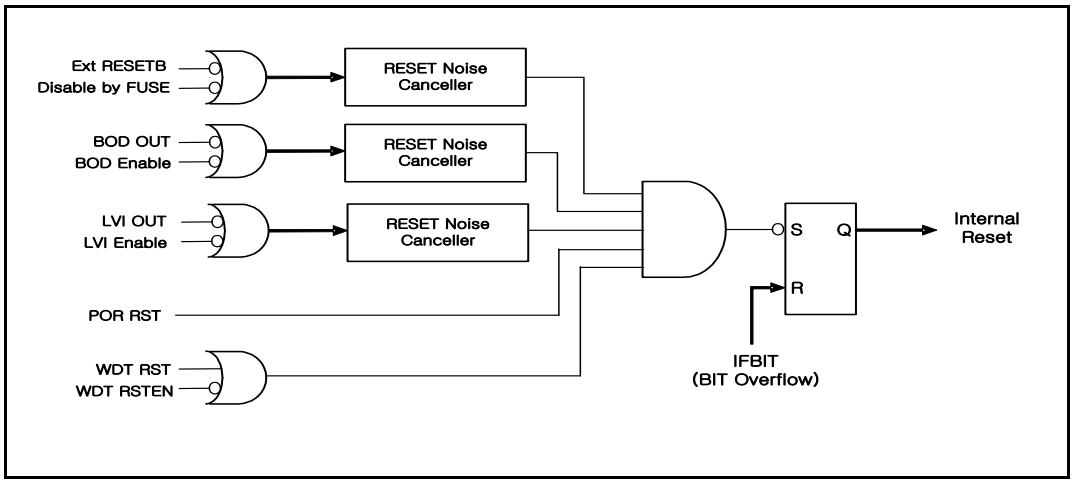


Figure 13-1 RESET Block Diagram

13.4 RESET Noise Canceller

The Figure 13-2 is the Noise canceller diagram for Noise cancel of RESET. It has the Noise cancel value of about 32us (@V_{DD}=5V) to the low input of System Reset.

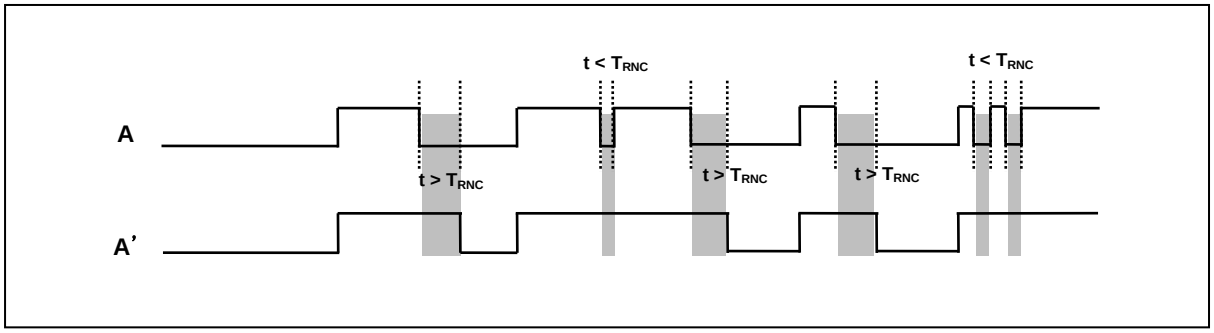


Figure 13-2 Reset Noise Canceller Time Diagram

13.5 Power ON Reset

When rising device power, the POR (Power ON Reset) have 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 input pin.

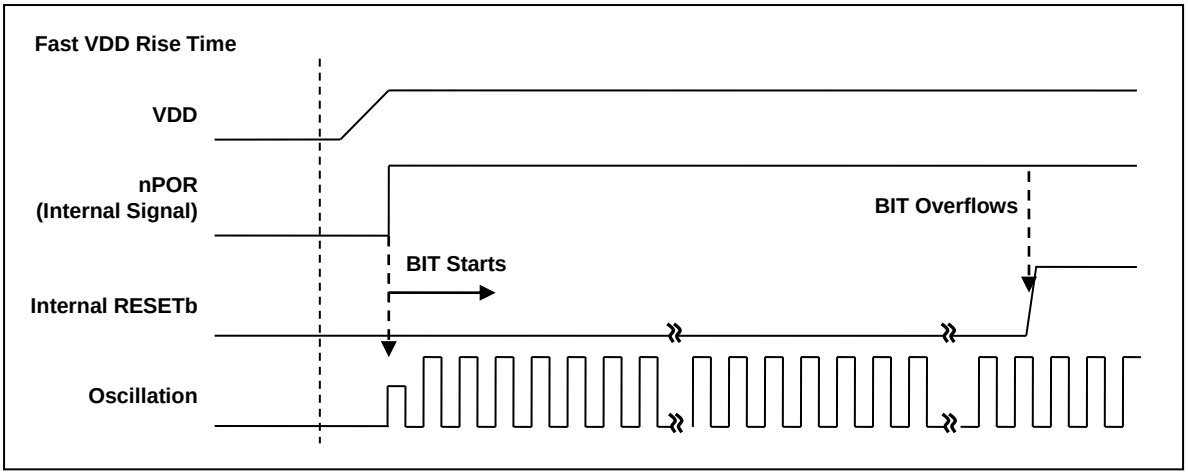


Figure 13-3 Fast VDD Rising Time

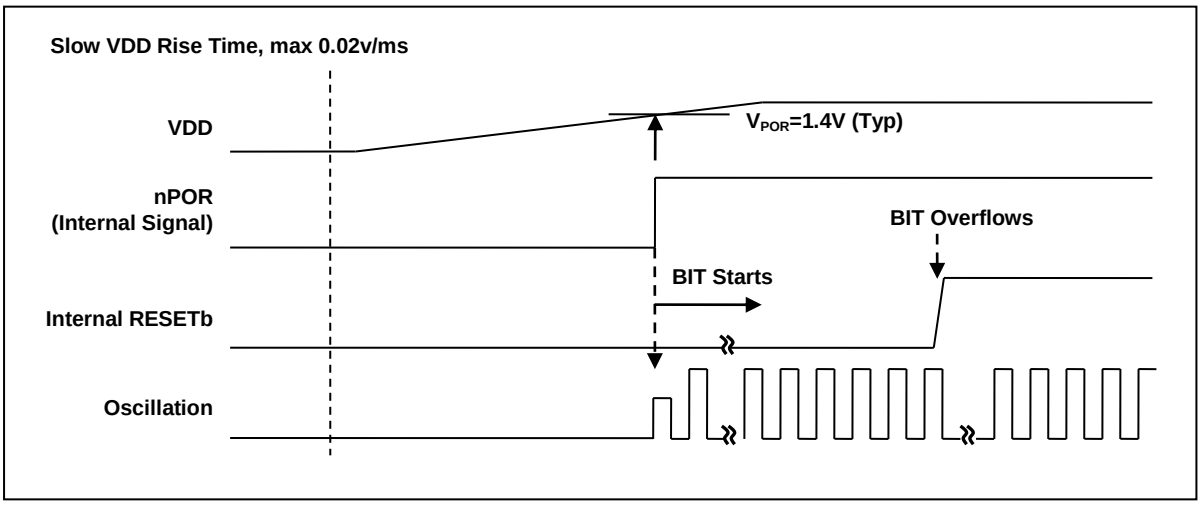


Figure 13-4 Internal RESET Release Timing on Power-Up

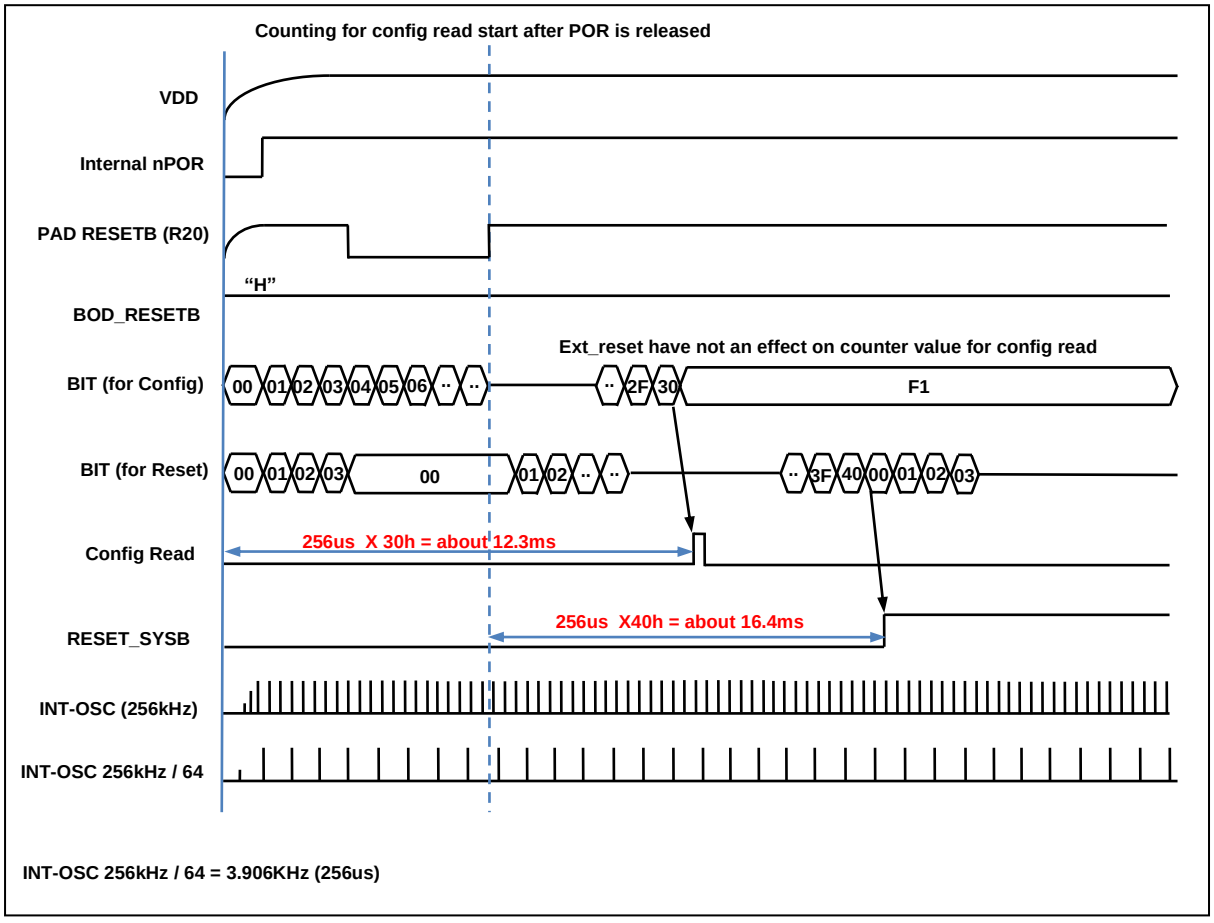


Figure 13-5 Configuration Timing when Power-on

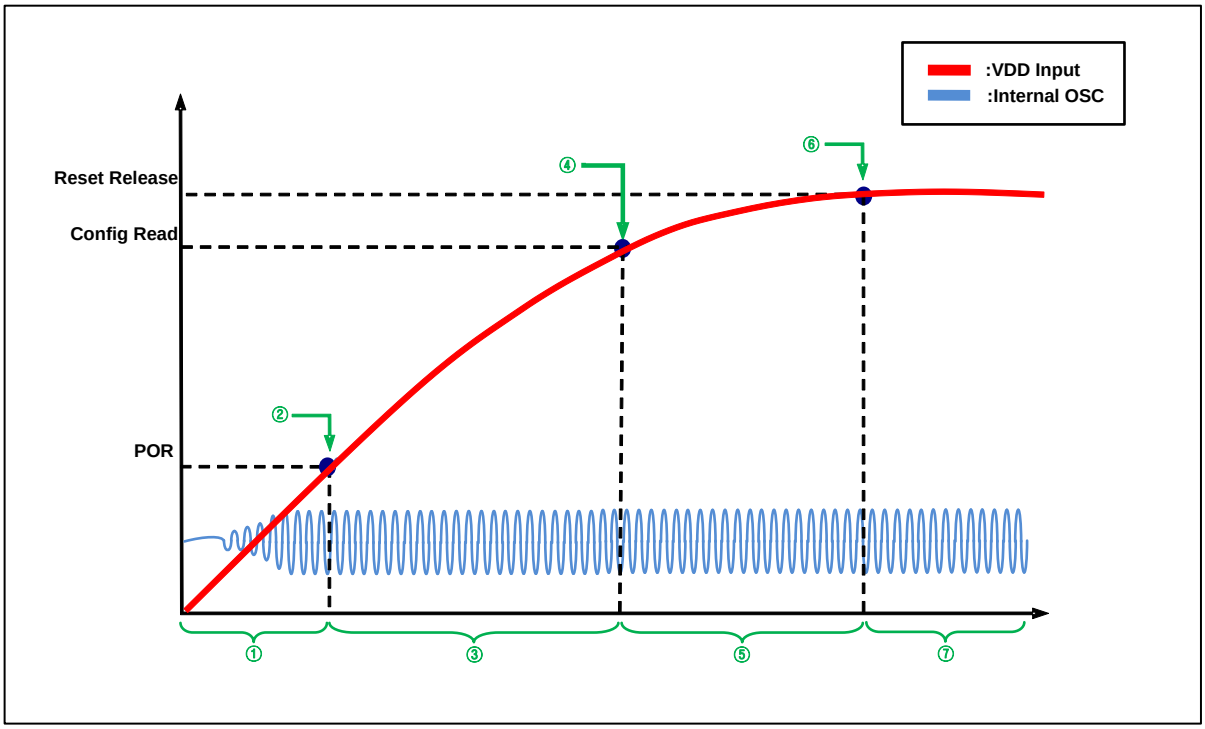


Figure 13-6 Boot Process Waveform

Table 13-2 Boot Process Description

Process	Description	Remarks
①	No operation Internal OSC (256kHz) ON	-about 0.8V
②	1st POR level Detection	-about 1.4V ~ 1.5V
③	(INT-OSC 256kHz/64)×30h Delay section (=12ms) 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	

13.6 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 32us over, within the operating voltage range and oscillation stable, it is applied, and the internal state is initialized. After reset state becomes ‘1’, it needs the stabilization time with 16ms and after the stable state, the internal RESET becomes ‘1’. The Reset process step needs 5 oscillator clocks. And the program execution starts at the vector address stored at address 0000H.

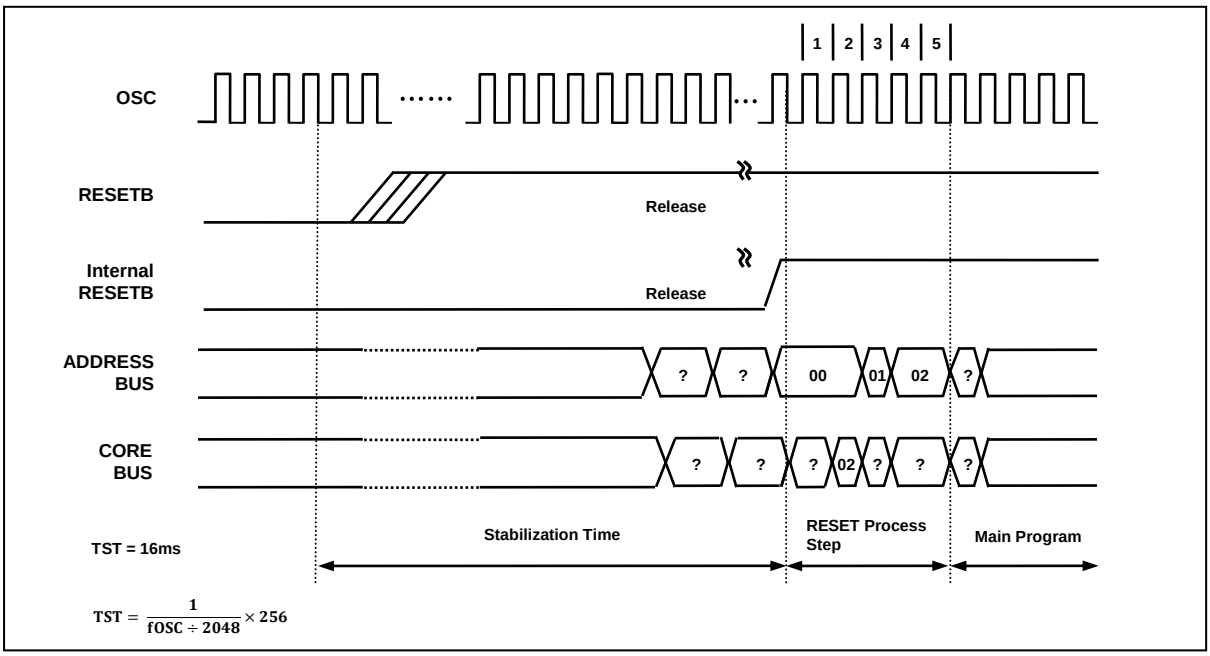


Figure 13-7 Timing Diagram after RESET

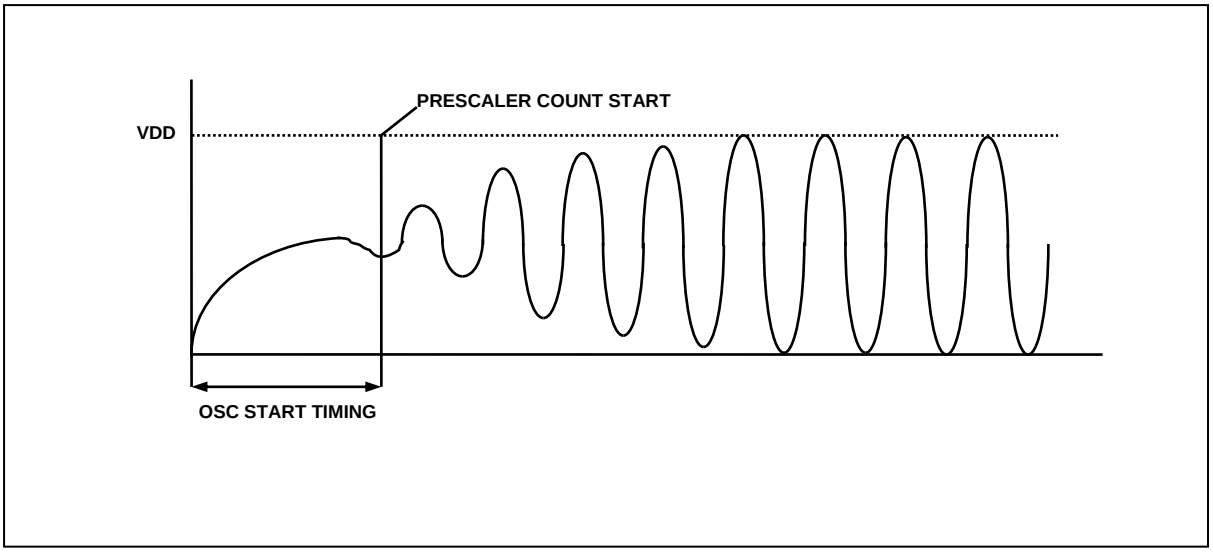


Figure 13-8 Oscillator Generating Waveform Example

Note) as shown Figure 13-8, the stable generating time is not included in the start-up time.

13.7 Brown Out Detector Processor

The A96T336 has an On-chip Brown-out detection and LVI circuit for monitoring the VDD level during operation by comparing it to a fixed trigger level. The trigger level for the BOD and LVI can be selected by BODEN and LVILS[1:0] bit to be 1.6V, 2.5V, 3.6V or 4.2V. In the STOP mode, this will contribute significantly to the total current consumption. So to minimize the current consumption, the BODEN bit is set to off by software.

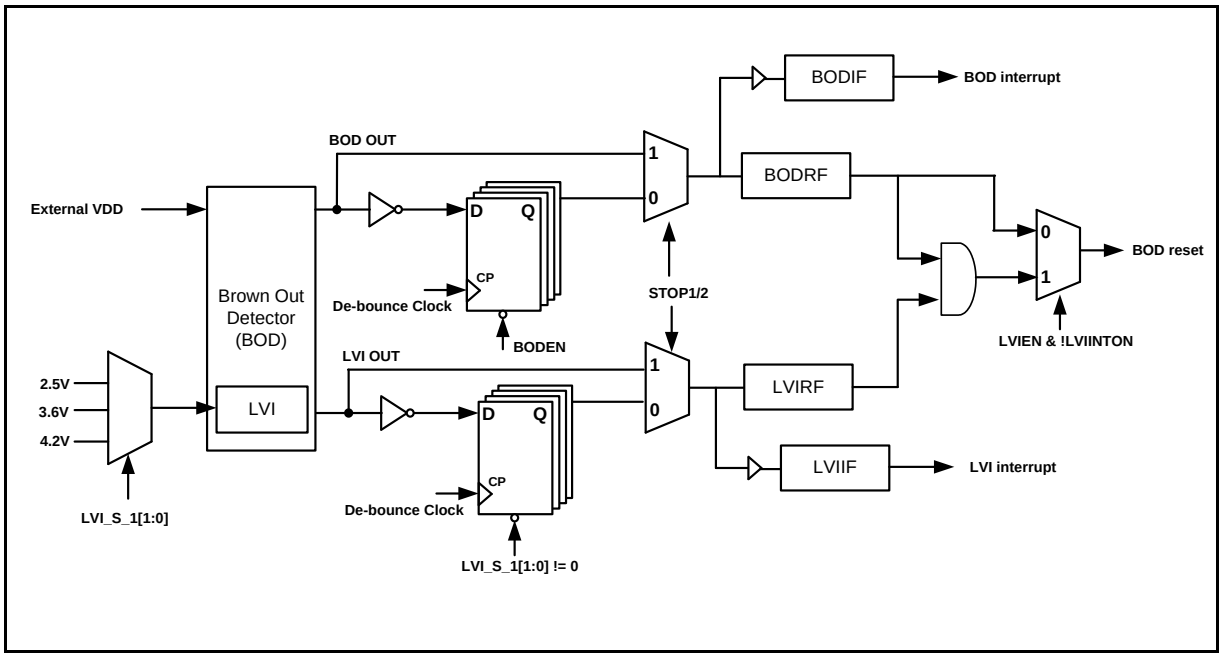


Figure 13-9 Block Diagram of BOD

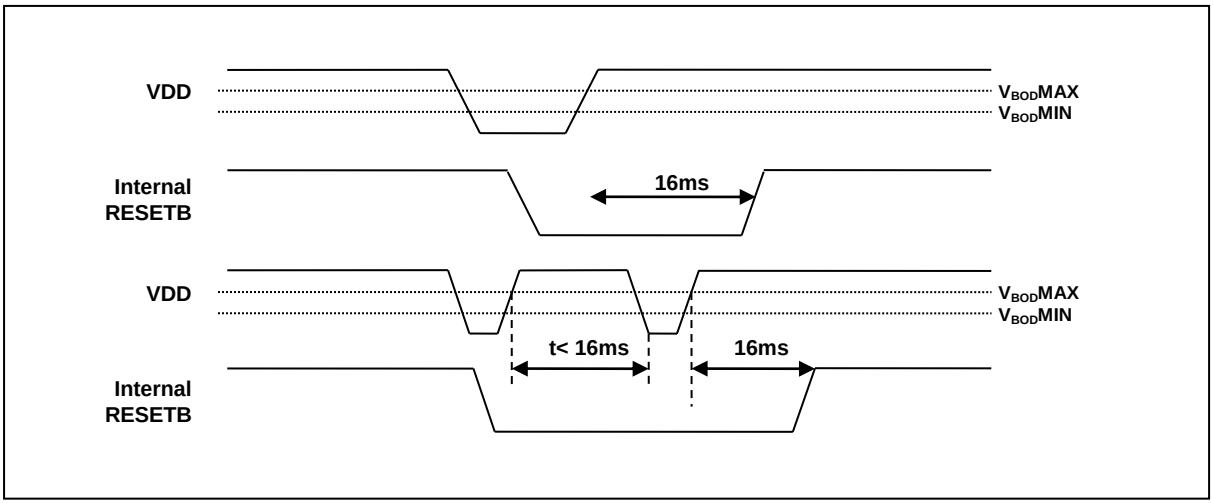


Figure 13-10 Internal Reset at the Power Fail Situation

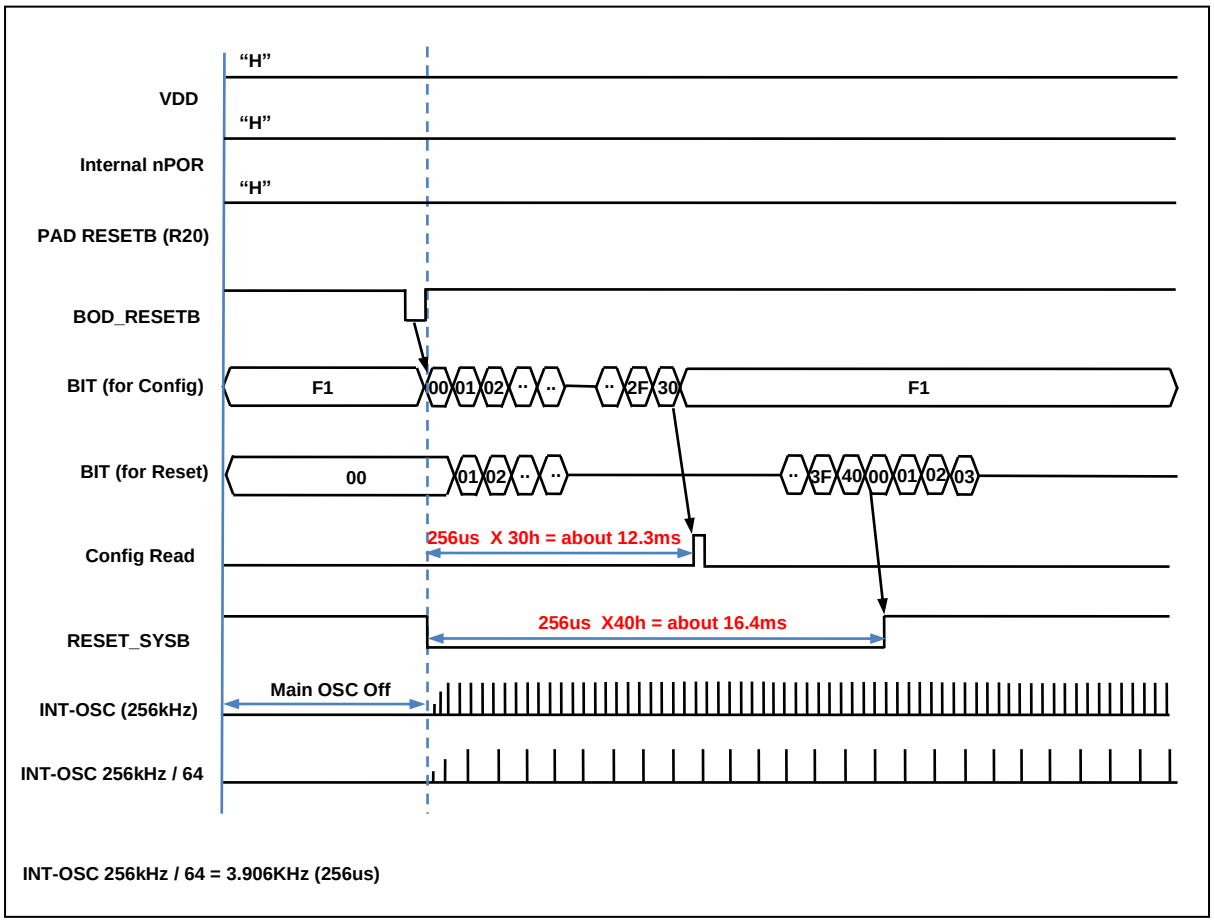


Figure 13-11 Configuration Timing when BOD RESET

13.7.1 Register Map

Table 13-3 Register Map

Name	Address	Dir	Default	Description
BODR	86H	R/W	09H	BOD Control Register
RSFR	8FH	R/W	80H	RESET Source FLAG Register

13.7.2 Reset Operation Register Description

Reset and Interrupt control Register consists of the BOD Control Register (BODR).

13.7.3 Register Description for Reset Operation

BODR (BOD Control Register) : 86H

7	6	5	4	3	2	1	0
-	-	-	-	LVIINTON	LVILS[1]	LVILS[0]	BODEN
-	-	-	-	R/W	R/W	R/W	R/W

Initial value : 09H

LVIINTON Select Interrupt or Reset

0 Reset

1 Interrupt (default)

LVILS[1:0] LVI level Voltage

LVILS[1]	LVILS[0]	Description
0	0	LVI disable (default)
0	1	2.5V
1	0	3.6V
1	1	4.2V

BODEN 1.6V BOD operation

0 BOD disable

1 BOD enable (default)

RSFR (RESET Source FLAG Register) : 8FH

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

Initial value : 80H

PORF Power-On Reset flag bit. The bit is reset by writing '0' to this bit.

0 No detection

1 Detection

EXTRF External Reset 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

BODRF Brown-Out Reset flag bit. The bit is reset by writing '0' to this bit or by Power ON reset.

0 No detection

1 Detection

LVIRF LVI Reset flag bit. The bit is reset by writing '0' to this bit or by Power ON reset.

0 No detection

1 Detection

14. On-chip Debug System

14.1 Overview

14.1.1 Description

On-chip debug System (OCD) of A96T336 can be used for programming the non-volatile memories and on-chip debugging. Detailed descriptions for programming via the OCD interface can be found in the following chapter.

Figure 14-1shows a block diagram of the OCD interface and the On-chip Debug system.

14.1.2 Feature

- Two-wire external interface: 1-wire serial clock input, 1-wire bi-directional serial data bus
- Debugger Access to:
  - All Internal Peripheral Units
  - Internal data RAM
  - Program Counter
  - Flash Memory
- Extensive On-chip Debug Support for Break Conditions, Including
  - Break Instruction
  - Single Step Break
  - Program Memory Break Points on Single Address
  - Programming of Flash, Fuses, and Lock Bits through the two-wire Interface
  - On-chip Debugging Supported by Dr.Choice®
- Operating frequency

  Supports the maximum frequency of the target MCU

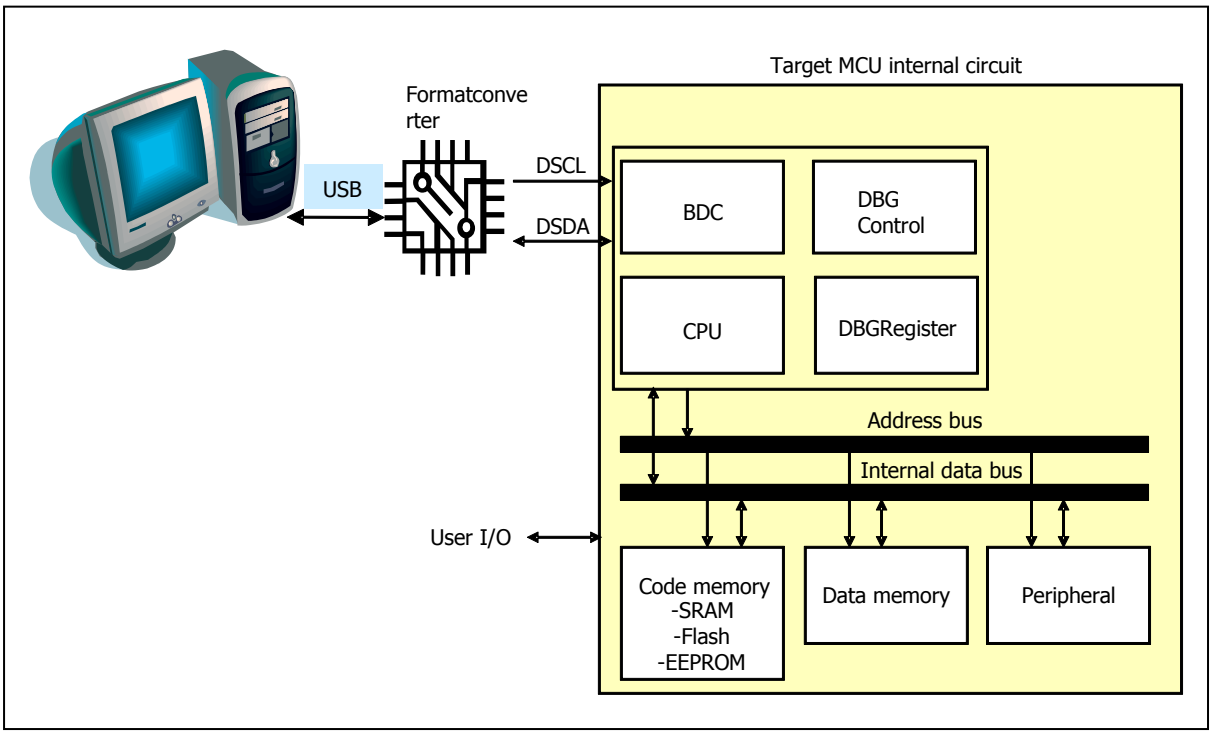


Figure 14-1 Block Diagram of On-chip Debug System

14.2 Two-pin External Interface

14.2.1 Basic Transmission Packet

- 10-bit packet transmission using two-pin interface.
- 1-packet consists of 8-bit data, 1-bit parity and 1-bit acknowledge.
- Parity is even of '1' for 8-bit data in transmitter.
- Receiver generates acknowledge bit as '0' when transmission for 8-bit data and its parity has no error.
- When transmitter has no acknowledge (Acknowledge bit is '1' at tenth clock), error process is executed in transmitter.
- When acknowledge error is generated, host PC makes stop condition and transmits command which has error again.
- Background debugger command is composed of a bundle of packet.
- Star condition and stop condition notify the start and the stop of background debugger command respectively.

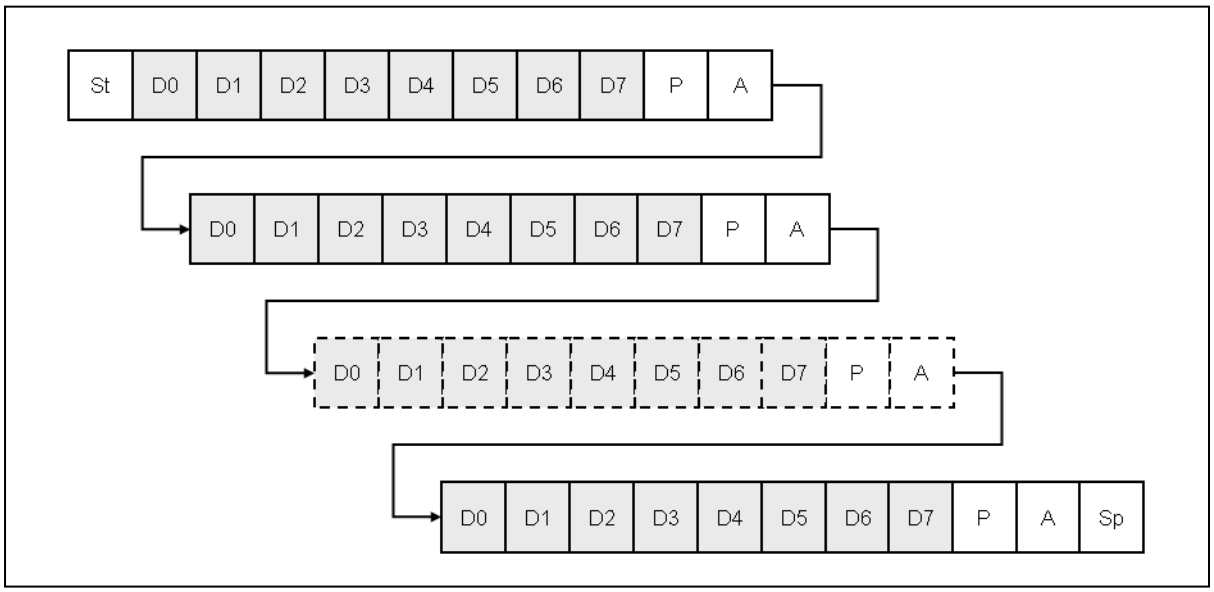


Figure 14-2 10-bit Transmission Packet

14.2.2 Packet Transmission Timing

14.2.2.1 Data transfer

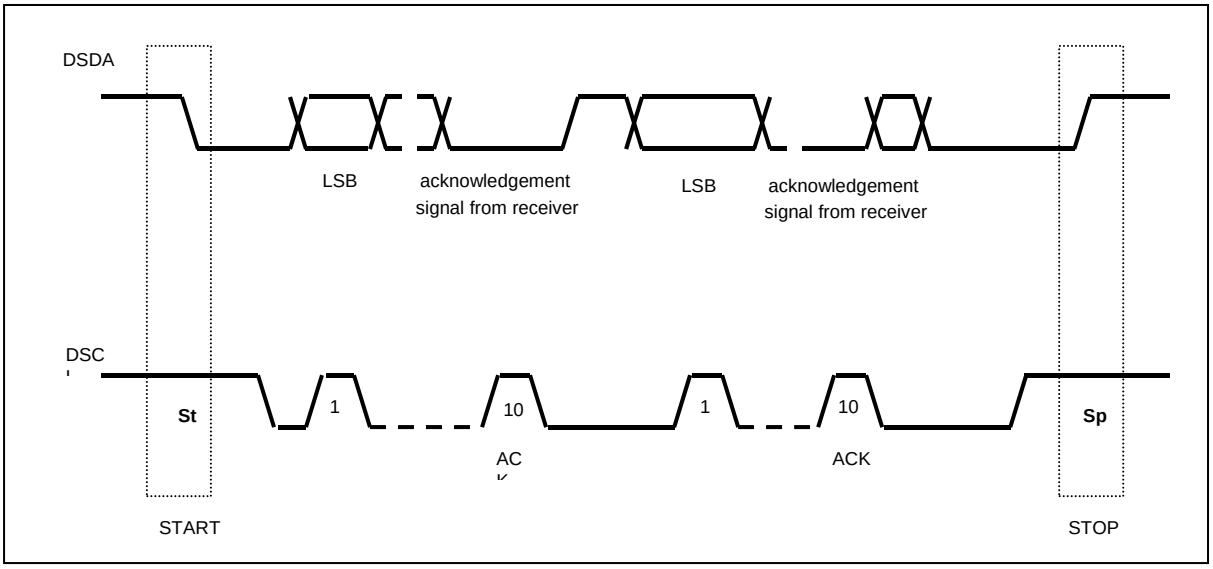


Figure 14-3 Data Transfer on the Twin Bus

14.2.2.2 Bit transfer

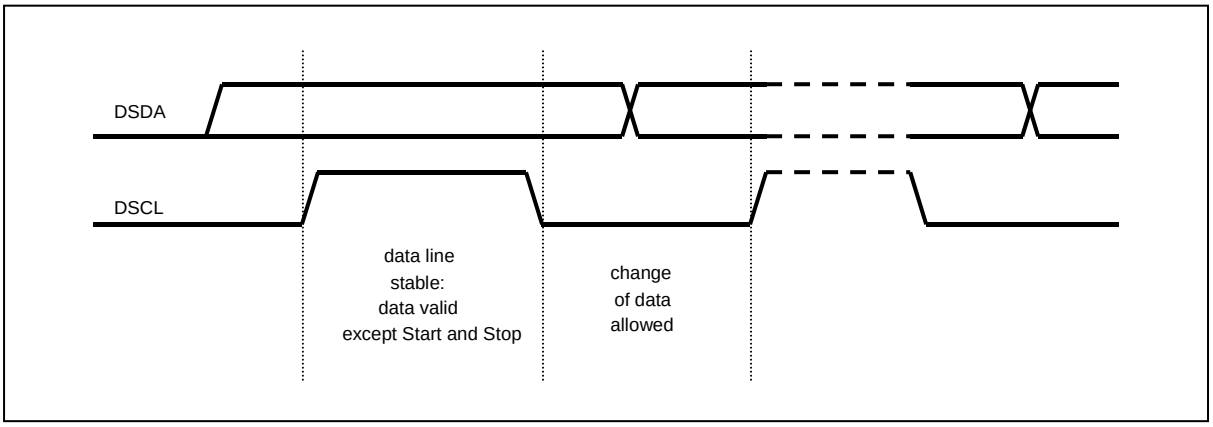


Figure 14-4 Bit Transfer on the Serial Bus

14.2.2.3 Start and stop condition

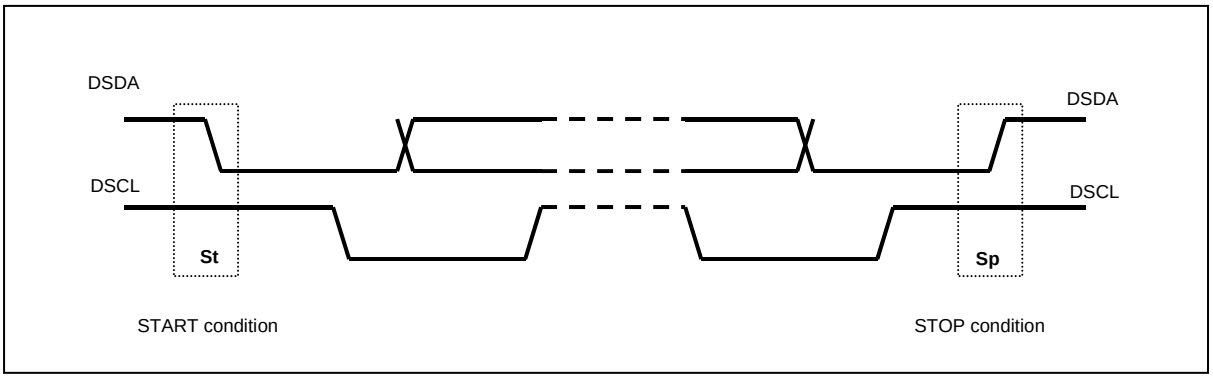


Figure 14-5 Start and Stop Condition

14.2.2.4 Acknowledge bit

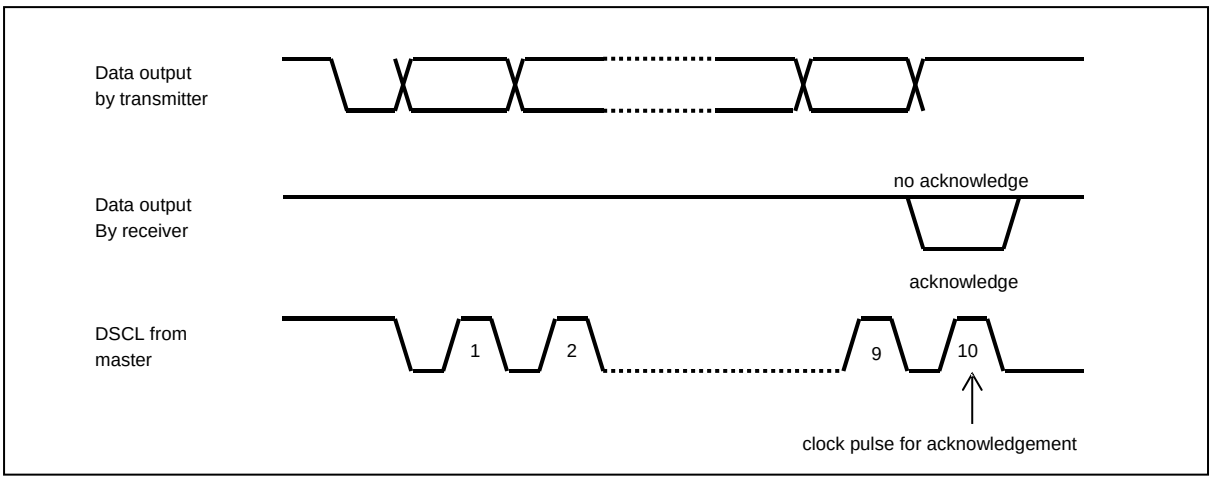


Figure 14-6 Acknowledge on the Serial Bus

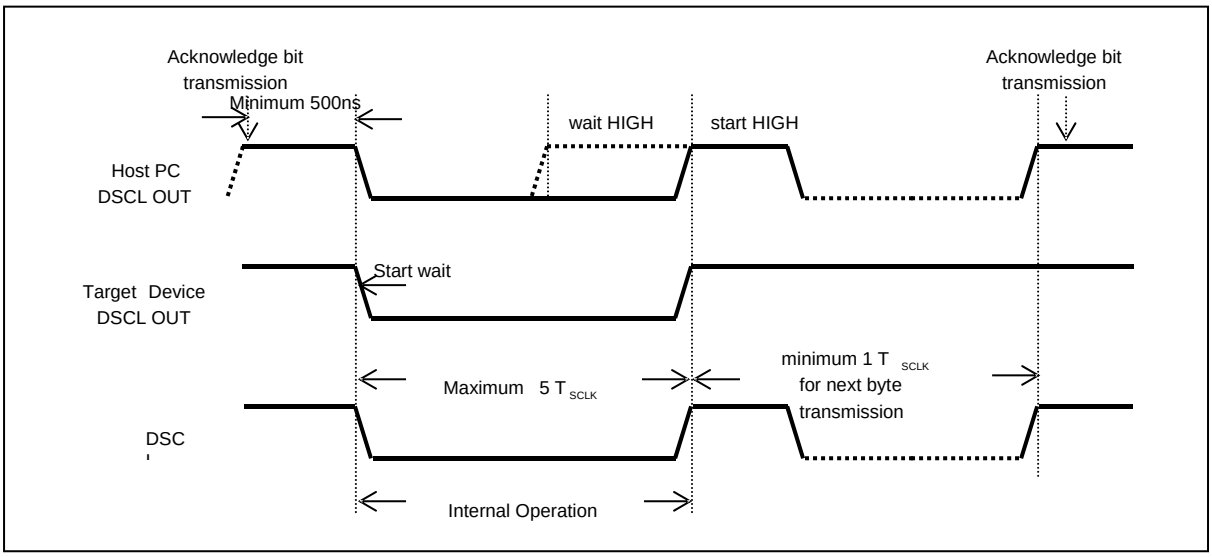


Figure 14-7 Clock Synchronization during Wait Procedure

14.2.3 Connection of Transmission

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

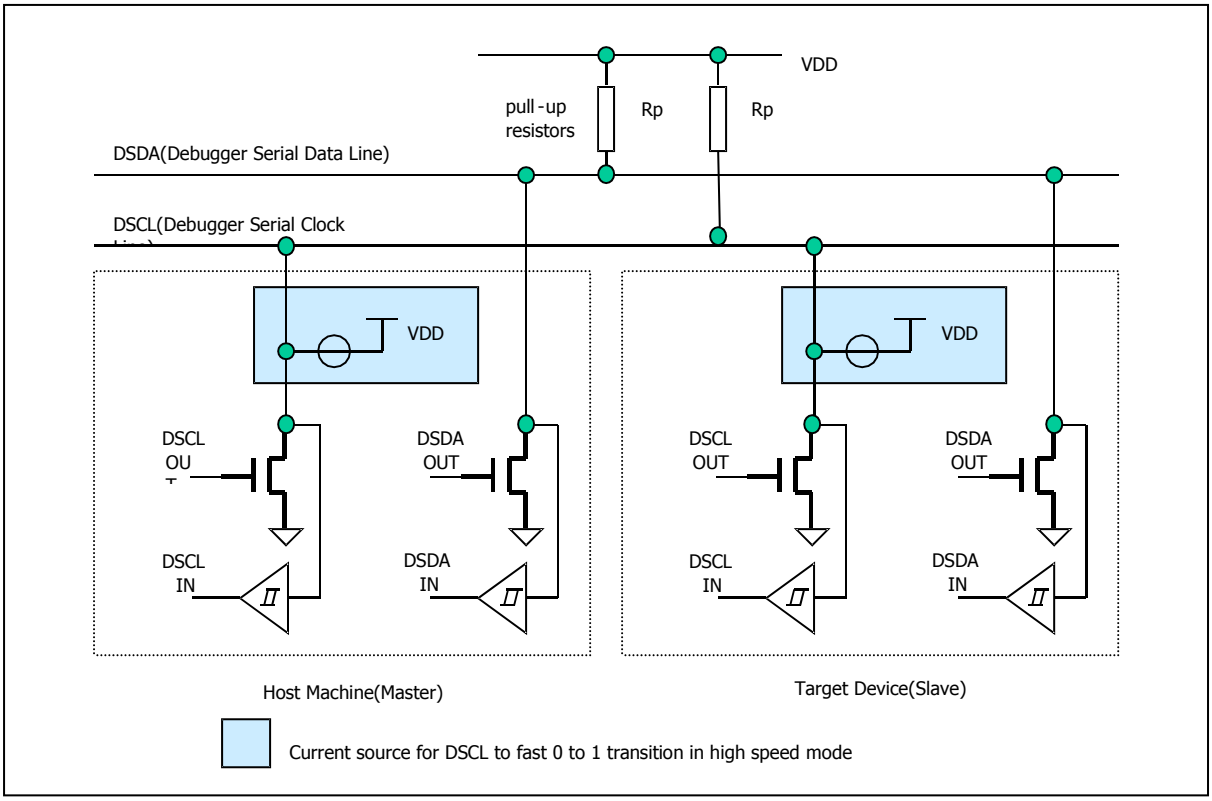


Figure 14-8 Connection of Transmission

15. Memory Programming

15.1 Overview

15.1.1 Description

A96T336 has flash memory to which a program can be written, erased, and overwritten while mounted on the board. Serial ISP mode is supported.

15.1.2 Features

- Flash Size : 16Kbytes
- 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 x (FEARLx), Address Middle Register x (FEARMx), address High Register (FEARH). They are mapped to SFR area and can be accessed only in programming mode.

15.2.1 Register Map

Table 15-1 Register Map

Name	Address	Dir	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
FEARL1	EEH	R/W	00H	Flash Address Low Register 1
FEARM1	EFH	R/W	00H	Flash Address Middle Register 1
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

15.2.2 Register Description for Flash

FEMR (Flash Mode Register) : EAH

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

Initial value : 00H

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

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

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

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

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

VFY Set program or erase verify mode with PGM or ERASE

Program Verify: PGM=1, VFY=1

Erase Verify: ERASE=1, VFY=1

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

FECR (Flash Control Register) : EBH

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

Initial value : 03H

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

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

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

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

1 Start to program or erase of Flash

READ Start auto-verify of Flash. It is cleared automatically after 1 clock

0 No operation

1 Start auto-verify of Flash (Checksum or CRC16)

nFERST Reset Flash control logic. It is set automatically after 1 clock

0 Reset Flash control logic

1 No operation (default)

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

PBUFF	nPBRST	Description
0	0	Page buffer reset
1	0	Page buffer select register reset
X	1	No operation (default)

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	LOCKC	PCRCRD	-	ROMINT	WMODE	EMODE	VMODE
R	R/W	R/W	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

LOCKC Enable Code area write protection from top of BSIZE to {FEARM1,FEARL1}.

0 Disable (default)

1 Enable

PCRCRD Enable CRC16-CCITT or CheckSum features at Flash auto-verify mode.

0 Checksum (default)

1 CRC16-CCITT

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

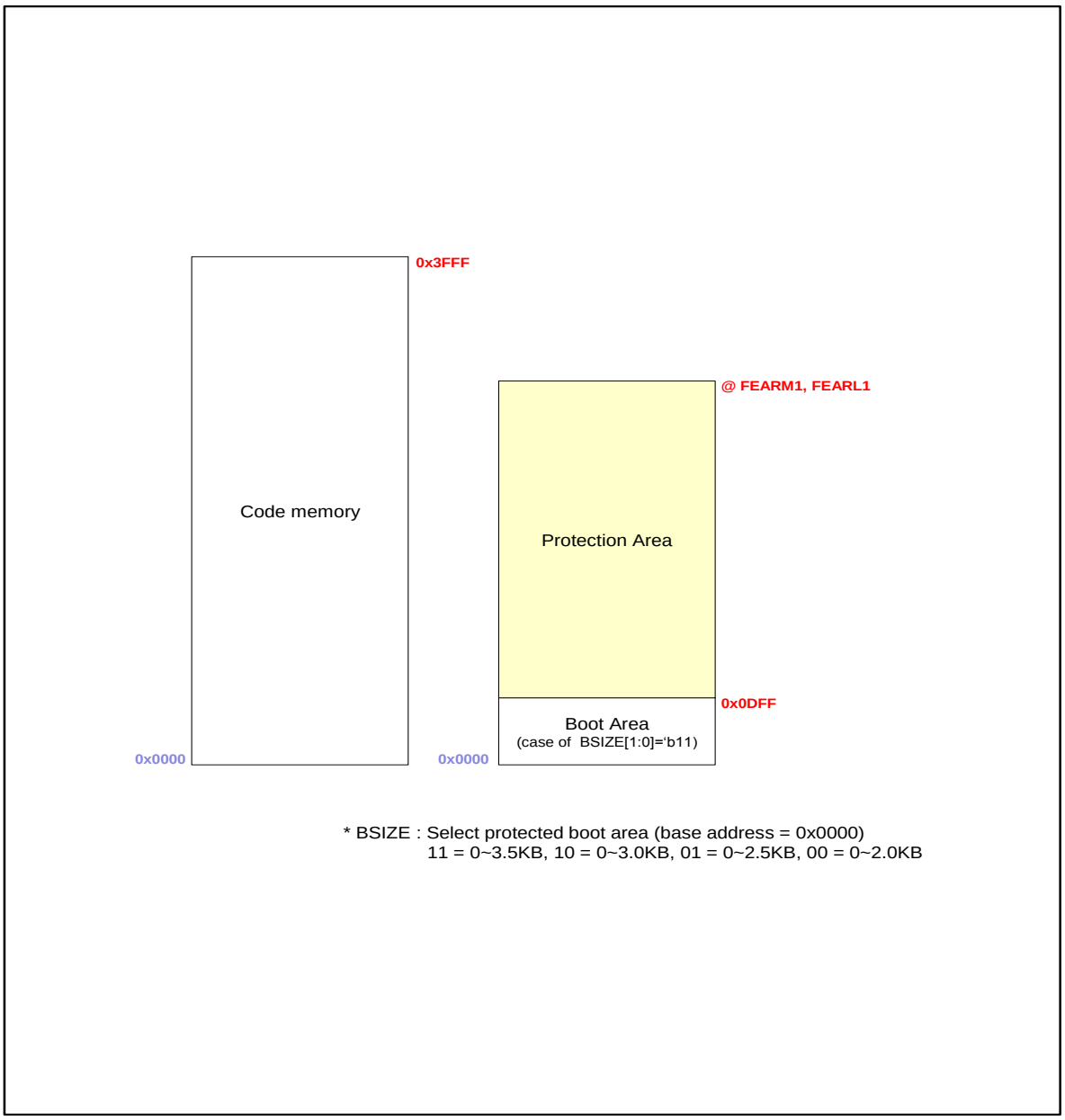


Figure 15-1 Code area write protection by LOCKC (from top of BSIZE to FEARM1, FEARL1)

FEARL1 (Flash address low Register 1) : EEH

7	6	5	4	3	2	1	0
ARL17	ARL16	ARL15	ARL14	ARL13	ARL12	ARL11	ARL10
W	W	W	W	W	W	W	W

Initial value : 00H

ARL1[7:0] Flash address low 1

FEARM1 (Flash address middle Register 1) : EFH

7	6	5	4	3	2	1	0
ARM17	ARM16	ARM15	ARM14	ARM13	ARM12	ARM11	ARM10
W	W	W	W	W	W	W	W

Initial value : 00H

ARM1[7:0] Flash address middle 1

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 or 16-bit CRC result.

This device can support internal CRC or CheckSum calculation, device verification time will be decreased dramatically.

CRC or CheckSum cannot detect error address or error bit, but it is quite good feature in mass product programming.

Device data read out time takes few seconds. However, you can read out device CRC or CheckSum within 10's ~ 100's of milliseconds. It is 100's times faster than normal data read.

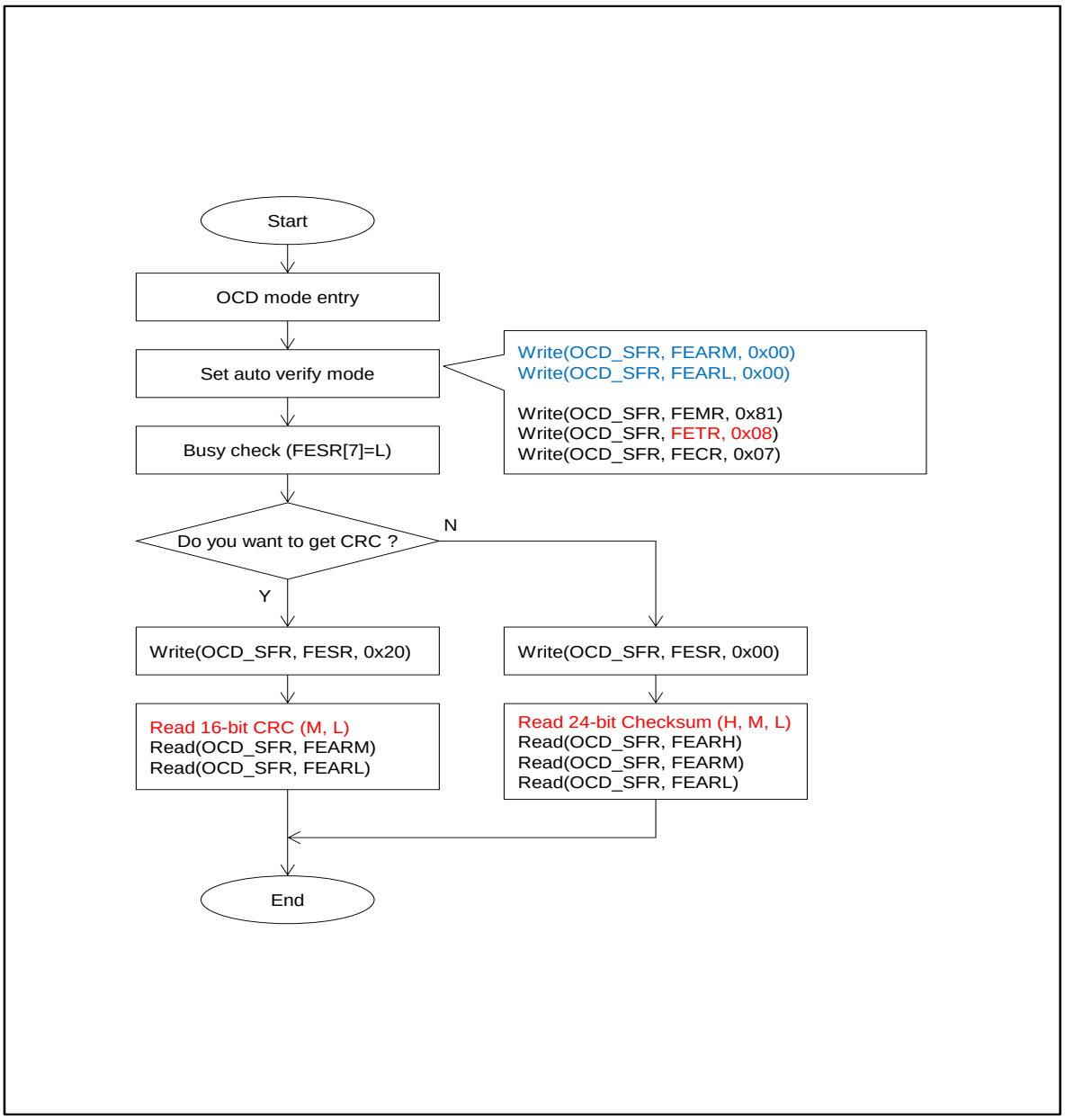


Figure 15-2 Read device internal 16-bit CRC or CkeckSum (Full size)

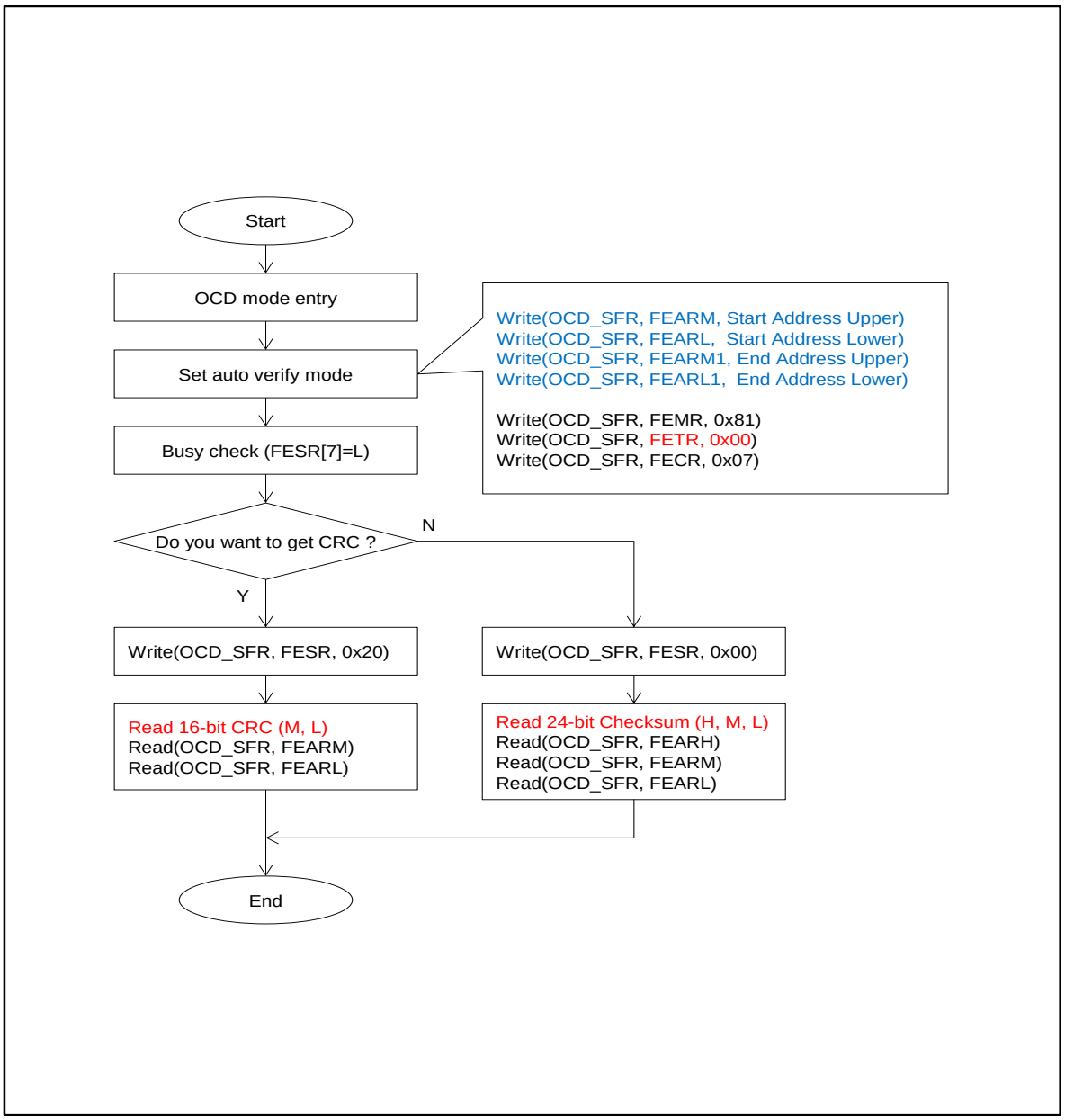


Figure 15-3 Read device internal 16-bit CRC or CkeckSum (User define size)

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 RING clock frequency (f_{RING}=128kHz). 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 RING clock : (255+1) * 2 * (7.8125us) = 4.0ms

In the case of ±10% of error rate of counter source clock, program or erase time is 3.6~4.4ms

* Program/erase time calculation

for page write or erase, Tpe = (TCON+1) * 2 * (f_{RING})

for bulk erase, Tbe = (TCON+1) * 4 * (f_{RING})

Recommended bulk erase time : FETCR = 57h

Recommended program / page erase time : FETCR = AFh

Table 15-2 Program/erase Time

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

15.3 Memory Map

15.3.1 Flash Memory Map

Program memory uses 16K bytes of Flash memory. It is read by byte and written by byte or page. One page is 32 bytes

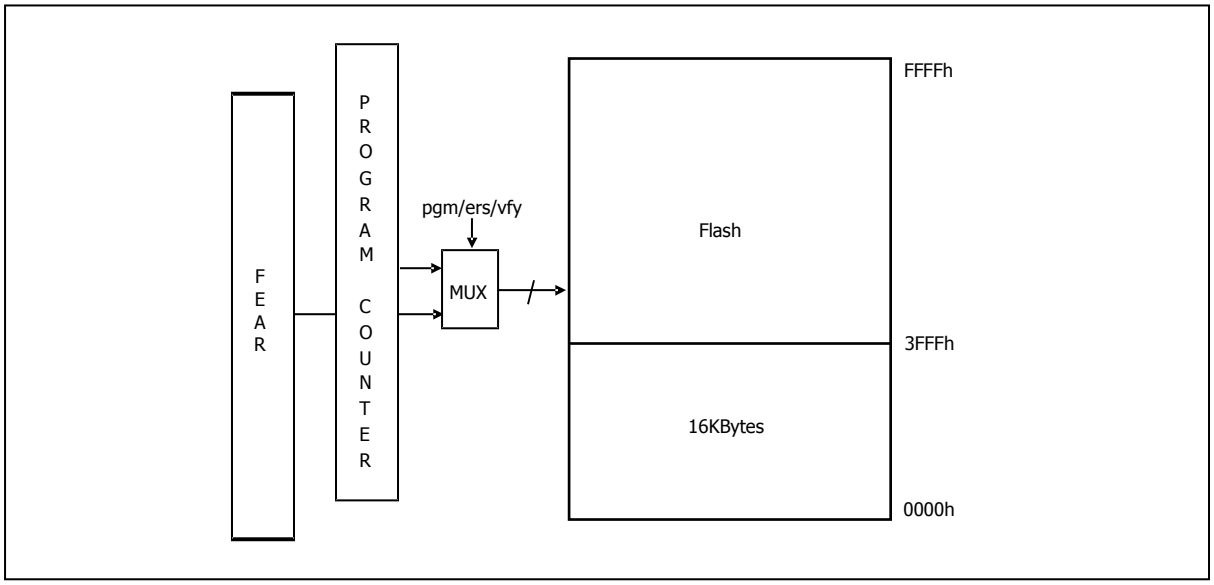


Figure 15-4 Flash Memory Map

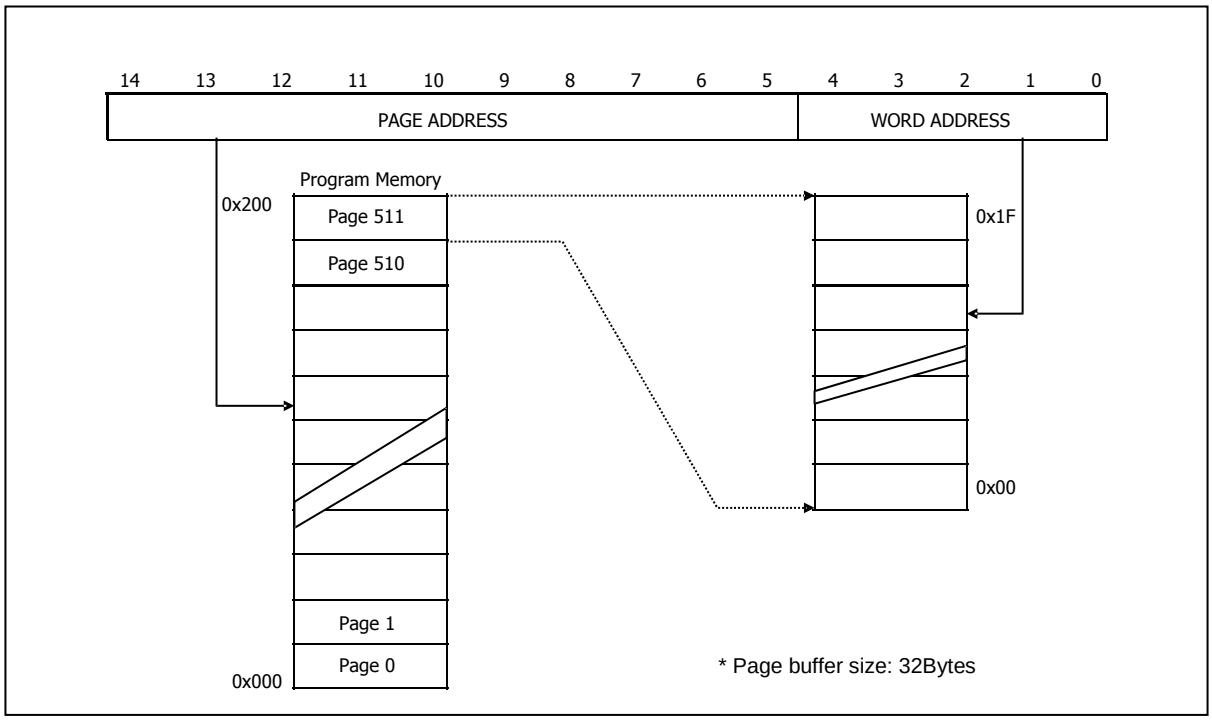


Figure 15-5 Address Configuration of Flash Memory

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

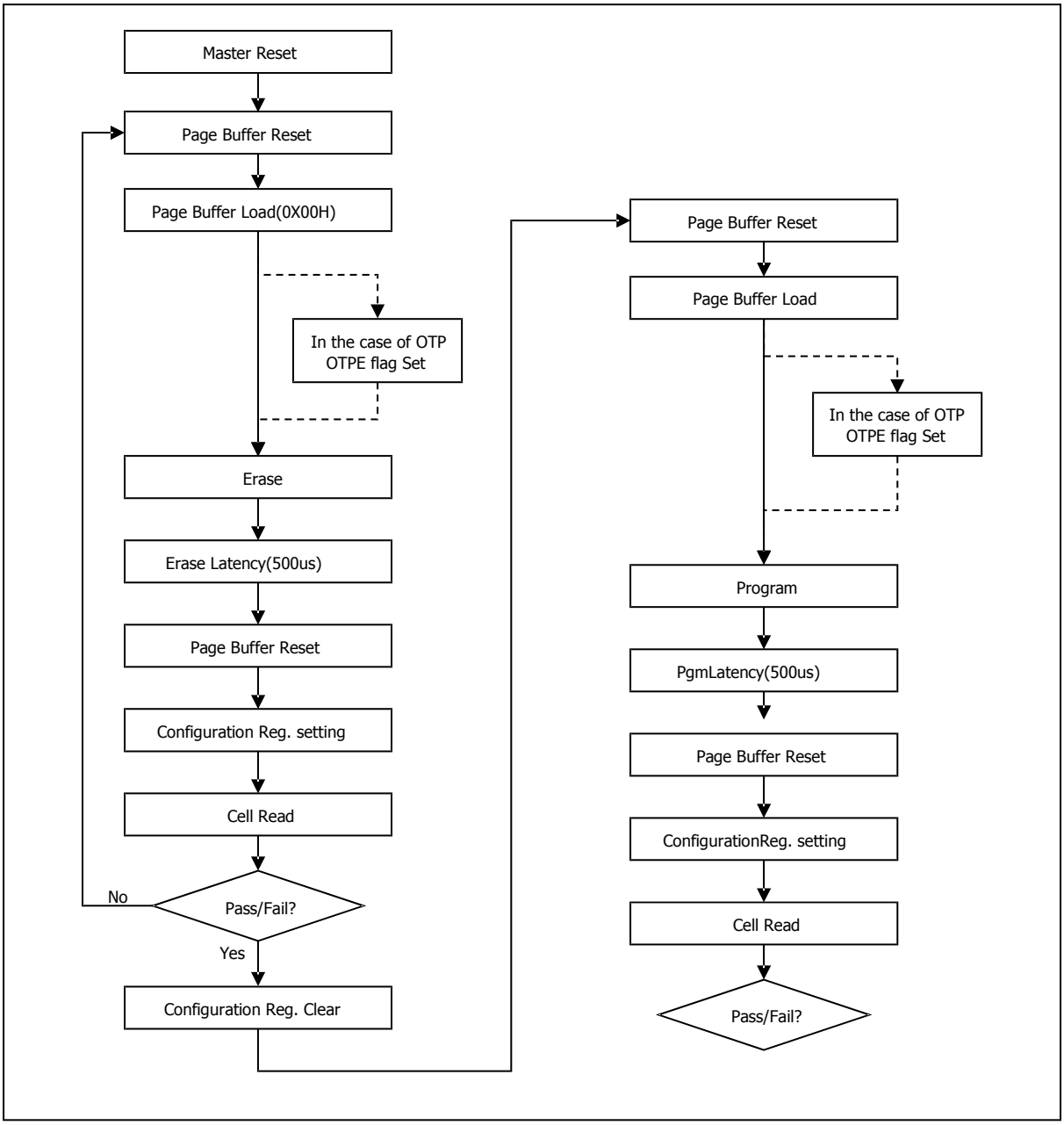


Figure 15-6 The Sequence of Page Program and Erase of Flash Memory

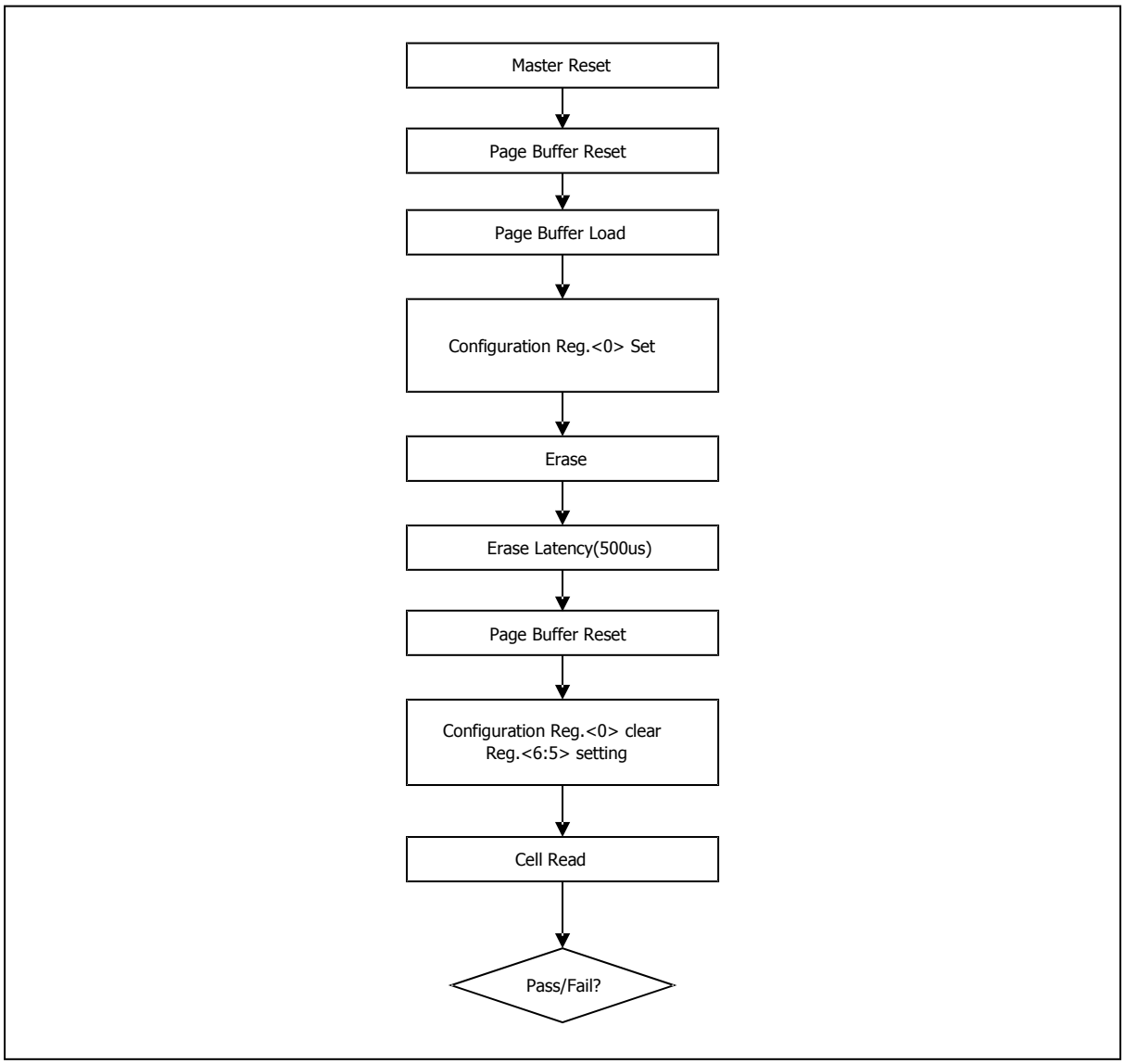


Figure 15-7 The Sequence of Bulk Erase of Flash Memory

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 Summary of Flash Program/Erase Mode

Table 15-3 Operation 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.

15.5 Mode Entrance Method of ISP Mode

15.5.1 Mode Entrance Method for ISP

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

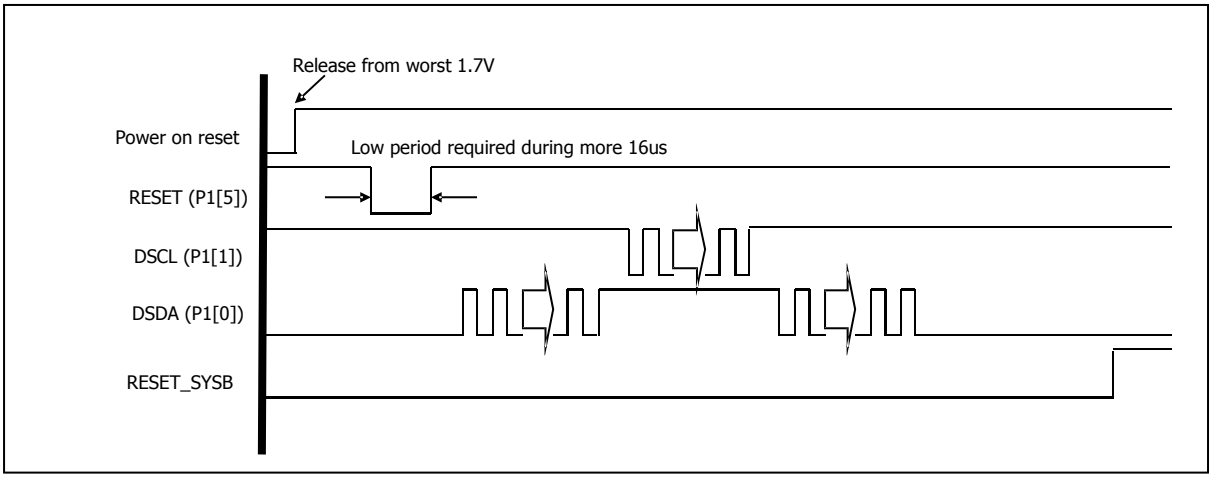


Figure 15-8 ISP Mode

15.6 Security

A96T336 provides Lock bits which can be left un-programmed (“0”) or can be programmed (“1”) to obtain the additional features listed in Table 15-4. The Lock bit can only be erased to “0” with the bulk erase command and a value of more than 0x40 at FETCR.

Table 15-4 Security policy using lock-bits

LOCK MODE	USER MODE								ISP MODE							
	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

- 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) : 2F50H

7	6	5	4	3	2	1	0
BSIZE1	BSIZE0	-	-	GPIOSEL	LOCKB	-	LOCKF
R	R	-	-	R	R	-	R

Initial value : 00H

BSIZE Select Specific Area for Write Protection
Note) When LOCKB is set, it is applied.
00 000h~7FFh (2KB)
01 000h~9FFh (2.5KB)
10 000h~BFFh (3KB)
11 000h~DFFh (3.5KB)

GPIOSEL Select GPIO or External Reset
0 Select External Reset (default)
1 Select GPIO (P15)

LOCKB Enable Specific Area (Boot Area) Write Protection
0 LOCK Disable
1 LOCK Enable

LOCKF CODE Read Protection
0 LOCK Disable
1 LOCK Enable

17. APPENDIX

A. Instruction Table

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

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

ARITHMETIC				
Mnemonic	**Description**	**Bytes**	**Cycles**	**Hex code**
ADD A,Rn	Add register to A	1	1	28-2F
ADD A,dir	Add direct byte to A	2	1	25
ADD A,@Ri	Add indirect memory to A	1	1	26-27
ADD A,#data	Add immediate to A	2	1	24
ADDC A,Rn	Add register to A with carry	1	1	38-3F
ADDC A,dir	Add direct byte to A with carry	2	1	35
ADDC A,@Ri	Add indirect memory to A with carry	1	1	36-37
ADDC A,#data	Add immediate to A with carry	2	1	34
SUBB A,Rn	Subtract register from A with borrow	1	1	98-9F
SUBB A,dir	Subtract direct byte from A with borrow	2	1	95
SUBB A,@Ri	Subtract indirect memory from A with borrow	1	1	96-97
SUBB A,#data	Subtract immediate from A with borrow	2	1	94
INC A	Increment A	1	1	04
INC Rn	Increment register	1	1	08-0F
INC dir	Increment direct byte	2	1	05
INC @Ri	Increment indirect memory	1	1	06-07
DEC A	Decrement A	1	1	14
DEC Rn	Decrement register	1	1	18-1F
DEC dir	Decrement direct byte	2	1	15
DEC @Ri	Decrement indirect memory	1	1	16-17
INC DPTR	Increment data pointer	1	2	A3
MUL AB	Multiply A by B	1	4	A4
DIV AB	Divide A by B	1	4	84
DA A	Decimal Adjust A	1	1	D4

LOGICAL				
Mnemonic	**Description**	**Bytes**	**Cycles**	**Hex code**
ANL A,Rn	AND register to A	1	1	58-5F
ANL A,dir	AND direct byte to A	2	1	55
ANL A,@Ri	AND indirect memory to A	1	1	56-57
ANL A,#data	AND immediate to A	2	1	54
ANL dir,A	AND A to direct byte	2	1	52
ANL dir,#data	AND immediate to direct byte	3	2	53
ORL A,Rn	OR register to A	1	1	48-4F
ORL A,dir	OR direct byte to A	2	1	45
ORL A,@Ri	OR indirect memory to A	1	1	46-47
ORL A,#data	OR immediate to A	2	1	44
ORL dir,A	OR A to direct byte	2	1	42
ORL dir,#data	OR immediate to direct byte	3	2	43
XRL A,Rn	Exclusive-OR register to A	1	1	68-6F
XRL A,dir	Exclusive-OR direct byte to A	2	1	65

XRL A, @Ri	Exclusive-OR indirect memory to A	1	1	66-67
XRL A,#data	Exclusive-OR immediate to A	2	1	64
XRL dir,A	Exclusive-OR A to direct byte	2	1	62
XRL dir,#data	Exclusive-OR immediate to direct byte	3	2	63
CLR A	Clear A	1	1	E4
CPL A	Complement A	1	1	F4
SWAP A	Swap Nibbles of A	1	1	C4
RL A	Rotate A left	1	1	23
RLC A	Rotate A left through carry	1	1	33
RR A	Rotate A right	1	1	03
RRC A	Rotate A right through carry	1	1	13

DATA TRANSFER				
Mnemonic	**Description**	**Bytes**	**Cycles**	**Hex code**
MOV A,Rn	Move register to A	1	1	E8-EF
MOV A,dir	Move direct byte to A	2	1	E5
MOV A,@Ri	Move indirect memory to A	1	1	E6-E7
MOV A,#data	Move immediate to A	2	1	74
MOV Rn,A	Move A to register	1	1	F8-FF
MOV Rn,dir	Move direct byte to register	2	2	A8-AF
MOV Rn,#data	Move immediate to register	2	1	78-7F
MOV dir,A	Move A to direct byte	2	1	F5
MOV dir,Rn	Move register to direct byte	2	2	88-8F
MOV dir,dir	Move direct byte to direct byte	3	2	85
MOV dir,@Ri	Move indirect memory to direct byte	2	2	86-87
MOV dir,#data	Move immediate to direct byte	3	2	75
MOV @Ri,A	Move A to indirect memory	1	1	F6-F7
MOV @Ri,dir	Move direct byte to indirect memory	2	2	A6-A7
MOV @Ri,#data	Move immediate to indirect memory	2	1	76-77
MOV DPTR,#data	Move immediate to data pointer	3	2	90
MOVC A,@A+DPTR	Move code byte relative DPTR to A	1	2	93
MOVC A,@A+PC	Move code byte relative PC to A	1	2	83
MOVX A,@Ri	Move external data(A8) to A	1	2	E2-E3
MOVX A,@DPTR	Move external data(A16) to A	1	2	E0
MOVX @Ri,A	Move A to external data(A8)	1	2	F2-F3
MOVX @DPTR,A	Move A to external data(A16)	1	2	F0
PUSH dir	Push direct byte onto stack	2	2	C0
POP dir	Pop direct byte from stack	2	2	D0
XCH A,Rn	Exchange A and register	1	1	C8-CF
XCH A,dir	Exchange A and direct byte	2	1	C5
XCH A,@Ri	Exchange A and indirect memory	1	1	C6-C7
XCHD A,@Ri	Exchange A and indirect memory nibble	1	1	D6-D7

BOOLEAN				
Mnemonic	**Description**	**Bytes**	**Cycles**	**Hex code**
CLR C	Clear carry	1	1	C3
CLR bit	Clear direct bit	2	1	C2
SETB C	Set carry	1	1	D3
SETB bit	Set direct bit	2	1	D2
CPL C	Complement carry	1	1	B3
CPL bit	Complement direct bit	2	1	B2

ANL C,bit	AND direct bit to carry	2	2	82
ANL C,/bit	AND direct bit inverse to carry	2	2	B0
ORL C,bit	OR direct bit to carry	2	2	72
ORL C,/bit	OR direct bit inverse to carry	2	2	A0
MOV C,bit	Move direct bit to carry	2	1	A2
MOV bit,C	Move carry to direct bit	2	2	92

BRANCHING				
Mnemonic	**Description**	**Bytes**	**Cycles**	**Hex code**
ACALL addr 11	Absolute jump to subroutine	2	2	11→F1
LCALL addr 16	Long jump to subroutine	3	2	12
RET	Return from subroutine	1	2	22
RETI	Return from interrupt	1	2	32
AJMP addr 11	Absolute jump unconditional	2	2	01→E1
LJMP addr 16	Long jump unconditional	3	2	02
SJMP rel	Short jump (relative address)	2	2	80
JC rel	Jump on carry = 1	2	2	40
JNC rel	Jump on carry = 0	2	2	50
JB bit,rel	Jump on direct bit = 1	3	2	20
JNB bit,rel	Jump on direct bit = 0	3	2	30
JBC bit,rel	Jump on direct bit = 1 and clear	3	2	10
JMP @A+DPTR	Jump indirect relative DPTR	1	2	73
JZ rel	Jump on accumulator = 0	2	2	60
JNZ rel	Jump on accumulator ≠0	2	2	70
CJNE A,dir,rel	Compare A,directjne relative	3	2	B5
CJNE A,#d,rel	Compare A,immediatejne 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 op-codes 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.