Smart Proximity Sensing 8-bit MCU for SAR

Main features

8-bit Microcontroller with high performance M8051 CPU

8CH Self Capactitive Touch Sensing

- Fast Initial Self-Calibration
- The Improvement of the SNR by Bias-Calibration in Analog Sensing Block
- Support Offset Compensation
- 16-bit Sensing Resolution
- Support Wakeup at Power Down
- Internal 3.0V/2.5V LDO Reference (Option)
- Max. Operation Shunt Cap 200pF

Basic MCU Function

- 16K bytes Flash Code Memory
- Code Area Protection
- 256 bytes SRAM Data Memory
- 1,792 bytes XRAM

Built-in Analog Function

- Power-On Reset and Brown Out Detecter Reset
- Internal 16MHz/128kHz RC Oscillator

8CH 12-bit AD Converter

Peripherals

- Timer/Counter : 16-bit X 1CH
- 1CH 16-bit PWM (using Timer0)
- I2C with 1.8V Interface

I/O and Packages

- Up to 11 Programmable I/O Lines with 16QFN
- Up to 8 Programmable I/O Lines with 10DFN
- Up to 6 Programmable I/O Lines with 8WLCSP

Operating Voltage

- 2.7V ~ 3.6V (@16MHz)

Operating Conditions

- -40°C to 85°C temperature range

Application

- SAR Sensing Application, Touch Key Application

A96T346

User's Manual

V 1.2

Revised 12 March, 2019

Revision history

Version	Date	Revision list
1.0	2018.03.19	First Release
1.1	2018.06.08	Added the information of A96T346HW (8 WLCSP)
1.2	2019.03.12	Midified information of A96T346DF (10 DFN)

Version 1.2

Published by AE team

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

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

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

1. Overview

1.1 Description

The A96T346 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 : 8CH self-capacitive touch sensing, 16K bytes of FLASH, 256 bytes of SRAM, 1792 bytes of XRAM, 8CH AD-converter, 16-bit timer/counter, watchdog timer, I^2C, on-chip POR, BOD and 16-bit PWM output, on-chip oscillator and clock circuitry. The A96T346 also supports power saving modes to reduce power consumption.

Device name	ROM size	SRAM size	Package
A96T346AUN	16Kbytes FLASH	I:256 bytes X:1792 bytes	16 QFN
A96T346DF			10 DFN
A96T346HW			8 WLCSP

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**
  - 11 Ports (P0[4:0], P1[5:0]) : 16 Pin Package
  - 8 Ports (P0[3:1], P1[4:0]) : 10 Pin Package
  - 6 Ports (P01, P1[4:0]) : 8 Pin Package
- **8-Ch SelfCapacitive Touch Sensing**
  - Fast Initial Self-Calibration.
  - The Improvement of the SNR by Bias-Calibration in Analog Sensing Block.
  - Support Offset Compensation
  - 16-bit Sensing Resolutions
  - Support Wakeup at Power down
  - Internal 3.0V/2.5V LDO reference (option)
  - Maximum Operation Shunt Cap 200pF
- **8-Ch 12-bit AD Converter**
  - Internal 3.0V/2.5V LDO reference (option)
- **Basic Interval Timer**
- **Timer/ Counter(with Timer-Out)**
  - 16Bit×1Ch
- **Watch Dog Timer**
- **I²C with 1.8V Interface**
- **Interrupt Sources**
  - External (1)
  - I²C (1)
  - BIT (1)
  - WDT (1)
  - TIMER (1)
  - TOUCH (1)
  - ROM (1)
- **On-Chip RC-Oscillator**
  - 16MHz OSC (±5%@-40~+85℃)
- **On-Chip WDT-Oscillator**
  - 128kHz OSC (±20%@-40~+85℃)
- **Power On Reset & Brown-Out Detector**
  - POR 1.2V
  - BOD 1.6V
- **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 ~ 3.6V (@ 16MHz)
- **Operating Temperature : -40 ~ +85℃**
- **Package Type**
  - 16 QFN
  - 10 DFN
  - 8 WLCSP

1.3 Ordering Information

Table 1-1 Ordering Information of A96T346

Device name	ROM size	SRAM size	Package
A96T346AUN	16Kbytes FLASH	I:256 bytes X:1792 bytes	16 QFN
A96T346DF			10 DFN
A96T346HW			8 WLCSP

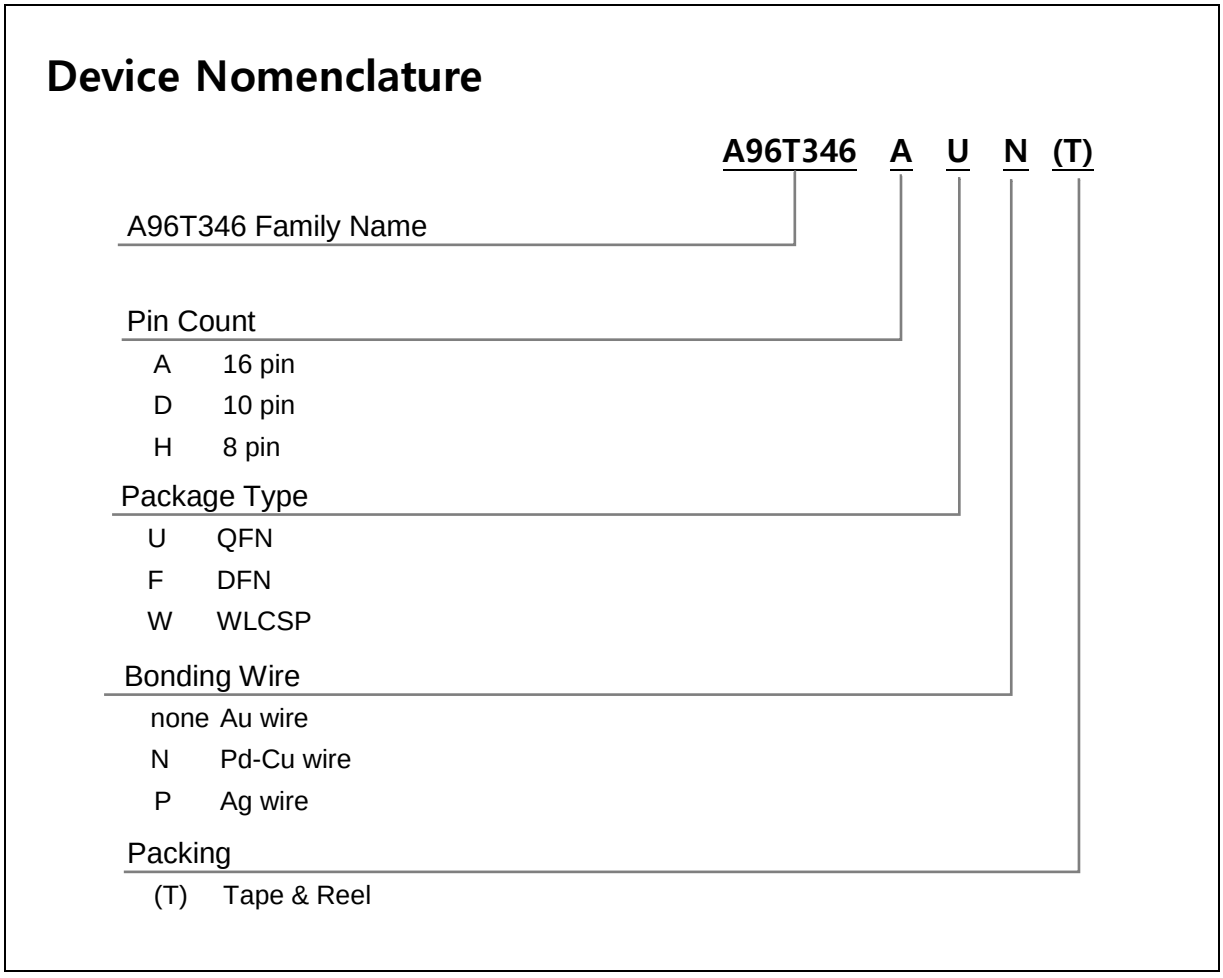


Figure 1-1 Device Nomenclature of A96T346

1.4 Development Tools

1.4.1 Compiler

ABOV semiconductor does not provide any compiler for the A96T346. But the CPU core of A96T346 is M8051 core, you can use all kinds of third party's standard 8051 compiler like Keil C Compiler, Open Source SDCC (Small Device C Compiler). These compilers' output debug information can be integrated with our OCD1 emulator and debugger. Refer to OCD1 manual for more details.

1.4.2 OCD1 Emulator and Debugger

The OCD1 emulator supports ABOV Semiconductor's 8051 series MCU emulation.

The OCD1 interface uses two wires interfacing between PC and MCU which is attached to user's system. The OCD1 can read or change the value of MCU internal memory and I/O peripherals. And also the OCD1 controls MCU internal debugging logic, it means OCD1 controls emulation, step run, monitoring etc.

The OCD1 Debugger program works on Microsoft-Windows NT, 2000, XP, Vista (32-bit), Windows 8, Windows 10 operating system.

If you want to see more details, please refer OCD1 debugger manual. You can download debugger S/W and manual from our web-site.

- P11 (A96T346 DSCL pin)
- P10 (A96T346 DSDA pin)

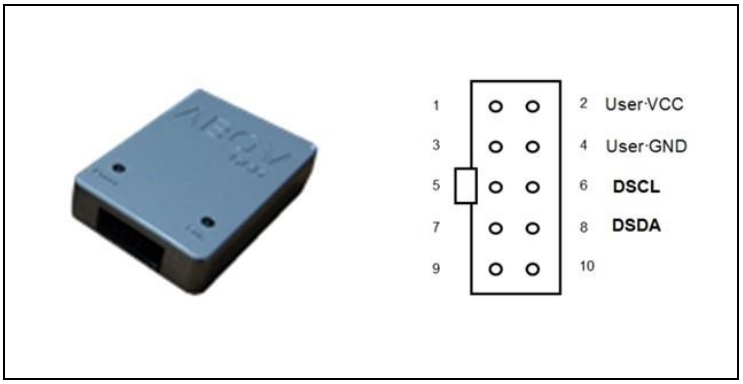


Figure 1-2 On Chip Debugger 1 and Pin Description

1.4.3 Debugger

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

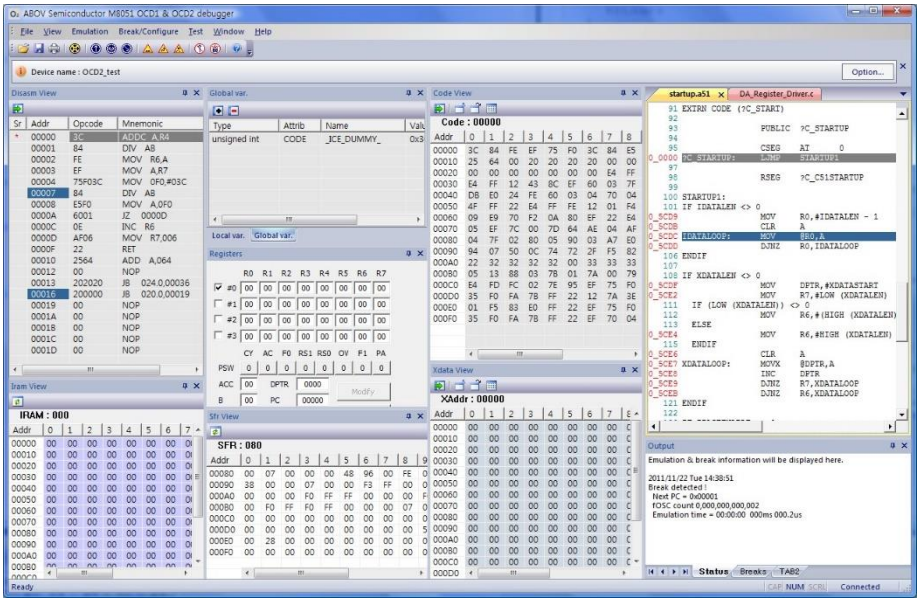


Figure 1-3 OCD Debugger

1.4.4 Programmer

E-PGM +

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

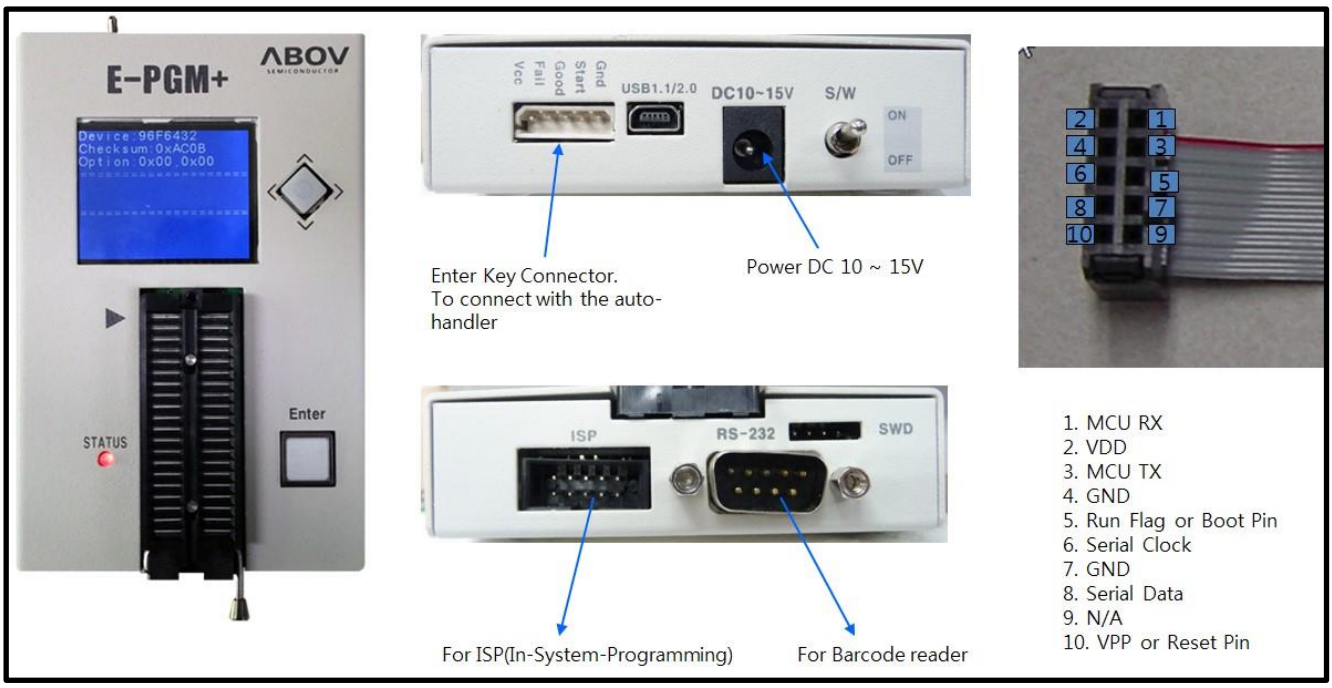


Figure 1-4 PGMplus USB

PGMPlusLC 2

Description

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

Features

- PGMplusLC2 is low cost writing Tool.
- USB interface is supported.
- Not need USB driver installation.
- Connect the external power adaptor (5v@2A).
- Fast 32-bit Cortex-M3 MCU is used.
- Supported high voltage Max 18V.
- PGMplusLC2 is based on PC environment.
- PGMplusLC2 is faster than PGMplusLC.
- Transmission speed is 64Kbyte/s

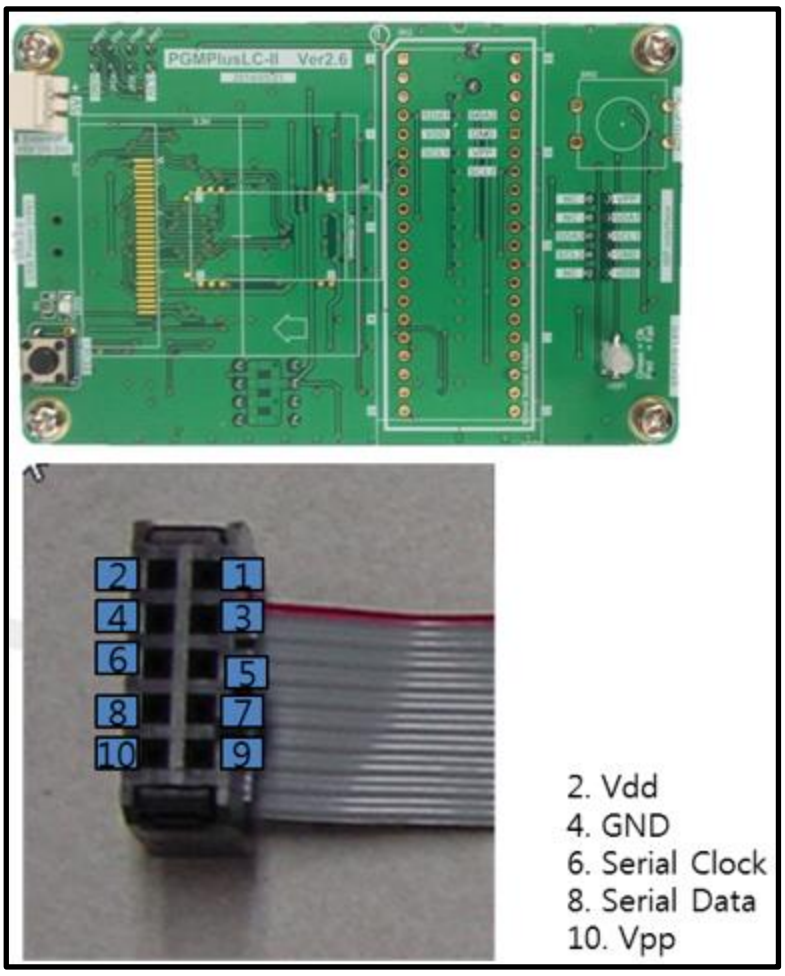


Figure 1-5 PGMplusLC Writer

E-PGM+ Gang4/6

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

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

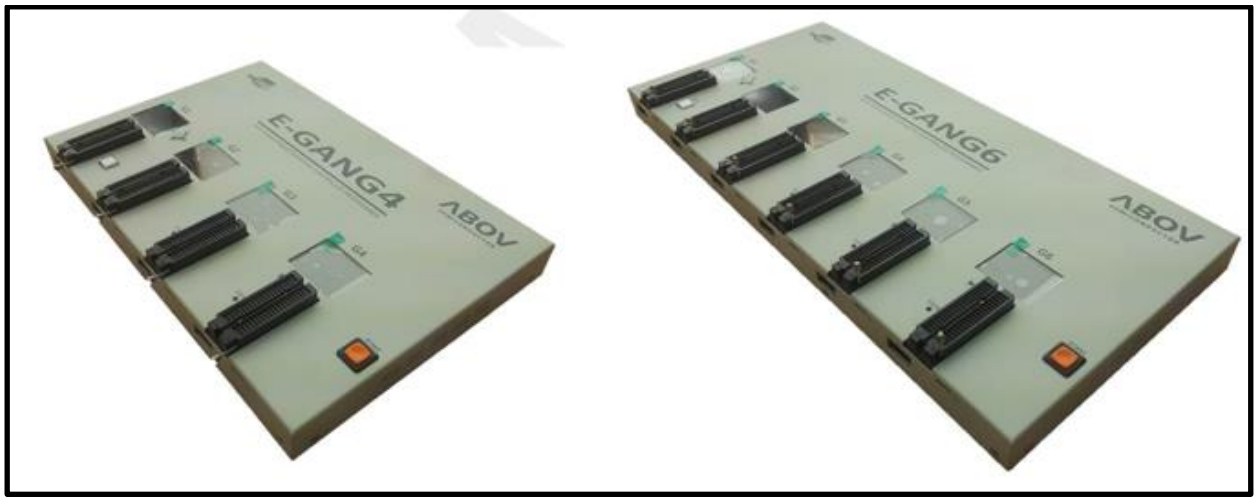


Figure 1-6 Gang Programmer

2. Block Diagram

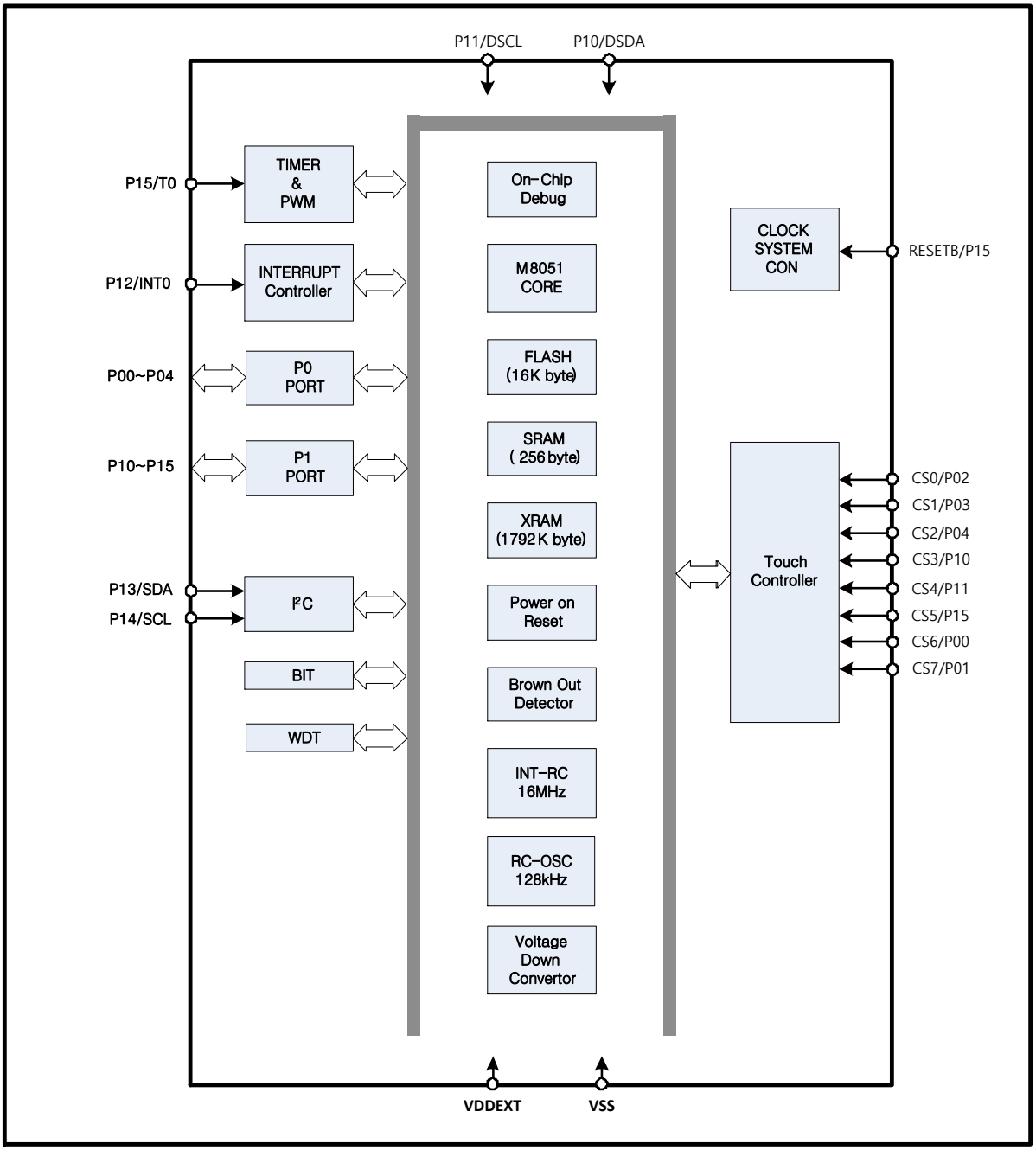


Figure 2-1 A96T346 Block Diagram

3. Pin Assignment

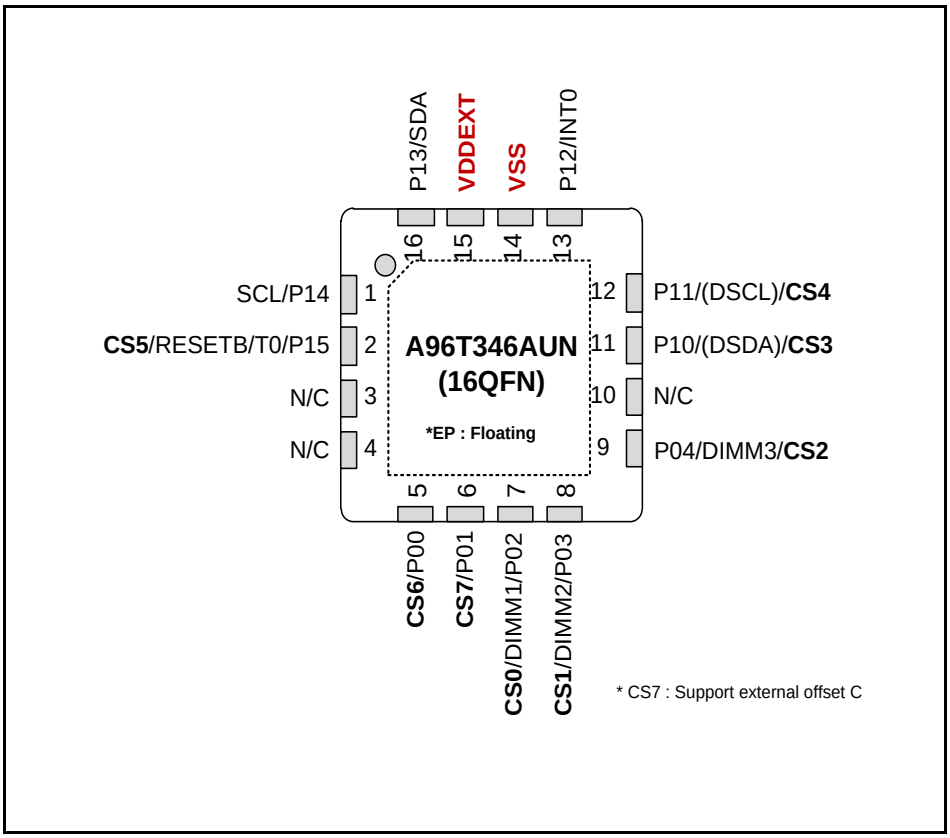


Figure 3-1 16 QFN Pin Assignment

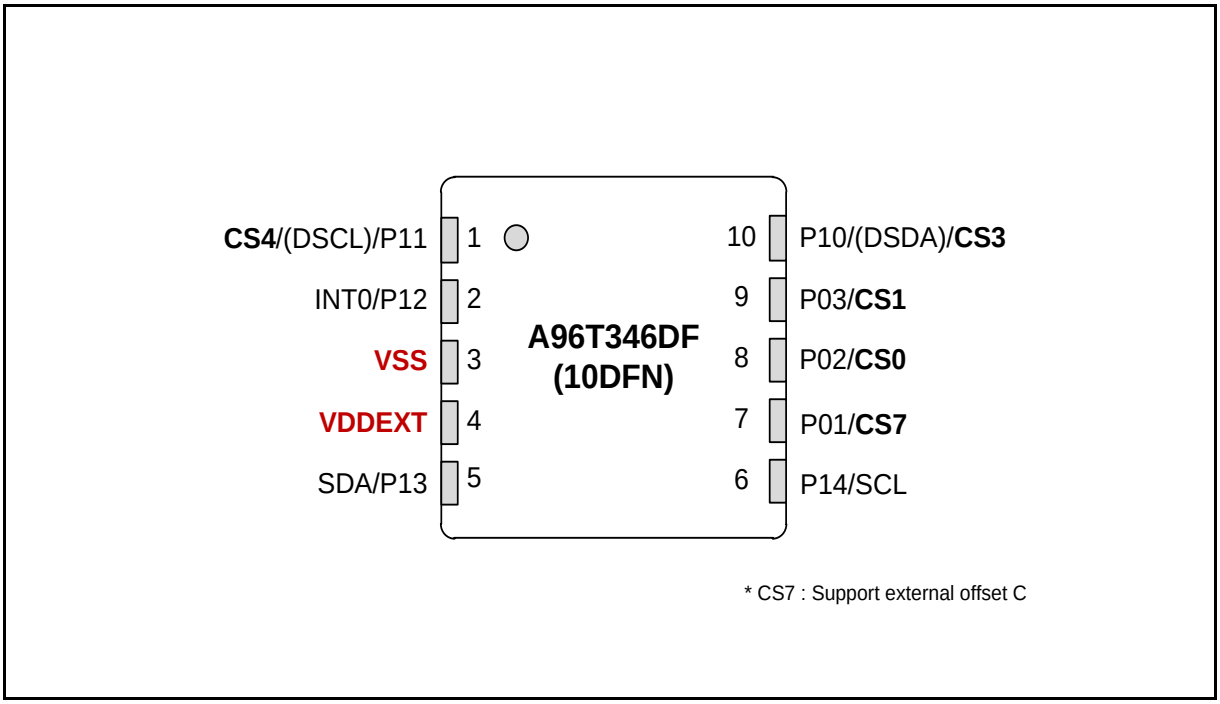
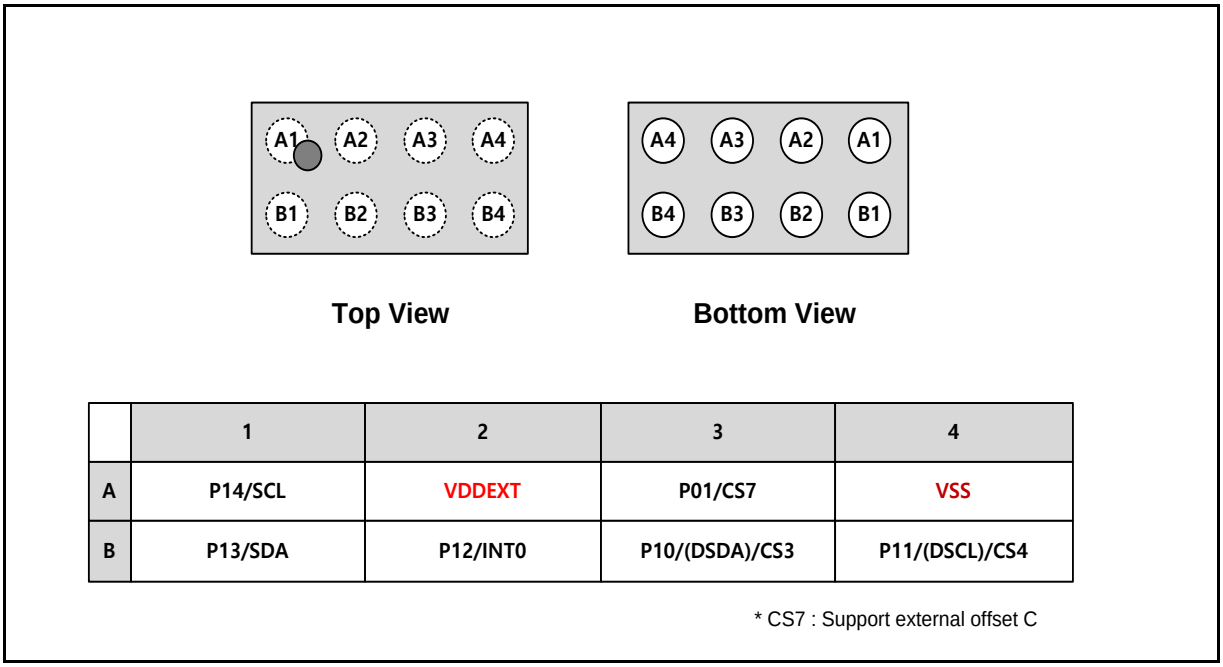


Figure 3-2 10-DFN Pin Assignment



	1	2	3	4
A	P14/SCL	VDDEXT	P01/CS7	VSS
B	P13/SDA	P12/INT0	P10/(DSDA)/CS3	P11/(DSCL)/CS4

* CS7 : Support external offset C

Figure 3-3 8-WLCSP Pin Assignment

4. Package Diagram

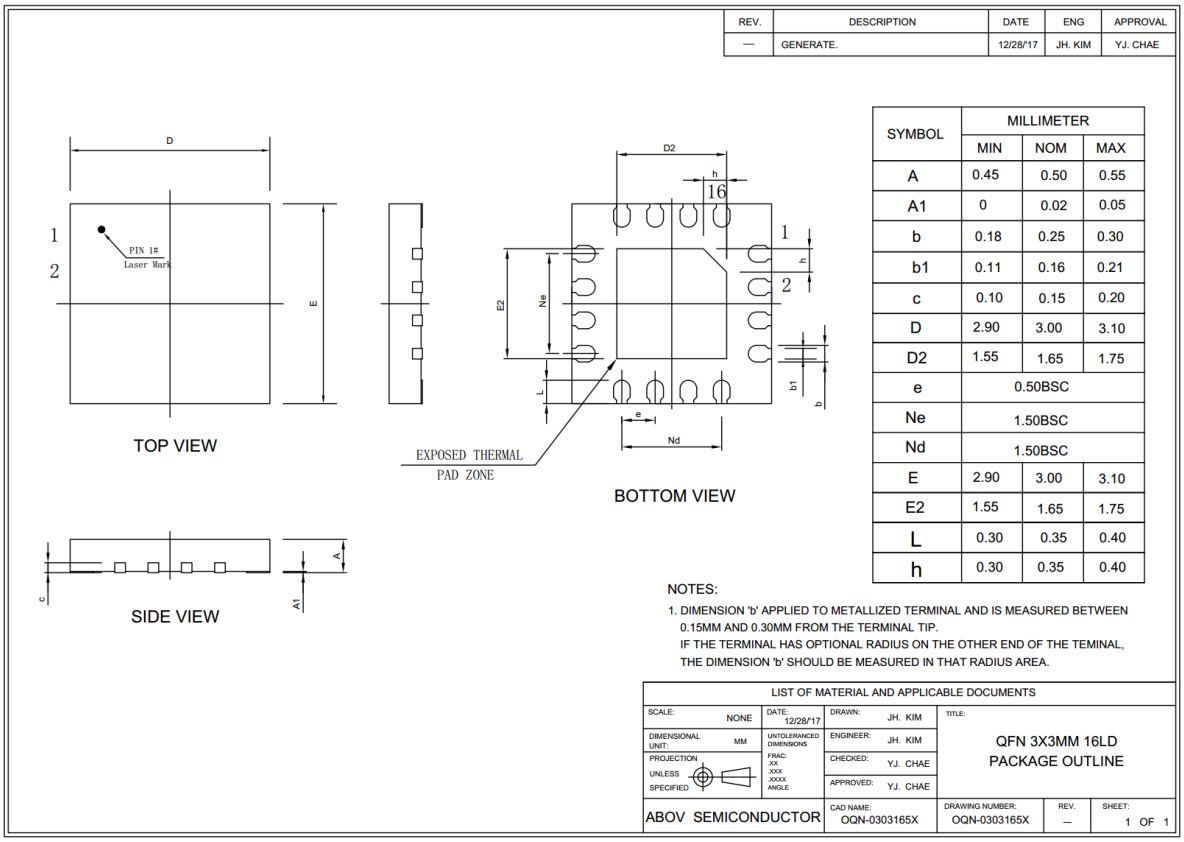
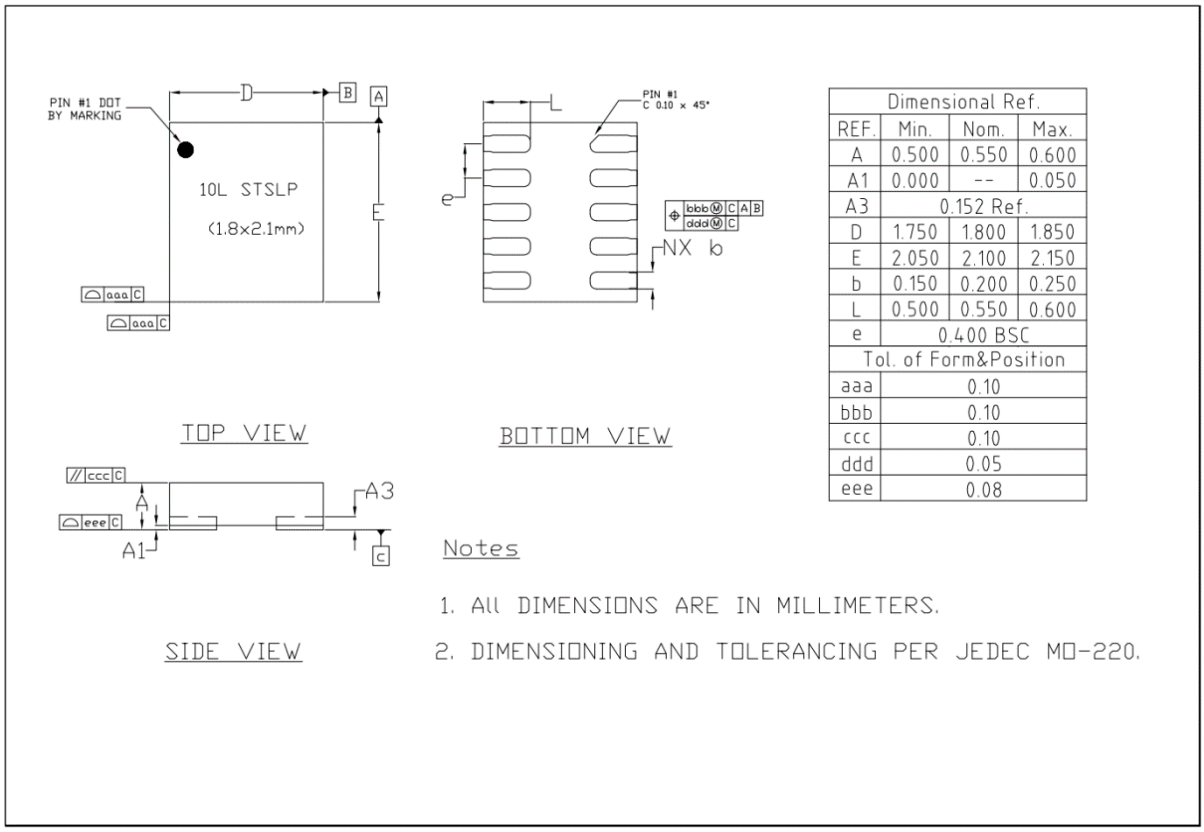
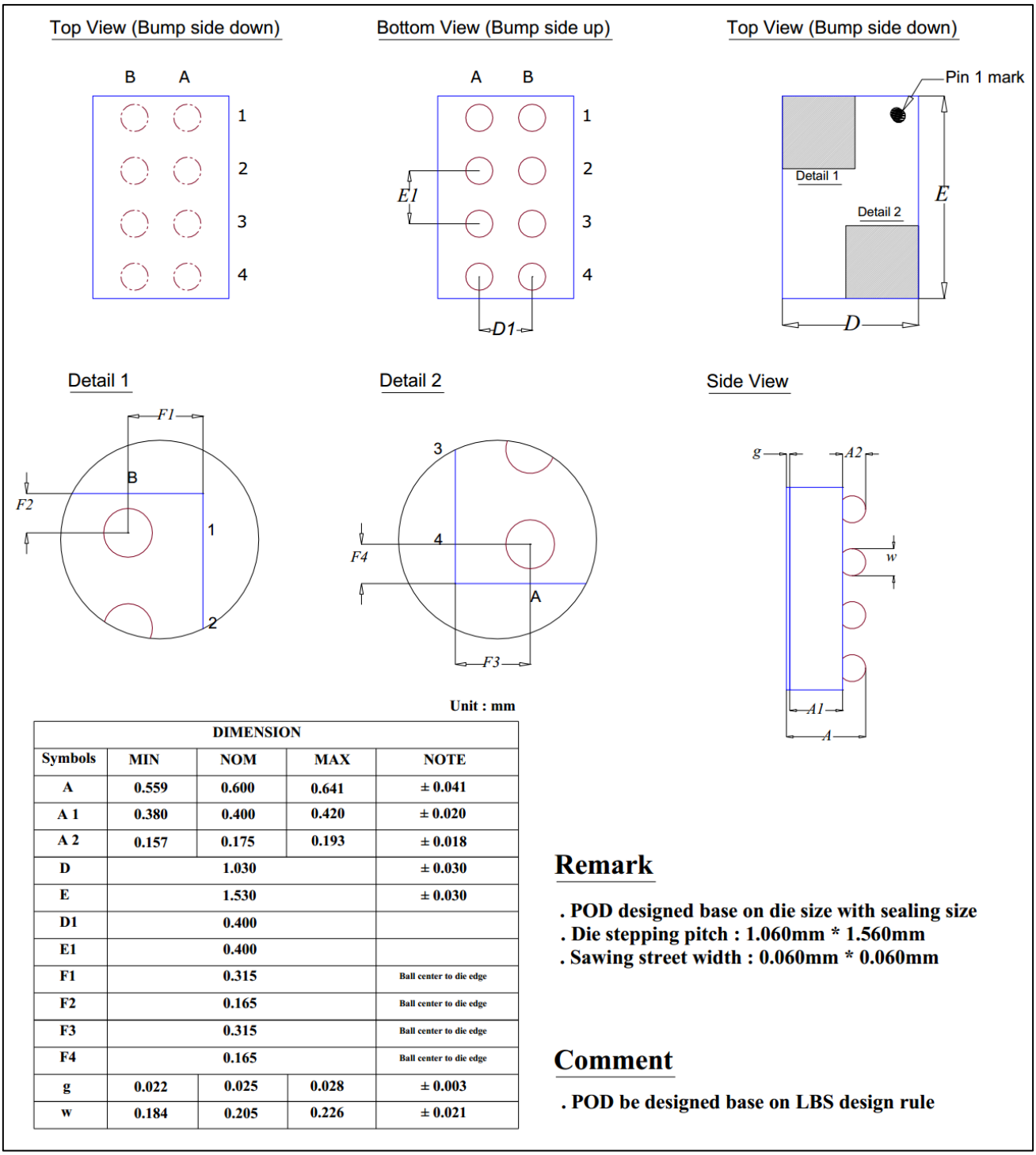


Figure 4-1 16-pin QFN Package



Dimensional Ref.			
REF.	Min.	Nom.	Max.
A	0.500	0.550	0.600
A1	0.000	--	0.050
A3	0.152 Ref.		
D	1.750	1.800	1.850
E	2.050	2.100	2.150
b	0.150	0.200	0.250
L	0.500	0.550	0.600
e	0.400 BSC		
Tol. of Form&Position			
aaa	0.10		
bbb	0.10		
ccc	0.10		
ddd	0.05		
eee	0.08		

Figure 4-2 10-pin DFN Package



Unit : mm

DIMENSION				
Symbols	MIN	NOM	MAX	NOTE
A	0.559	0.600	0.641	± 0.041
A 1	0.380	0.400	0.420	± 0.020
A 2	0.157	0.175	0.193	± 0.018
D		1.030		± 0.030
E		1.530		± 0.030
D1		0.400		
E1		0.400		
F1		0.315		Ball center to die edge
F2		0.165		Ball center to die edge
F3		0.315		Ball center to die edge
F4		0.165		Ball center to die edge
g	0.022	0.025	0.028	± 0.003
w	0.184	0.205	0.226	± 0.021

Remark

. POD designed base on die size with sealing size
. Die stepping pitch : 1.060mm * 1.560mm
. Sawing street width : 0.060mm * 0.060mm

Comment

. POD be designed base on LBS design rule

Figure 4-3 8-Pin WLCSP Package

5. Pin Description

Table 5-1 Normal Pin description

PIN Name		I/O	Function	@RESET	Shared with
10PIN	16PIN				
-	P00	I/O	Port P0 5-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	Input	CS6
P01	P01				CS7
P02	P02				DIMM1 / CS0
P03	P03				DIMM2 / CS1
-	P04				DIMM3/CS2
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	Input	(DSDA) / CS3
P11	P11				(DSCL) / CS4
P12	P12				INT0
P13	P13				SDA
P14	P14				SCL
-	P15				RESETB / T0 / CS5

6. Port Structures

6.1 General Purpose I/O Port

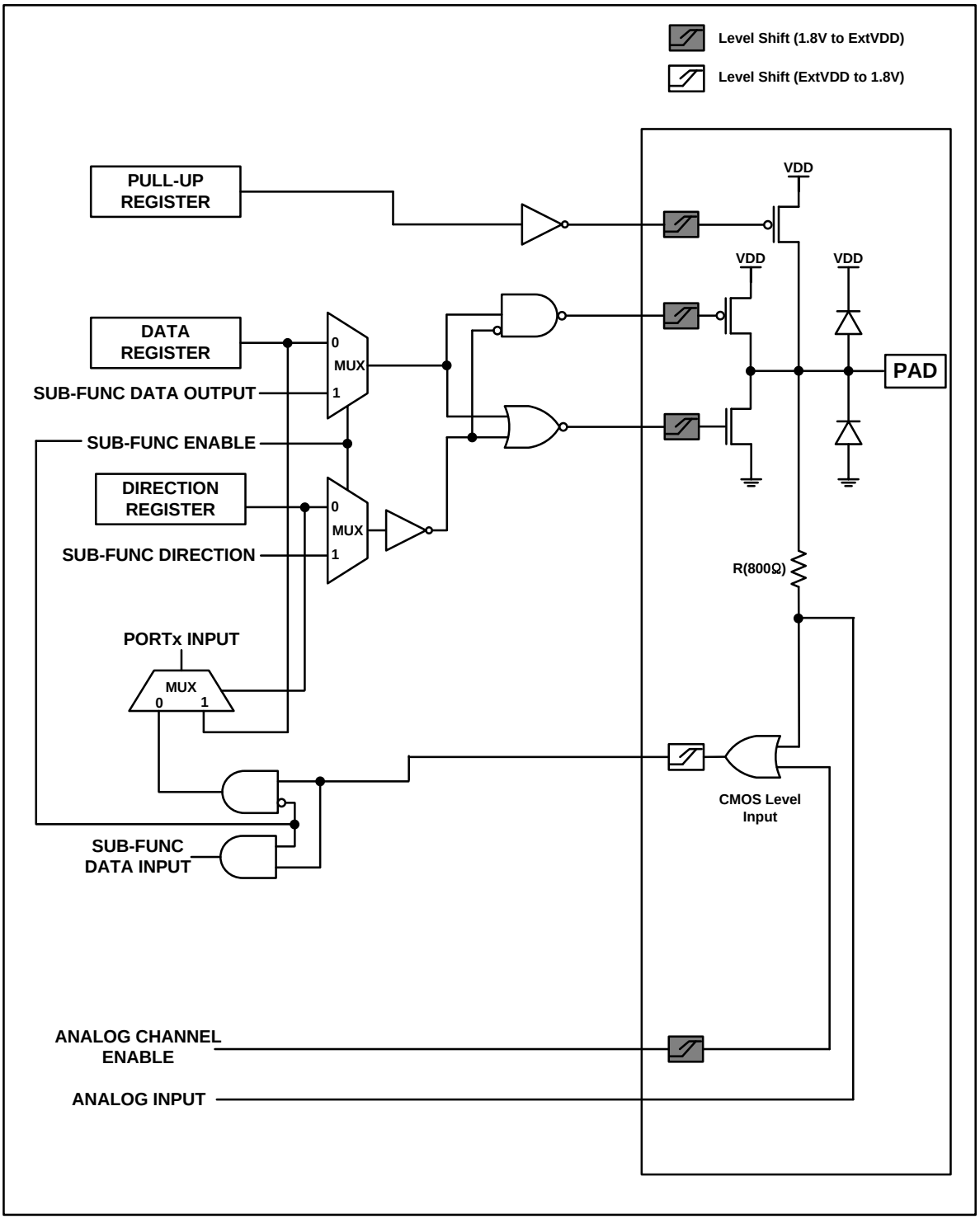


Figure 6-1 General Purpose I/O Port

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

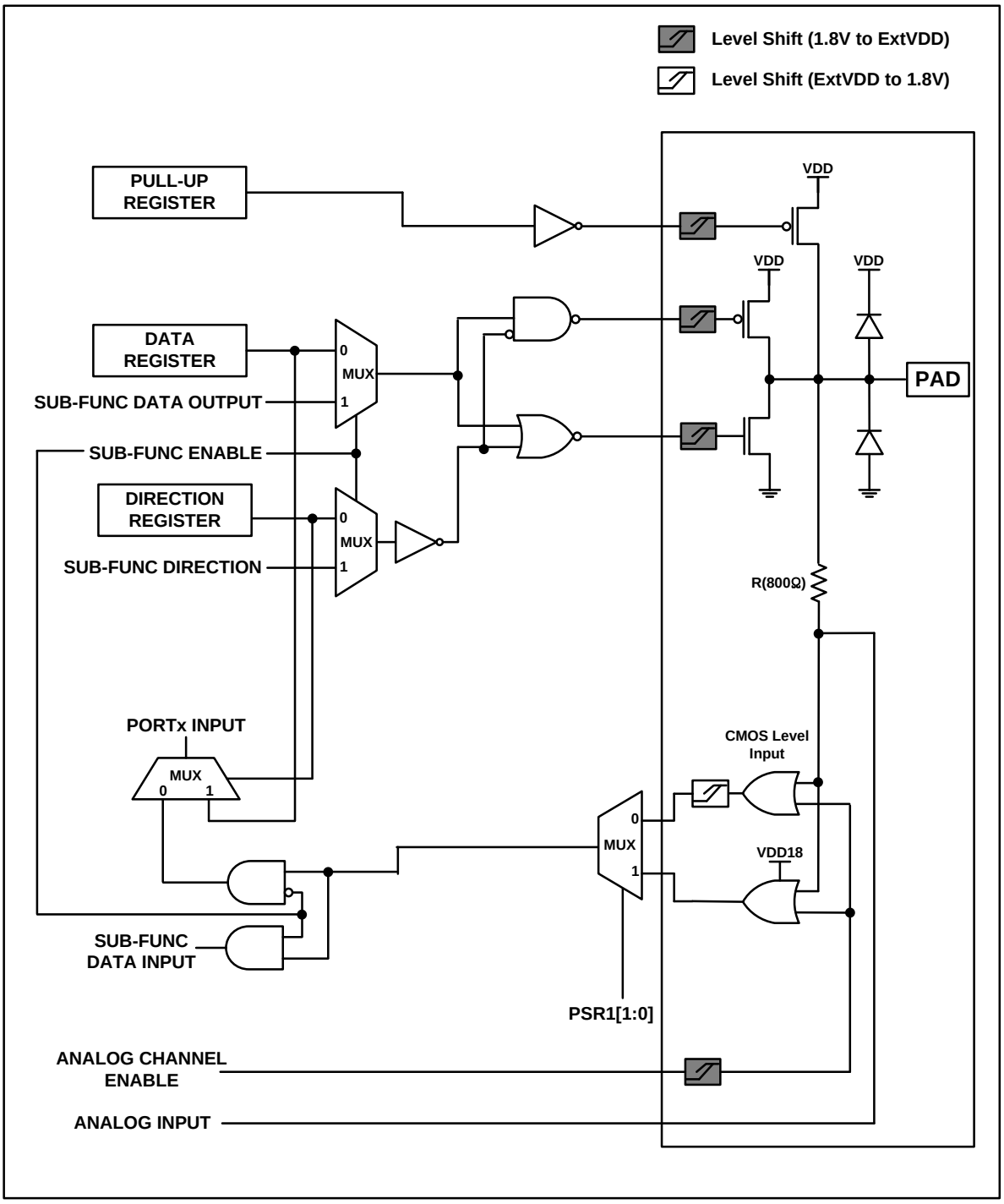


Figure 6-2 General Purpose I/O Port with 1.8V Interface (P10, P11)

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

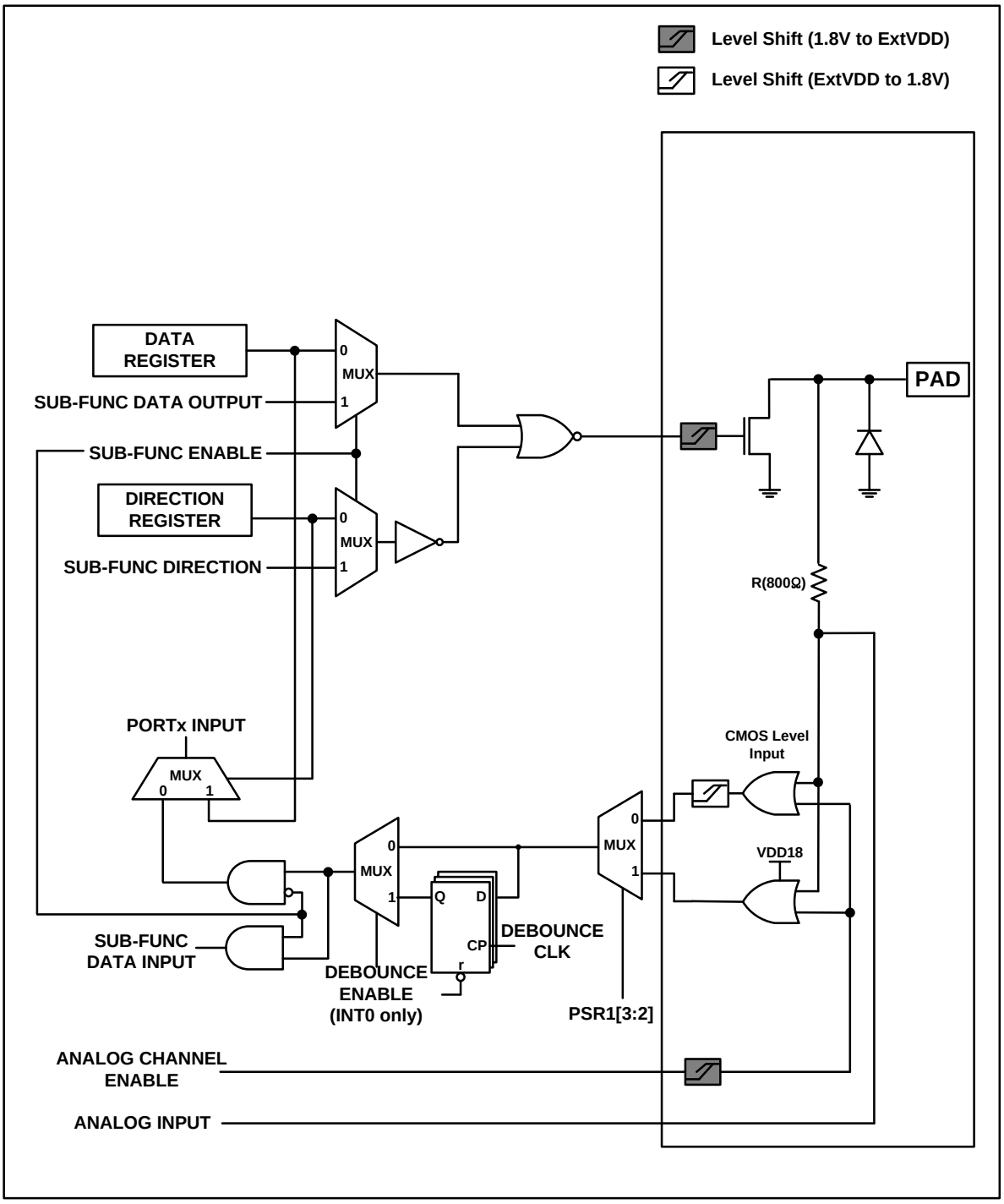


Figure 6-3 Open-Drain I/O Port with 1.8V Interface (INT0/P12, SDA/P13, SCL/P14)

7. Electrical Characteristics

7.1 Absolute Maximum Ratings

Table 7-1 Absolute Maximum Ratings

Parameter	Symbol	Rating	Unit
Supply Voltage	VDD	-0.3~+4.2	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	-8	mA
	∑IOH	-64	mA
	IOL	17	mA
	∑IOL	136	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.

7.2 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	-	3.6	V
Operating Temperature	TOPR	VDD=2.7~3.6V	-40	-	85	℃
Operating Frequency	FIRC	Internal RC-OSC	-	16	-	MHz
	FWDT	Internal WDT Ring-OSC	-	128	-	kHz

7.3 Voltage Dropout Converter Characteristics

Table 7-3 Voltage Dropout Converter Characteristics

Parameter	Symbol	Condition	MIN	TYP	MAX	Unit
Operating Voltage		-	2.7	-	3.6	V
Operating Temperature		-	-40	-	+85	℃
Regulation Voltage		-	-	1.55	-	V
Drop-out Voltage		-	-	0.08	-	V
Current Drivability		RUN/IDLE	-	-	10	mA
		SUB-ACTIVE	-	-	0.2	mA
		STOP2	-	-	10	uA
Operating Current	IDD1	RUN/IDLE	-	0.14	0.3	mA
	IDD2	SUB-ACTIVE	-	21	56	uA
	SIDD2	STOP2	-	0.35	1	uA
Drivability Transition Time	TRAN1	SUB to RUN	-	1	-	us
	TRAN2	STOP to RUN	-	100	-	us

7.4 Low Drop Out Characteristics

Table 7-4 Voltage Dropout Converter Characteristics

Parameter	Symbol	Condition	MIN	TYP	MAX	Unit
Operating Voltage		-	2.7	-	3.6	V
Operating Temperature		-	-40	-	+85	℃
Regulation Voltage		-	-	3.0	-	V
		-	-	2.5	-	V
Drop-out Voltage		-	-	50	-	mV
Current Drivability		RUN	-	-	5	mA
Operating Current	IDD	RUN	-	-	180	uA
	SIDD	STOP	-	-	0.1	uA

7.5 Power-On Reset Characteristics

Table 7-5 Power-On Reset Characteristics

Parameter	**Symbol**	**Condition**	**MIN**	**TYP**	**MAX**	**Unit**
Operating Voltage		-	VSS	-	3.6	V
Operating Temperature		-	-40	-	+85	℃
RESET Release Level	Vpor	-	1.0	1.2	1.4	V
Operating Current	IDD	-	-	0.1	-	uA
VDD Rise Rate	VRR		0.05		50	V/ms

7.6 Brown Out Detector

Table 7-6 Brown Out Detector Characteristics

Parameter	Symbol	Condition	MIN	TYP	MAX	Unit
Operating Voltage		-	VSS	-	3.6	V
Operating Temperature		-	-40	-	+85	℃
Detection Level	Vbod	BOD	1.38	1.6	1.77	V
Hysteresis		-	-	100	-	mV
Operating Current	IDD	-	-	2	-	uA

7.7 Internal RC Oscillator Characteristics

Table 7-7 Internal RC Oscillator Characteristics

Parameter	Symbol	Condition	MIN	TYP	MAX	Unit
Operating Voltage		-	2.7	-	3.6	V
Operating Temperature		-	-40	25	+85	℃
Frequency		-	15.2	16	16.8	MHz
Stabilization Time		-	-	1	-	ms
Operating Current	IDD	-	-	180	230	uA

7.8 Ring-Oscillator Characteristics

Table 7-8 Ring-Oscillator Characteristics

Parameter	Symbol	Condition	MIN	TYP	MAX	Unit
Operating Voltage		-	2.7	-	3.6	V
Operating Temperature		-	-40	-	+85	℃
Frequency		25℃	121.6	128	134.4	kHz
		-40~85℃	102.4	128	153.6	kHz
Stabilization Time		-	-	1	-	ms
Operating Current	IDD	-	-	1	2	uA

7.9 Touch Sensing Characteristics

Table 7-9 Touch Sensing Characteristics

Parameter	Symbol	Condition		MIN	TYP	MAX	Unit
Measurement Range	CRANGEL	PGA = x1 (SCI = 001, SCC = 111)	Digial Gain = x16	-	±1	-	pF
	CRANGEH		Digital Gain = x8	-	±2	-	pF
Measurement Resolution	NBIT	-		-	16	-	bits
	CRESL	PGA = x8 (SCI = 111, SCC = 011)	Digital Gain = x16	-	0.0038	-	fF
	CRESH		Digital Gain = x8	-	0.0076	-	fF
Sampling Frequencies	FS	Programmable (@Sensing Frequency = 16MHz)		-	-	727	kHz
Scan Periods	TSCAN	Programmable with F/W		-	400	-	ms
Sampling & Scan Frequencies Accuracy	FTEMP	Full TA Range, VDD = 2.7V to 3.6V		-	-	±5	%
		TA = 25℃, VDD = 2.7V to 3.6V		-	-	±2	%
External Shunt Capacitor to GND	CDC	-		-	-	200	pF

7.10 A/D Converter Characteristics

Table 7-10 A/D Converter Characteristics

Parameter	Symbol	Pin/Condition	MIN	TYP	MAX	Unit
AV_{DD} Input Voltage	AV_{DD}	AVDD	2.7	-	3.6	V
Analog Input Voltage Range	V_{AIN}	AN0~AN7	VSS	-	AVDD	V
Resolution	N_R	-	-	-	12	Bit
Analog Input Capacitance	C_{AIN}	AN0~AN7	-	-	30	pF
Integral Non Linearity Error	N_{INL}	ADC reference voltage = 2.7~3.6V, F_{XIN}=4MHz	-	±8	-	LSB
Differential Non linearity Error	N_{DNL}		-	±2	-	
Zero Offset Error	N_{ZOE}		-	±4	-	
Full Scale Error	N_{TOE}		-	±28	-	
Conversion Time	T_{CONV}	F_{XIN}=4MHz	-	15	-	us
Operation Current	I_{AVDD}	F_{XIN}=4MHz / AVDD=3.0V	-	550	-	uA

Note) 1. When ADC reference voltage used LDO, the ADC resolution may be worse.

7.11 DC Characteristics

Table 7-11 DC Characteristics

(VDD =2.7~3.6V, 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	P14~P10 (1.8V interface case)	-0.5	-	0.2*1.8V	
Input High Voltage	VIH0	ALL I/O	0.7*VDD	-	VDD	V
	VIH1	P14~P10 (1.8V interface case)	0.7*1.8V	-	VDD	
Output Low Voltage	VOL0	ALL I/O (IOL=11.0mA, VDD=3.3V)	VSS	-	0.3*VDD	V
	VOL1	Dimming PAD (IOL=11.0mA, VDD=3.3V, @default setting)	VSS	-	0.3*VDD	V
Output High Voltage	VOH0	ALL I/O (IOH=-5.7mA, VDD=3.3V)	0.7*VDD	-	VDD	V
Input High Leakage Current	IIH	ALL PAD	-1	-	1	uA
Input Low Leakage Current	IIL	ALL PAD	-1	-	1	uA
Pull-Up Resister	RPU	ALL PAD @3.3V	30	-	84	kΩ
Power Supply Current	IDD1	Run Mode, 16MHz @3.3V	0.7	-	4	mA
	IDD2	Sleep Mode, 16MHz @3.3V	0.5	-	3	mA
	IDD3	STOP1 Mode, WDT Active @3.3V (BOD disable)	-	-	22	uA
	IDD4	STOP2 Mode, WDT Disable @3.3V (BOD disable)	-	-	7	uA

Note) 1. P14~P10 support 1.8V interface by PSR1.

2. Dimming PAD (P04, P03, P02), Open-Drain PAD (P14, P13, P12).

3. STOP1: WDT only running, STOP2: All function disable.

7.12 AC Characteristics

Table 7-12 AC Characteristics

(VDD=3.3V±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	-	-	-	1	ms
Debounce Clock Cycle Time	tRING	-	6.25	-	9.375	us
External Interrupt Input Width	tIW	INT0	8	-	-	tRING
External Interrupt Transition Time	tFI,tRI	INT0	-	-	20	ns
nRESET Input Pulse “L” Width	tRST	nRESET	8	-	-	tRING

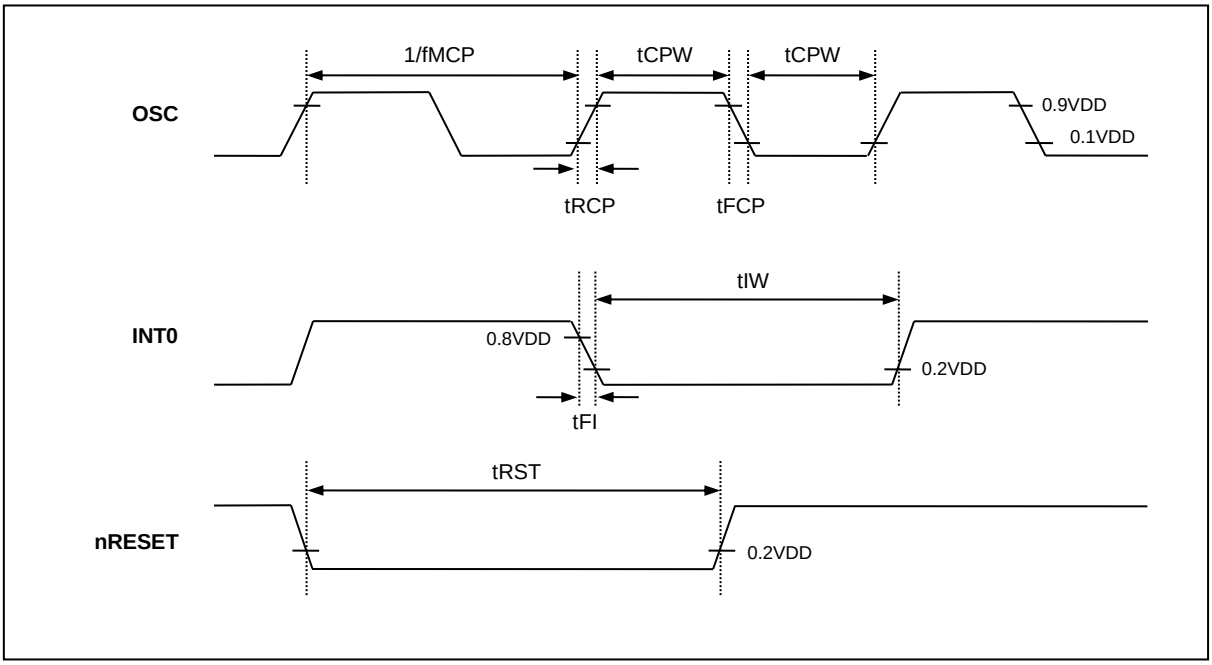


Figure 7-1 AC Timing

7.13 I²C Characteristics

The following table and figure show the timing condition of SDA and SCL bus lines for I²C bus devices.

Parameter	Symbol	STANDARD MODE		FAST MODE		Unit
		Min	Max	Min	Max	
SCL clock frequency	f_{SCL}	0	100	0	400	kHz
Hold time after (repeated) START condition. After this period, the first clock pulse is generated	$t_{HD;STA}$	4.0	-	0.6	-	us
LOW period of the SCL clock	t_{LOW}	4.7	-	1.3	-	us
HIGH period of the SCL clock	t_{HIGH}	4.0	-	0.6	-	us
Setup time for a repeated START condition	$t_{SU;STA}$	4.7	-	0.6	-	us
Data hold time	$t_{HD;DAT}$	0	3.45	0	0.9	us
Data setup time	$t_{SU;DAT}$	100	-	100	-	ns
Clock/data fall time	t_F	0	300	0	300	ns
Clock/data rise time	t_R	0	1000	0	300	ns
Setup time for STOP condition	$t_{SU;STO}$	4.0	-	0.6	-	us
Bus free time between a STOP and START condition	t_{BUF}	4.7	-	1.3	-	us

Table 7-13 Timing characteristics of I²C

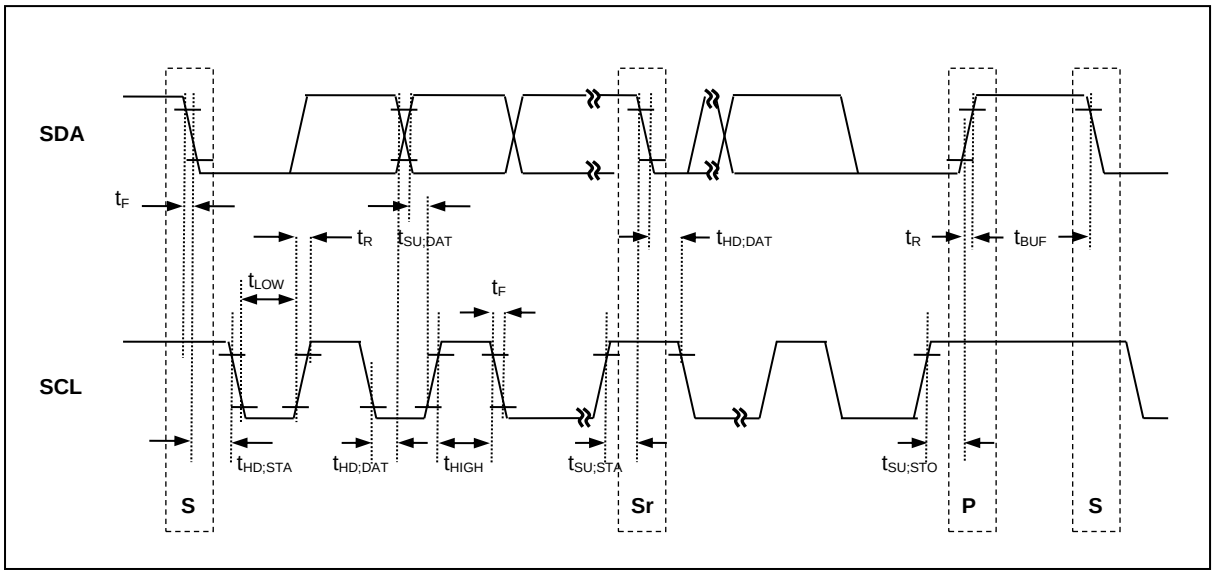


Figure 7-2 Timing diagram of I²C

7.14 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 A96T346 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 A96T346 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.

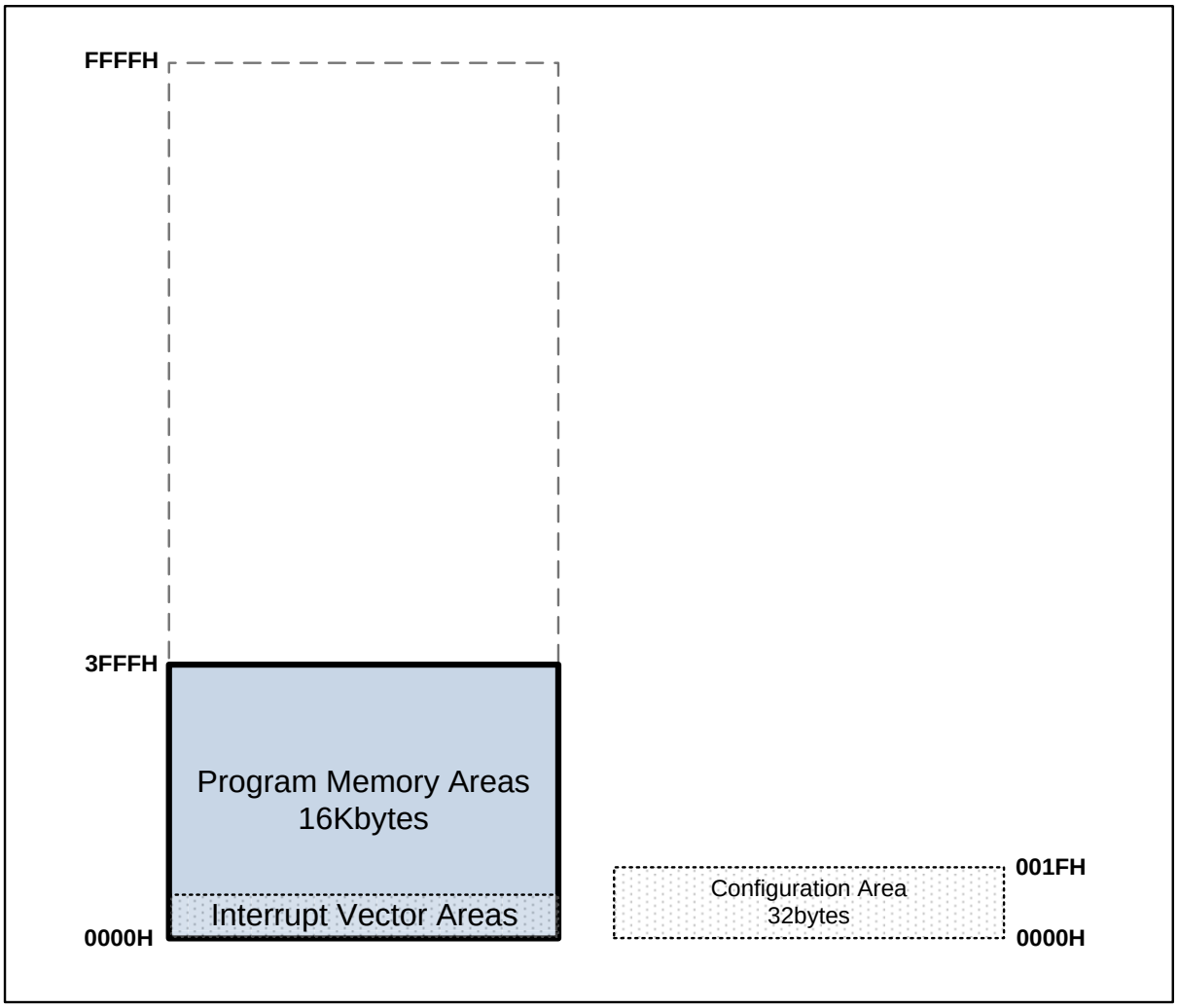


Figure 8-1 Program Memory

- 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.

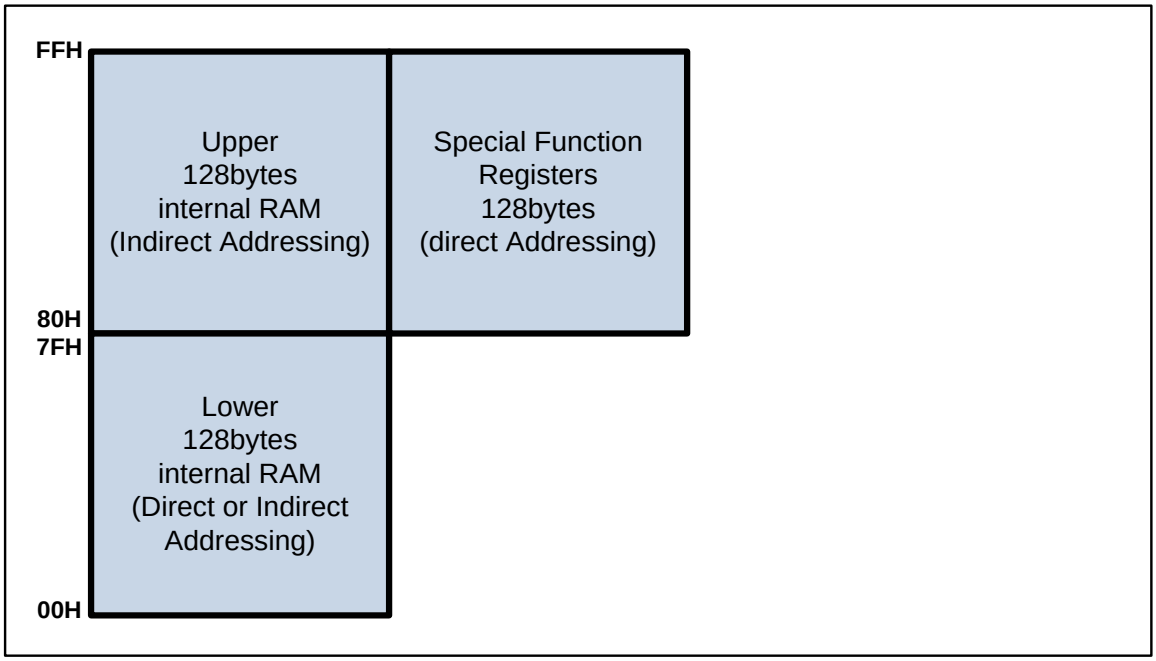


Figure 8-2 Data Memory Map

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.

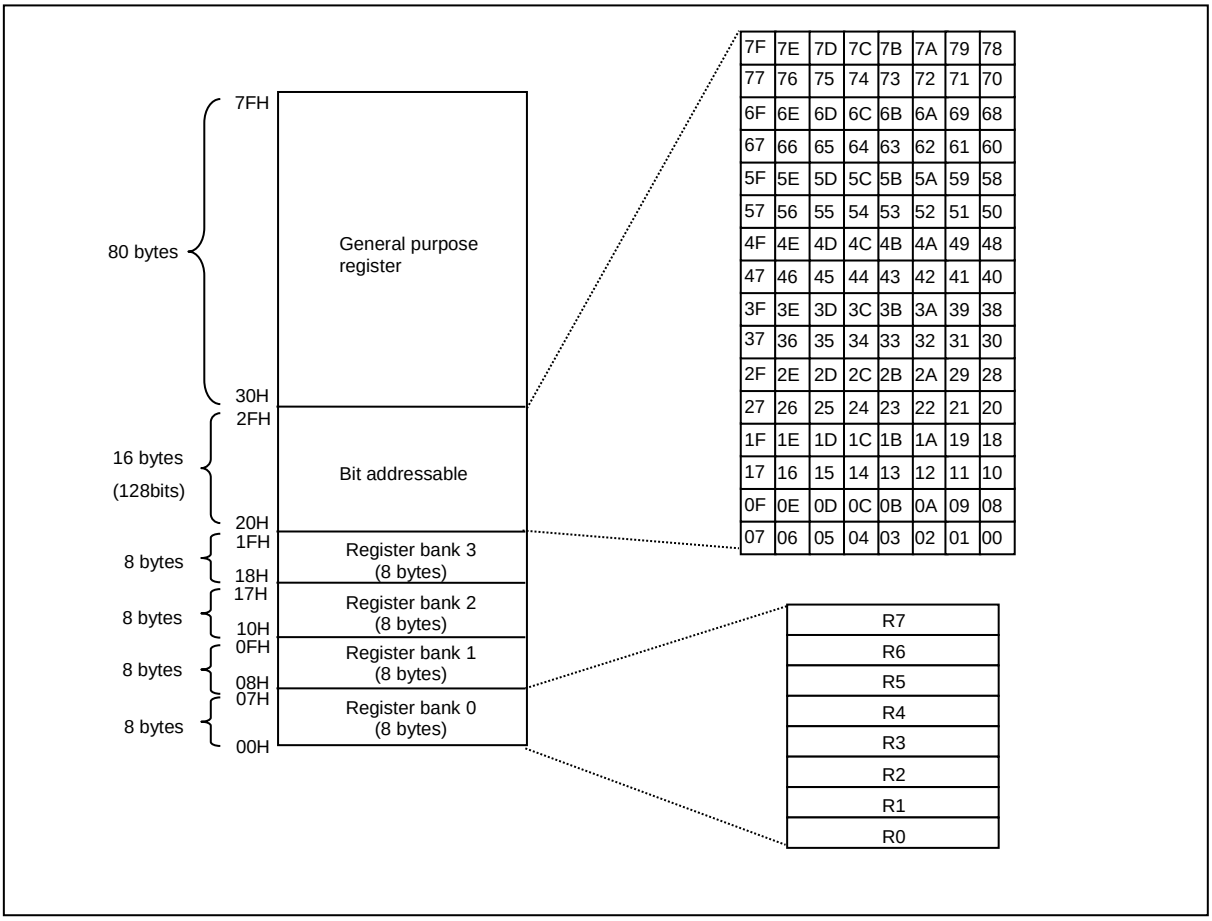


Figure 8-3 Lower 128 bytes RAM

8.3 XSFR

A96T346 has 1792BytesXSRAM. This area has no relation with RAM/FLASH. It can read and write through SFR with 8-bit unit.

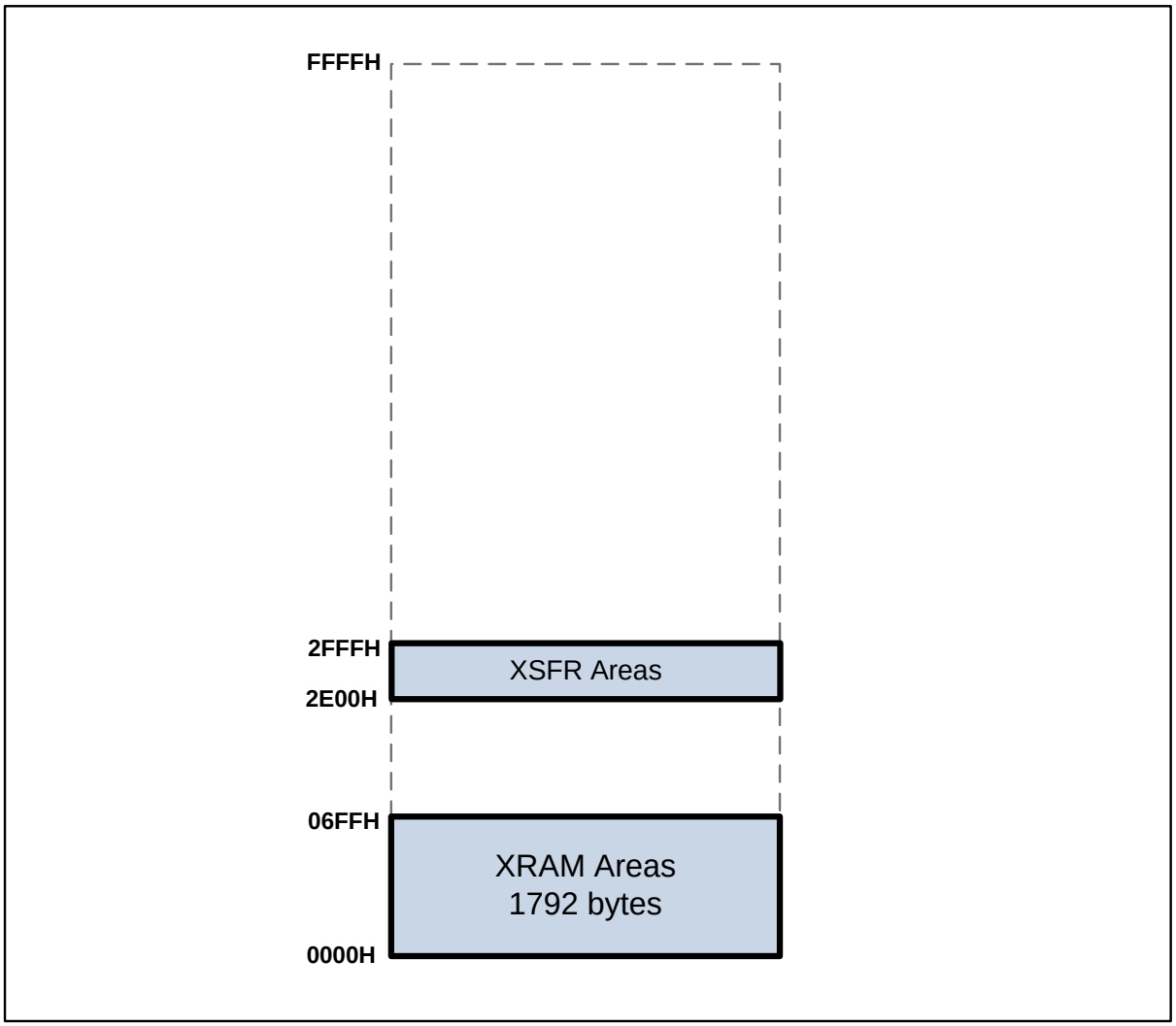


Figure 8-4 XDATA Memory Area

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
2F58H	-	-	-	-	-	-	TEST_B	TEST_A
2F50	FUSE_CONF	FUSE_CAL0	FUSE_CAL1	FUSE_RING	FUSE_BGR3	FUSE_FLAG	FUSE_TOUC	FUSE_PKGx
2E30	TAR	TRST	TDRV	TINT	TD			
2E28	TS_FREQ_NUM	TS_FREQ_DEL	TS_CLK_CFG	TRIM_OSC	TRIM_ADC_OSC	SCI	SCC	SVREF
2E20	TS_CON	TS_MODE	TS_SUM_CNT	TS_CH_SEL	TS_SLP_CON	ADC_CH_SEL_H	ADC_CH_SEL_L	TS_INTEG_CNT
2E18	SCO4_H	SCO4_L	SCO5_H	SCO5_L	SCO6_H	SCO6_L	SCO7_H	SCO7_L
2E10	SCO0_H	SCO0_L	SCO1_H	SCO1_L	SCO2_H	SCO2_L	SCO3_H	SCO3_L
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	-	-	-	-	-	-	-
F0H	B	-	FEARL	FEARM	FEARH	-	-	-
E8H	-	-	FEMR	FECR	FESR	FETCR	-	-
E0H	ACC	-	-	-	-	-	-	-
D8H	-	-	I2CMR	I2CSR	I2CSCLLR	I2CSCLHR	I2CSDAHR	I2CDR
D0H	PSW	-	DIMM1	DIMM2	DIMM3	-	I2CSAR1	I2CSAR
C8H	-	-	-	-	-	-	-	-
C0H	-	-	-	-	-	-	-	-
B8H	IP	-	-	-	-	-	-	-
B0H	-	-	T0CR	-	PWM0DRL CDR0L / T0L	PWM0DRH CDR0H / T0H	PWM0PRL T0DRL	PWM0PRH T0DRH
A8H	IE	IE1	-	PSR0	PSR1	-	-	-
A0H	-	-	EO	EIENAB	EIFLAG	-	-	-
98H	-	-	-	-	-	-	-	-
90H	-	P1IO	P1PU	-	-	P0PU	-	-
88H	P1	P0IO	SCCR	BCCR	BITR	WDTMR	WDTR /WDTCR	LDOCR
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 A96T346 has 11 I/O ports (P0 ~ P1). Each port can be easily configured by software as I/O pin, internal pull up pin to meet various system configurations and design requirements.

9.2 Port Register

9.2.1 Data Register (P0~P1)

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~P1IO)

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~P1PU)

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.

9.2.3.1 Port Selection Register 0 (PSR0)

PSR0 registers prevent the input leakage current when ports are connected to analog inputs. If the bit of PSR0 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[0] is ‘0’, Enable P0[0] function (Digital input mode)
The bit of PSR0[0] is ’1’, Disable P0[0] function (Digital input of the corresponding port is always “1”)

The bit of PSR0[1] is ‘0’, Enable P0[1] function (Digital input mode)
The bit of PSR0[1] is ’1’, Disable P0[1] function (Digital input of the corresponding port is always “1”)

The bit of PSR0[2] is ‘0’, Enable P0[2] function (Digital input mode)
The bit of PSR0[2] is ’1’, Disable P0[2] function (Digital input of the corresponding port is always “1”)

The bit of PSR0[3] is ‘0’, Enable P0[3] function (Digital input mode)
The bit of PSR0[3] is ’1’, Disable P0[3] function (Digital input of the corresponding port is always “1”)

The bit of PSR0[4] is ‘0’, Enable P0[4] function (Digital input mode)
The bit of PSR0[4] is ’1’, Disable P0[4] function (Digital input of the corresponding port is always “1”)

The bit of PSR0[5] is ‘0’, Enable P1[0] function (Digital input mode)
The bit of PSR0[5] is ’1’, Disable P1[0] function (Digital input of the corresponding port is always “1”)

The bit of PSR0[6] is ‘0’, Enable P1[1] function (Digital input mode)
The bit of PSR0[6] is ’1’, Disable P1[1] function (Digital input of the corresponding port is always “1”)

The bit of PSR0[7] is ‘0’, Enable P1[5] function (Digital input mode)
The bit of PSR0[7] is ’1’, Disable P1[5] function (Digital input of the corresponding port is always “1”)

9.2.3.2 Port Selection Register 1 (PSR1)

The bit of PSR1[0] is ‘0’, VDD/VSS input level use P1[0].
The bit of PSR1[0] is ‘1’, 1.8V/VSS input level use P1[0].

The bit of PSR1[1] is ‘0’, VDD/VSS input level use P1[1].
The bit of PSR1[1] is ‘1’, 1.8V/VSS input level use P1[1].

The bit of PSR1[2] is ‘0’, VDD/VSS input level use P1[2].
The bit of PSR1[2] is ‘1’, 1.8V/VSS input level use P1[2].

The bit of PSR1[3] is ‘0’, VDD/VSS input level use P1[4:3].
The bit of PSR1[3] is ‘1’, 1.8V/VSS input level use P1[4:3].

The bit of PSR1[4] is ‘0’, I2C is SCL (P1[4]) / SDA (P1[3]).
The bit of PSR1[4] is ‘1’, I2C is SCL (P1[3]) / SDA (P1[4]).

9.2.3.3 Port Dimming Register (DIMM1~3)

The DIMM1 register select N-TR current P0[2].
The DIMM2 register select N-TR current P0[3].
The DIMM3 register select N-TR current P0[4].

9.2.4 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
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
PSR0	ABH	R/W	00H	P0 Port Selection Register
PSR1	ACH	R/W	00H	P1 Port Selection Register
DIMM1	D2H	R/W	11H	Dimming Value Select Register
DIMM2	D3H	R/W	11H	Dimming Value Select Register
DIMM3	D4H	R/W	11H	Dimming Value Select Register

9.3 P0, P1 Port

9.3.1 Px Port Description

P0 is 5-bit I/O port and P1 is 6bit I/O port. Px control registers consist of Data register (Px), direction register (PxIO), pull-up register selection register (PxPU).

9.3.2 Register Description for Px

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

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

Initial value : 00H

P0[4:0] I/O Data

P1[5:0] I/O Data

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

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

Initial value : 00H

P0IO[4:0] P0 data I/O direction.
- 0 Input
- 1 Output

P1IO[5:0] P1 data I/O direction.
- 0 Input
- 1 Output

P0PU, P1PU (Px Pull-up Resistor Selection Register) : 95H, 92H

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

Initial value : 00H

P0PU[4:0] Configure pull-up resistor of P0 port
- 0 Disable
- 1 Enable

P1PU[5:0] Configure pull-up resistor of P1 port
- 0 Disable
  (P1[4:2] : Only Pull-up disable)
- 1 Enable

PSR0 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 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

PSR1 (P1 Port Selection Register) : ACH

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

Initial value : 00H

PSR14 I2C SDA, SCL Change register
0 SCL : P14, SDA : P13 (default)
1 SCL : P13, SDA : P14

PSR13 VIH Level Selector for P14, P13 Input
0 VDD Level (default)
1 1.8V Level

PSR12 VIH Level Selector for P12 Input
0 VDD Level (default)
1 1.8V Level

PSR11 VIH Level Selector for P11 Input
0 VDD Level (default)
1 1.8V Level

PSR10 VIH Level Selector for P10 Input
0 VDD Level (default)
1 1.8V Level

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 : 11H

DIMM1[4:0] Select N-TR Total Current value of Dimming port P02.

DIMM14	DIMM13	DIMM12	DIMM11	DIMM10	
0	0	0	0	0	N/A
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 : 11H

DIMM2[4:0] Select N-TR Total Current value of Dimming port P03.

DIMM24	DIMM23	DIMM22	DIMM21	DIMM20	
0	0	0	0	0	N/A
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 : 11H

DIMM3[4:0] Select N-TR Total Current value of Dimming port P04.

DIMM34	DIMM33	DIMM32	DIMM31	DIMM30	
0	0	0	0	0	N/A
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} = 19.6mA (typical VOL@0.42V, VDD@3.3V), LED drive current table.

	Control	Current (mA)
1	00001	0.65
2	00010	1.28
3	00011	1.93
4	00100	2.52
5	00101	3.17
6	00110	3.80
7	00111	4.45
8	01000	5.05
9	01001	5.70
10	01010	6.33
11	01011	6.98
12	01100	7.57
13	01101	8.22
14	01110	8.85

15	01111	9.50
16	10000	10.10
17	10001	10.75 (default)
18	10010	11.38
19	10011	12.03
20	10100	12.62
21	10101	13.27
22	10110	13.90
23	10111	14.55
24	11000	15.15
25	11001	15.80
26	11010	16.43
27	11011	17.08
28	11100	17.67
29	11101	18.32
30	11110	18.95
31	11111	19.60

10. Interrupt Controller

10.1 Overview

The A96T346 supports up to 7 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 7 interrupt source
- 2 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). Bits of IE, IE1 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 A96T346 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.

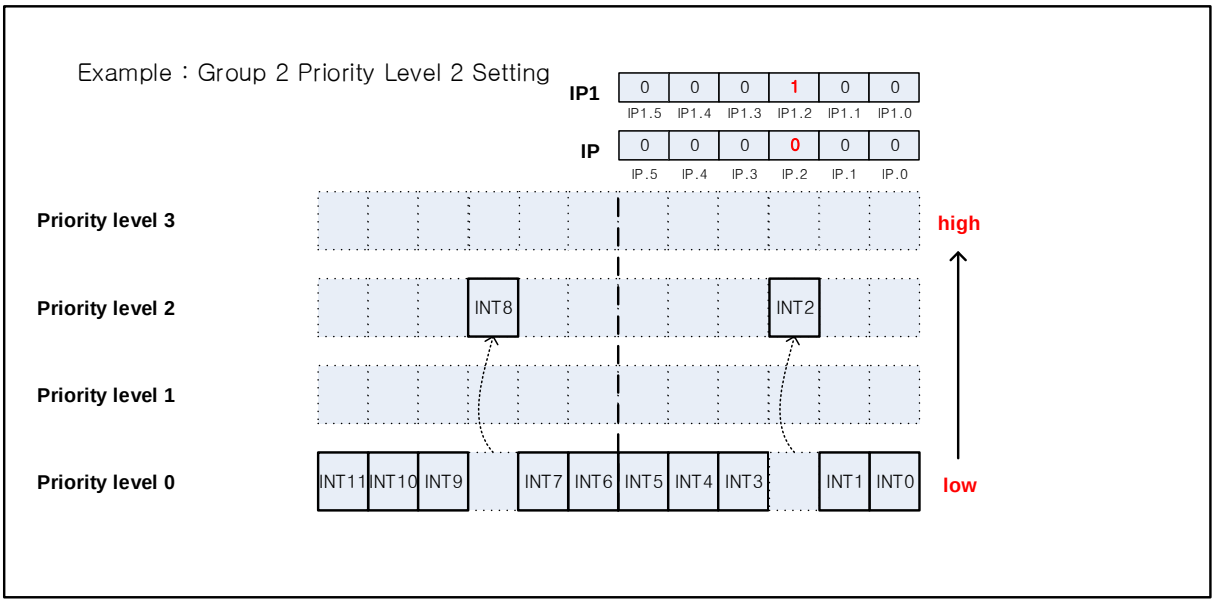


Figure 10-1 Interrupt Group Priority Level

10.2 Block Diagram

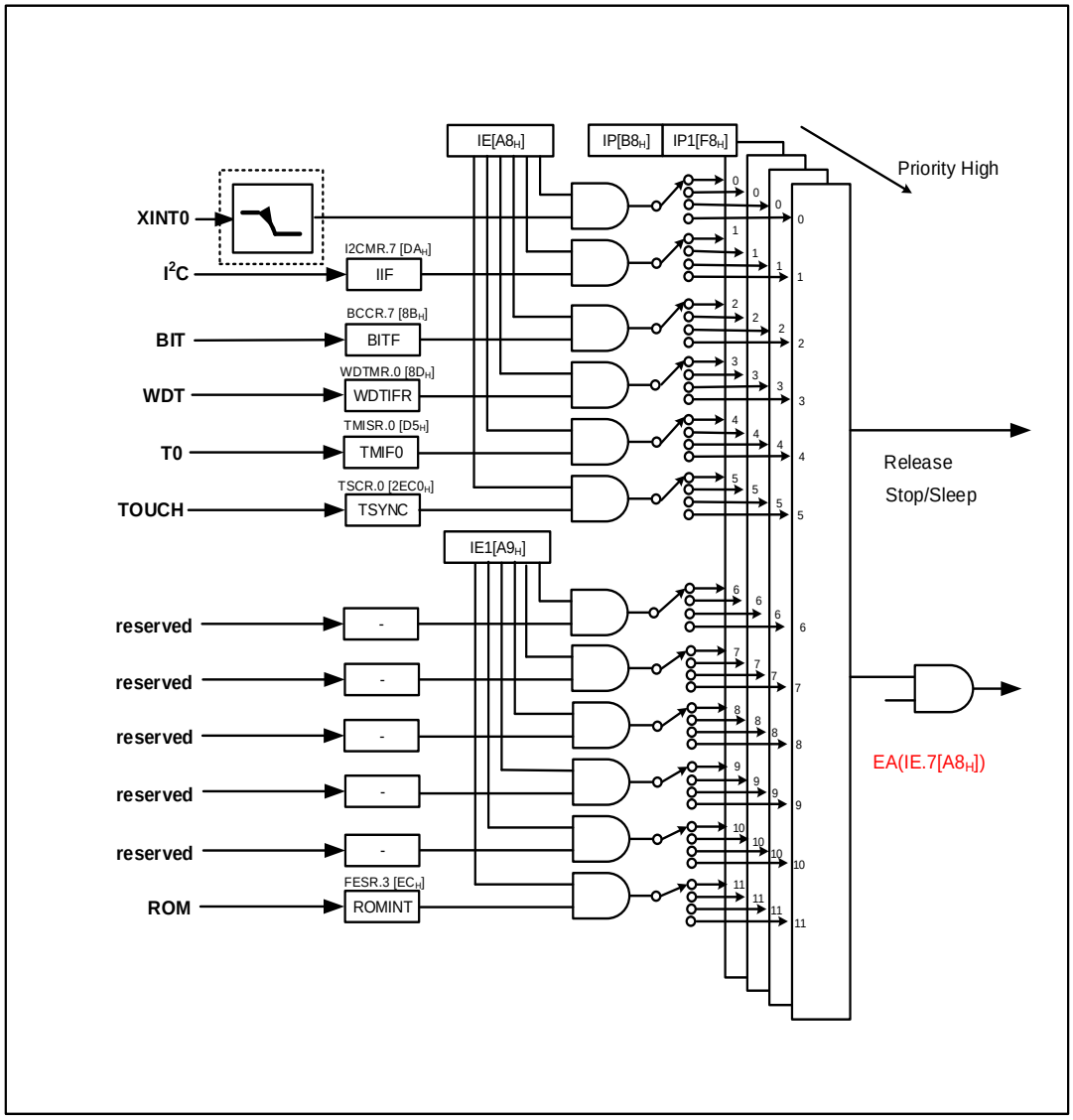


Figure 10-2 Block Diagram of Interrupt

10.3 Interrupt Vector Table

The interrupt controller supports 7 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 8 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
I^2C	INT1	IE0.1	2	Maskable	000BH
BIT	INT2	IE0.2	3	Maskable	0013H
WDT	INT3	IE0.3	4	Maskable	001BH
T0	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
-	INT8	IE1.2	9	Maskable	0043H
	INT9	IE1.3	10	Maskable	004BH
-	INT10	IE1.4	11	Maskable	0053H
ROM	INT11	IE1.5	12	Maskable	005BH

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

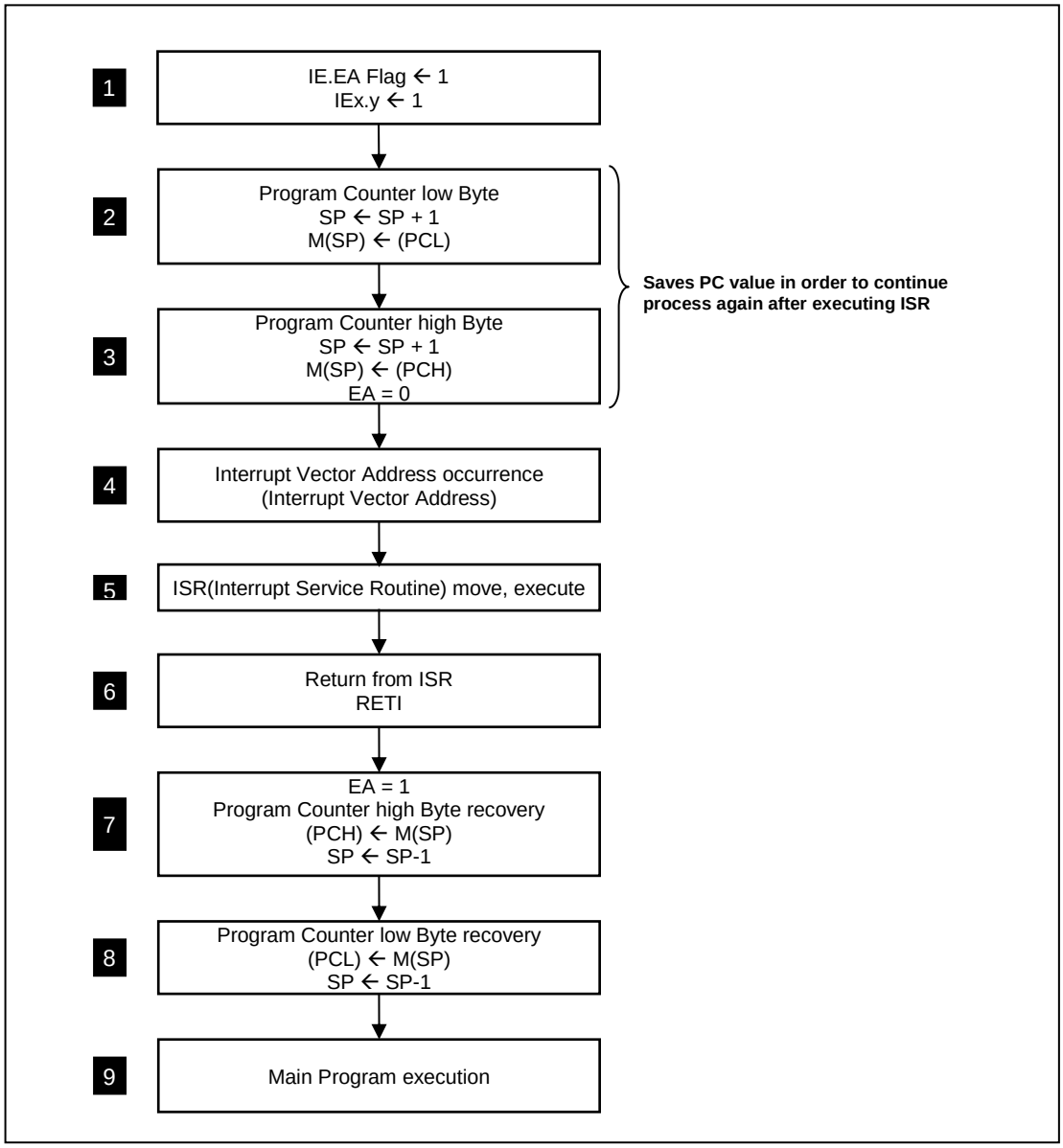


Figure 10-3 Interrupt Sequence Flow

10.5 Effective Timing after Controlling Interrupt bit

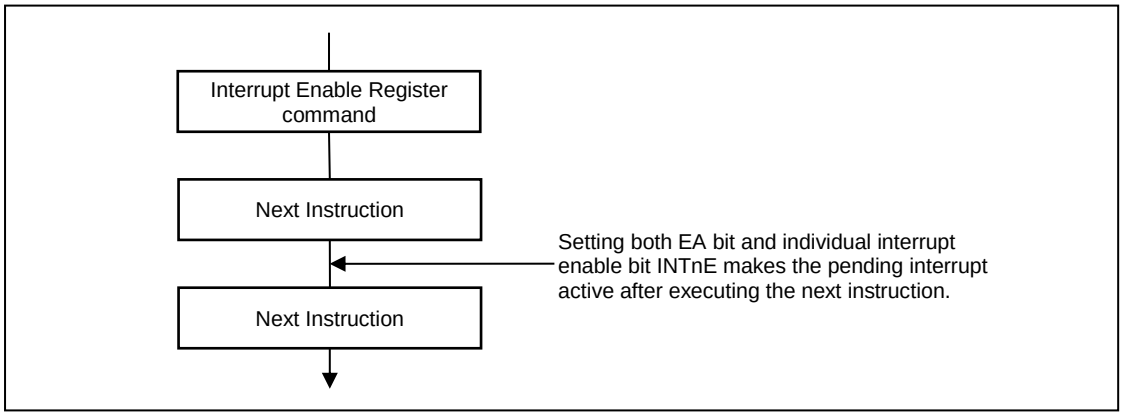


Figure 10-4 Interrupt Enable Register Effective Timing

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.

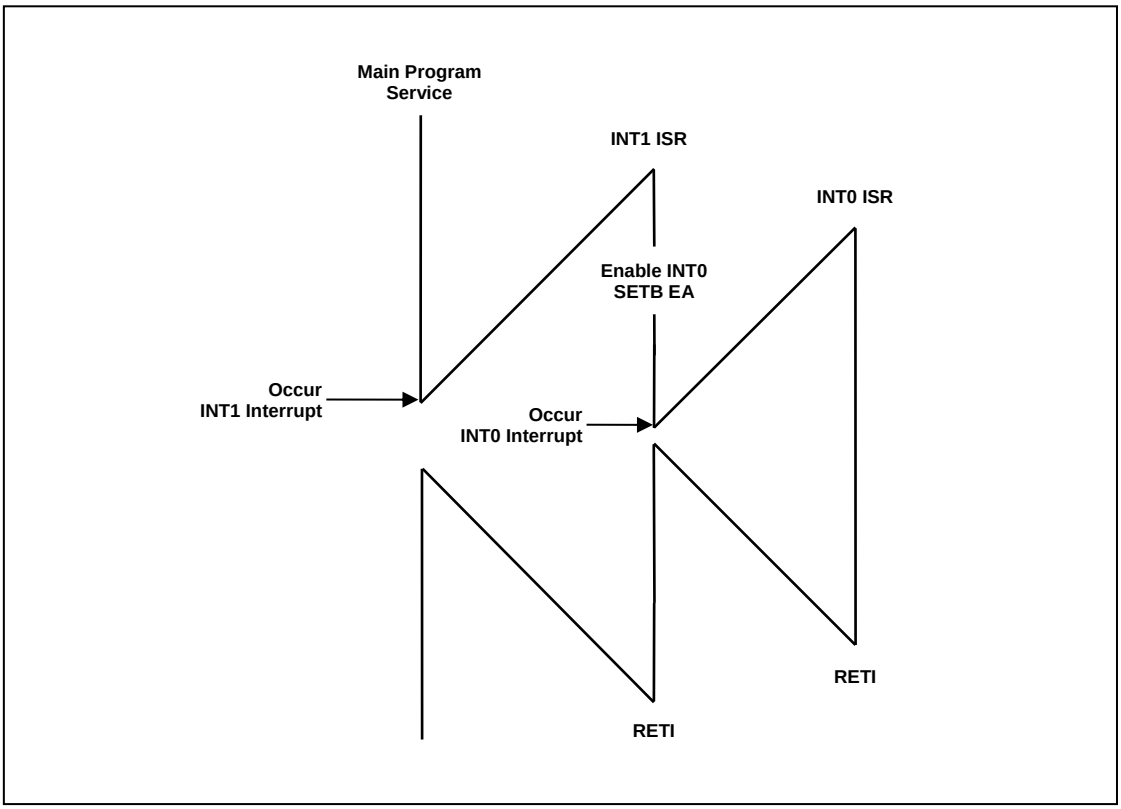


Figure 10-5 Execution of Multi Interrupt

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

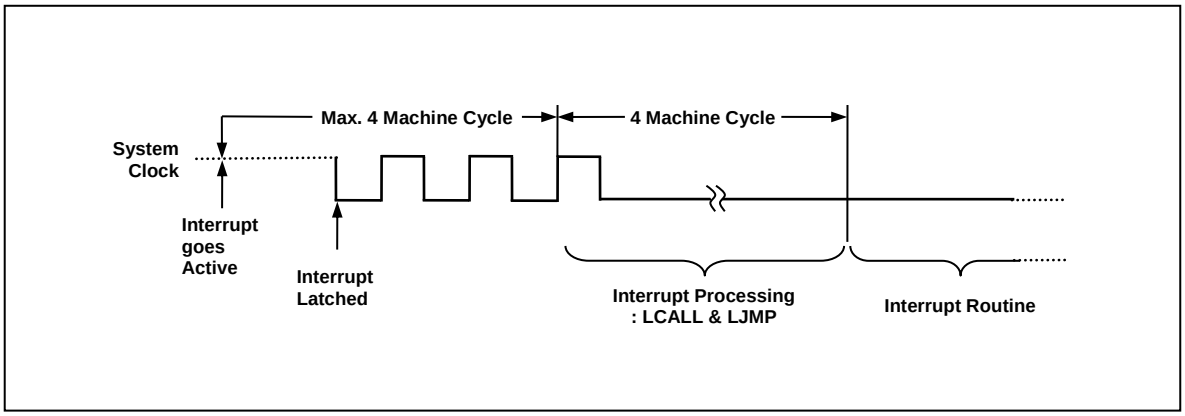


Figure 10-6 Interrupt Response Timing Diagram

10.8 Interrupt Service Routine Address

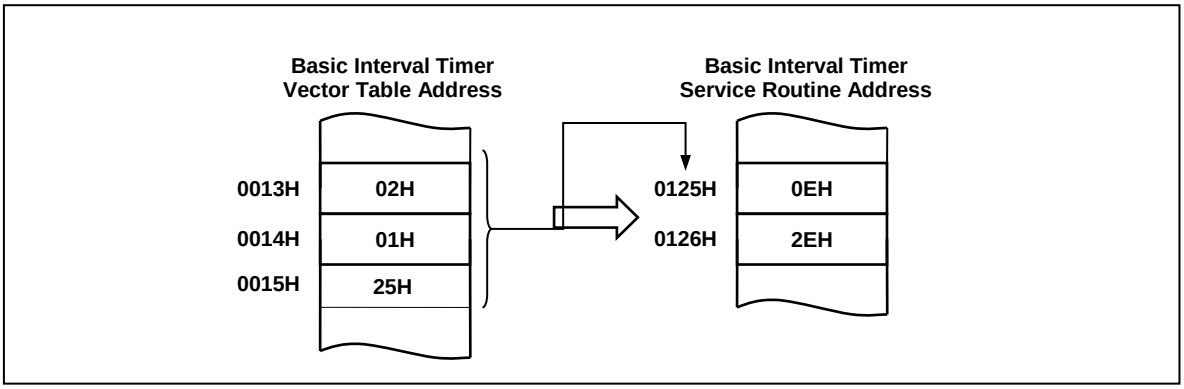
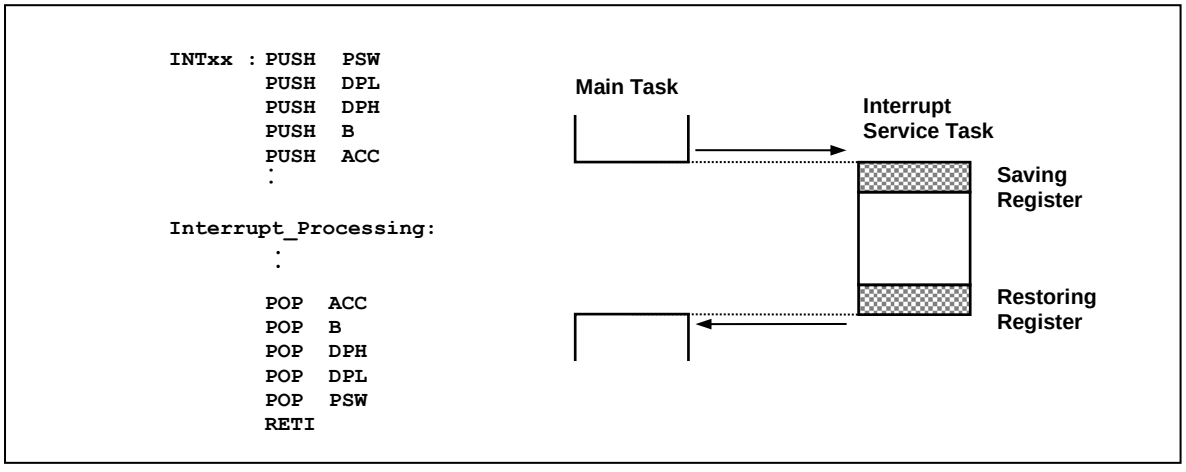


Figure 10-7 Correspondence between Vector Table Address and the Entry Address of ISR

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

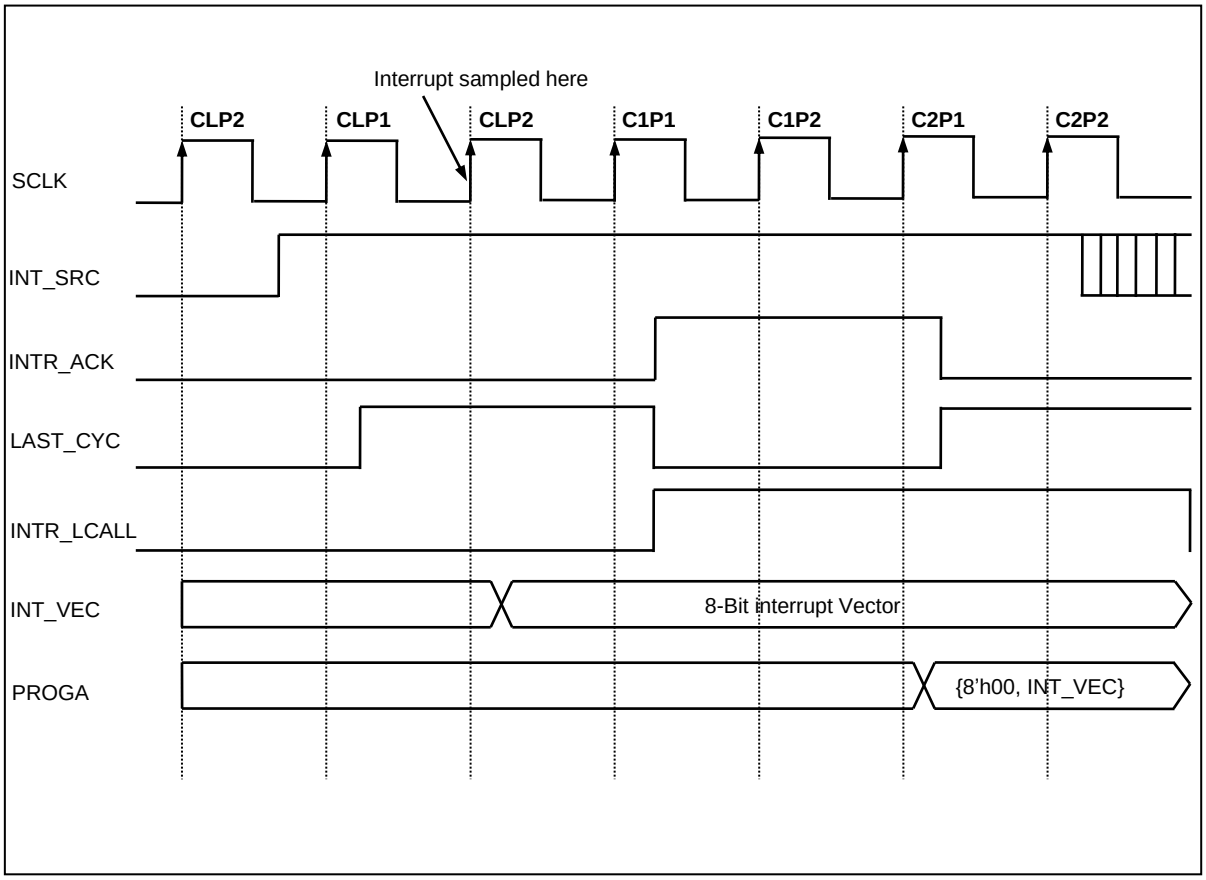


Figure 10-9 Timing Chart of Interrupt Acceptance and Interrupt Return Instruction

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)

Interrupt enable register consists of Global interrupt control bit (EA) and peripheral interrupt control bits. Totally 7 peripheral are able to control interrupt.

10.11.2 Interrupt Priority Register (IP, IP1)

The 7 interrupt divides 2 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
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

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 Priority Register (IP), Interrupt Priority Register 1 (IP1).

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

INT3E Enable or disable Watch Dog Timer Interrupt

0 Disable

1 Enable

INT2E Enable or disable BIT Interrupt

0 Disable

1 Enable

INT1E Enable or disable I^2C Interrupt

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 ROM Interrupt (not for user)

0 Disable

1 Enable

INT10E Reserved

0 Disable

1 Enable

INT9E Reserved

0 Disable

1 Enable

INT8E Reserved

0 Disable

1 Enable

INT7E Reserved

0 Disable

1 Enable

INT6E Reserved

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
-	-	-	-	-	-	-	ENAB0
-	-	-	-	-	-	-	R/W

Initial value: 00H

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
-	-	-	-	-	-	-	FLAG0
-	-	-	-	-	-	-	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.

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

0 External Interrupt 0 is not occurred

1 External Interrupt 0 is occurred

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 128kHz RING oscillator for BIT, WDT.

- 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

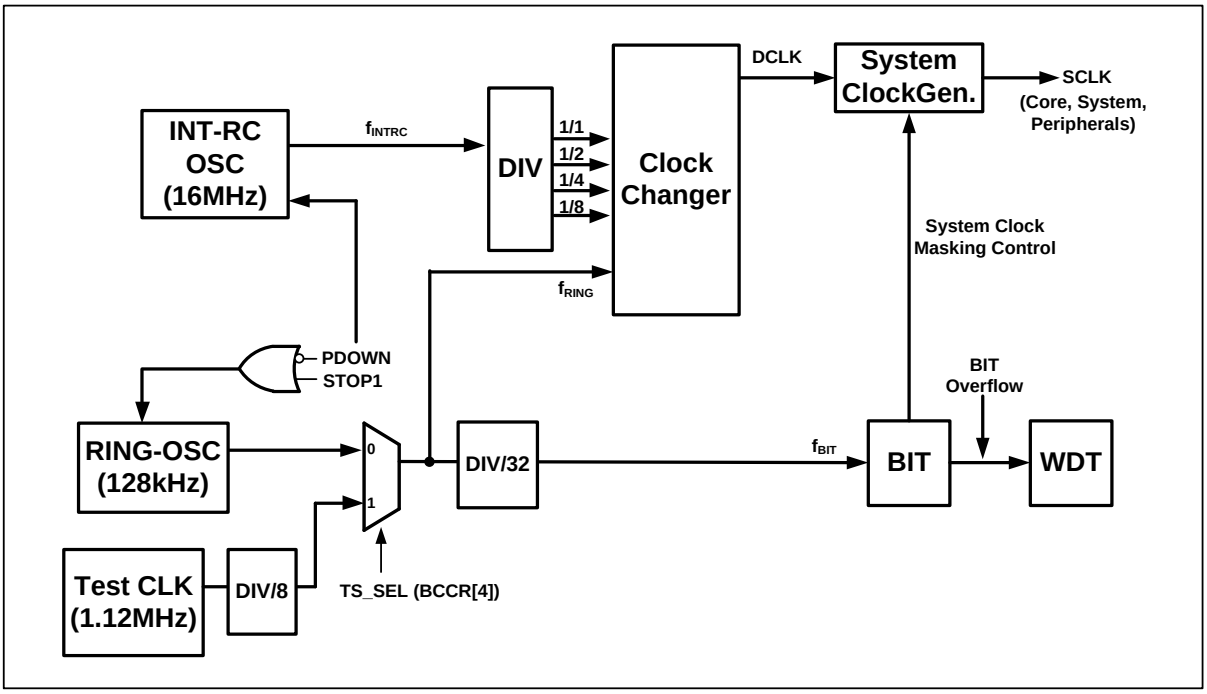


Figure 11-1 Clock Generator Block Diagram

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	-	-	CS
R/W	R/W	R/W	R/W	R/W	R	R	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

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

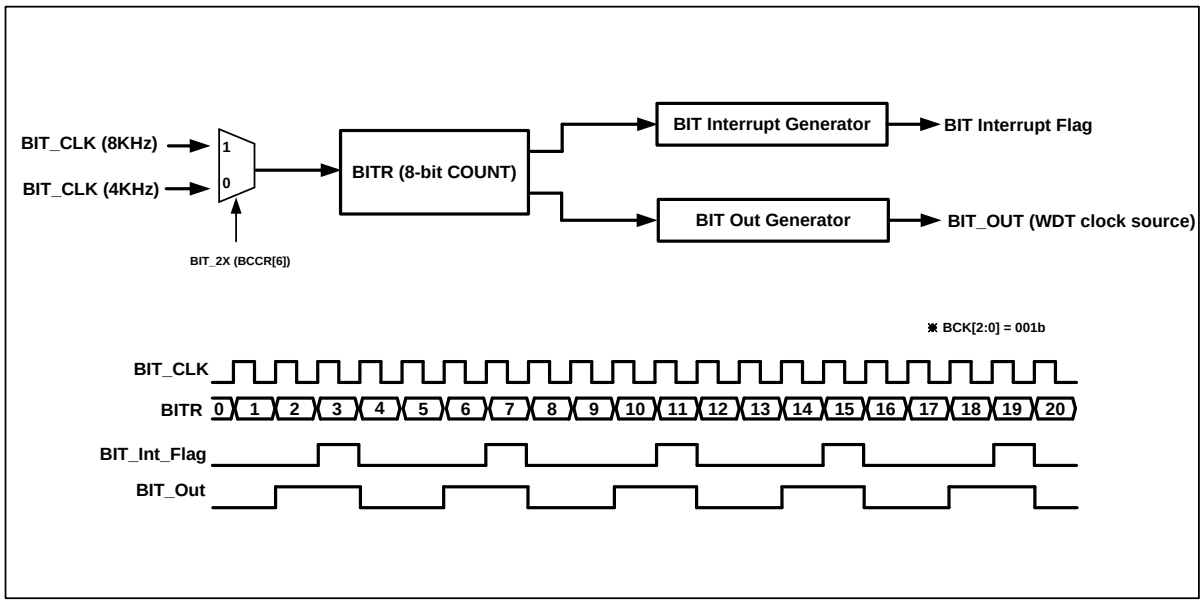


Figure 11-2 BIT Block Diagram

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	BIT_2X	-	TS_SEL	BCLR	BCK2	BCK1	BCK0
R/W	R	R	R/W	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 or when CPU accept specific interrupt request.

0 no generation

1 generation

BIT_2X Double speed mode enable register.

0 1 x mode. (BIT_CLK is 4kHz, default)

1 2 x mode (BIT_CLK is 8kHz)

TS_SEL Clock selection bit for BIT clock sources

0 RC-RING (128kHz, default)

1 Test clock (.12MHz)

BCLR If BCLR Bit is written to '1', BIT Counter is cleared as '0'.

0 Free Running

1 Clear Counter (It is cleared automatically after 1 clock)

BCK[2:0] Select BIT overflow period (when BIT Clock is 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

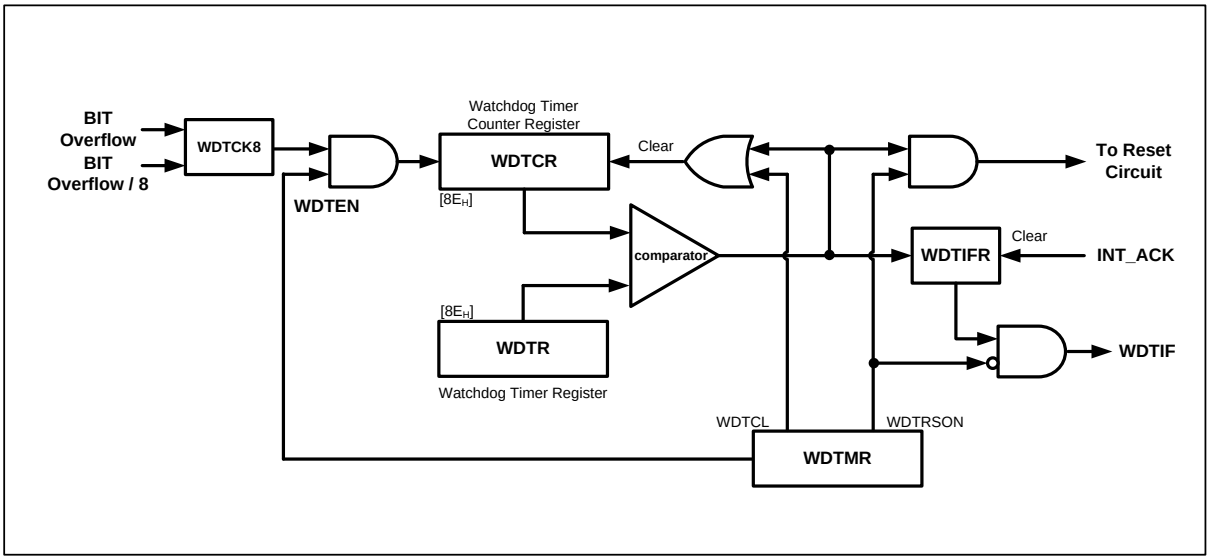


Figure 11-3 WDT Block Diagram

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	WDTCK8	-	-	-	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)

WDTCK8	Control WDT clock source
	0 BIT overflow output is clock source (default)
	1 BIT overflow output / 8 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

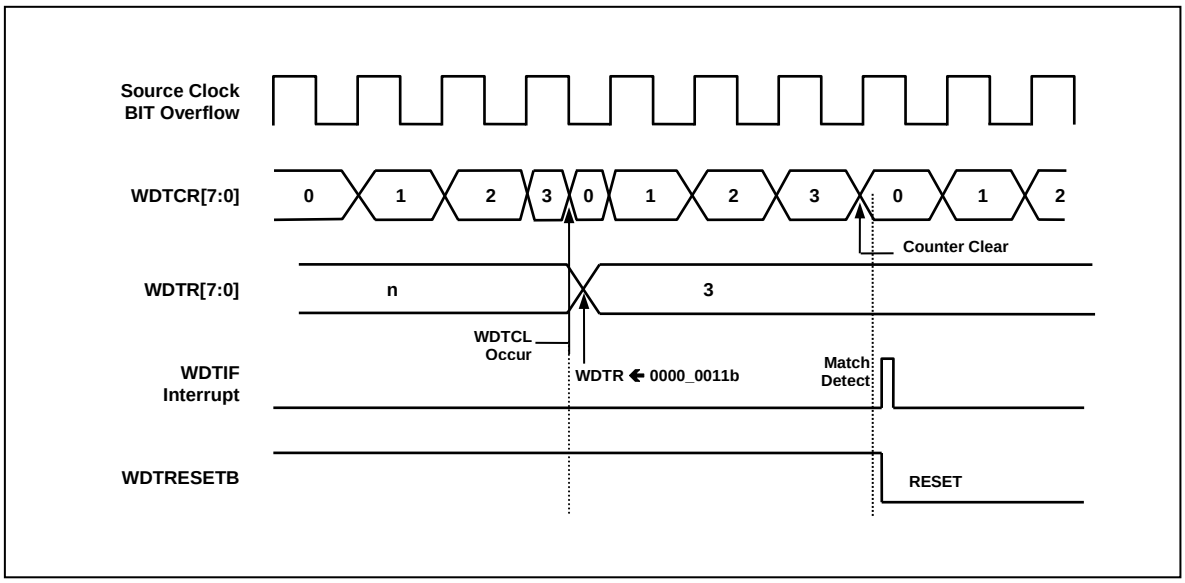


Figure 11-4 WDT Interrupt Timing Waveform

11.4 Timer/PWM

11.4.1 16-bit Timer/Event Counter 0

11.4.1.1 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.

11.4.1.2 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 11-5.

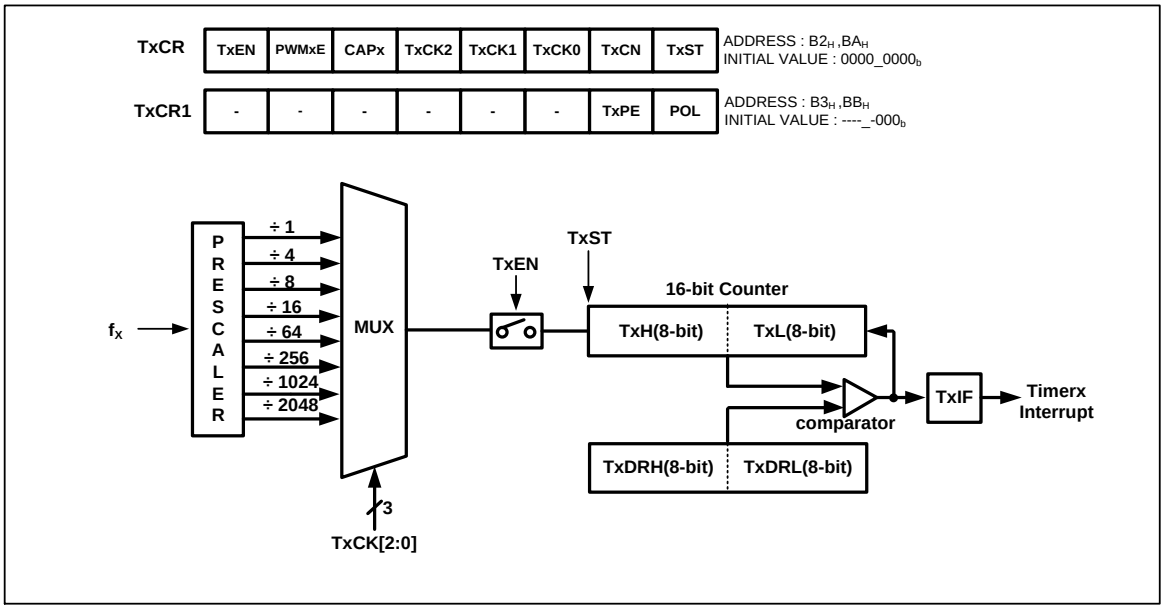


Figure 11-5 Timer x 16-bit Mode Block Diagram

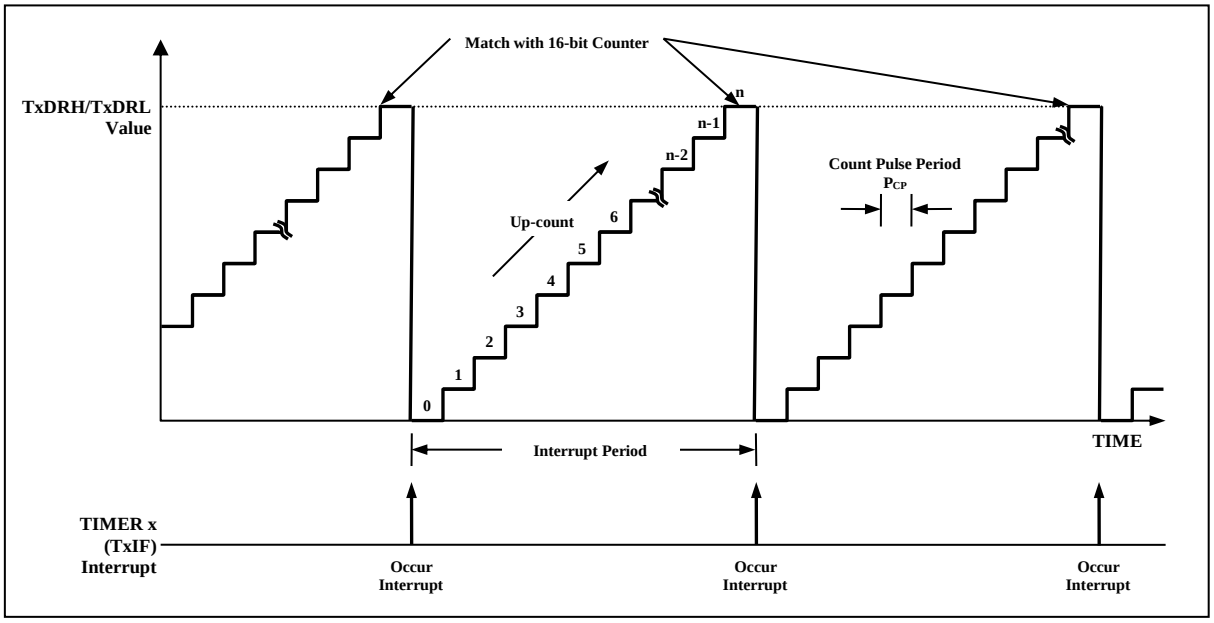


Figure 11-6 Interrupt of Timer/Counter Mode

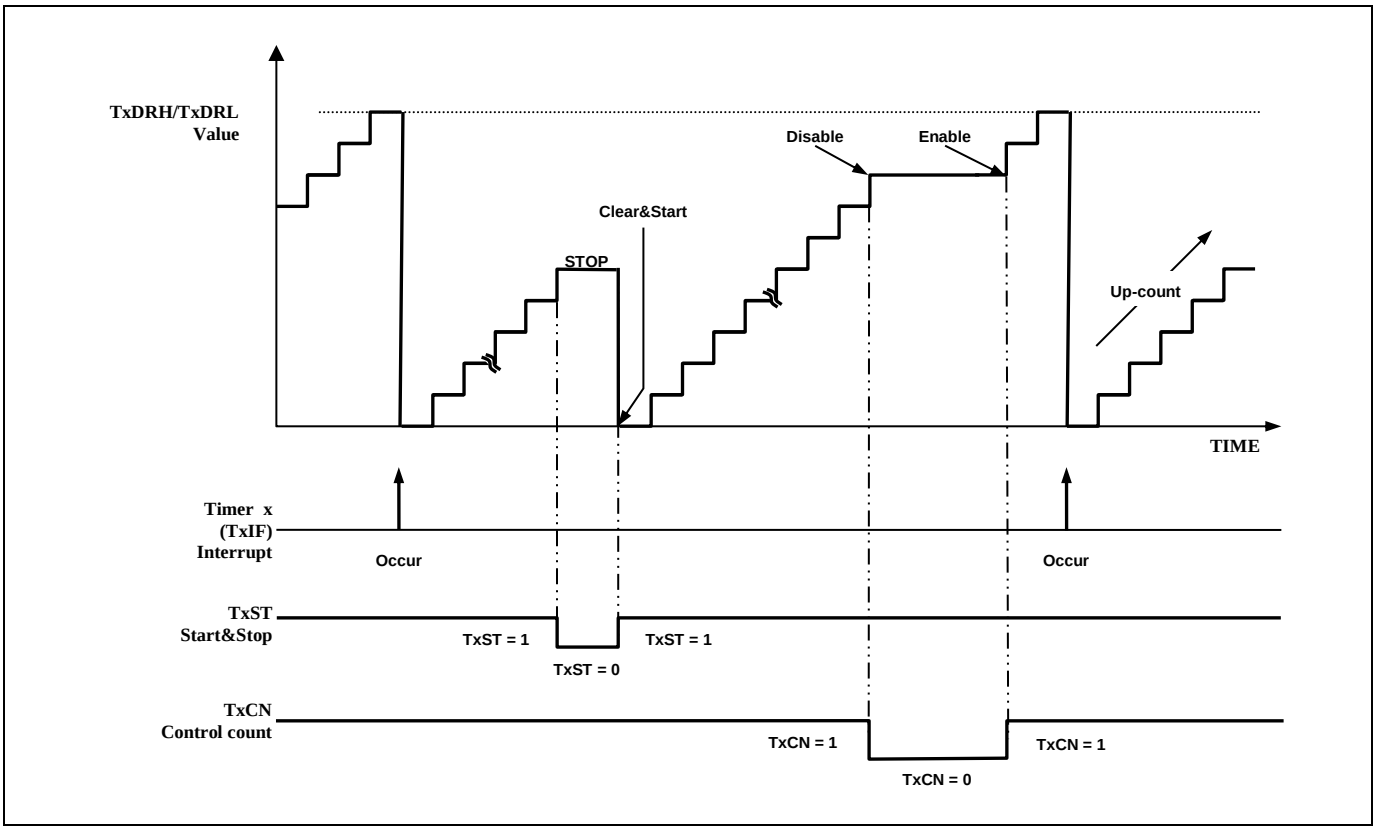


Figure 11-7 Operation Example of Timer/Event Counter Mode

11.4.1.3 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 BCCR register setting, the BIT interrupt 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.

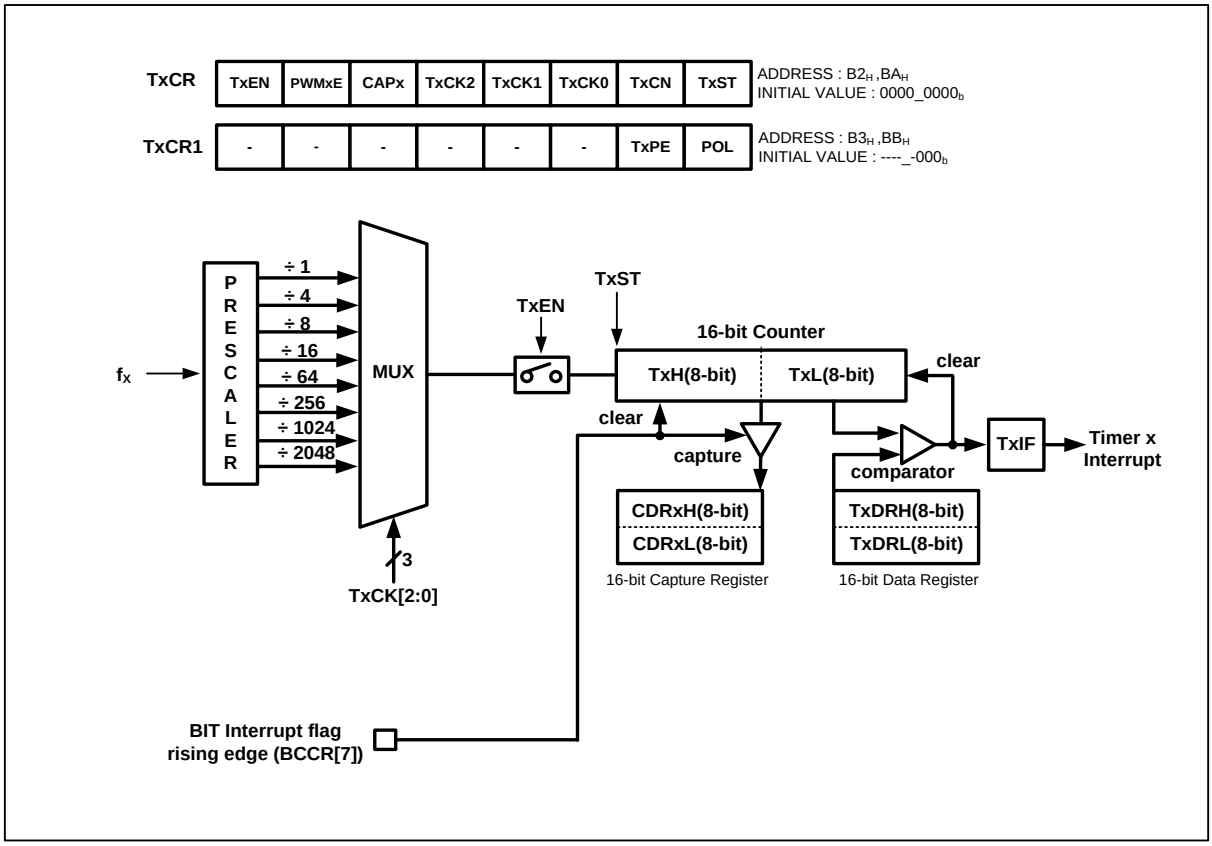


Figure 11-8 Timer x 16bit Capture Mode

.

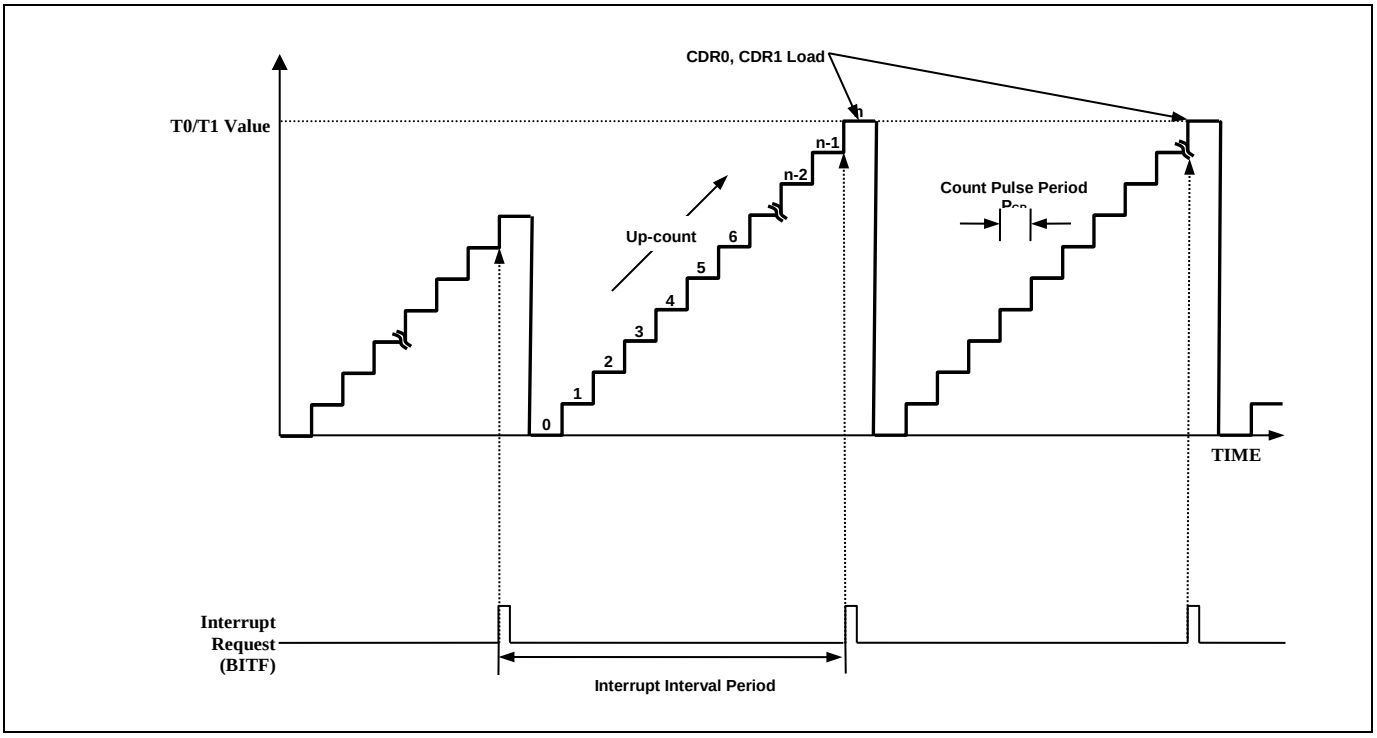


Figure 11-9 Input Capture Mode Operation of Timer 0, 1

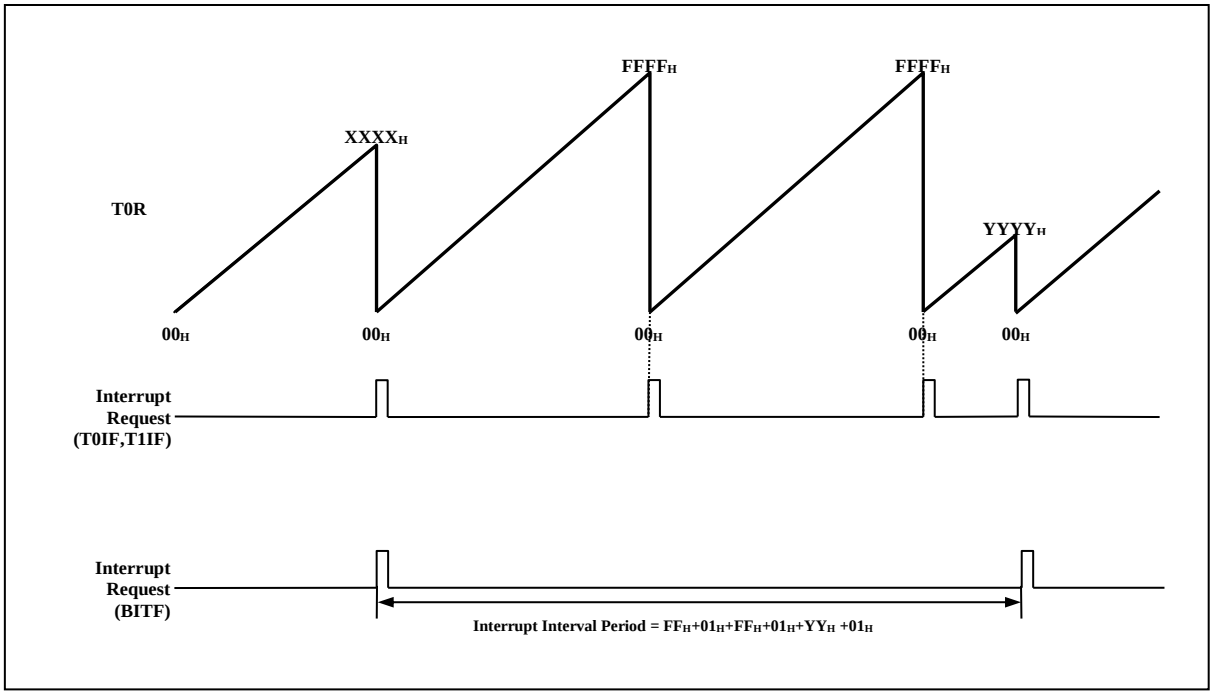


Figure 11-10 Express Timer Overflow in Capture Mode

11.4.1.4 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 11-13 Example of PWM at 8MHz

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

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

Table 11-4 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).

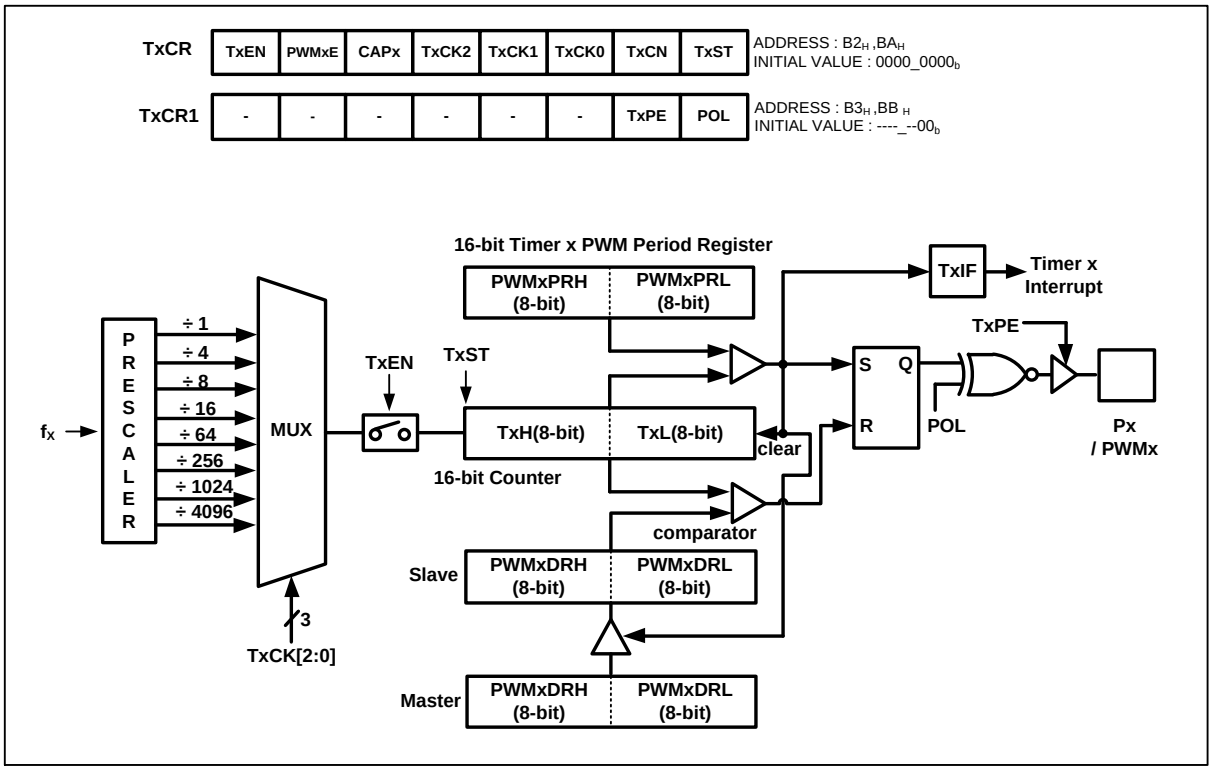


Figure 11-11 PWM Mode

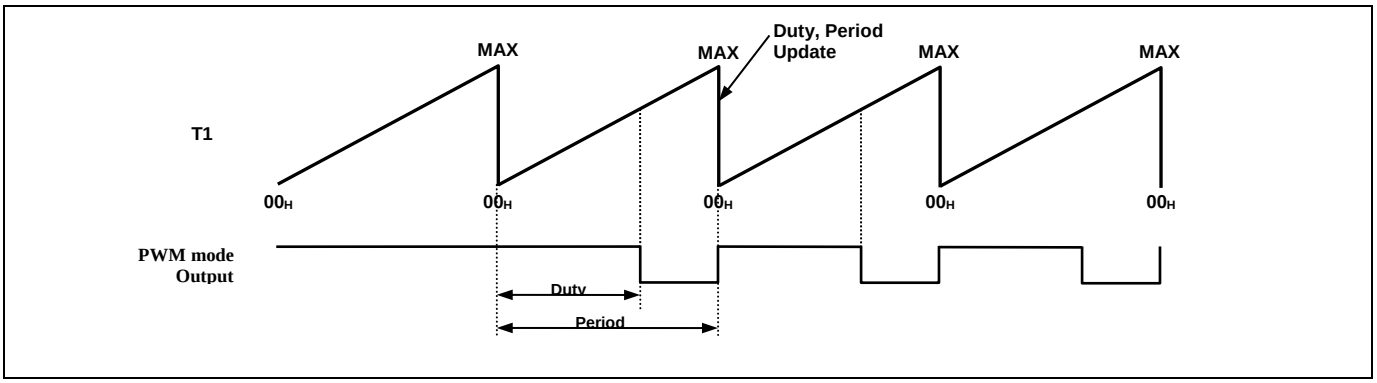


Figure 11-12 Example of PWM Output Waveform

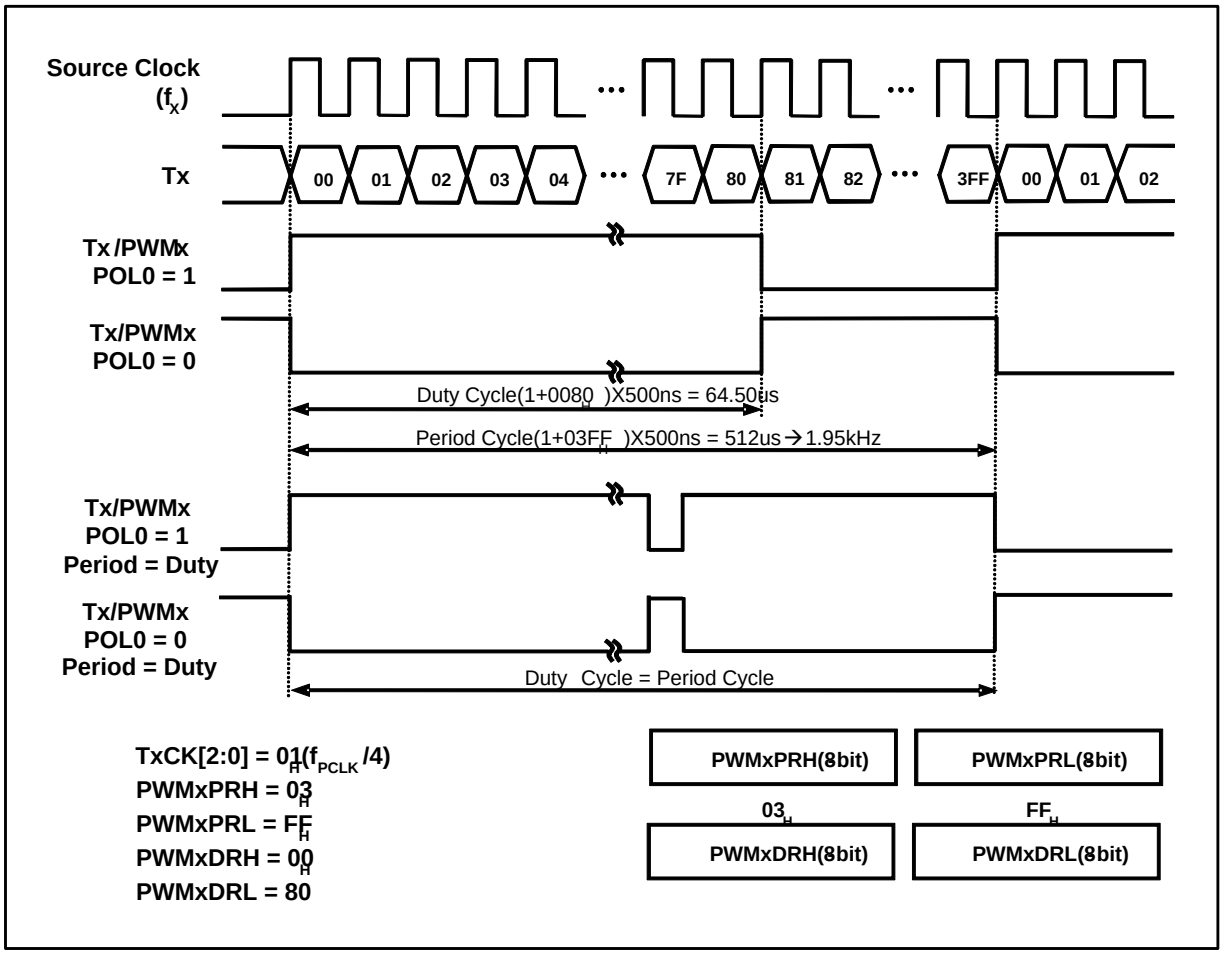


Figure 11-13 Example of PWM at 8MHz

11.4.1.5 Register Map

Table 11-5 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

11.4.1.6 Timer/Counter x Register Description

The Timer 0 Register consists of Timer 0 Mode Control Register (T0CR) Timer 0 Mode Control Register 1 (T0CR1), Timer 0 Low Register (T0L),Timer 0 Data Register Low (T0DRL), Timer 0 High Register (T0H), Timer 0 Data Register High (T0DRH), Timer 0 Capture Data Low Register (CDR0L), Timer 0 Capture Data High Register (CDR0H), PWM0 Low Duty Register (PWM0DRL), PWM0 High Duty Register (PWM0DRH), PWM0 Low Period Register (PWM0PRL), PWM0 High Period Register (PWM0PRH).

11.4.1.7 Register Description for Timer/Counter x

T0CR (Timer 0 Mode Control Register): B2H

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 (Timer 0 Mode Control Register 1) : B3H

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 (Timer 0 Register Low, Read Case) : B4H

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 (Capture 0 Data Register Low, Read Case) : B4H

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 (PWM0 Duty Register Low, Write Case) : B4H

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 (Timer 0 Register High, Read Case) : B5H

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 (Capture 0 Data High Register, Read Case) : B5H

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 (PWM0 Duty Register High, Write Case) : B5H

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 (Timer 0 Data Register Low, Write Case) : B6H

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 (PWM 0 Period 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 (Timer 0 Data Register High, Write Case) : B7H

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 (PWM 0 Period Register High, Write Case) : B7H

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 I²C

11.5.1 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

11.5.2 Block Diagram

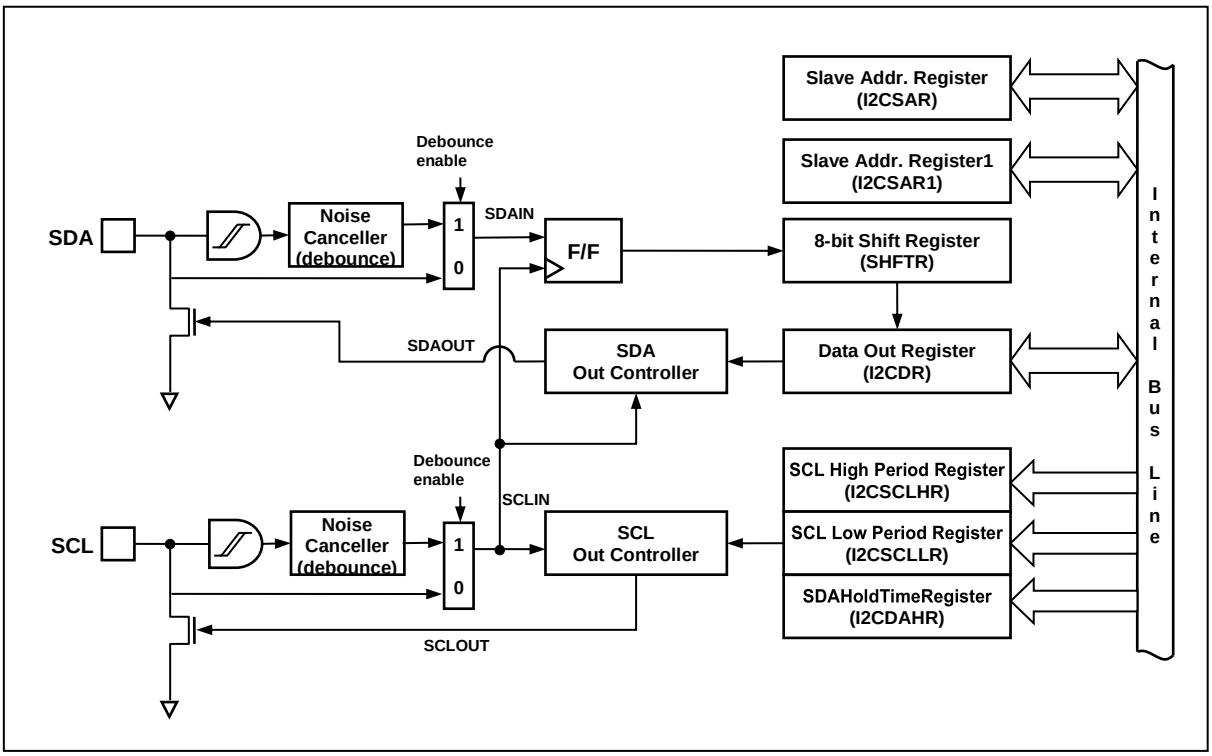


Figure 11-14 I²C Block Diagram

11.5.3 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.

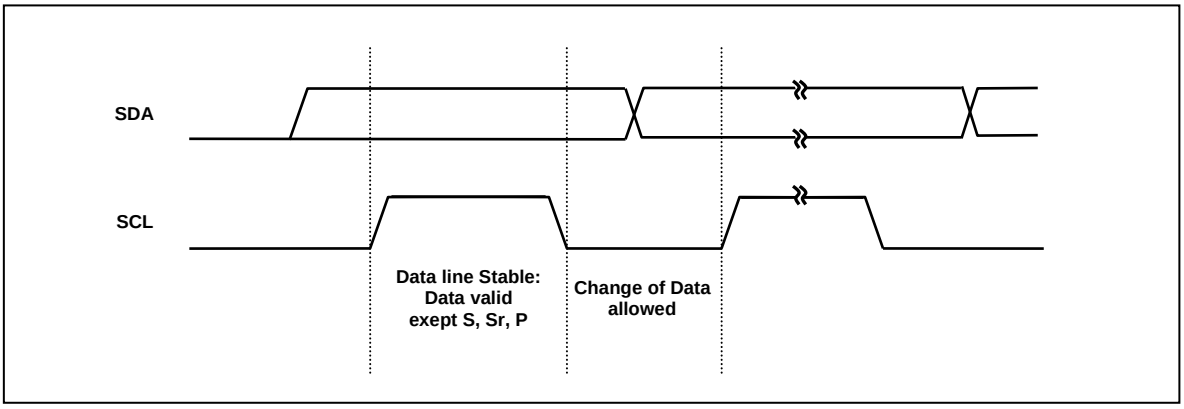


Figure 11-15 Bit Transfer on the I²C-Bus

11.5.4 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.

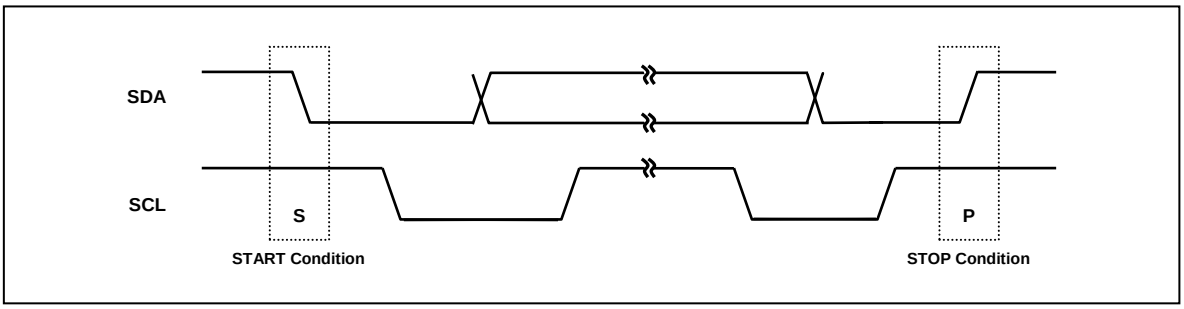


Figure 11-16 START and STOP Condition

11.5.5 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.

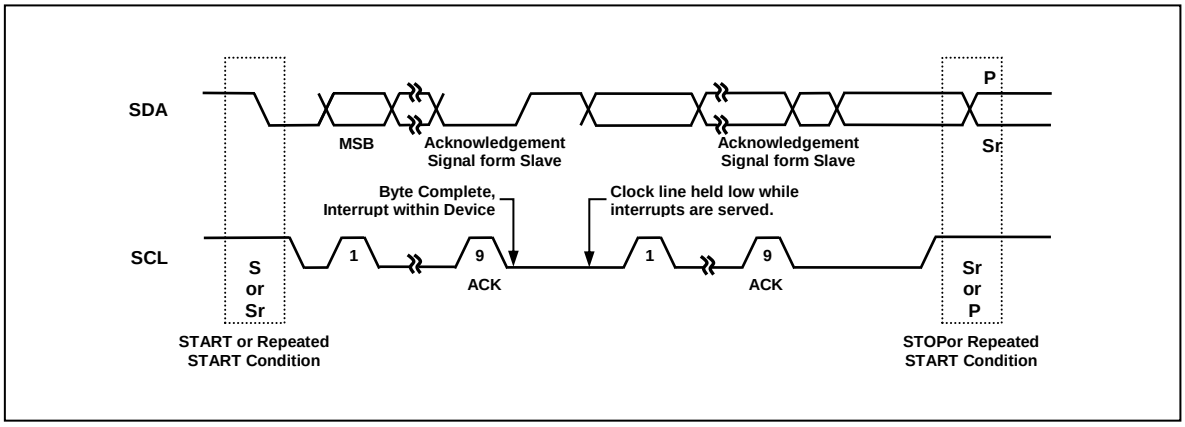


Figure 11-17 Data Transfer on the I²C-Bus

11.5.6 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.

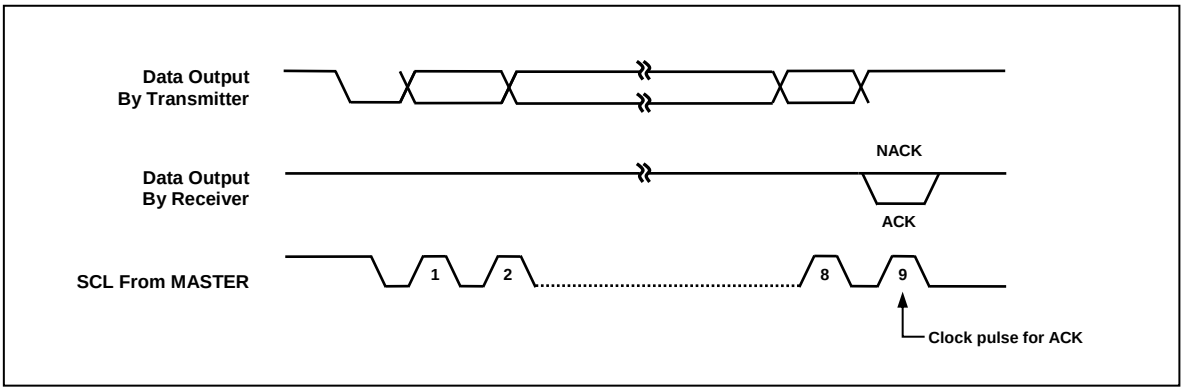


Figure 11-18 Acknowledge on the I²C-Bus

11.5.7 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.

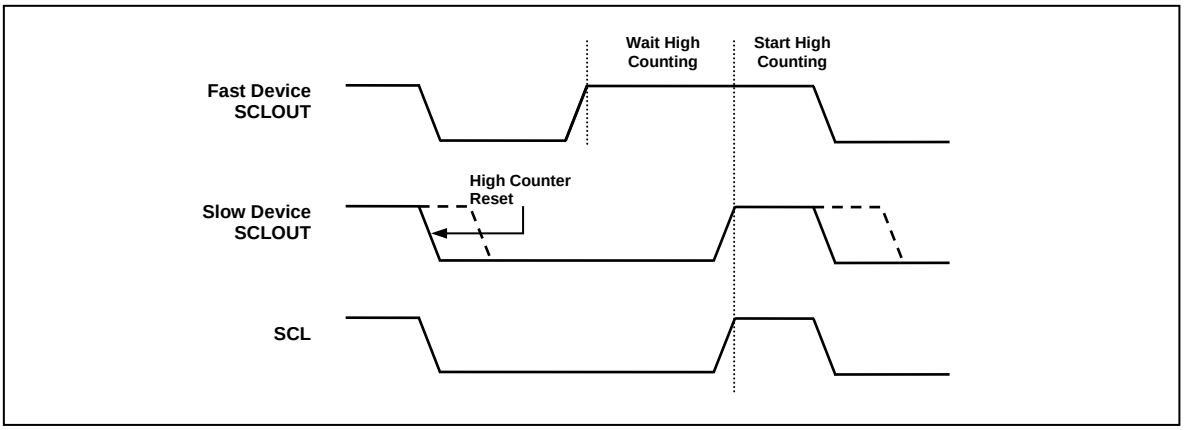


Figure 11-19 Clock Synchronization during Arbitration Procedure

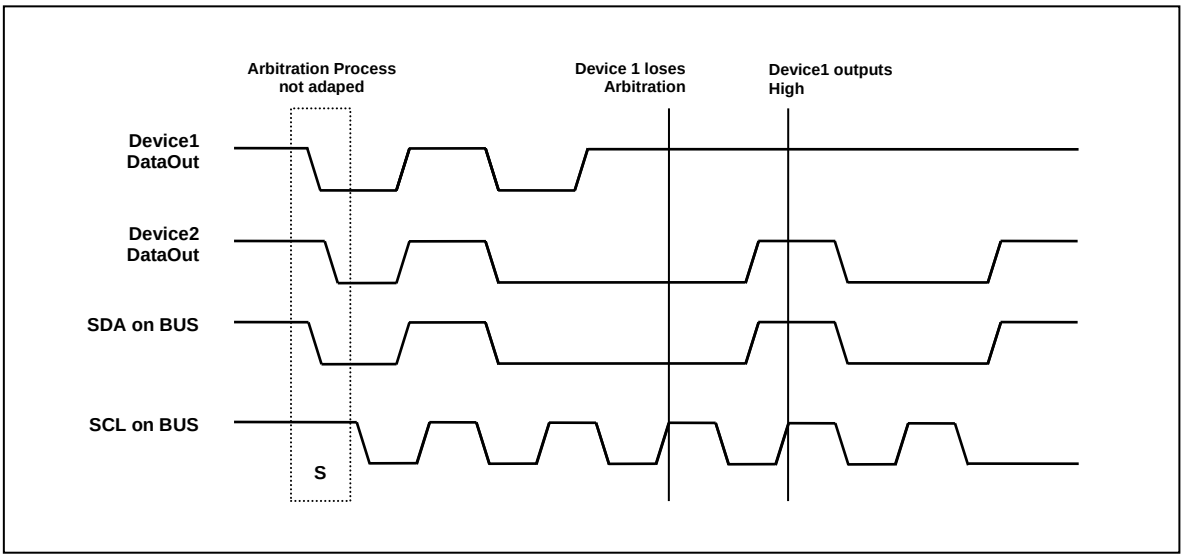


Figure 11-20 Arbitration Procedure of Two Masters

11.5.8 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 9^{th} 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 I²C interrupt is generated by I2C block, IIF flag in I2CMR register is set and it is cleared by writing any value to I2CSR. When I²C 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.

I²C 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

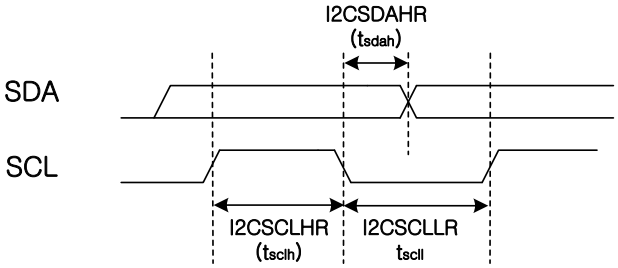
① I2C block will start operation (operation clock active) by setting IICEN bit on I2CMR register.

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

t_{scll} =t_{sysclk} (4xI2CSCLLR +1) → SCL clock low time

t_{sclh} =t_{sysclk} (4xI2CSCLHR +3) → SCL clock High time

t_{sdah} =t_{sysclk} (I2CSDAHR +4) → 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 = 6;

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;
```

11.5.9 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

11.5.10 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).

11.5.11 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 I²C operates in master mode. The base clock is SCLK, the system clock, and the period is calculated by the formula :$t_{SCLK} \times (4 \times SCLL + 1)$ where t_{SCLK} is the period of SCLK.

I2CSCLHR (SCL High Period Register) : 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 I²C operates in master mode. The base clock is SCLK, the system clock, and the period is calculated by the formula :$t_{SCLK} \times (4 \times SCLH + 3)$ where t_{SCLK} is the period of SCLK.

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

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

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 $t_{SCLK} \times$ SDAH. In master mode, load half the value of SCLL to this register to make SDA change in the middle of SCL. In slave mode, configure this register regarding the frequency of SCL from master. The SDA is changed after $t_{SCLK} \times$ (SDAH + 4). So, to insure normal operation in slave mode, the value $t_{SCLK} \times$ (SDAH + 4) must be smaller than the period of SCL.

I2CDR (I²C 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 I²C is configured as a transmitter, load this register with data to be transmitted. When I²C is a receiver, the received data is stored into this register.

I2CSAR (I²C 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 I²C module when I²C operates in slave mode.

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

0 Ignore general call address

1 Allow general call address

I2CSAR1 (I²C Slave Address Register 1) : 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 I²C module when I²C operates in slave mode.

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

0 Ignore general call address

1 Allow general call address

11.6 8-Channel Touch Sensing

11.6.1 Overview

. Self-Capacitive Touch Key Sensor.
. Total 8-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 ~ 3.6V.
. Current Consumption : T.B.D.
. Current Consumption@STOPmode :< 1uA.
. Operation Temperature : -40℃ ~ 85℃.

11.6.2 Block Diagram

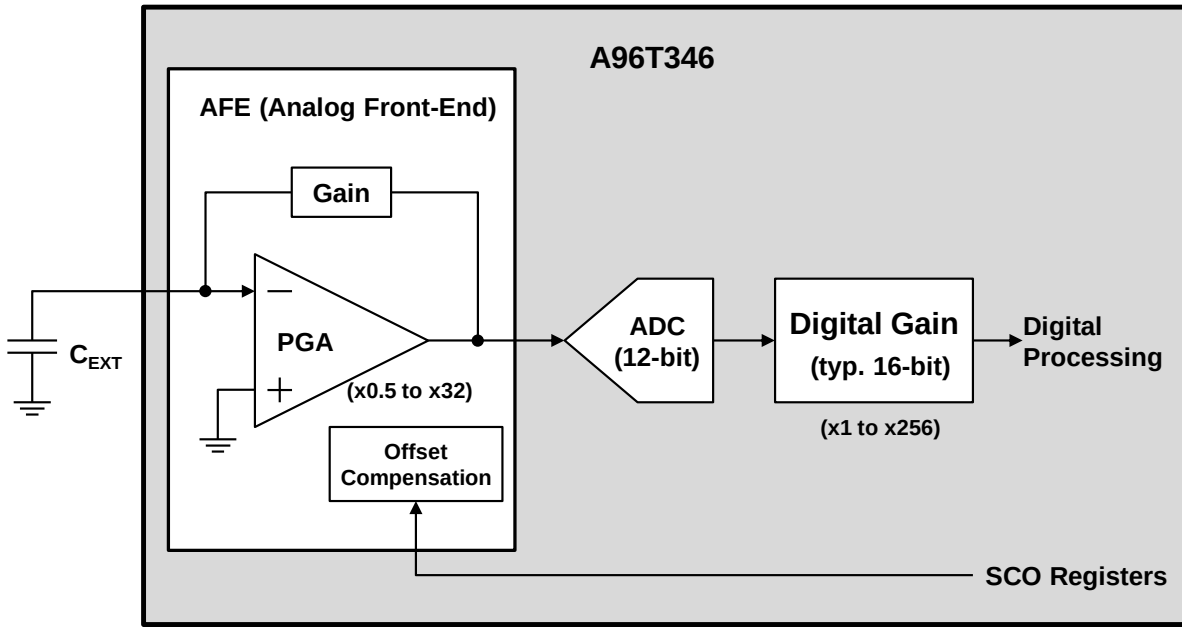


Figure 11-21 Block Diagram of Touch Sensing

1) AFE (Analog Front-End)

- Offset Compensation : Searchs for a cap identical to the external cap, then F/W automatically changes setting
- PGA (Programmable Gain Amplifier) : Gain configurable from minimum x0.5 up to maximum x32 by setting SCI Register and SCC Register (SCI : Input Capacitance Setting, SCC : Feedback Capacitance Setting)

SCI[2:0]	SCC[2:0]	Gain	SCI[2:0]	SCC[2:0]	Gain
111	000	32	111	111	4
111	001	16	110	111	3.5
111	010	10.7	101	111	3
111	011	8	100	111	2.5
111	100	6.4	011	111	2
111	101	5.3	010	111	1.5
111	110	4.6	001	111	1
111	111	4	000	111	0.5

2) ADC (Analog-to-Digital Conversion)

- ADC supports 12-bit resolution. Please refer to section 11.7 for detailed explanation. In order to use the ADC for Touch Sensor, set SAP[1:0] value of TS_MODE Register to 0x01.

TS_MODE (A/D Converter Mode Register) : 2E21H

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

Initial value : 10H

SAP[1:0] Touch Sensor / ADC Selection

- 01 Touch Sensor Select (Default)
- 10 ADC Select. The result of ADC is stored only at SUM_CH0 registers.

PORT[1:0] Port Configuration During Inactive Status

- 00 Floating
- 01 Low
- 10 High

- Configure TS_CH_SEL Register to select touch channels. Please refer to section 11.6.4 for detailed explanation on TS_CH_SEL (2E23H) Register.

3) Digital Gain

- Gain can be configured from x1 up to x256 and typically represented in 16-bit data width.

11.6.3 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_CH7H	2E0EH	R	00H	Touch Sensor Ch7 Sum High-Byte Register
SUM_CH7L	2E0FH	R	00H	Touch Sensor Ch7 Sum Low-Byte Register
SCO0_H	2E10H	R/W	00H	Touch Sensor Offset Capacitor Selection High-Byte Register for Ch0
SCO0_L	2E11H	R/W	00H	Touch Sensor Offset Capacitor Selection Low-Byte Register for Ch0
…				
SCO7_H	2E1EH	R/W	00H	Touch Sensor Offset Capacitor Selection High-Byte Register for Ch7
SCO7_L	2E1FH	R/W	00H	Touch Sensor Offset Capacitor Selection Low-Byte Register for Ch7
TS_CON	2E20H	R/W	00H	Touch Sensor Control Register
TS_MODE	2E21H	R/W	10H	Touch Sensor Mode Register
TS_SUM_CNT	2E22H	R/W	01H	Touch Sensor Sum Repeat Count Register
TS_CH_SEL	2E23H	R/W	00H	Touch Sensor Channel Selection Register
TS_SLP_CR	2E24H	R/W	01H	Touch Sensor Low Pass Filter Control Register
TS_INTEG_CNT	2E27H	R/W	32H	Touch Sensor Sensing Integration Count Register
TS_FREQ_NUM	2E28H	R/W	FFH	Touch Sensor Frequency Number Register
TS_FREQ_DEL	2E29B	R/W	00H	Touch Sensor Frequency Delta Register
TS_CLK_CFG	2E2AH	R/W	10H	Touch Sensor Clock Configuration Register
TRIM_OSC	2E2BH	R/W	B8H	Touch Sensor RING OSC. Trimming Selection Register
TRIM_A_OSC	2E2CH	R/W	FFH	Touch Sensor RING OSC. Trimming for ADC Register
SCI	2E2DH	R/W	34H	Touch Sensor Input Capacitor Selection Register
SCC	2E2EH	R/W	04H	Touch Sensor Conversion Capacitor Selection Register
SVREF	2E2FH	R/W	04H	Touch Sensor VREF Selection Register
TAR	2E30H	R/W	20H	Touch Sensor Integration AMP Reset Time Register
TRST	2E31H	R/W	03H	Touch Sensor Reset Time of Sensing Register
TDRV	2E32H	R/W	03H	Touch Sensor Driving Time of Sensing Register
TINT	2E33H	R/W	14H	Touch Sensor Integration Time of Sensing Register
TD	2E34H	R/W	20H	Touch Sensor Differential AMP Sampling Register
LDOCR	8FH	R/W	00H	LDO Control Register

11.6.4 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

SCO0_H (Touch Sensor Offset Capacitor Selection High-Byte Register for CH0) : 2E10H

7	6	5	4	3	2	1	0
SCO0_H							
						R/W	R/W

Initial value : 00H

SCO0_L (Touch Sensor Offset Capacitor Selection Low-Byte Register for CH0) : 2E11H

7	6	5	4	3	2	1	0
SCO0_L							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

SCO1_H (Touch Sensor Offset Capacitor Selection High-Byte Register for CH1) : 2E12H
SCO1_L (Touch Sensor Offset Capacitor Selection Low-Byte Register for CH1) : 2E13H
SCO2_H (Touch Sensor Offset Capacitor Selection High-Byte Register for CH2) : 2E14H
SCO2_L (Touch Sensor Offset Capacitor Selection Low-Byte Register for CH2) : 2E15H
SCO3_H (Touch Sensor Offset Capacitor Selection High-Byte Register for CH3) : 2E16H
SCO3_L (Touch Sensor Offset Capacitor Selection Low-Byte Register for CH3) : 2E17H
SCO4_H (Touch Sensor Offset Capacitor Selection High-Byte Register for CH4) : 2E18H
SCO4_L (Touch Sensor Offset Capacitor Selection Low-Byte Register for CH4) : 2E19H
SCO5_H (Touch Sensor Offset Capacitor Selection High-Byte Register for CH5) : 2E1AH
SCO5_L (Touch Sensor Offset Capacitor Selection Low-Byte Register for CH5) : 2E1BH
SCO6_H (Touch Sensor Offset Capacitor Selection High-Byte Register for CH6) : 2E1CH
SCO6_L (Touch Sensor Offset Capacitor Selection Low-Byte Register for CH6) : 2E1DH
SCO7_H (Touch Sensor Offset Capacitor Selection High-Byte Register for CH7) : 2E1EH
SCO7_L (Touch Sensor Offset Capacitor Selection Low-Byte Register for CH7) : 2EFH

TS_CON (Touch Sensor Control Register) : 2E20H

7	6	5	4	3	2	1	0
-	-	-	OSC_EN	BGR_EN	TS_IF	-	TS_RUN
-	-	-	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 This flag indicates that the new sensing results are generated.

TS_RUN Touch Sensor Enable
- 0 Touch Sensor Disable (Default)
- 1 Touch Sensor Enable

TS_MODE (Touch Sensor Mode Register) : 2E21H

7	6	5	4	3	2	1	0
SREF	SC_GAIN	SAP1	SAP0	-	-	PORT1	PORT0
-	-	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

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 : TEST only) : 2E22H

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 (Touch Sensor Channel Selection Register) : 2E23H

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 P01

CH6_SEL Channel 6 Enable
- 0 Disable (Default)
- 1 Enable Touch Key by P00

CH5_SEL Channel 5 Enable
- 0 Disable (Default)
- 1 Enable Touch Key by P15

CH4_SEL Channel 4 Enable
- 0 Disable (Default)
- 1 Enable Touch Key by P11

CH3_SEL Channel 3 Enable
- 0 Disable (Default)
- 1 Enable Touch Key by P10

CH2_SEL Channel 2 Enable
- 0 Disable (Default)
- 1 Enable Touch Key by P04

CH1_SEL Channel 1 Enable
- 0 Disable (Default)
- 1 Enable Touch Key by P03

CH0_SEL Channel 0 Enable
- 0 Disable (Default)
- 1 Enable Touch Key by P02

TS_SLP_CR (Touch Sensor Low Pass Filter Control Register) : 2E24H

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) : 2E27H

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) : 2E28H

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) : 2E29H

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) : 2E2AH

7	6	5	4	3	2	1	0
CLKSEL-	ACLKDIV2	ACLKDIV1	ACLKDIV0	TSCLKOE	TSCLKDIV2	TSCLKDIV1	TSCLKDIV0
-	-	-	-	RW	RW	RW	RW

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

ACLKDIV[2:0] ADC Clock Divider (Refer to TRIM_A_OSC)

000 Reserved

001 Reserved (default)

010 Reserved

011 OSC_{ts} / 8

100 Reserved

101 Reserved

110 Reserved

111 Reserved

CLKSEL Touch Clock Source Select

0 Touch Sensor Clock

1 System MCU Clock

TRIM_OSC (Touch Sensor RING OSC. Trimming Selection Register) : 2E2BH

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

Initial value : B8H

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

00H 40MHz (maximum)

…

B8H 18.6MHz (default)

…

FFH 3.2MHz (minimum)

TRIM_A_OSC (Touch Sensor RING OSC. Trimming for ADC Register) : 2E2CH

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

Initial value : FFH

TRIM_A_OSC[7:0] Touch Sensor RING OSC. Trimming for ADC Selection

00H 40MHz (maximum)

…

B8H 18.6MHz (typical)

…

FFH 3.2MHz (minimum, default)

SCI (Touch Sensor Input Capacitor Selection Register) : 2E2DH

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 0.8uA

0001 1.2uA

0010 1.6uA

0011 2.0uA

0111 2.4uA

1011 3.6uA

1111 6.8uA

SCI[2:0] Touch Sensor Input Capacitor

000 2.4pF

001 4.8pF

010 7.2pF

011 9.6pF

100 12.0pF

101 14.4pF

110 16.8pF

111 19.2pF

SCC (Touch Sensor Conversion Capacitor Selection Register) : 2E2EH

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) : 2E2FH

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

TAR(Touch Sensor Intergration AMP Reset Time Register) : 2E30H

7	6	5	4	3	2	1	0
TAR7	TAR6	TAR5	TAR4	TAR3	TAR2	TAR1	TAR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 20H

TRST(Touch Sensor Reset Time of Sensing Register) : 2E31H

7	6	5	4	3	2	1	0
TRST7	TRST6	TRST5	TRST4	TRST3	TRST2	TRST1	TRST0
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.

TDRV(Touch Sensor Driving Time of Sensing Register) : 2E32H

7	6	5	4	3	2	1	0
TDRV7	TDRV6	TDRV5	TDRV4	TDRV3	TDRV2	TDRV1	TDRV0
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.

TINT(Touch Sensor Integration Time of Sensing Register) : 2E33H

7	6	5	4	3	2	1	0
TINT7	TINT6	TINT5	TINT4	TINT3	TINT2	TINT1	TINT0
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.

TD(Touch Sensor Differential AMP Sampling Time Register) : 2E34H

7	6	5	4	3	2	1	0
TD7	TD6	TD5	TD4	TD3	TD2	TD1	TD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 20H

LDOCR (LDO Control Register) : 8FH

7	6	5	4	3	2	1	0
-	-	-	-	-	MODESEL	LEVELSEL	LDODIS
-	-	-	-	-	R/W	R/W	R/W

Initial value : 00H

MODESEL LDO mode selection
- 0 LDO mode (default)
- 1 Bypass mode

LEVELSEL LDO level selection
- 0 3.0V (default)
- 1 2.5V

LDODIS LDO Disable (test only)
- 0 Enable (default)
- 1 Disable

11.6.5 User Programming Procedure

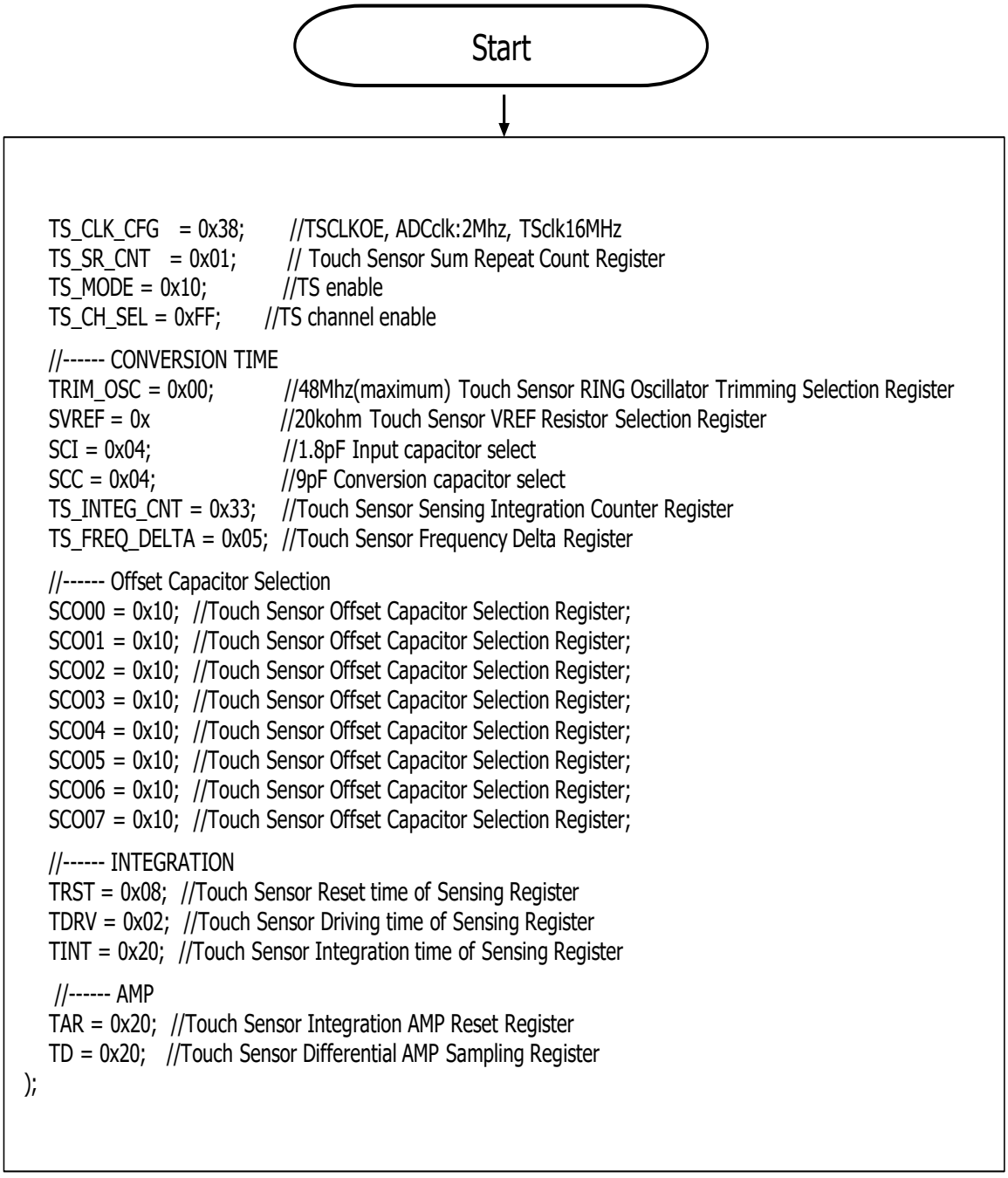


Figure 11-22 User Programming Procedure

11.7 12-bit A/D Converter

11.7.1 Overview

The analog-to-digital converter(A/D) allows conversion of an analog input signal to corresponding 12-bit digital value. The A/D module has 8 analog inputs. The output of the multiplexer is the input into the converter with generates the result through successive approximation. The A/D module has 3 registers which are the A/D converter control register (TS_CON), A/D converter datahigh register (SUM_CH0H), and A/D converter datalow register (SUM_CH0L). The channels to be conveter are selected by setting A/D converter channel selection high register (ADC_CH_SEL_H) and A/D converter channel selection low register (ADC_CH_SEL_L). The register SUM_CH0H and SUM_CH0L contains the results of the A/D conversion. When the conversion is completed, the result is loaded into the SUM_CH0H and SUM_CH0L, the A/D conversion status AFLAG bit is set to '1', and the A/D interrupt is set. Durung A/D conversion, AFLAG bit read as '0'.

11.7.2 Conversin Timing

The A/D conversion process requires 1 step (1 clock edge) to convert each bit and 6 clocks to set up A/D conversion. Therefore, total of 16 clocks are required to complete a 10-bit conversion. The maximum clock frequency for A/D conversion is 4MHz , so the maximum conversion rate is:

1 clock/bit X 10 bits + set-up time(6 clocks) = 16 clocks

16 clocks X 0.25us(4MHz) = 4us (max. 250KSPS)

11.7.3 Block Diagram

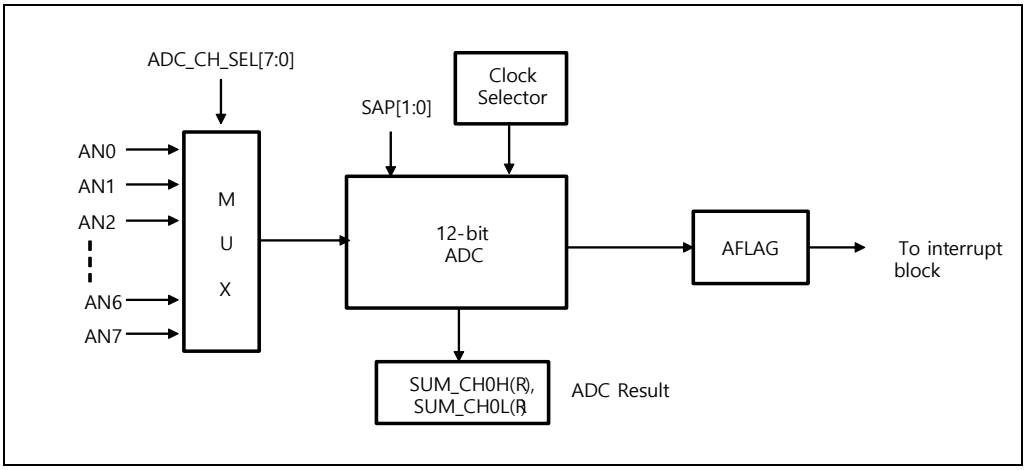


Figure 11-23 Block Diagram of A/D Converter

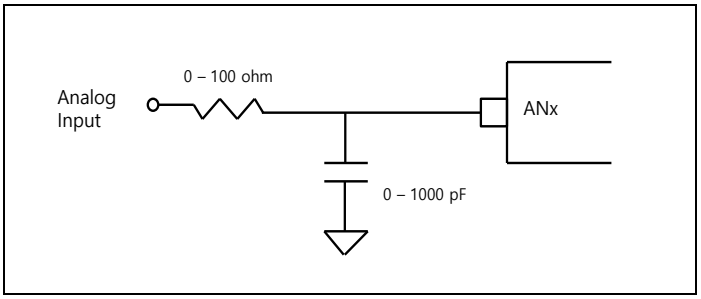


Figure 11-24 A/D Analog Input Pin with Capacitor

11.7.4 ADC Operation

To execute A/D converter, the following procedure is neeed:

1) SET ADC_CLK_CFG : Select ADC clock
2) SET TS_CON & ADC_CH_SEL: ADC enable and select AN input channel
3) Start ADC Conversion
4) If Conversion is completed, AFLAG is set to ‘1’ and ADC interrupt is occurred.
5) After conversion is completed, read SUM_CH0H and SUM_CH0L.

11.7.5 Register Map

Name	Address	Dir	Default	Description
SUM_CH0H	2E00H	R	00H	A/D Converter Data High Register
SUM_CH0L	2E01H	R	00H	A/D Converter Data Low Register
TS_CON	2E20H	R/W	00H	Touch Sensor Control Register
TS_MODE	2E21H	R/W	10H	Touch Sensor Mode Register
ADC_CH_SELH	2E25H	R/W	00H	A/D Converter Channel Selection High-Byte Register
ADC_CH_SELL	2E26H	R/W	00H-	A/D Converter Channel Selection Low-Byte Register
TS_CLK_CFG	2E2AH	R/W	20H	Touch Sensor Clock Configuration Register
TRIM_A_OSC	2E2CH	R/W	FFH	A/D Converter RING OSC. Trimming Register
LDOCR	8F	R/W	00H	LDO Control Register

11.7.6 ADC Register Description

SUM_CH0H (A/D Converter Data High Register) : 2E00H

7	6	5	4	3	2	1	0
-	-	-	-	ADCDR11	ADCDR10	ADCDR9	ADCDR8
-	-	-	-	R	R	R	R

Initial value : 00H

SUM_CH0L (A/D Converter Data Low Register) : 2E01H

7	6	5	4	3	2	1	0
ADCDR7	ADCDR6	ADCDR5	ADCDR4	ADCDR3	ADCDR2	ADCDR1	ADCDR0
R	R	R	R	R	R	R	R

Initial value : 00H

TS_CON (A/D Converter Control Register) : 2E20H

7	6	5	4	3	2	1	0
-	-	-	OSC_EN	BGR_EN	AFLAG	-	ADCEN
-	-	-	R/W	R/W	R/W	-	R/W

Initial value : 00H

OSC_EN Oscillator Enable (Do not use when AD conversion !!)

0 Oscillator Disable (Default)

1 Oscillator Enable

BGR_EN Band Gap Reference Enable (Do not use when AD conversion !!)

0 BGR Disable (Default)

1 BGR Enable

AFLAG ADC Interrupt Flag

0 No new ADC results

1 This flag indicates that the new ADC results are generated. For next ADC execution, this flag must be cleared.

ADCEN ADC Enable

0 ADC Disable (Default)

1 ADC Enable

TS_MODE (A/D Converter Mode Register) : 2E21H

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

Initial value : 10H

SAP[1:0] Touch Sensor / ADC Selection

01 Touch Sensor Select (Default)

10 ADC Select. The result of ADC is stored only at SUM_CH0 registers.

PORT[1:0] Port Configuration During Inactive Status

00 Floating

01 Low

10 High

ADC_CH_SEL_H (A/D Converter Channel Selection High-Byte Register) : 2E25H

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	CH8_SEL
-	-	-	-	-	-	-	R/W

Initial value : 00H

CH8_SEL Channel 8 Enable

0 Disable (Default)

1 Enable ADC by VDC (test only)

ADC_CH_SEL_L (A/D Converter Channel Selection Low-Byte Register) : 2E26H

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 ADC by P01

CH6_SEL Channel 6 Enable
- 0 Disable (Default)
- 1 Enable ADC by P00

CH5_SEL Channel 5 Enable
- 0 Disable (Default)
- 1 Enable ADC by P15

CH4_SEL Channel 4 Enable
- 0 Disable (Default)
- 1 Enable ADC by P11

CH3_SEL Channel 3 Enable
- 0 Disable (Default)
- 1 Enable ADC by P10

CH2_SEL Channel 2 Enable
- 0 Disable (Default)
- 1 Enable ADC by P04

CH1_SEL Channel 1 Enable
- 0 Disable (Default)
- 1 Enable ADC by P03

CH0_SEL Channel 0 Enable
- 0 Disable (Default)
- 1 Enable ADC by P02

TS_CLK_CFG (A/D Converter Clock Configuration Register) : 2E2AH

7	6	5	4	3	2	1	0
ACLKSEL	ACLKDIV2	ACLKDIV1	ACLKDIV0	-	-	-	-
R/W	R/W	R/W	R/W	-	-	-	-

Initial value : 10H

ACLKSEL ADC Clock Source Select

0 Touch Sensor Clock (Do not use when AD conversion !!)

1 System MCU Clock

ACLKDIV[2:0] ADC Clock Divider

000 Reserved (OSC_{sys} / 1)

001 Reserved (OSC_{sys} / 2, default)

010 Reserved (OSC_{sys} / 4)

011 OSC_{sys} / 8

100 Reserved (OSC_{sys} / 16)

101 Reserved (OSC_{sys} / 32)

110 Reserved (OSC_{sys} / 64)

111 Reserved (OSC_{sys} / 128)

TRIM_A_OSC (Touch Sensor RING OSC. Trimming for ADC Register) : 2E2CH

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

Initial value : FFH

TRIM_A_OSC[7:0] Touch Sensor RING OSC. Trimming for ADC Selection

00H 40MHz (maximum)

…

B8H 18.6MHz (typical)

…

FFH 3.2MHz (minimum, default)

LDOCR (LDO Control Register) : 8FH

7	6	5	4	3	2	1	0
-	-	-	-	-	MODESEL	LEVELSEL	LDODIS
-	-	-	-	-	R/W	R/W	R/W

Initial value : 00H

MODESEL LDO mode selection

0 LDO mode (default)

1 Bypass mode

LEVELSEL LDO level selection

0 3.0V (default)

1 2.5V

LDODIS LDO Disable (test only)

0 Enable (default)

1 Disable

12. Power Down Operation

12.1 Overview

The A96T346 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 (128kHz)	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	By RESET, External Interrupt, I^2C (slave mode)

12.3 IDLE mode

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

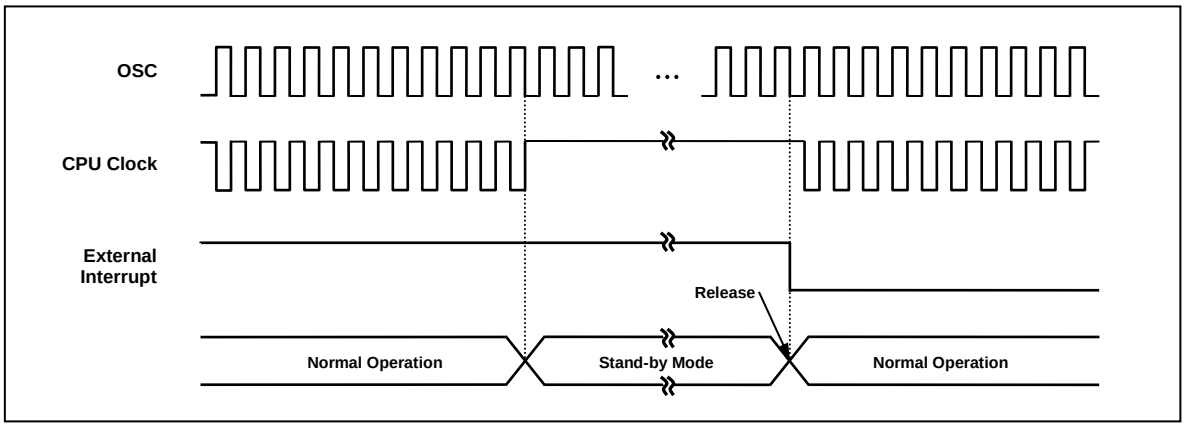


Figure 12-1 IDLE Mode Release Timing by External Interrupt

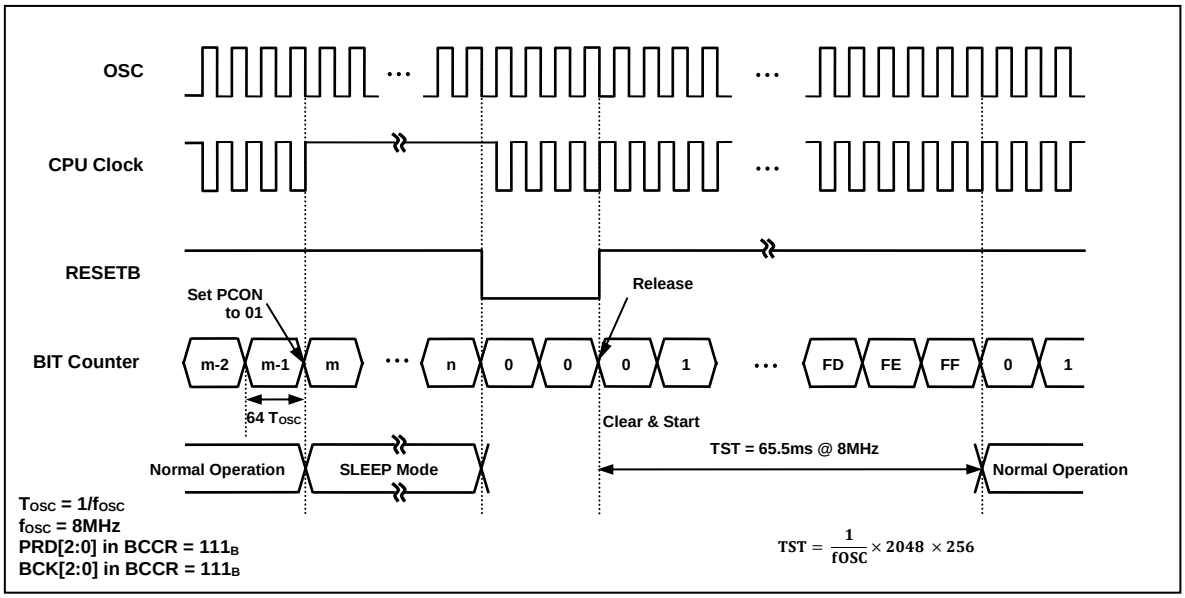


Figure 12-2 IDLE Mode Release Timing by RESETB

(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.

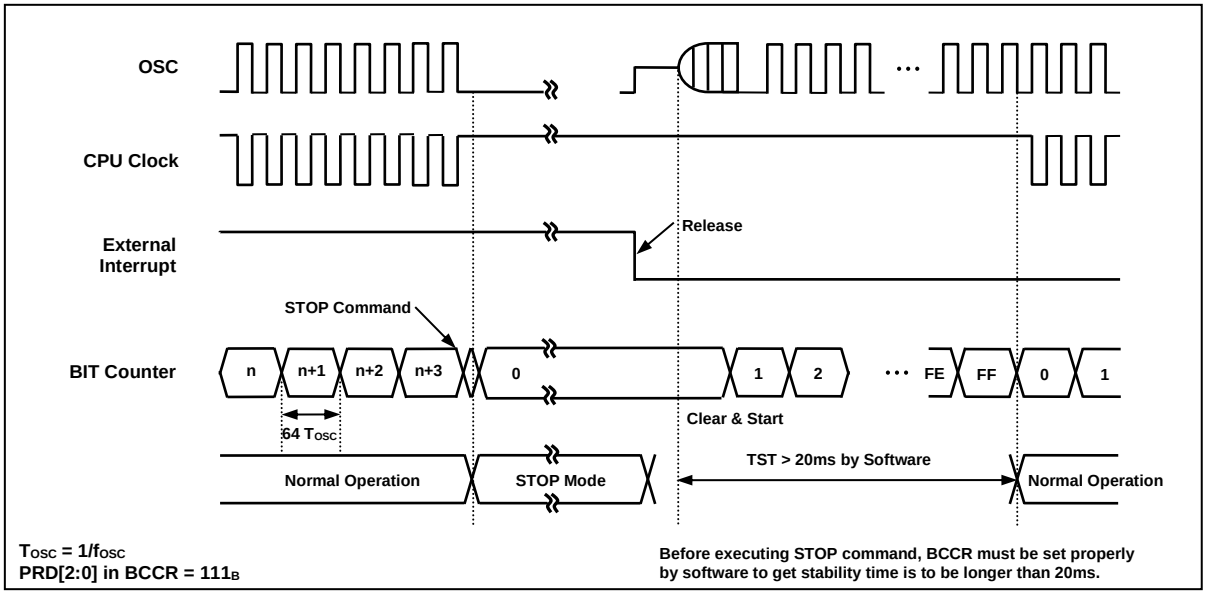


Figure 12-3 STOP Mode Release Timing by External Interrupt

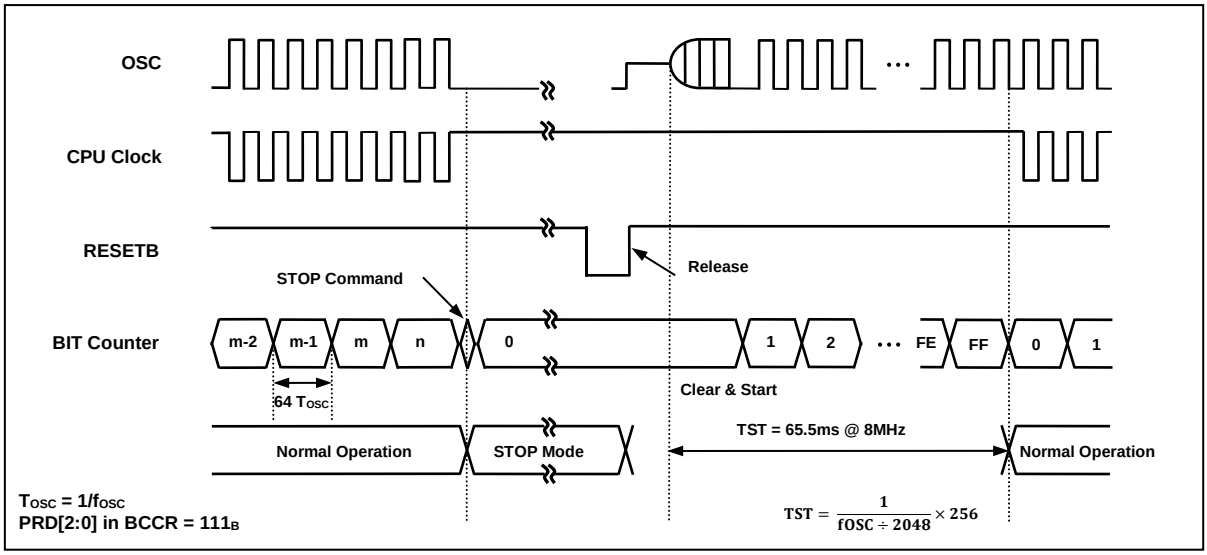


Figure 12-4 STOP Mode Release Timing by RESETB

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.

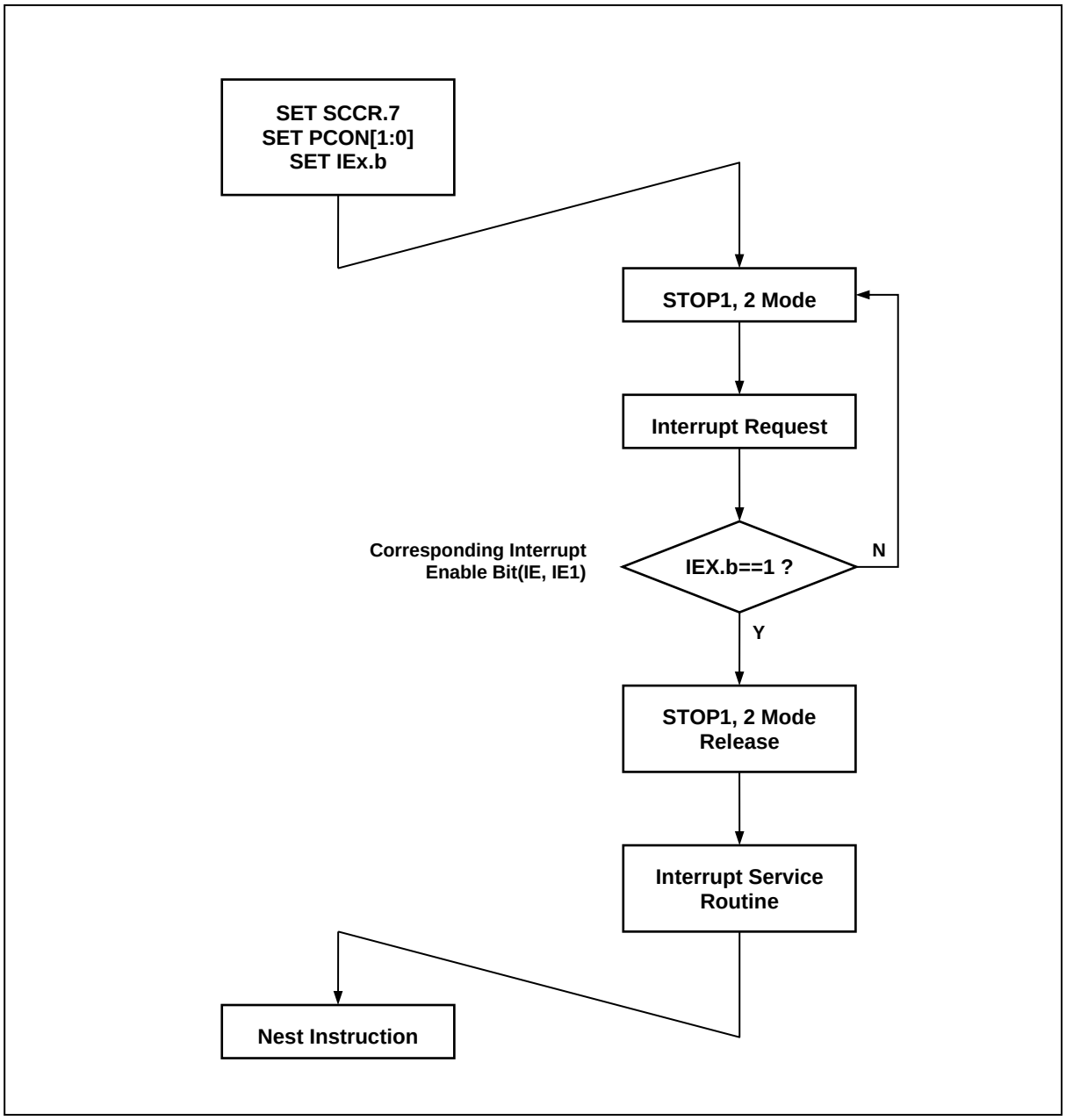


Figure 12-5 STOP1, 2 Mode Release Flow

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 128kHz-OSC during STOP mode operating.

13. RESET

13.1 Overview

The A96T346 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 A96T346 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 `)
- OCD Reset

13.3 Block Diagram

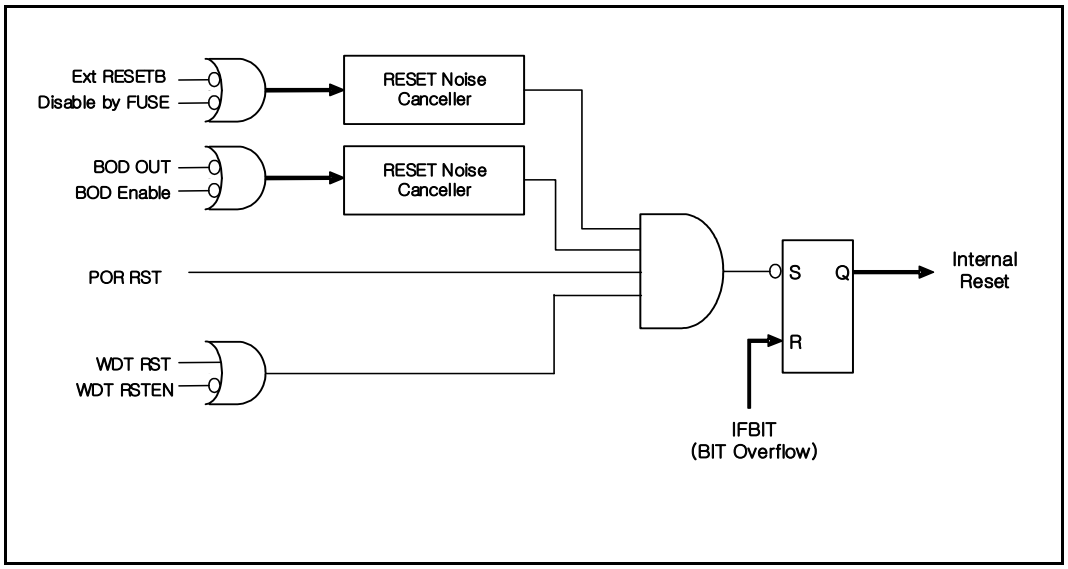


Figure 13-1 RESET Block Diagram

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 40~24us (@V_{DD}=3V) to the low input of System Reset.

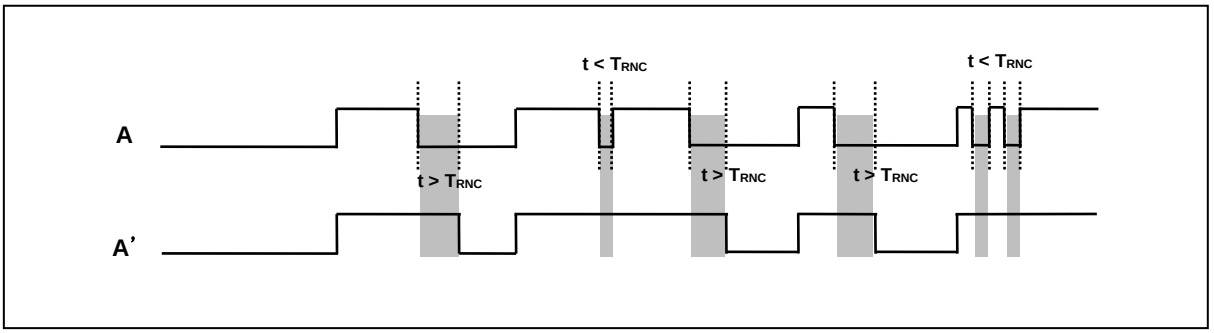


Figure 13-2 Reset Noise Canceller Time Diagram

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.

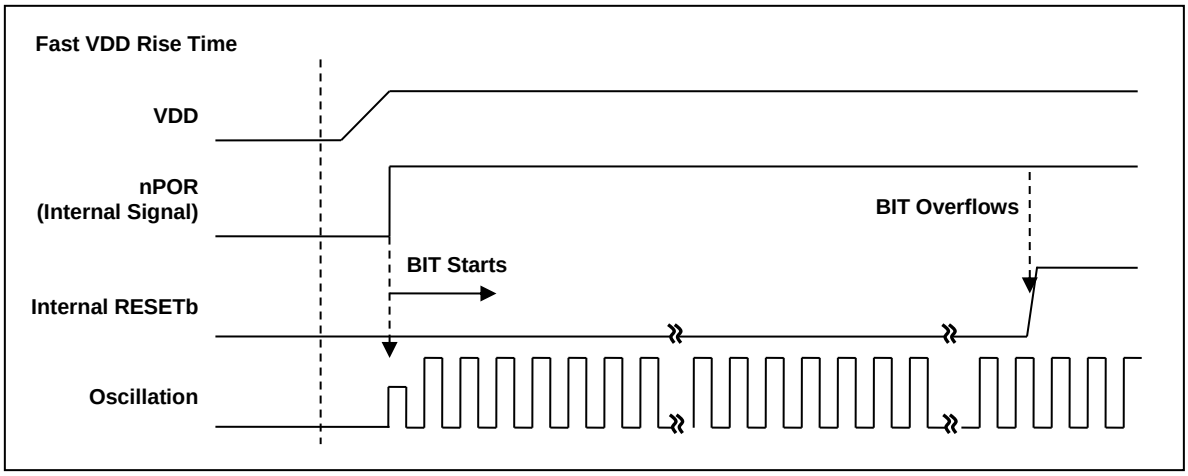


Figure 13-3 Fast VDD Rising Time

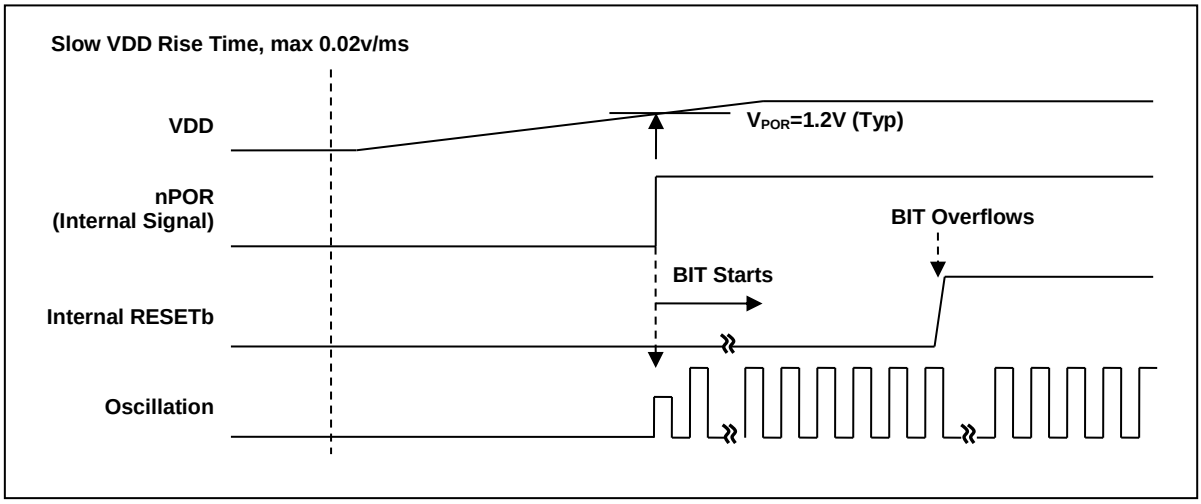


Figure 13-4 Internal RESET Release Timing on Power-Up

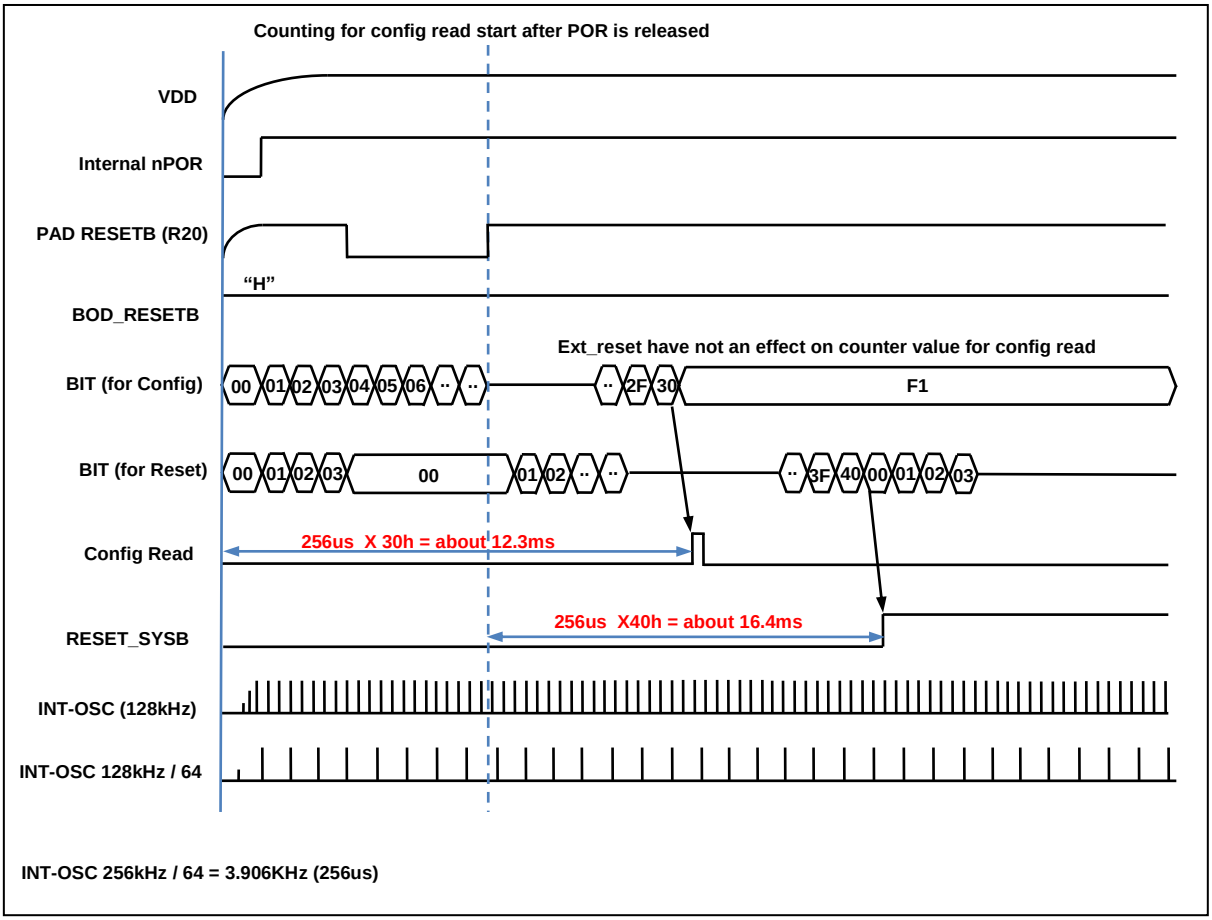


Figure 13-5 Configuration Timing when Power-on

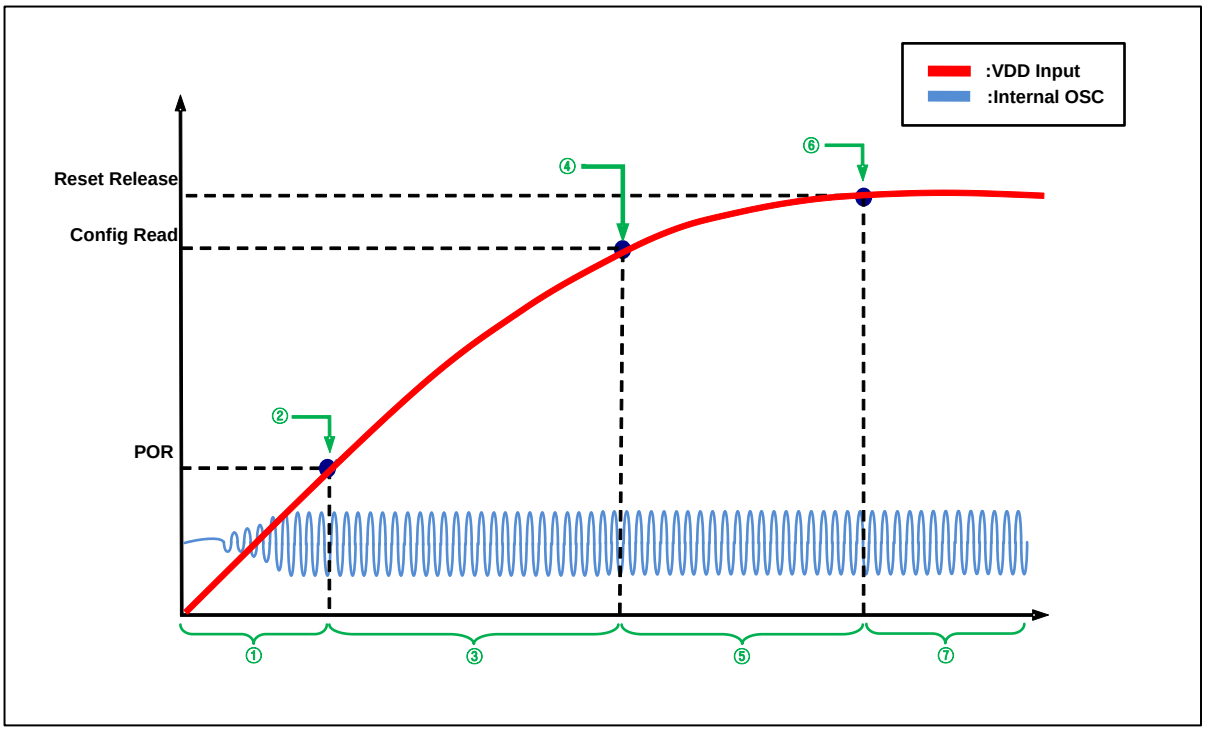


Figure 13-6 Boot Process Waveform

Table 13-2 Boot Process Description

Process	Description	Remarks
①	No operation Internal OSC (128kHz) ON	- 0.7V~0.9V
②	1st POR level Detection	- 1.1V~1.3V
③	(INT-OSC 128kHz/32)×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	- 1.6V ~ 1.8V -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 50us 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.

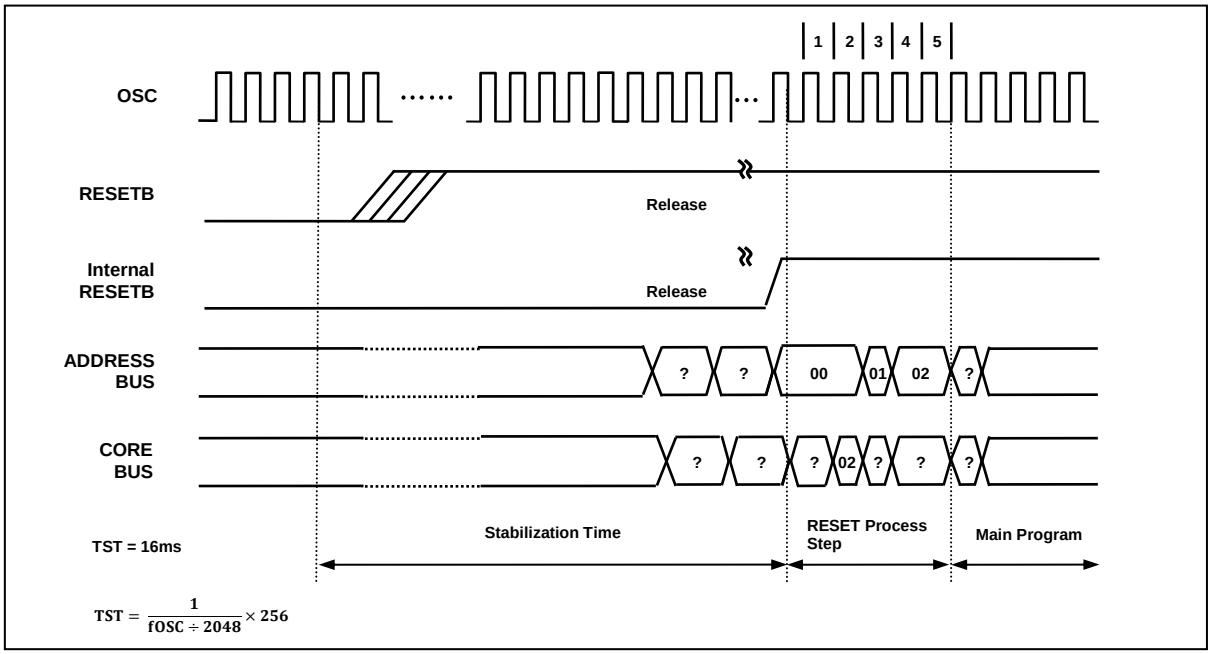


Figure 13-7 Timing Diagram after RESET

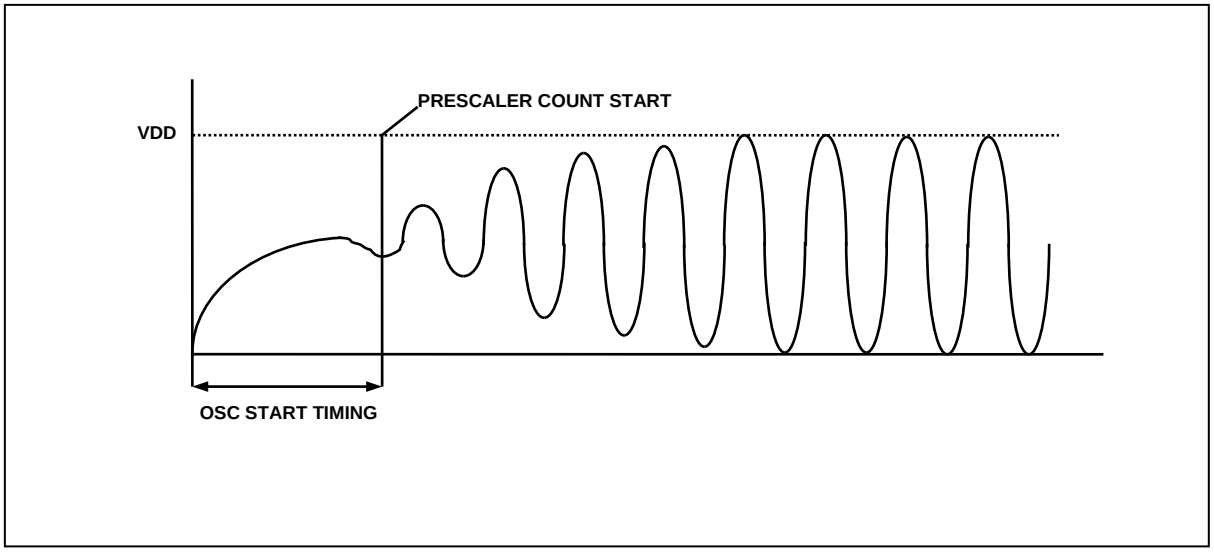


Figure 13-8 Oscillator Generating Waveform Example

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 A96T346 has an On-chip Brown-out detection. In the STOP mode, this will contribute significantly to the total current consumption. So to minimize the current consumption, the BOD is set to off by hardware.

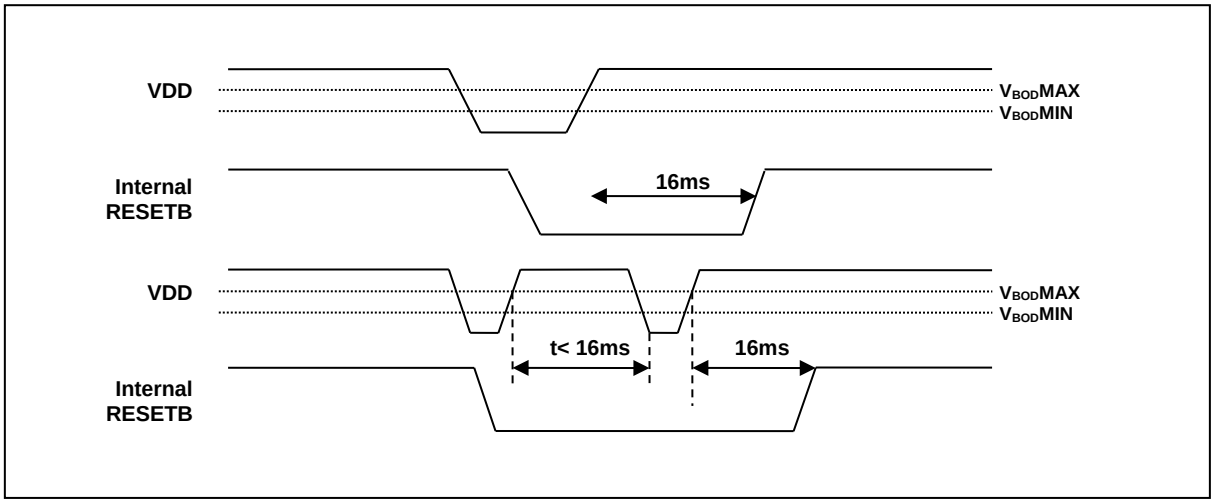


Figure 13-9 Internal Reset at the Power Fail Situation

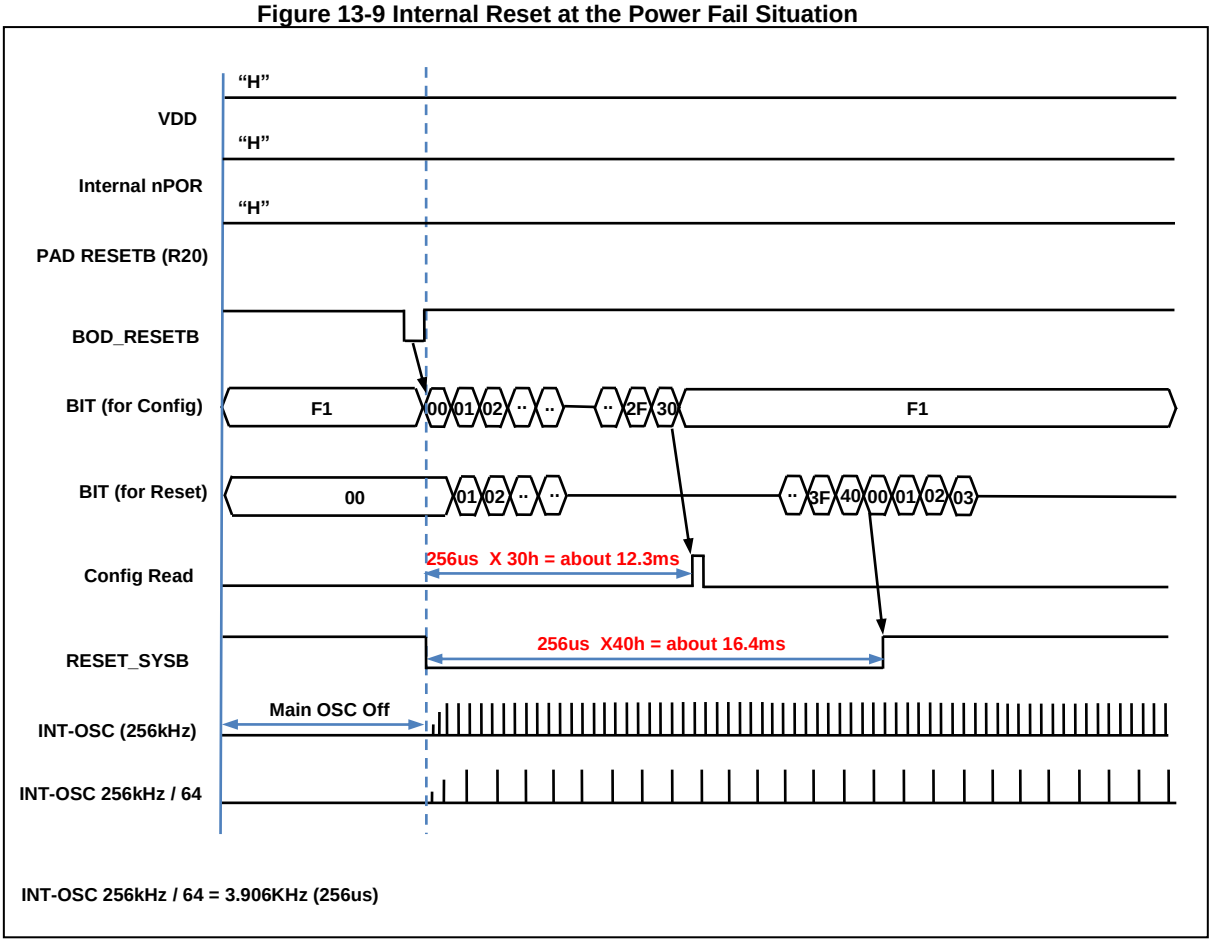


Figure 13-10 Configuration Timing when BOD RESET

13.7.1 Register Map

Table 13-3 Register Map

Name	Address	Dir	Default	Description
BODR	86H	R/W	80H	BOD status Register

13.7.2 Reset Operation Register Description

Reset Flag Register consists of the BOD status Register (BODR).

13.7.3 Register Description for Reset Operation

BODR (BOD Status Register) : 86H

7	6	5	4	3	2	1	0
PORF	EXTRF	WDTRF	OCDRF	BODRF	-	-	-
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

14. On-chip Debug System

14.1 Overview

14.1.1 Description

On-chip debug System (OCD) of A96T346 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

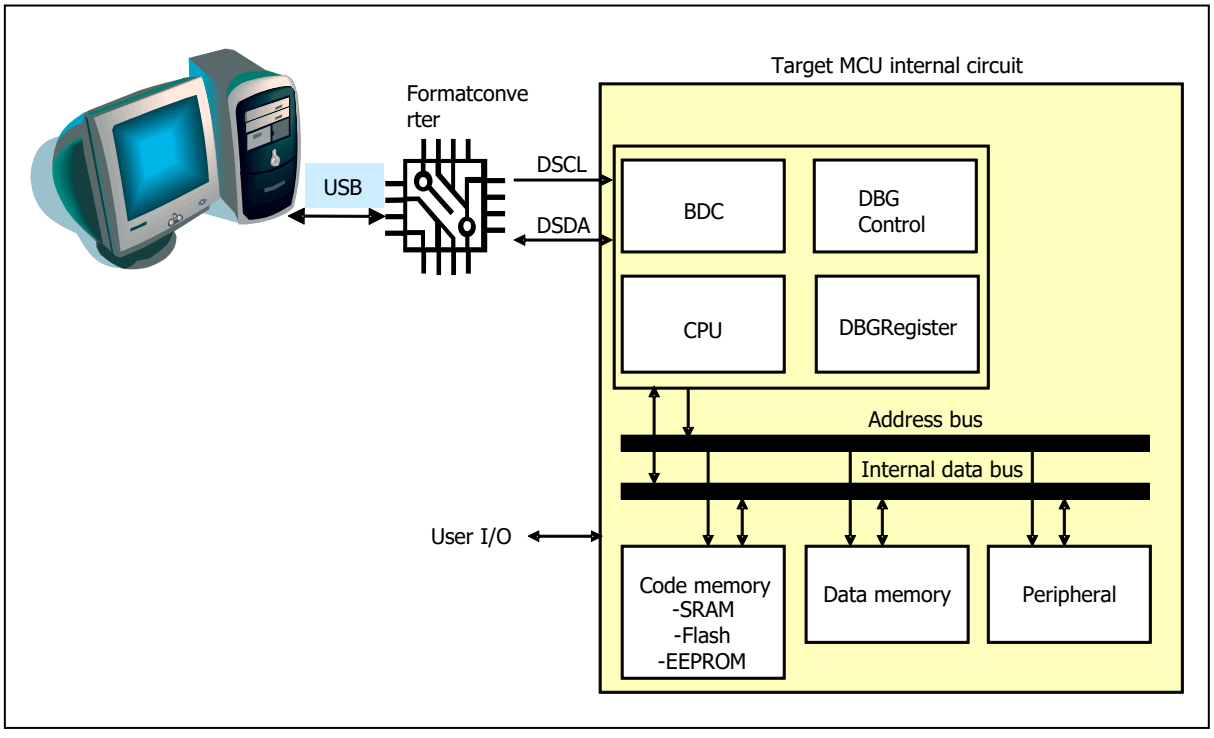


Figure 14-1 Block Diagram of On-chip Debug System

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.

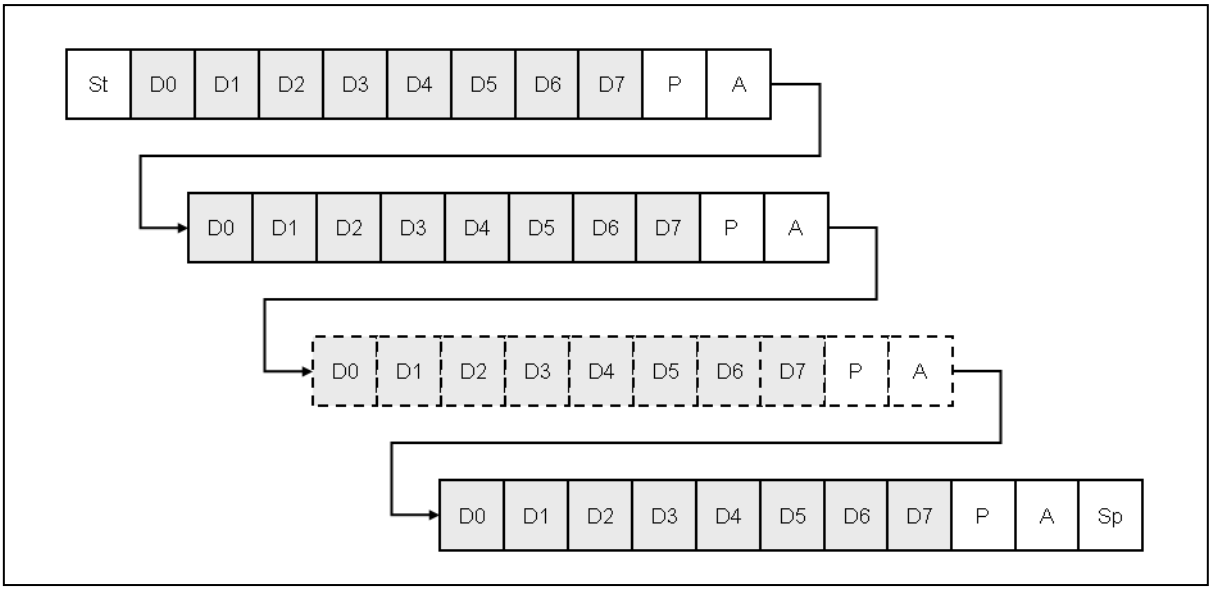


Figure 14-2 10-bit Transmission Packet

14.2.2 Packet Transmission Timing

14.2.2.1 Data transfer

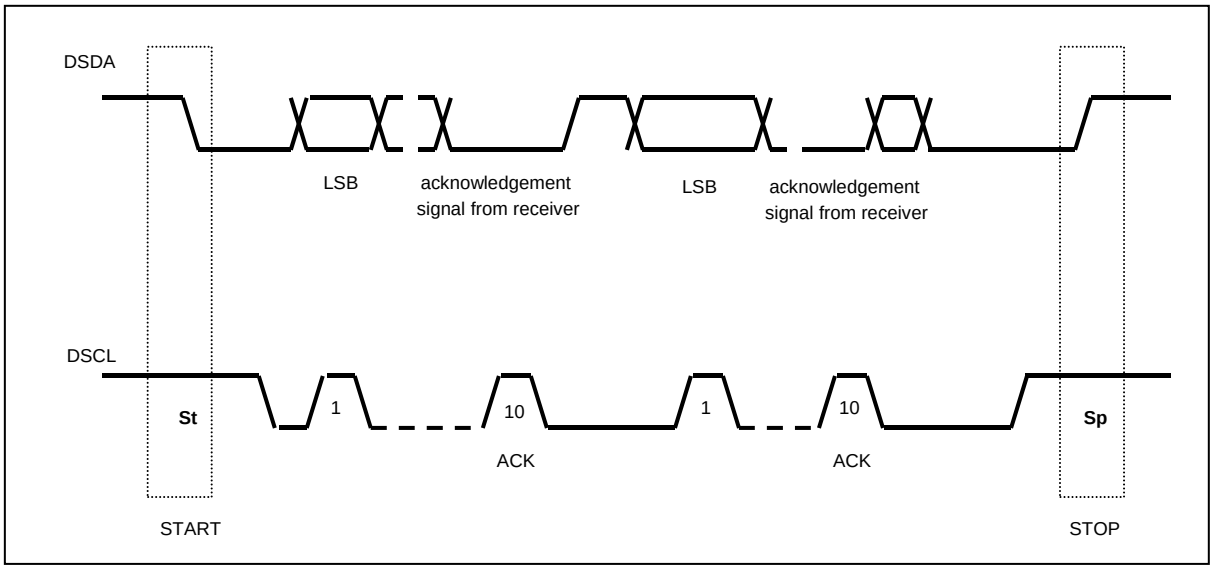


Figure 14-3 Data Transfer on the Twin Bus

14.2.2.2 Bit transfer

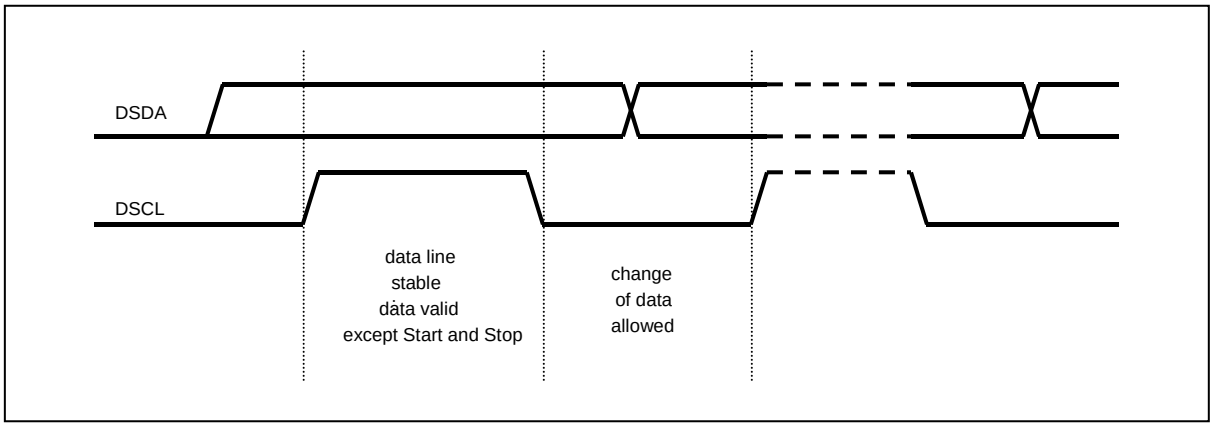


Figure 14-4 Bit Transfer on the Serial Bus

14.2.2.3 Start and stop condition

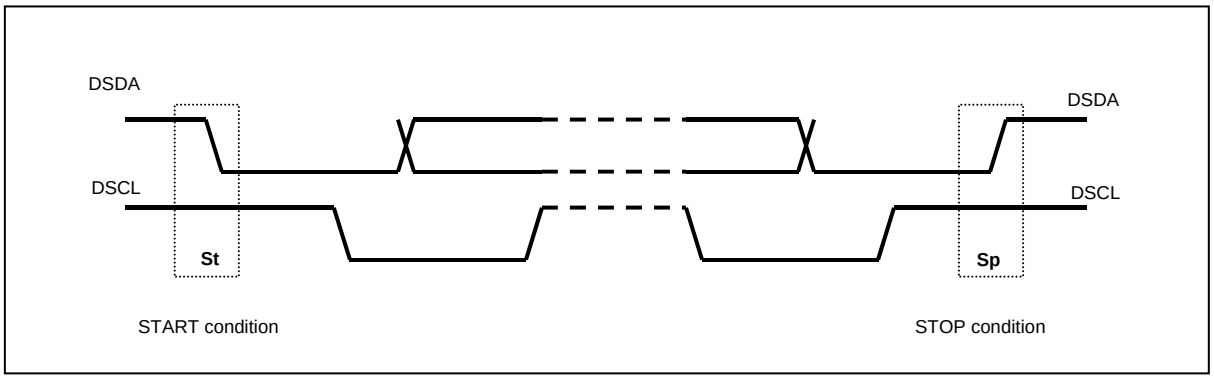


Figure 14-5 Start and Stop Condition

14.2.2.4 Acknowledge bit

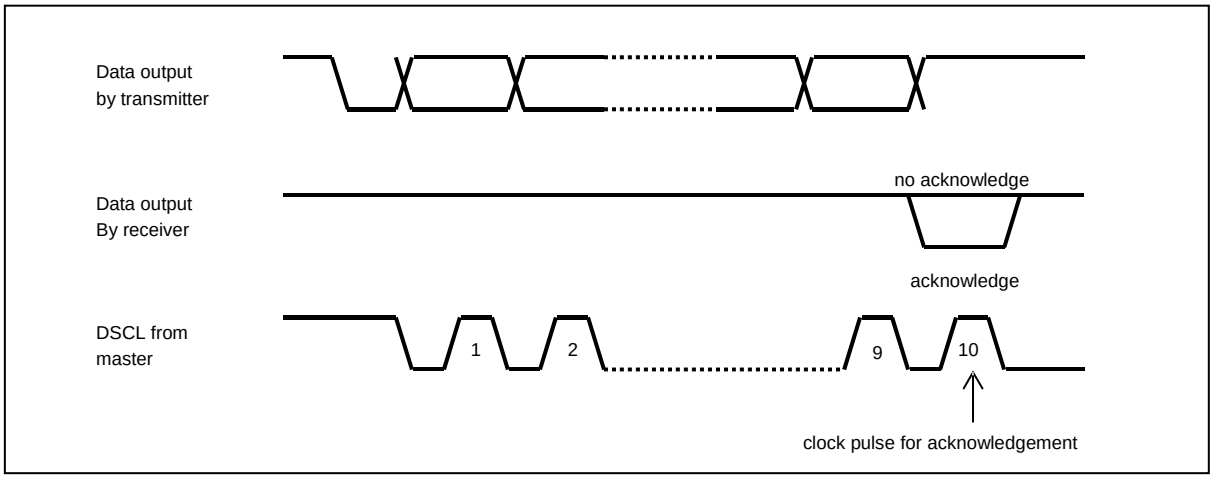


Figure 14-6 Acknowledge on the Serial Bus

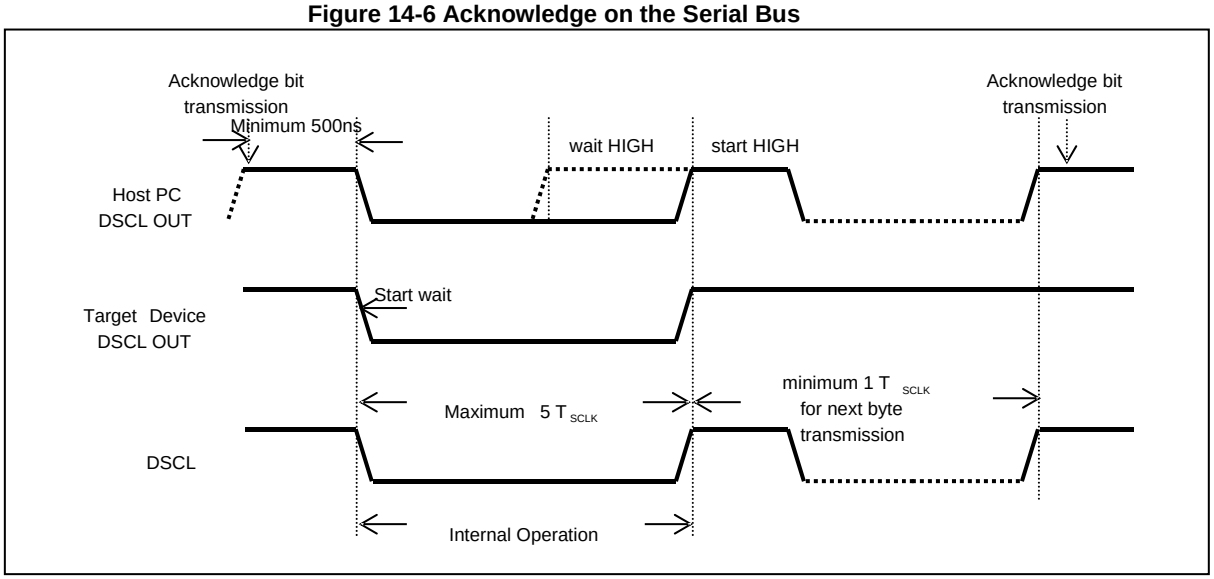


Figure 14-7 Clock Synchronization during Wait Procedure

14.2.3 Connection of Transmission

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

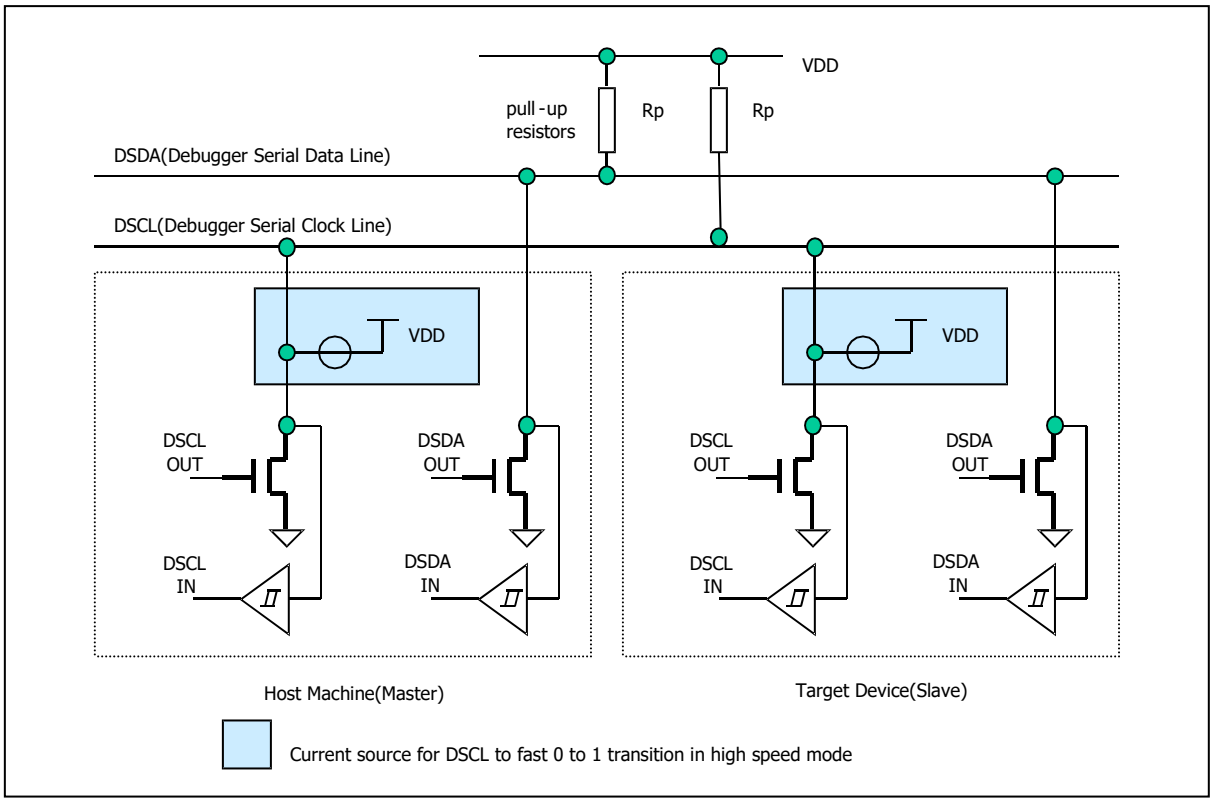


Figure 14-8 Connection of Transmission

15. Memory Programming

15.1 Overview

15.1.1 Description

A96T346 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
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)

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	REMAP	-	-	ROMINT	WMODE	EMODE	VMODE
R	R/W	R	R	R/W	R	R	R

Initial value : 80H

PEVBSY Operation status flag. It is cleared automatically when operation starts. Operations are program, erase or verification

0 Busy (Operation processing)

1 Complete Operation

REMAP Remap enable register for OTP.

0 Disable (default)

1 Enable

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

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 result.

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

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 CheckSum within 10’s ~ 100’s of milliseconds. It is 100’s times faster than normal data read.

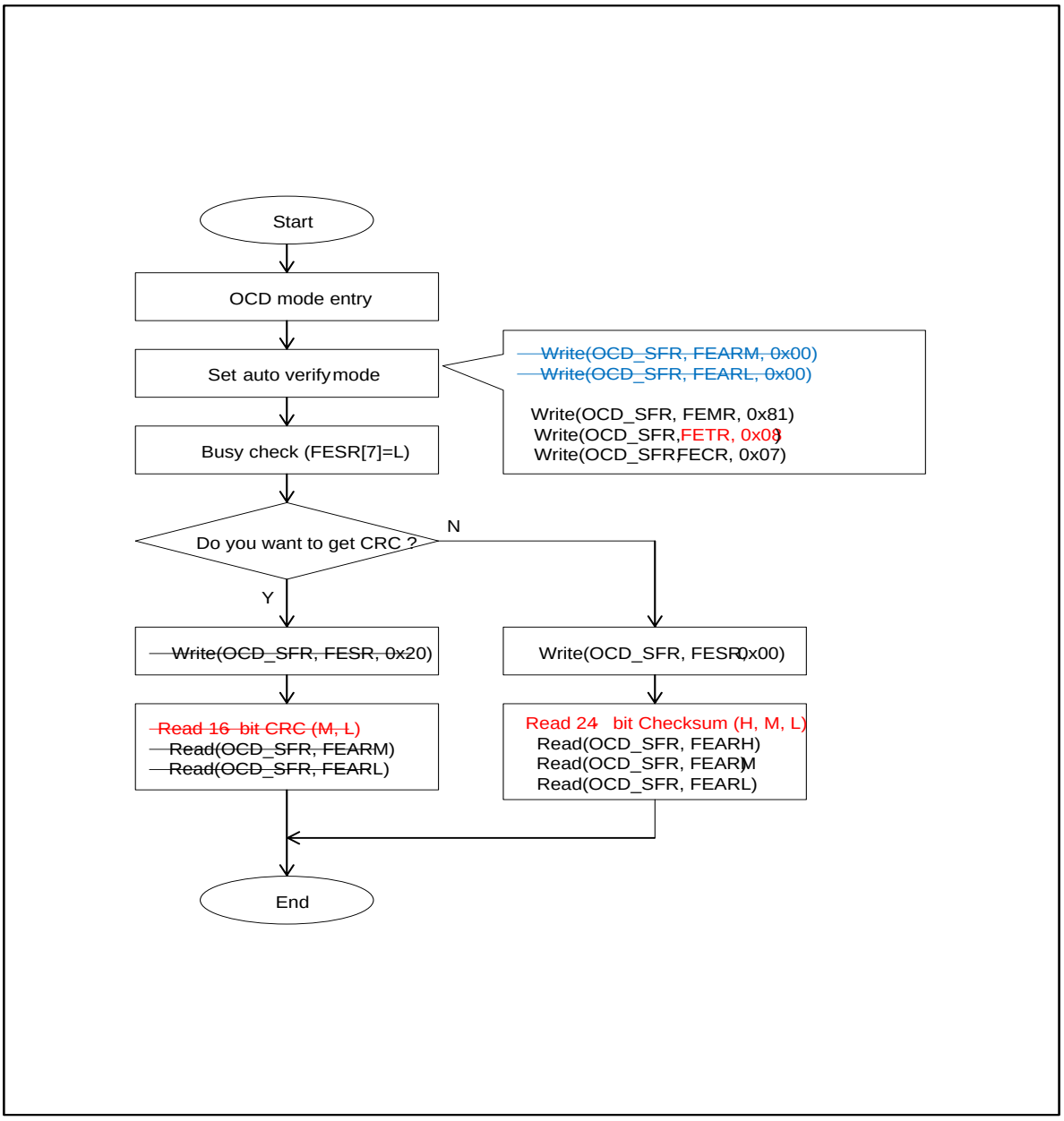


Figure 15-1 Read device internal CkeckSum (Full size)

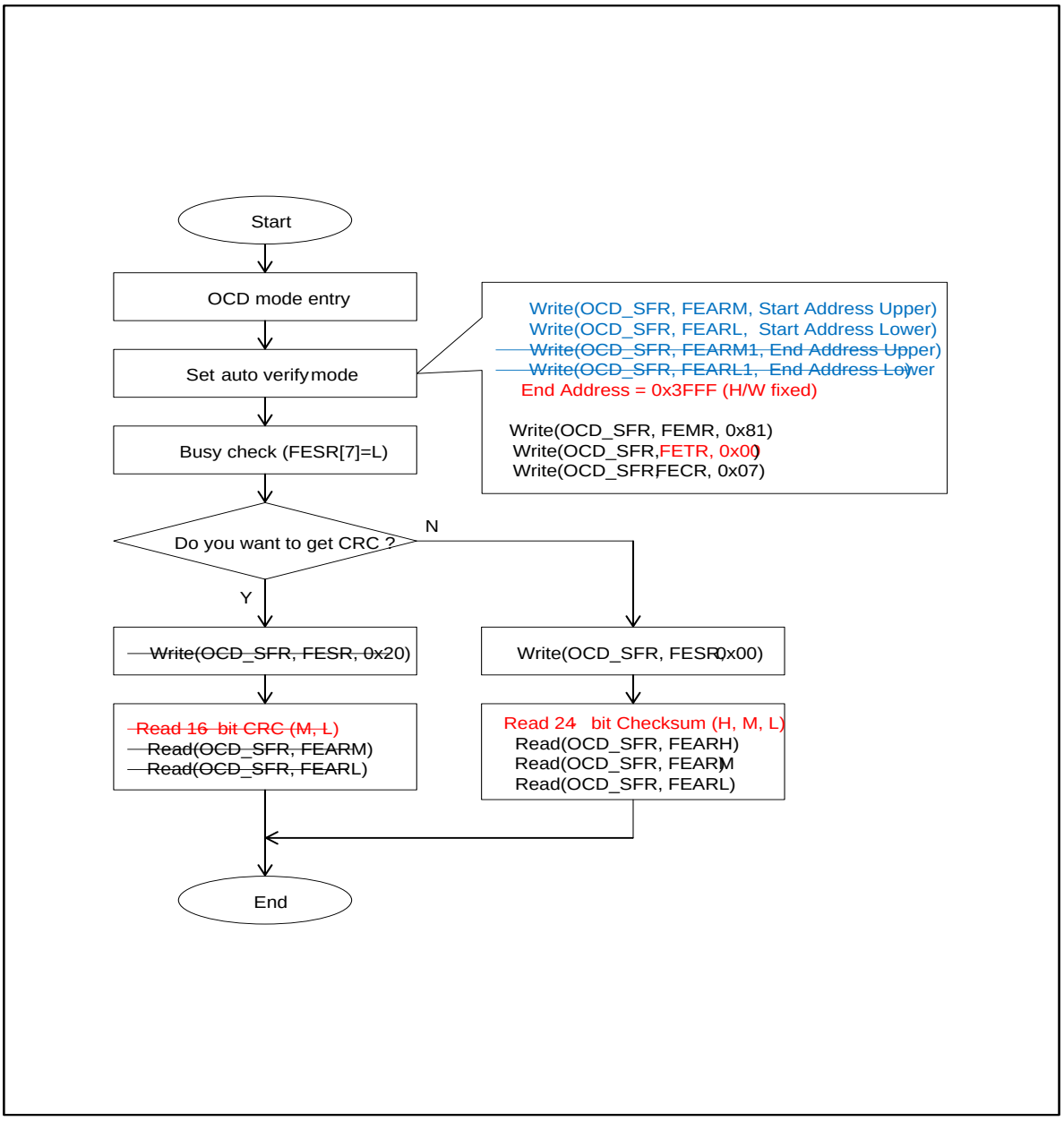


Figure 15-2 Read device internal CkeckSum (User define size)

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

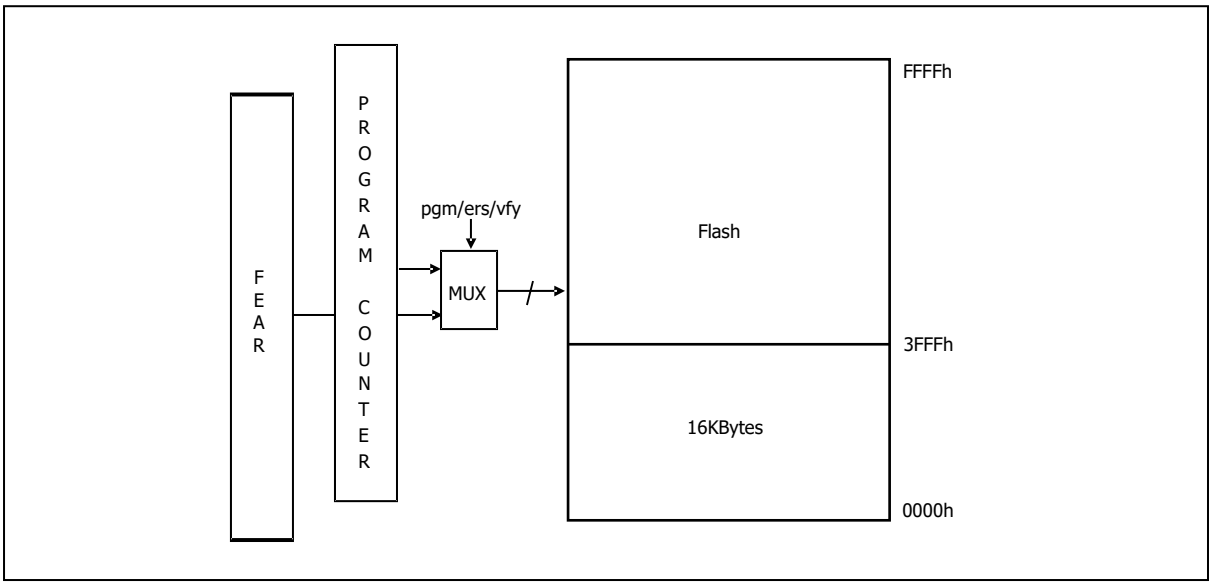


Figure 15-3 Flash Memory Map

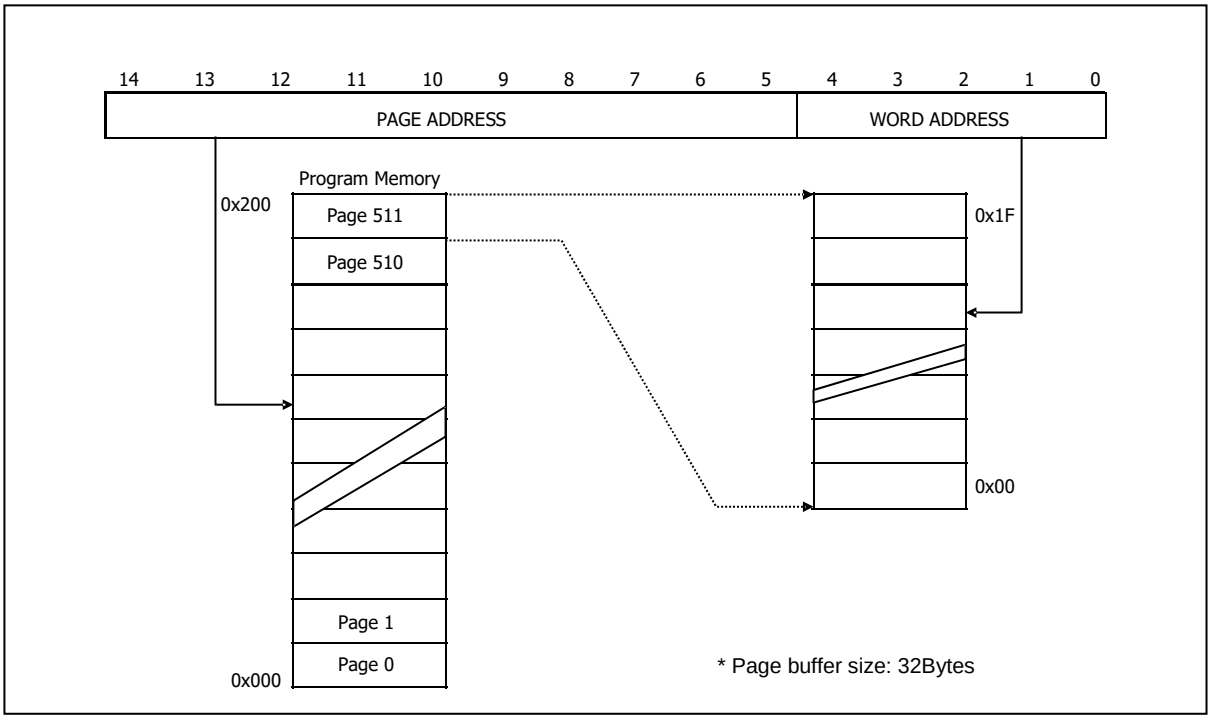


Figure 15-4 Address Configuration of Flash Memory

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

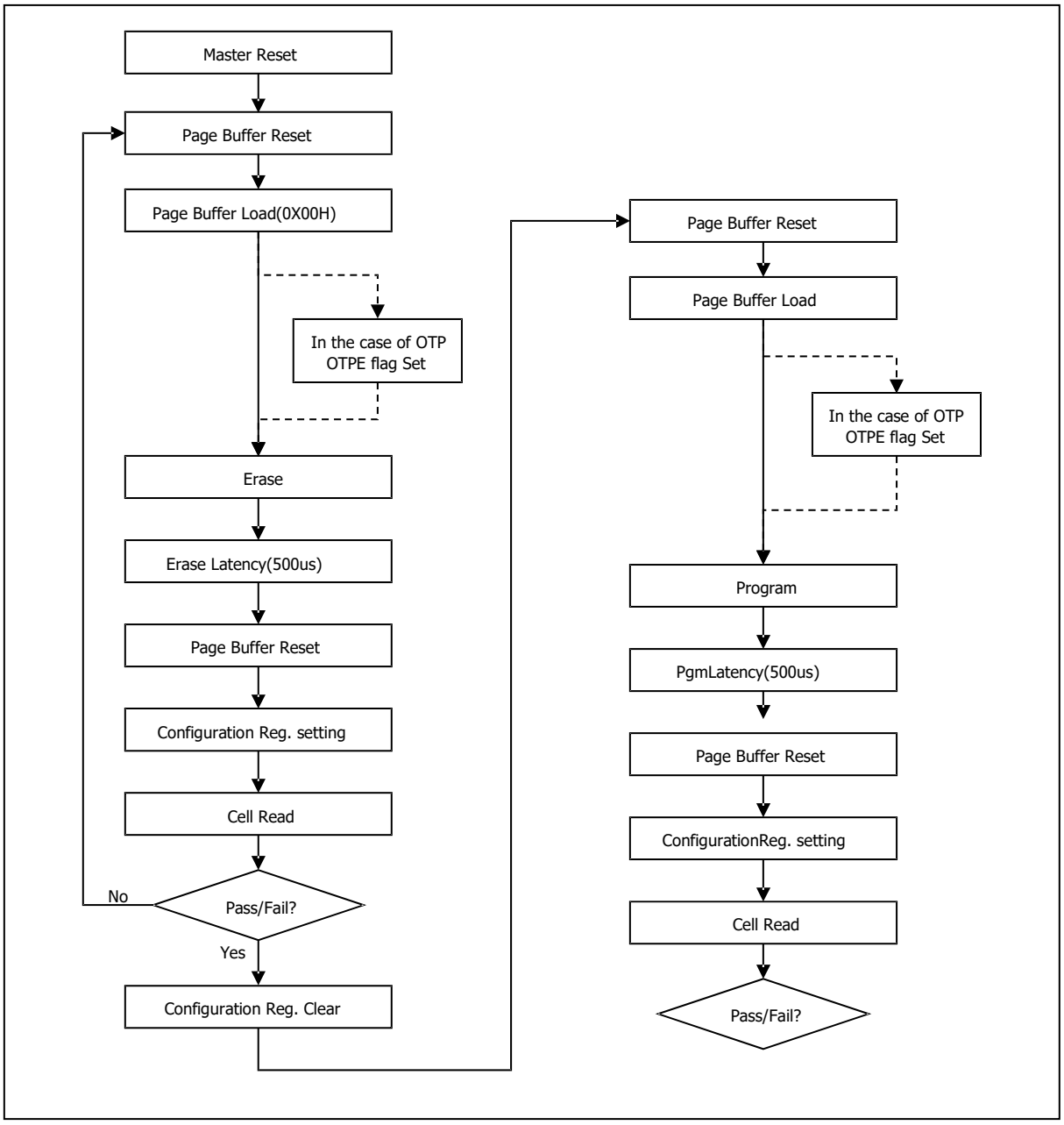


Figure 15-5 The Sequence of Page Program and Erase of Flash Memory

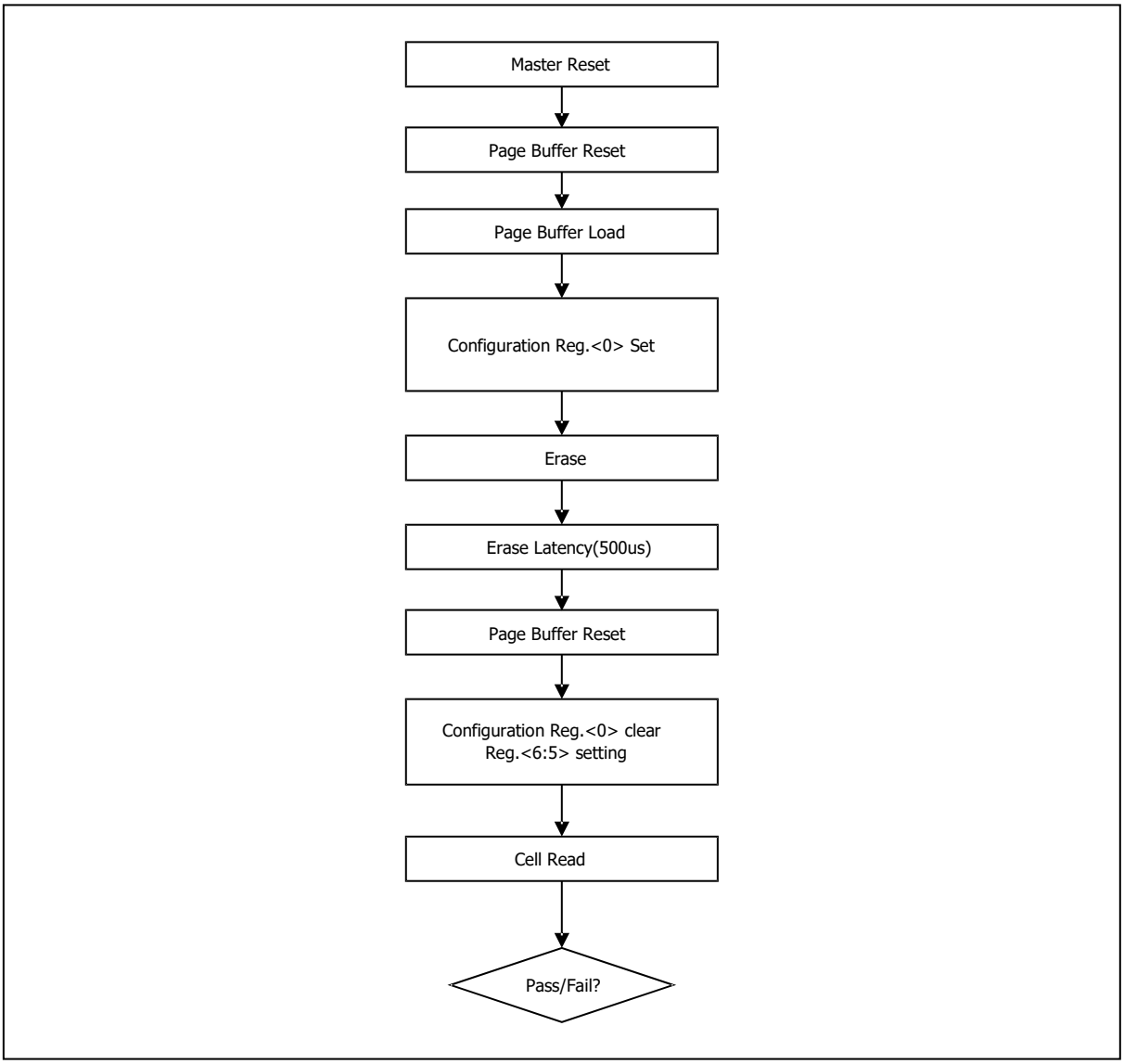


Figure 15-6 The Sequence of Bulk Erase of Flash Memory

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

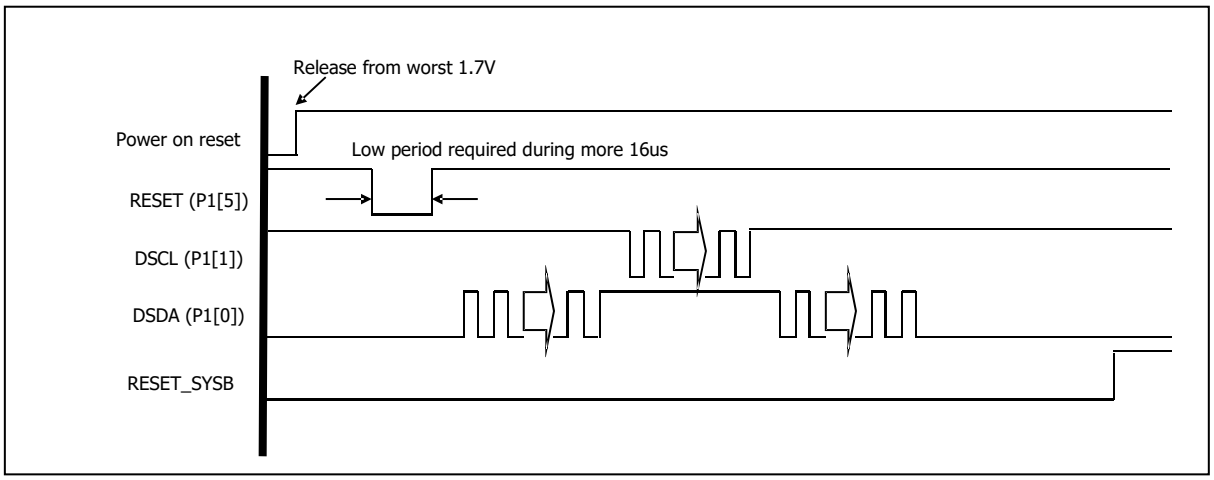


Figure 15-7 ISP Mode

15.6 Security

A96T346 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~3FFFh (16KB)

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.

Table of Contents

List of Figures