A96G174/A96S174
User's Manual

16 MHz 8-bit A96G174 Microcontroller
8 Kbyte Flash memory, 12-bit ADC, 3 Timers, USART, I2C, Window WDT

User's Manual Version 1.13

Introduction

This user's manual targets application developers who use A96G174/A96S174 for their specific needs. It provides complete information of how to use A96G174/A96S174 device. Standard functions and blocks including corresponding register information of A96G174/A96S174 are introduced in each chapter, while instruction set is in Appendix.

A96G174/A96S174 is based on M8051 core, and provides standard features of 8051 such as 8-bit ALU, PC, 8-bit registers, timers and counters, serial data communication, PSW, DPTR, SP, 8-bit data bus and 2x16-bit address bus, and 8/11/16-bit operations.

In addition, this device incorporates followings to offer highly flexible and cost-effective solutions:

- 8Kbytes of FLASH, 256bytes of IRAM, and 256bytes of XRAM
- Basic interval timer, watchdog timer, and 8/16-bit timer/counter
- 16-bit PPG output, 8-bit PWM output, 16-bit PWM output, USART, I2C, and 12-bit ADC
- On-chip POR, LVR, LVI
- On-chip oscillator and clock circuitry.

As a field proven best seller, A96G174/A96S174 introduces rich features such as excellent noise immunity, code optimization, cost effectiveness, and so on.

Reference document

- A96G174/A96S174 programming tools and manuals released by ABOV: They are available at ABOV website, www.abovsemi.com.
- SDK-51 User's guide (System Design Kit) released by Intel in 1982: It contains all of components of a single-board computer based on Intel's 8051 single-chip microcomputer
- Information on Mentor Graphics 8051 microcontroller: The technical document is provided at Mentor® website: https://www.mentor.com/products/ip/peripheral/microcontroller/

Contents

List of figures

List of tables

1 Description

A96G174/A96S174 is an advanced CMOS 8-bit microcontroller with 8Kbytes of FLASH. This is a powerful microcontroller which provides a highly flexible and cost effective solution to many embedded control applications.

1.1 Device overview

In this section, features of A96G174/A96S174 and peripheral counts are introduced.

Table 1. A96G174/A96S174 Device Features and Peripheral Counts

Peripherals		Description
Core	CPU	8-bit CISC core (M8051, 2 clocks per cycle)
	Interrupt	Up to 14 peripheral interrupts supported. • EINT0 to 2, PCI (4) • Timer (0/1/2) (3) • WDT (1) • BIT (1) • USART Rx/Tx (2) • I2C (1) • ADC (1) • LVI (1)
Memory	ROM (FLASH) capacity	• 8Kbytes FLASH with self-read and write capability • In-system programming (ISP) • Endurance: 30,000times
	IRAM	256Bytes
	XRAM	256Bytes
Programmable pulse generation		• Pulse generation (by T1/T2) • 8-bit PWM (by T0) • 16-bit Complementary PWM (by T1, Dead time)
Minimum instruction execution time		• 125ns (@ 16MHz main clock) • 61us (@ 32.768kHz sub clock)
Power down mode		• STOP mode • IDLE mode
General Purpose I/O (GPIO)		• Normal I/O: 18ports

Table 1. A96G174/A96S174 Device Features and Peripheral Counts (continued)

Peripherals		Description
Reset	Power on reset	• Reset release level: 1.32V
	Low voltage reset	• 5 levels detect • 1.61/1.77/2.13/2.46/3.56V
Low voltage indicator		• 3 levels detect • 1.77/2.13/2.46/3.56V
Timer/counter		• Basic interval timer (BIT) 8-bit x 1-ch. • Window Watch Dog Timer (WWDT) 8-bit x 1-ch. • 8-bit x 1-ch (T0), 16-bit x 2-ch (T1/T2)
Communication function	USART	• 8-bit USART x 1-ch or 8-bit SPI x 1-ch • Receiver timer out (RTO) • 0% error baud rate
	I2C	• Compatible with I2C bus standard • Up to 400kHz
12-bit A/D converter		• 15 input channels
Internal RC oscillator		• HSIRC 32MHz ±1.5% (T_A= 0~ +50°C) • HSIRC 32MHz ±2.0% (T_A=-10~ +70°C) • HSIRC 32MHz ±2.5% (T_A=-40~ +85°C) • HSIRC 32MHz ±5.0% (T_A=-40~ +105°C) • LSIRC 128kHz ±20% (TA= -40~ +85°C) • LSIRC 128kHz ±30% (TA= -40~ +105°C)
Operating voltage and frequency		• 1.8V to 5.5V @ 32.768kHz with crystal • 1.8V to 5.5V @ 0.5MHz to 16.0MHz with internal RC
Operating temperature		• -40℃ to +85, -40℃ to 105℃
Package		• Pb-free packages • 20 SOP / TSSOP / QFN • 16 SOPN

1.2 A96G174/A96S174 block diagram

In this section, A96G174/A96S174 device with peripherals is described in a block diagram.

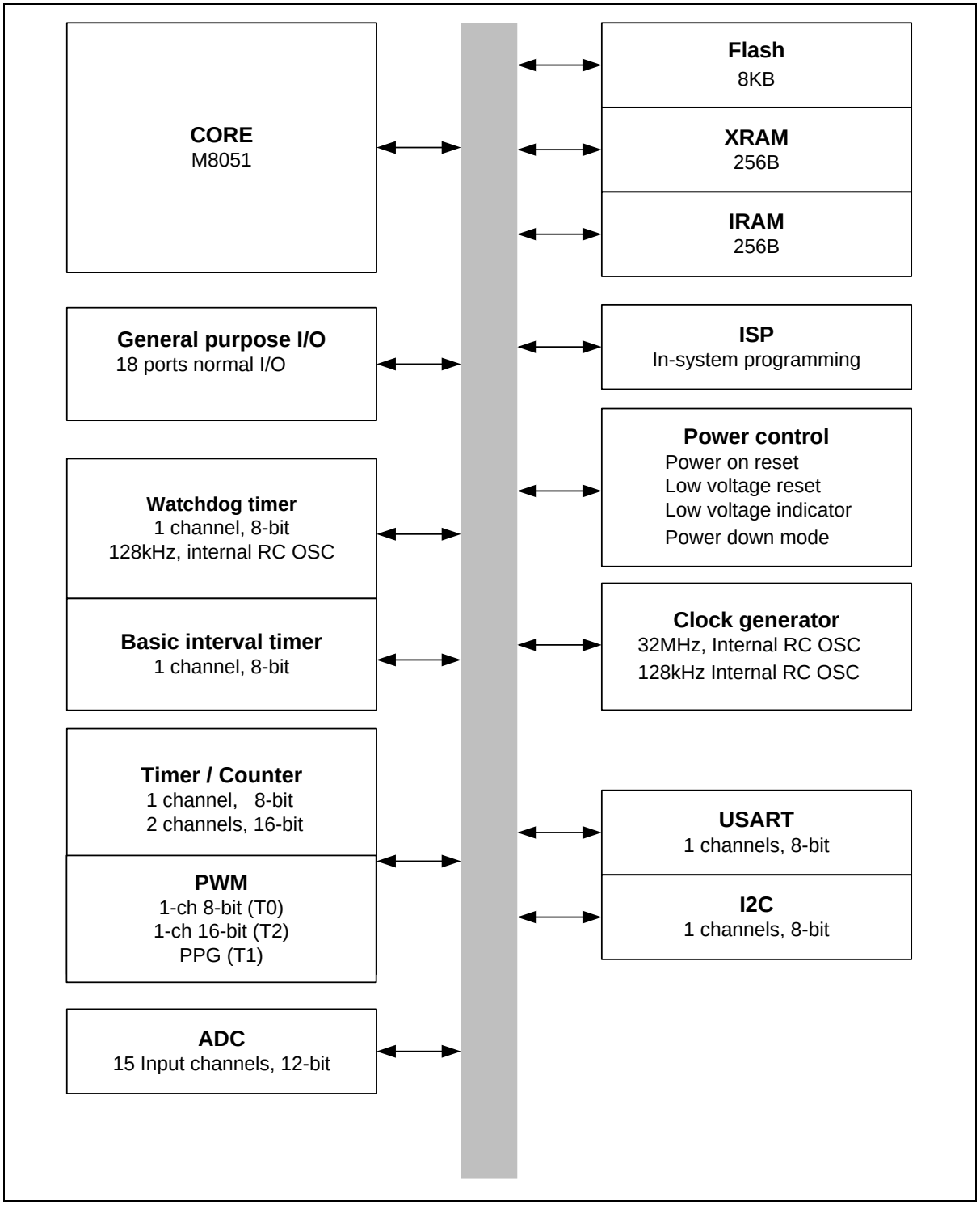
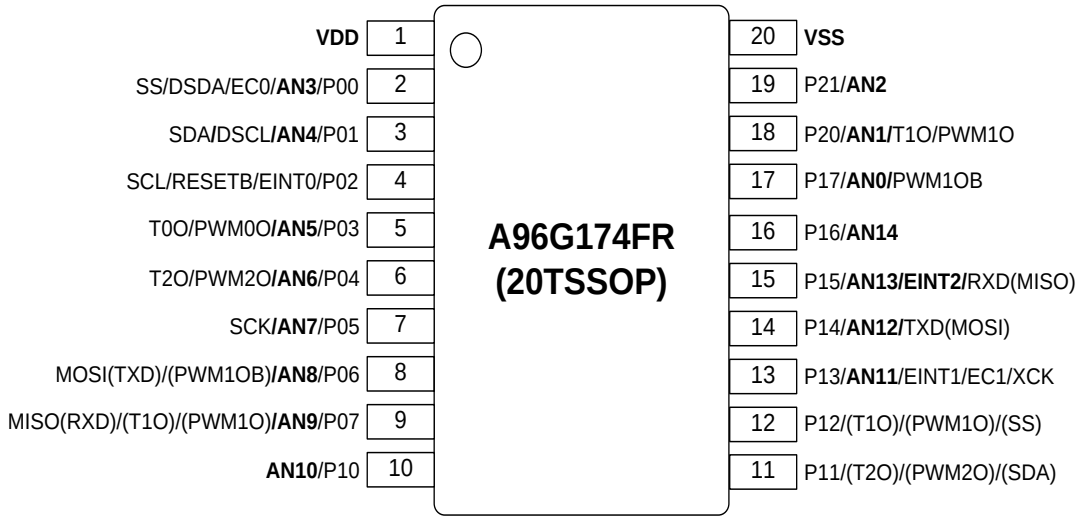


Figure 1. A96G174/A96S174 Block Diagram

2 Pinouts and pin description

Pinouts and pin descriptions of A96G174/A96S174 device are introduce in the following sections.

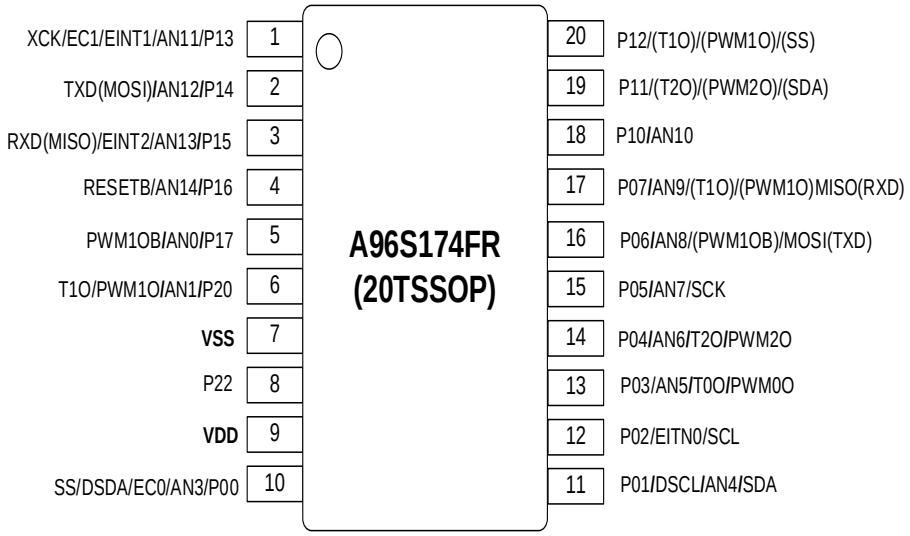
2.1 Pinouts



NOTES:

1. Programmer (E-PGM+, E-Gang4/6) uses P0[1:0] pin as DSCL, DSDA.
2. The P22 pins should be selected as a push-pull output or an input with pull-up resistor by software control when the 20-pin package is used.

Figure 2. A96G174 20TSSOP Pin Assignment

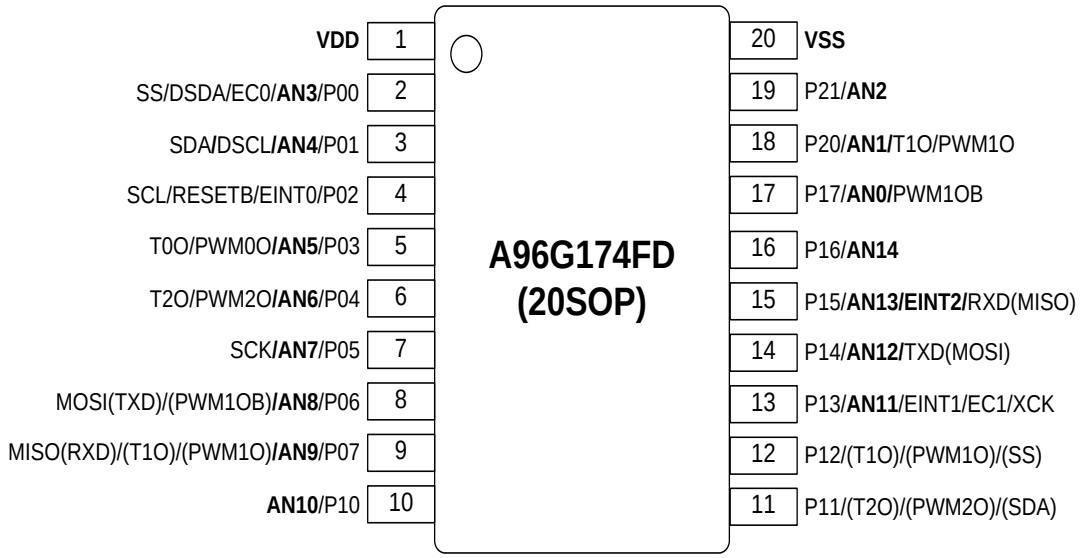


NOTES:

1. The programmer (E-PGM+, E-Gang4/6) uses P0[1:0] pin as DSCL, DSDA.
2. The P21 pins should be selected as a push-pull output or an input with pull-up resistor by software control when the 20-pin package is used.

3.

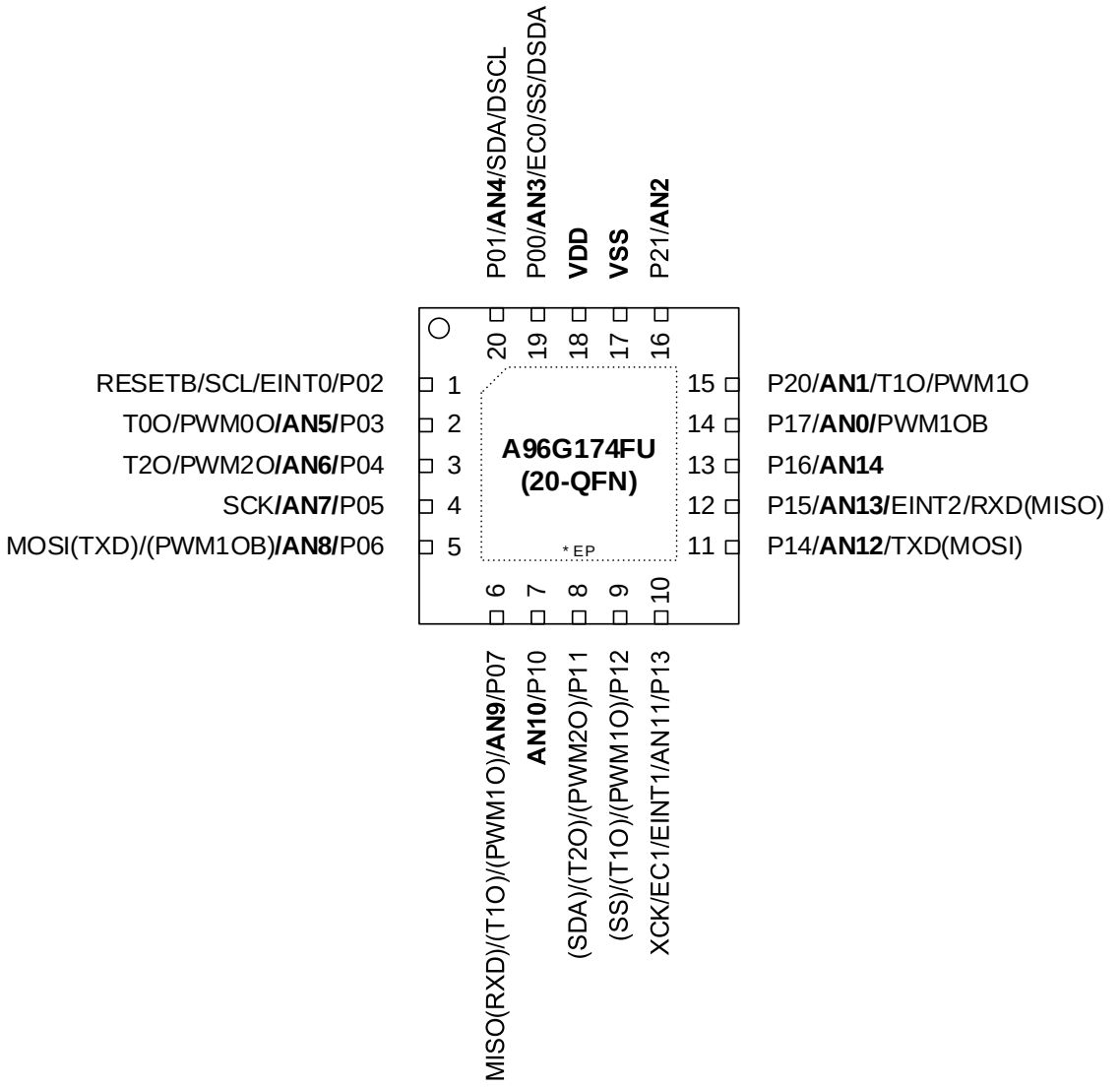
Figure 3. A96S174 20TSSOP pin assignment



NOTES:
1. The programmer (E-PGM+, E-Gang4/6) uses P0[1:0] pin as DSCL, DSDA.
2. The P22 pins should be selected as a push-pull output or an input with pull-up resistor by software control when the 20-pin package is used.

3.

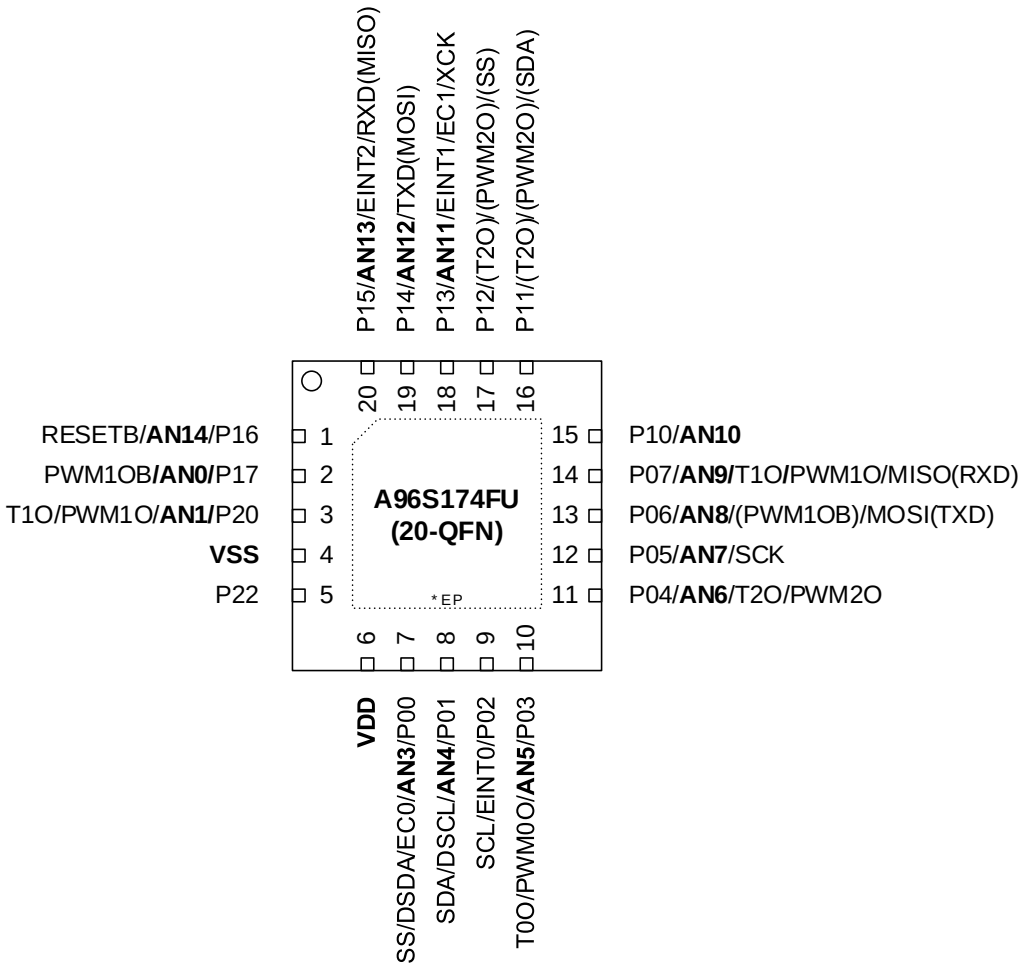
Figure 4. A96G174 20SOP Pin Assignment



NOTES:

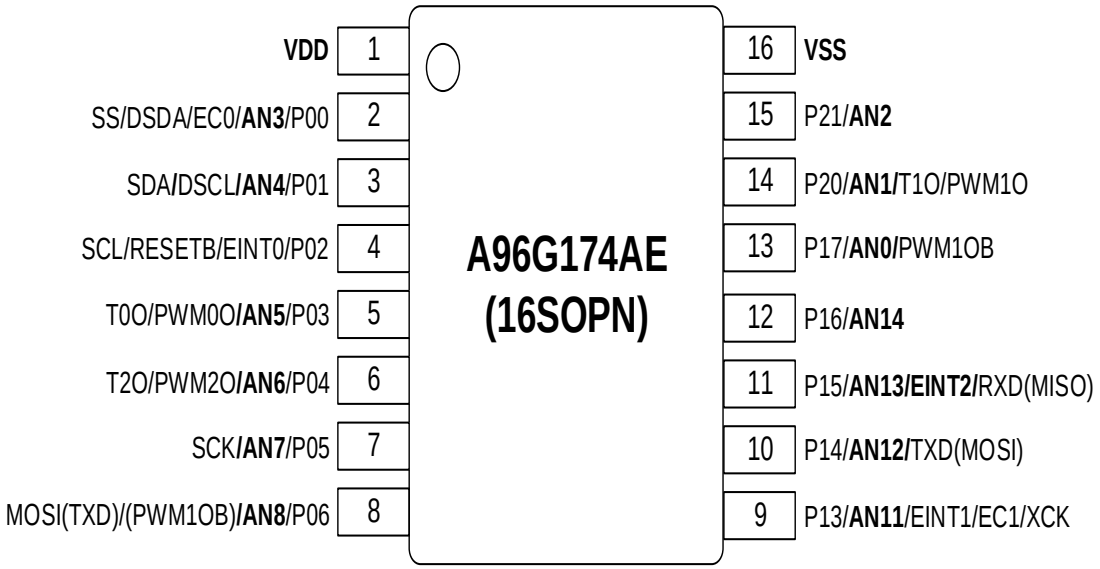
1. The programmer (E-PGM+, E-Gang4/6) uses P0[1:0] pin as DSCL, DSDA.
2. The P22 pins should be selected as a push-pull output or an input with pull-up resistor by software control when the 20-pin package is used.
3. *EP is Floating (Connectable to VSS)

Figure 5. A96G174 20QFN Pin Assignment



NOTES:
1. The programmer (E-PGM+, E-Gang4/6) uses P0[1:0] pin as DSCL, DSDA.
2. The P21 pins should be selected as a push-pull output or an input with pull-up resistor by software control when the 20-pin package is used.
3. *EP is Floating (Connectable to VSS)

Figure 6. A96S174 20QFN Pin Assignment



NOTE: The programmer (E-PGM+, E-Gang4/6) uses P0[1:0] pin as DSCL, DSDA

Figure 7. A94G174 16SOPN Pin Assignment

2.2 Pin description

Table 2. Normal Pin Description

Pin no.			PIN Name	I/O[2]	Description	Remark
20TSSOP	20QFN	16SOPN				
2(10) [1]	19(7)[1]	2	P00*	IOUS	Port 0 bit 0 Input/output	
			AN3	IA	ADC input ch-3	
			EC0	I	Timer 0(Event Capture) input	
			DSDA	IO	OCD debugger data input/output	Pull-up
			SS	IO	USART slave select signal	
3(11)	20(8)	3	P01*	IOUS	Port 0 bit 1 Input/output	
			AN4	IA	ADC input ch-4	
			DSCL	I	OCD debugger clock	Pull-up
			SDA	IO	I2C data signal	
4(12)	1(9)	4	P02*	IOUS	Port 0 bit 2 Input/output	
			RESETB	IU	A96G174 only, Reset pin	Pull-up
			EINT0	I	External interrupt input ch-0	
			SCL	IO	I2C clock signal	
5(13)	2(10)	5	P03*	IOUS	Port 0 bit 3 Input/output	
			AN5	IA	ADC input ch-5	
			T0O	O	Timer 0 interval output	
			PWM0O	O	Timer 0 PWM output	
6(14)	3(11)	6	P04*	IOUS	Port 0 bit 4 Input/output	
			AN6	IA	ADC input ch-6	
			T2O	O	Timer 2 interval output	
			PWM2O	O	Timer 2 PWM output	
7(15)	4(12)	7	P05*	IOUS	Port 0 bit 5 Input/output	
			AN7	IA	ADC input ch-7	
			SCK	IO	USART external clock input/output	
8(16)	5(13)	8	P06*	IOUS	Port 0 bit 6 Input/output	
			AN8	IA	ADC input ch-8	
			MOSI (TXD)	IO	USART data transmit /SPI MOSI	
			PWM1OB	O	Timer 1 PWM complementary output	
9(17)	6(14)	-	P07*	IOUS	Port 0 bit 7 Input/output	
			AN9	IA	ADC input ch-9	
			MISO (RXD)	IO	USART data receive /SPI MISO	

Table 2. Normal Pin Description (continued)

Pin no.			PIN Name	I/O[2]	Description	Remark
20TSSOP	20QFN	16SOPN				
9(17)	6(14)	-	T1O	O	Timer 1 interval output	
			PWM1O	O	Timer 1 PWM output	
10(18)	7(15)	-	P10*	IOUS	Port 1 bit 0 Input/output	
			AN10	IA	ADC input ch-10	
11(19)	8(16)	-	P11*	IOUS	Port 1 bit 1 Input/output	
			T2O	O	Timer 2 interval output	
			PWM2O	O	Timer 2 PWM output	
			SDA	IO	I2C data signal	

12(20)	9(17)	-	P12*	IOUS	Port 1 bit 2 Input/output	
			T1O	O	Timer 1 interval output	
			PWM1O	O	Timer 1 PWM output	
			SS	IO	USART slave select signal	
13(1)	10(18)	9	P13*	IOUS	Port 1 bit 3 Input/output	
			AN11	IA	ADC input ch-11	
			EINT1	I	External interrupt input ch-1	
			EC1	I	Timer 1(Event Capture) input	
			XCK	IO	USART clock signal	
14(2)	11(19)	10	P14*	IOUS	Port 1 bit 4 Input/output	
			AN12	IA	ADC input ch-12	
			TXD (MOSI)	IO	USART data transmit /SPI MOSI	
15(3)	12(20)	11	P15*	IOUS	Port 1 bit 5 Input/output	
			AN13	IA	ADC input ch-13	
			EINT2	I	External interrupt input ch-2	
			RXD (MISO)	IO	USART data receive /SPI MISO	
16(4)	13(1)	12	P16*	IOUS	Port 1 bit 6 Input/output	
			AN14	IA	ADC input ch-14	
			RESETB	IU	A96S174 only, Reset pin	
17(5)	14(2)	13	P17*	IOUS	Port 1 bit 7 Input/output	
			AN0	IA	ADC input ch-0	
			PWM1OB	O	Timer 1 PWM Complementary Output	

Table 2. Normal Pin Description (continued)

Pin no.			PIN Name	I/O(2)	Description	Remark
20TSSOP	20QFN	16SOPN				
20(6)	15(3)	14	P20*	IOUS	Port 2 bit 0 Input/output	
			AN1	IA	ADC input ch-1	
			T1O	O	Timer 1 interval output	
			PWM1O	O	Timer 1 PWM output	
21(-)	16(-)	15	P21*	IOUS	Port 2 bit 1 Input/output	
			AN2	IA	ADC input ch-2	
-(8)	-(5)	-	P22 I/O*	IOUS	Port 2 bit 2 Input/output	
1(9)	18(6)	1	VDD	P	VDD	
20(7)	17(4)	16	VSS	P	VSS	

NOTE:

1. (1) It is applied to A96S174.
2. (2) I=Input, O=Output, U=Pull-up, D=Pull-down, S=Schmitt-Trigger Input Type, C=CMOS Input Type, A=Analog, P=Power
3. (3) The * means 'Selected pin function after reset condition

3 Port structures

In this chapter, two port structures are introduced in Figure 8 & Figure 9 regarding general purpose I/O port and external interrupt I/O port respectively.

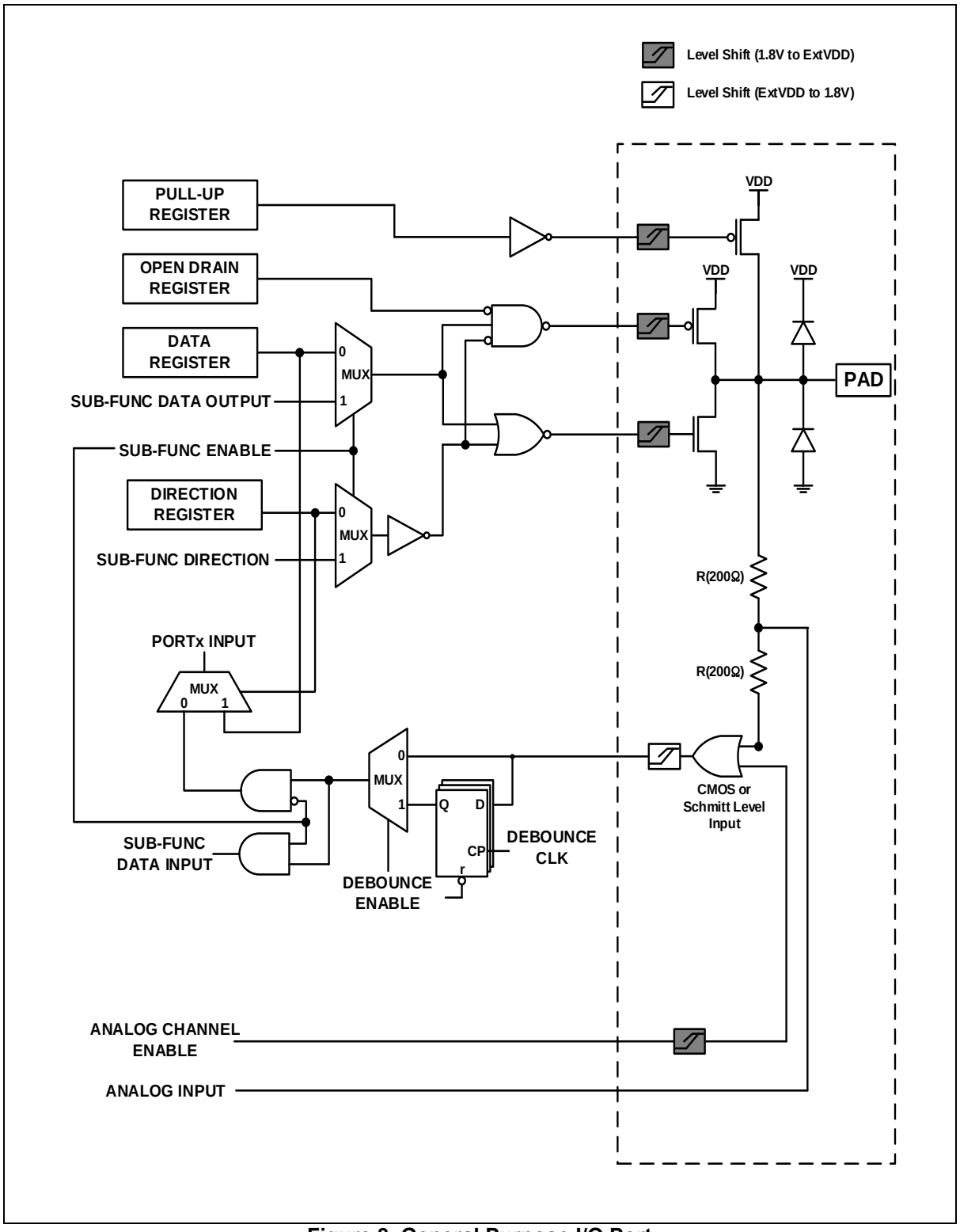


Figure 8. General Purpose I/O Port

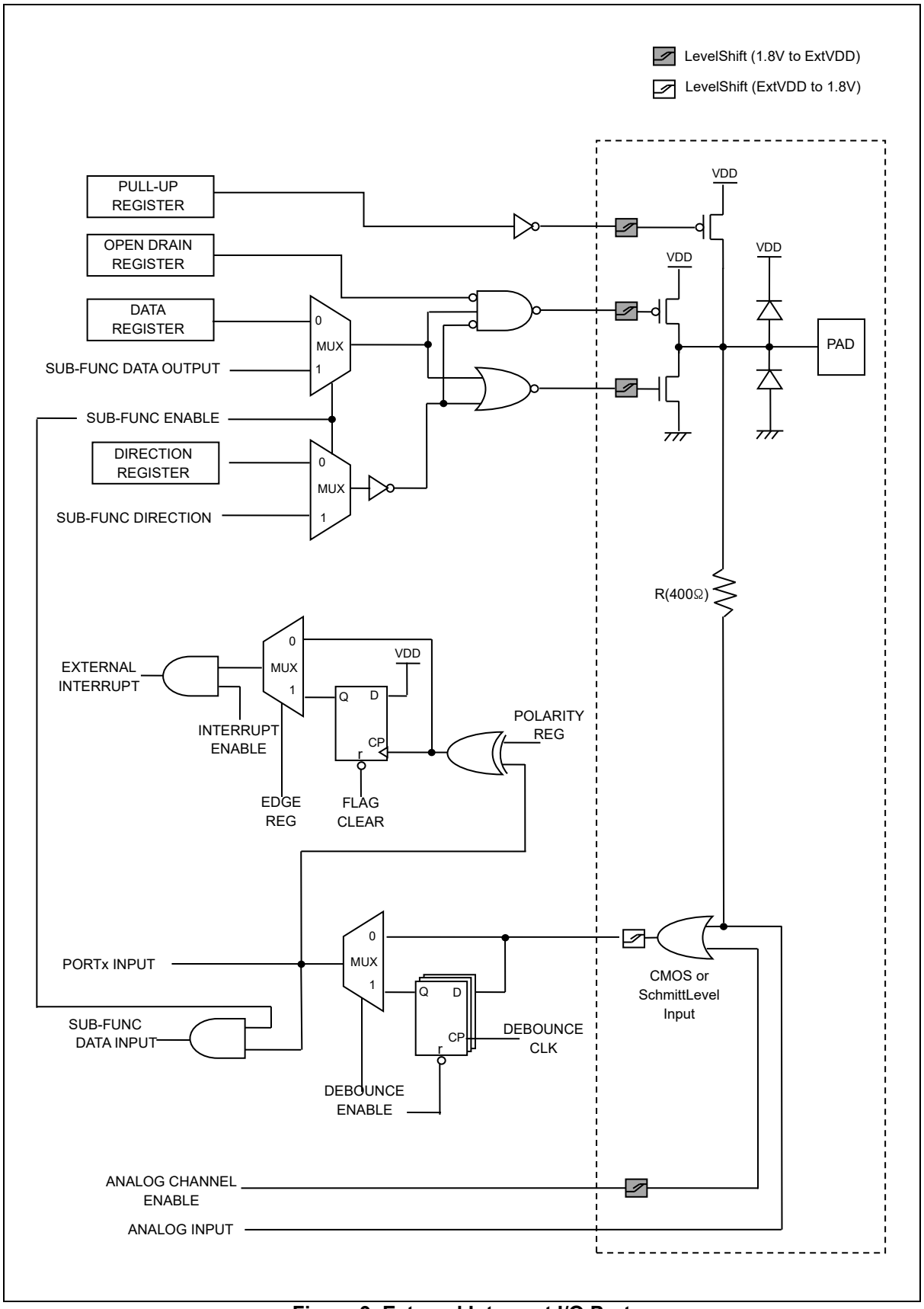


Figure 9. External Interrupt I/O Port

4 Central Processing Unit(CPU)

Central Processing Unit (CPU) of A96G174/A96S174 is based on Mentor Graphics M8051EW core, which offers improved code efficiency and high performance.

4.1 Architecture and registers

Figure 10 shows a block diagram of the M8051EW architecture. As shown in the figure, the M8051EW supports both Program Memory and External Data Memory. In addition, it features a Debug Mode in which it can be driven through a dedicated debug interface.

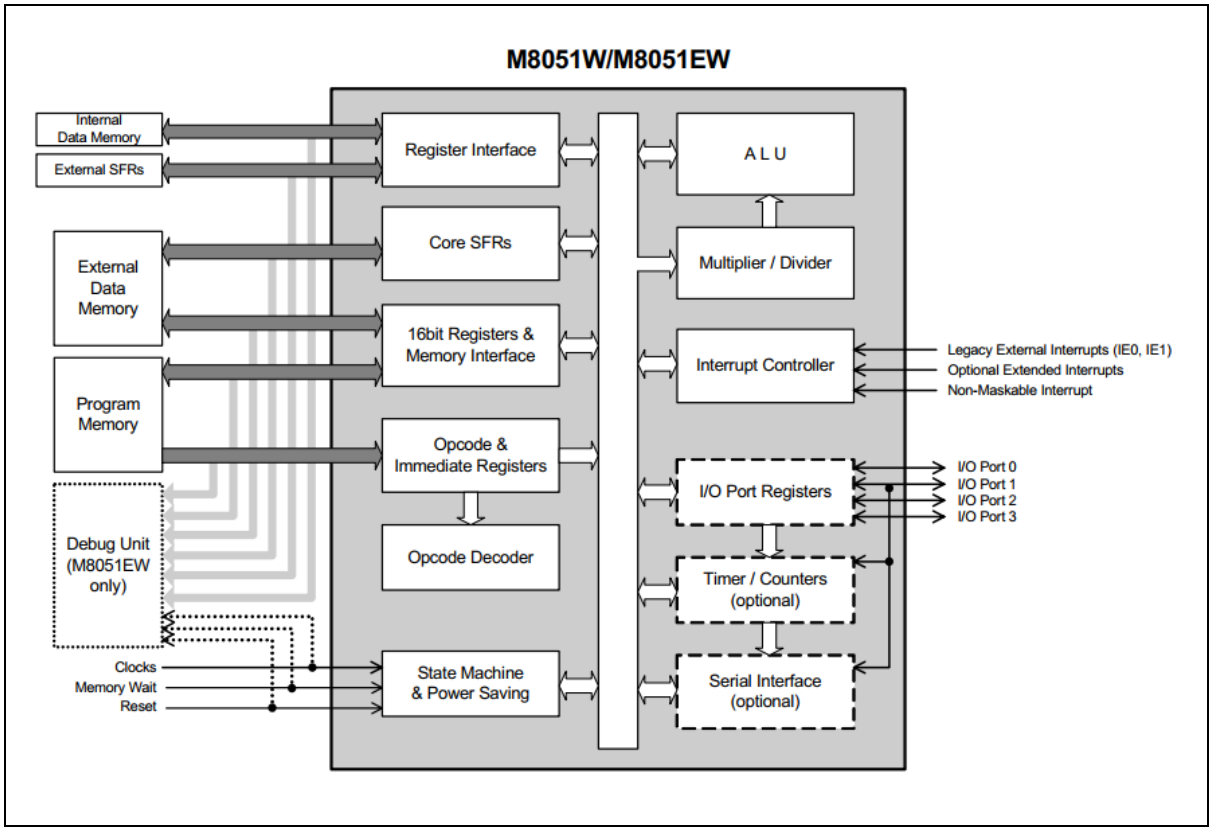


Figure 10. M8051EW Architecture

Main features of the M8051EW are listed below:

- Two clocks per machine cycle architecture:

  This allows the device either to run up to six times faster with the same power consumption or to consume one sixth of the power when running at standard speed. All instructions have zero-wait-state execution times that are exactly 1/6 of the time each standard part takes.

- Debug support (OCD and OCD II):

  The M8051EW offers a Debug Mode together with a set of dedicated debug signals which can be used by external debug hardware, OCD and OCD II, to provide start/stop program execution in response to both hardware and software triggers, single step operation and program execution tracing.

- Separate program and external data memory interfaces or a single multiplexed interface
  - Up to 1Mbyte of External Data Memory, accessible by selecting one from interfaces
  - Up to 256bytes of Internal Data Memory
  - Up to 1Mbyte of RAM or ROM Program Memory, accessible by selecting one from interfaces
- Support for synchronous and asynchronous Program, External Data and Internal Data Memory
- Wait states support for slow Program and External Data Memory
- 16-bit Data Memory address is generated through the Data Pointer register(DPTR register).
- 16-bit program counter is capable of addressing up to Flash size in Each device
- A single data pointer, two memory-mapped data pointers, or 2 banked data pointers
- Support for 2 or 4 level of priority scheme – Up to 24 maskable Interrupt sources
- External Special Function Register (SFR) are memory mapped into Direct Memory at the addresses between 80 hex and FF hex

4.2 Addressing

The M8051EW supports six types of addressing modes as listed below:

1. Direct addressing mode: In this mode, the operand is specified by the 8-bit address field. Only internal data and SFRs can be accessed using this mode.
2. Indirect addressing mode: In this mode, the operand is specified by addresses contained in a register. Two registers (R0 and R1) from the current bank or the Data Pointer may be used for addressing in this mode. Both internal and external Data Memory may be indirectly addressed.
3. Register addressing mode: In this mode, the operand is specified by the top 3 bits of the opcode, which selects one of the current bank of registers. Four banks of registers are available. The current bank is selected by the 3rd and 4th bits of the PSW.
4. Register specific addressing mode: In this mode, some instructions only operate on specific registers. This is defined by the opcode. In particular many accumulator operations and some stack pointer operations are defined in this manner.
5. Immediate DATA mode: In this mode, Instructions which use Immediate Data are 2 or more bytes long and the Immediate operand is stored in Program Memory as part of the instruction.

   Example) `MOV A, #100`

   It loads the Accumulator with the decimal number 100. The same number could be specified in hex digits as 64H.
6. Indexed addressing mode: In this mode, only Program Memory can be addressed. It is intended for simple implementation of look-up tables. A 16-bit base register (either the PC or the DPTR) is combined with an offset stored in the accumulator to access data in Program Memory.

4.3 Instruction set

An instruction is a single operation of a processor that is defined by the instruction set. The M8051EW uses the instruction set of 8051 that is broadly classified into five functional categories:

1. Arithmetic instructions
2. Logical instructions
3. Data transfer instructions
4. Boolean instructions
5. Branching instructions

Major features of the instruction set are listed below. If you need detailed information about the instruction table, please refer to **Appendix** or **Instruction table**

- Instructions are either 1, 2 or 3 bytes long as listed in the 'Bytes'.
- Each instruction takes either 1, 2 or 4 machine cycles to execute. 1 machine cycle comprises 2 CCLK clock cycles.
- An M8051EW-specific instruction "`MOVC @(DPTR++), A`" is provided to enable software to be downloaded into Program Memory where this is implemented as RAM. This instruction can also be used subsequently to modify contents of the Program Memory RAM.
- Arithmetic Instruction: The M8051EW implements ADD, ADDC (Add with Carry), SUBB (Subtract with Borrow), INC (Increment) and DEC (Decrement) functions, which can be used in most addressing modes. There are three accumulator-specific instructions, DA A (Decimal Adjust A), MUL AB (Multiply A by B) and DIV AB (Divide A by B).
- Logical Instruction: The M8051EW implements ANL (AND Logical), ORL (OR Logical), and XRL (Exclusive-OR Logical) functions, which can be used in most addressing modes. There are seven accumulator-specific instructions, CLR A (Clear A), CPL A (Complement A), RL A (Rotate Left A), RLC A (Rotate Left through Carry A), RR A (Rotate Right A), RRC A (Rotate Right through Carry A), and SWAP A (Swap Nibbles of A).
- Internal data memory: Data can be moved from the accumulator to any Internal Data Memory location, from any Internal Data Memory location to the accumulator, and from any Internal Data Memory location to any SFR or other Internal Data Memory location.

- External data memory: Data can be moved between the accumulator and the external memory location in one of two addressing modes. In 8-bit addressing mode, the external location is addressed by either R0 or R1; in 16-bit addressing mode, the location is addressed by the DPTR.
- Unconditional Jumps: Four sorts of unconditional jump instructions are available.
  - Short jumps (SJMP) are relative jumps (limited from -128bytes to +127bytes).
  - Long jumps (LJMP) are absolute 16-bit jumps.
  - Absolute jumps (AJMP) are absolute 11-bit jumps (ex. within a 2Kbyte memory page).
  - Indexed jump, `JMP @A+DPTR`. This instruction jumps to a location of which address is stored in DPTR register and offset by a value stored in the accumulator.
- Subroutine calls and returns: There are only two sorts of subroutine call, ACALL and LCALL, which are Absolute and Long. Two return instructions are provided, RET and RETI. The latter is for interrupt service routines.
- Conditional jumps: All conditional jump instructions use relative addressing, so they are limited to the range of -128bytes to +127bytes.
- Boolean instructions: The bit-addressable registers in both direct and SFR space may be manipulated using Boolean instructions. Logical functions are available which use the carry flag and an addressable bit as operands. Each addressable bit can be set, cleared or tested in a jump instruction.
- Flag: Certain instructions affect one or more flags that are generated by ALU.

5 Memory organization

A96G174/A96S174 addresses two separate memory spaces:

- Program memory
- Data memory

By means of this logical separation of the memory, 8-bit CPU address can access the Data Memory more rapidly. 16-bit Data Memory address is generated through the DPTR register.

A96G174/A96S174 provides on-chip 8Kbytes of the ISP type flash program memory, which readable and writable. Internal data memory (IRAM) is 256bytes and it includes the stack area. External data memory (XRAM) is 256bytes.

5.1 Program memory

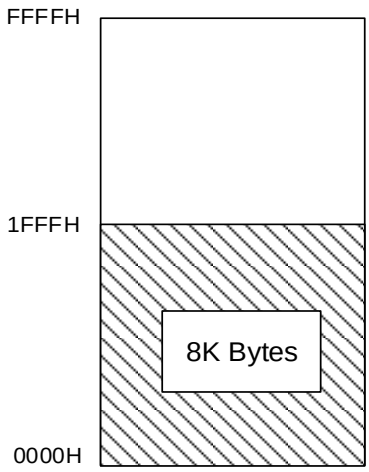
A 16-bit program counter is capable of addressing up to 64Kbytes, and A96G174/A96S174 has just 8Kbytes program memory space.

Figure 11 shows a map of the lower part of the program memory.

After reset, CPU begins execution from location 0000H. Each interrupt is assigned a fixed location in the program memory. An interrupt causes the CPU to jump to the corresponding location, where it commences execution of the service routine.

An external interrupt 11, for example, is assigned to location 000BH. If the external interrupt 11 is going to be used, its service routine must begin at location 000BH. If the interrupt is not going to be used, its service location is available as general purpose program memory. If an interrupt service routine is short enough (as is often the case in control applications), it can reside entirely within an interval of 8-bytes.

Longer service routines can use a jump instruction to skip over subsequent interrupt locations, if other interrupts are in use.



NOTE: The 8Kbytes includes the Interrupt Vector Region.

Figure 11. Program Memory Map

5.2 Data memory

Internal data memory space is divided into three blocks, which are generally referred to as lower 128bytes, upper 128bytes, and SFR space. Internal data memory addresses are always one byte wide, which implies an address space of 256bytes. In fact, the addressing modes for the internal data memory can accommodate up to 384bytes by using a simple trick. Direct addresses higher than 7FH access one memory space, while indirect addresses higher than 7FH access a different memory space. Thus as shown in Figure 12, the upper 128bytes and SFR space occupy the same block of addresses, 80H through FFH, although they are physically separate entities.

The lower 128bytes of RAM are present in all 8051 devices as mapped in Figure 13. The lowest 32bytes are grouped into 4 banks of 8 registers. Program instructions call out these registers as R0 through R7. Two bits in the Program Status Word select which register bank is in use. This allows more efficient use of code space, since register instructions are shorter than instructions that use direct addressing.

The next 16bytes 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 128bytes can be accessed by either direct or indirect addressing. The upper 128bytes of RAM can only be accessed by indirect addressing. These spaces are used for data RAM and stack.

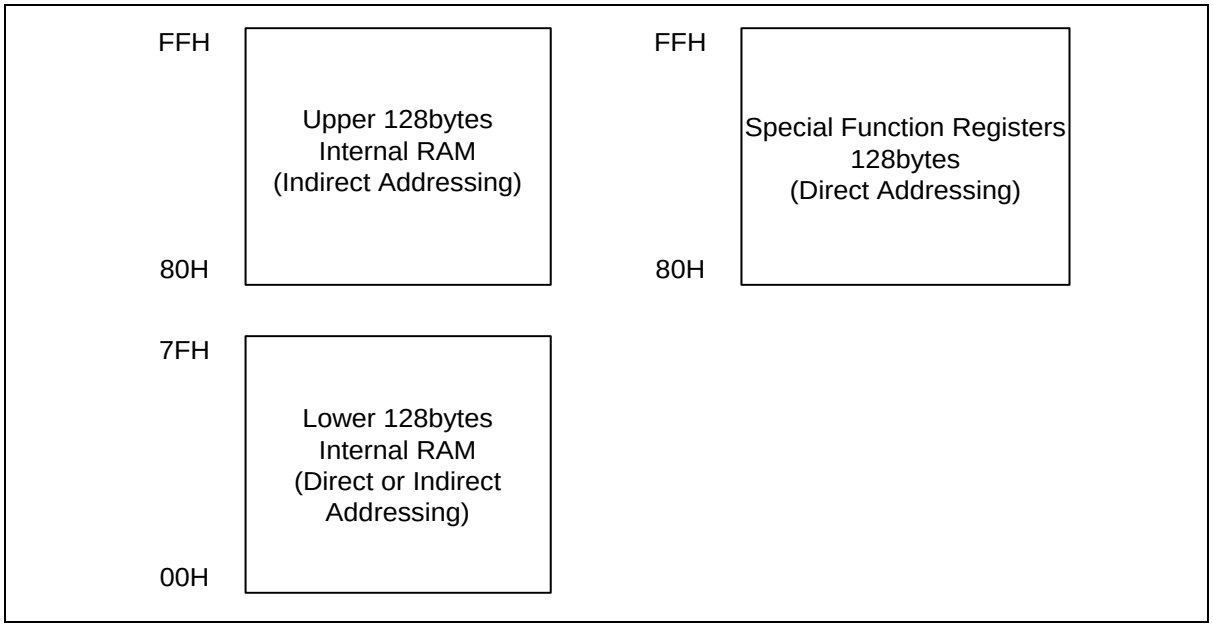


Figure 12. Data Memory Map

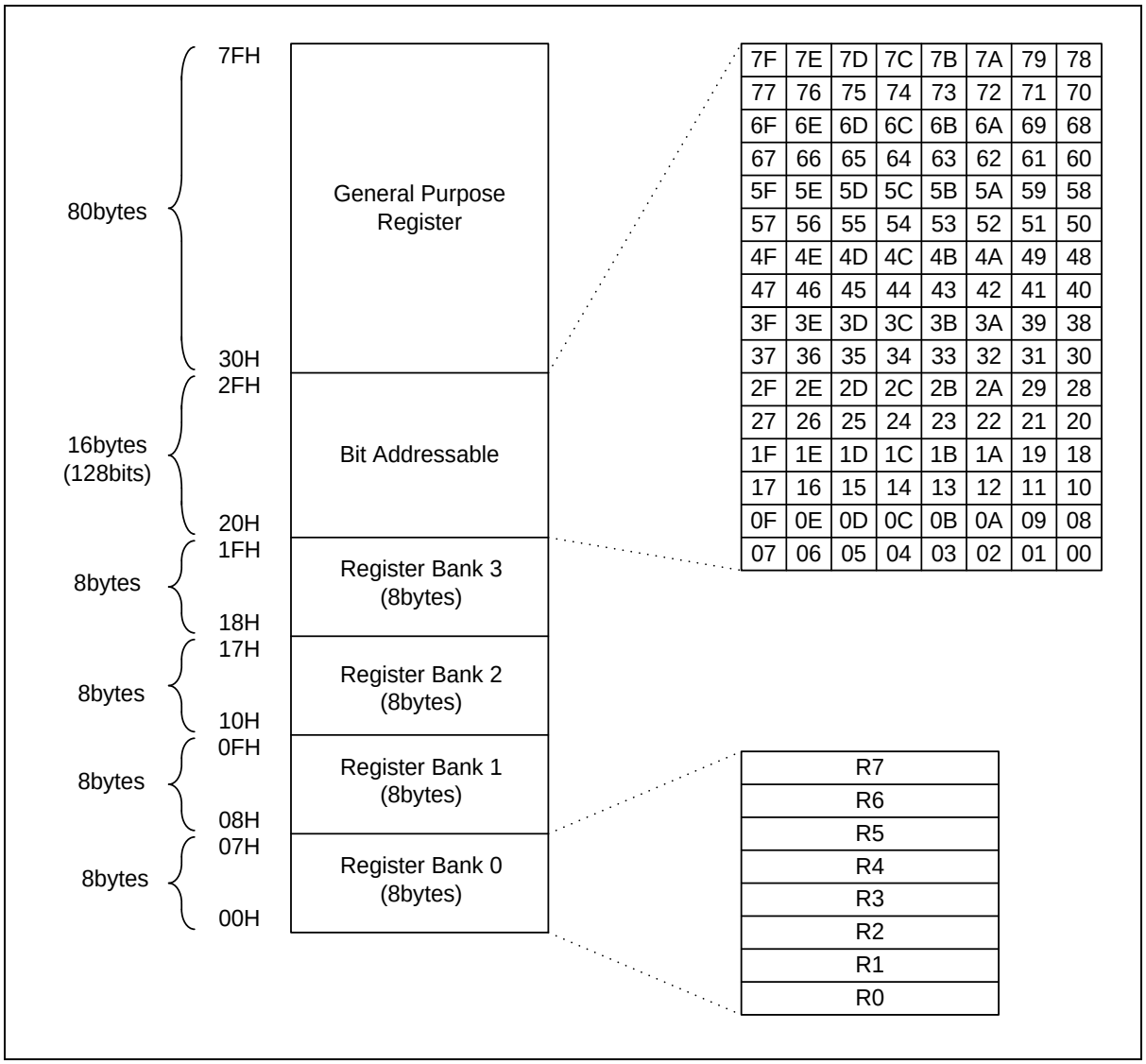


Figure 13. Lower 128bytes of RAM

5.3 External data memory

A96G174/A96S174 has 256bytes of XRAM and XSFR. This area has no relation with RAM/FLASH. It can be read and written to through SFR with 8-bit unit.

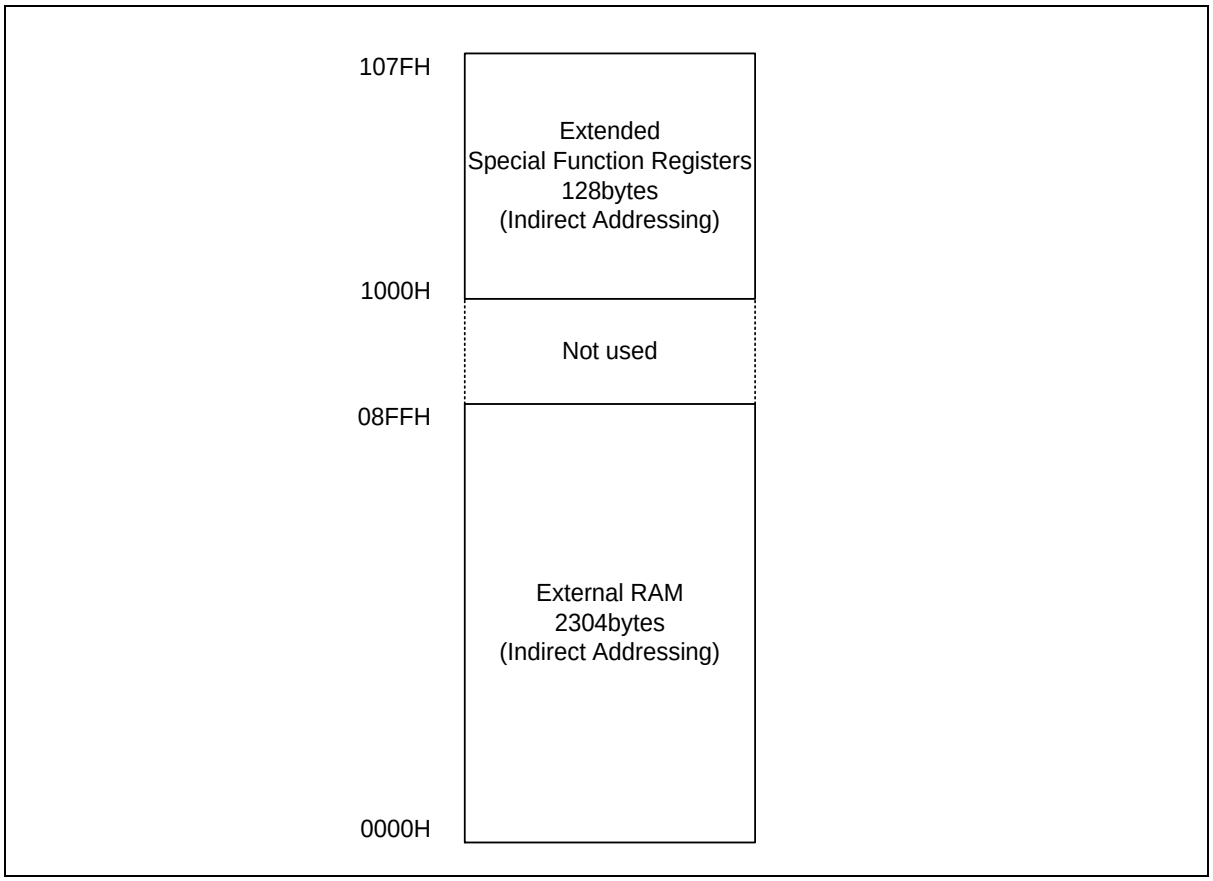


Figure 14. XDATA Memory Area

5.4 SFR map

5.4.1 SFR map summary

Table 3. SFR Map Summary

–	Reserved
	M8051 compatible

	00H/8H[1]	01H/9H	02H/0AH	03H/0BH	04H/0CH	05H/0DH	06H/0EH	07H/0FH
0F8H	IP1	–	–	–	UBAUD	UDATA	–	
0F0H	B							
0E8H	RSTFR	I2CSAR	I2CSAR1					
0E0H	ACC	I2CMR	I2CSR	I2CSCLLR	I2CSCLHR	I2CSDAHR	I2CDR	–
0D8H	LVRCR	T1CDRL	T1CDRH	T1DDRL	T1DDRH	–	P0DB	P1DB
0D0H	PSW		P0FSRL	P0FSRH	P1FSRL	P1FSRH	P2FSR	–
0C8H	OSCCR		–	UCTRL1	UCTRL2	UCTRL3	–	USTAT
0C0H	EIFLAG		T2CRL	T2CRH	T2ADRL	T2ADRH	T2BDRL	T2BDRH
0B8H	IP	P2IO	T1CRL	T1CRH	T1ADRL	T1ADRH	T1BDRL	T1BDRH
0B0H		P1IO	T0CR	T0CNT	T0DR/ T0CDR	–	–	–
0A8H	IE	IE1	IE2	–	P0PU	P1PU	P2PU	
0A0H		P0IO	EO	–	EIPOL	–	–	–
98H	–	–	–	–	ADCCRL	ADCCRH	ADCDRL	ADCDRH
90H	P2	P0OD	P1OD	P2OD	PCI1	–	–	
88H	P1	–	SCCR	BITCR	BITCNT	WDTCR	WDTIDR	
80H	P0	SP	DPL	DPH	DPL1	DPH1	LVICR	PCON

NOTE: 00H/8H, these registers are bit-addressable.

Table 4. XSFR Map Summary

	00H/8H[1]	01H/9H	02H/0AH	03H/0BH	04H/0CH	05H/0DH	06H/0EH	07H/0FH
1078H	–	–	–	–	–	–	–	–
1070H	–	–	–	–	–	–	–	–
1068H	–	–	–	–	–	–	–	–
1060H	–	–	–	–	–	–	–	–
1058H	–	–	–	–	–	–	–	–
1050H	–	–	–	–	–	–	–	–
1048H	–	–	–	–	–	–	–	–
1040H	–	–	–	–	–	–	–	–
1038H	–	–	–	–	–	–	–	–
1030H	–	–	–	–	–	–	–	–
1028H	FEARH	FEARM	FEARL	FEDR	FETR	–	–	–
1020H	FEMR	FECR	FESR	FETCR	FEARM1	FEARL1	–	–
1018H	UCTRL4	FPCR	RTOCH	RTOCL	–	–	–	–
1010H	WDTC	WDTSR	WDTCNTH	WDTCNTL	–	–	–	–
1008H	–	–	–	–	–	–	–	–
1000H	–	–	–	–	–	–	–	–

5.4.2 SFR map

Table 5. SFR Map

Address	Function	Symbol	R/W	@Reset							
				7	6	5	4	3	2	1	0
80H	P0 Data Register	P0	R/W	0	0	0	0	0	0	0	0
81H	Stack Pointer	SP	R/W	0	0	0	0	0	1	1	1
82H	Data Pointer Register Low	DPL	R/W	0	0	0	0	0	0	0	0
83H	Data Pointer Register High	DPH	R/W	0	0	0	0	0	0	0	0
84H	Data Pointer Register Low 1	DPL1	R/W	0	0	0	0	0	0	0	0
85H	Data Pointer Register High 1	DPH1	R/W	0	0	0	0	0	0	0	0
86H	Low Voltage Indicator Control Register	LVICR	R/W	–	–	0	0	0	0	0	0
87H	Power Control Register	PCON	R/W	0	–	–	–	0	0	0	0
88H	P1 Data Register	P1	R/W	0	0	0	0	0	0	0	0
8AH	System and Clock Control Register	SCCR	R/W	–	–	–	–	–	–	0	0
8BH	Basic Interval Timer Control Register	BITCR	R/W	0	1	0	0	0	1	0	1
8CH	Basic Interval Timer Counter Register	BITCNT	R	0	0	0	0	0	0	0	0
8DH	Watch Dog Timer Control Register	WDTCR	R/W	0	0	0	0	0	1	1	1
8EH	Watch Dog Timer Identification Register	WDTIDR	W	0	0	0	0	0	0	0	0
90H	P2 Data Register	P2	R/W	0	0	0	0	0	0	0	0
91H	P0 Open-drain Selection Register	P0OD	R/W	0	0	0	0	0	0	0	0
92H	P1 Open-drain Selection Register	P1OD	R/W	0	0	0	0	0	0	0	0
93H	P2 Open-drain Selection Register	P2OD	R/W	0	0	0	0	0	0	0	0
94H	P4 Open-drain Selection Register	P4OD	R/W	0	0	0	0	0	0	0	0
9CH	A/D Converter Control Low Register	ADCCRL	R/W	0	0	0	0	0	0	0	0
9DH	A/D Converter Control High Register	ADCCRH	R/W	0	0	0	0	0	0	0	1
9EH	A/D Converter Data Low Register	ADCDRL	R	x	x	x	x	x	x	x	x
9FH	A/D Converter Data High Register	ADCDRH	R	x	x	x	x	x	x	x	x

Table 5. SFR Map (continued)

Address	Function	Symbol	R/W	@Reset							
				7	6	5	4	3	2	1	0
A1H	P0 Direction Register	P0IO	R/W	0	0	0	0	0	0	0	0
A2H	Extended Operation Register	EO	R/W	–	–	–	0	–	0	0	0
A4H	External Interrupt Polarity 0 Low Register	EIPOL0L	R/W	0	0	0	0	0	0	0	0
A5H	External Interrupt Polarity 0 High Register	EIPOL0H	R/W	0	0	0	0	0	0	0	0
A6H	External Interrupt Flag 1 Register	EIFLAG1	R/W	0	0	–	–	0	0	0	0
A7H	External Interrupt Polarity 1 Register	EIPOL1	R/W	0	0	0	0	0	0	0	0
A8H	Interrupt Enable Register	IE	R/W	0	–	0	0	0	0	0	0
A9H	Interrupt Enable Register 1	IE1	R/W	–	–	0	0	0	0	0	0
AAH	Interrupt Enable Register 2	IE2	R/W	–	–	0	0	0	0	0	0
ACH	P0 Pull-up Resistor Selection Register	P0PU	R/W	0	0	0	0	0	0	0	0
ADH	P1 Pull-up Resistor Selection Register	P1PU	R/W	0	0	0	0	0	0	0	0
AEH	P2 Pull-up Resistor Selection Register	P2PU	R/W	0	0	0	0	0	0	0	0
B1H	P1 Direction Register	P1IO	R/W	0	0	0	0	0	0	0	0
B2H	Timer 0 Control Register	T0CR	R/W	0	–	0	0	0	0	0	0
B3H	Timer 0 Counter Register	T0CNT	R	0	0	0	0	0	0	0	0
B4H	Timer 0 Data Register	T0DR	R/W	1	1	1	1	1	1	1	1
	Timer 0 Capture Data Register	T0CDR	R	0	0	0	0	0	0	0	0
B8H	Interrupt Priority Register	IP	R/W	–	–	0	0	0	0	0	0
B9H	P2 Direction Register	P2IO	R/W	0	0	0	0	0	0	0	0
BAH	Timer 1 Control Low Register	T1CRL	R/W	0	0	0	0	–	0	0	0
BBH	Timer 1 Counter High Register	T1CRH	R/W	0	–	0	0	–	–	–	0
BCH	Timer 1 A Data Low Register	T1ADRL	R/W	1	1	1	1	1	1	1	1
BDH	Timer 1 A Data High Register	T1ADRH	R/W	1	1	1	1	1	1	1	1
BEH	Timer 1 B Data Low Register	T1BDRL	R/W	1	1	1	1	1	1	1	1
BFH	Timer 1 B Data High Register	T1BDRH	R/W	1	1	1	1	1	1	1	1
D9H	Timer 1 C Data Low Register	T1CDRL	R/W	1	1	1	1	1	1	1	1
DAH	Timer 1 C Data High Register	T1CDRH	R/W	1	1	1	1	1	1	1	1
DBH	Timer 1 D Data Low Register	T1DDRL	R/W	1	1	1	1	1	1	1	1
DCH	Timer 1 D Data High Register	T1DDRH	R/W	1	1	1	1	1	1	1	1

Table 5. SFR Map (continued)

Address	Function	Symbol	R/W	@Reset							
				7	6	5	4	3	2	1	0
C0H	External Interrupt Flag 0 Register	EIFLAG0	R/W	0	0	0	0	0	0	0	0
C2H	Timer 2 Control Low Register	T2CRL	R/W	0	0	0	0	–	0	–	0
C3H	Timer 2 Control High Register	T2CRH	R/W	0	–	0	0	–	–	–	0
C4H	Timer 2 A Data Low Register	T2ADRL	R/W	1	1	1	1	1	1	1	1
C5H	Timer 2 A Data High Register	T2ADRH	R/W	1	1	1	1	1	1	1	1
C6H	Timer 2 B Data Low Register	T2BDRL	R/W	1	1	1	1	1	1	1	1
C7H	Timer 2 B Data High Register	T2BDRH	R/W	1	1	1	1	1	1	1	1
C8H	Oscillator Control Register	OSCCR	R/W	–	0	1	0	1	0	0	0
CBH	USART Control Register 1	UCTRL1	R/W	0	0	0	0	0	0	0	0
CCH	USART Control Register 2	UCTRL2	R/W	0	0	0	0	0	0	0	0
CDH	USART Control Register 3	UCTRL3	R/W	0	0	0	0	–	0	0	0
CFH	USART Status Register	USTAT	R/W	1	0	0	0	0	0	0	0
D0H	Program Status Word Register	PSW	R/W	0	0	0	0	0	0	0	0
D2H	P0 Function Selection Low Register	P0FSRL	R/W	0	0	0	0	0	0	0	0
D3H	P0 Function Selection High Register	P0FSRH	R/W	0	0	0	0	0	0	0	0
D4H	P1 Function Selection Low Register	P1FSRL	R/W	0	0	0	0	0	0	0	0
D5H	P1 Function Selection High Register	P1FSRH	R/W	0	0	0	0	0	0	0	0
D6H	P2 Function Selection Register	P2FSR	R/W	–	–	–	0	0	0	0	0
D8H	Low Voltage Reset Control Register	LVRCR	R/W	–	–	–	0	0	0	0	0
DFH	P1/P5 De-bounce Enable Register	P15DB	R/W	–	–	0	0	0	0	0	0
E0H	Accumulator Register	ACC	R/W	0	0	0	0	0	0	0	0
E1H	I2C Mode Control Register	I2CMR	R/W	0	0	0	0	0	0	0	0
E2H	I2C Status Register	I2CSR	R	0	0	0	0	0	0	0	0
E3H	SCL Low Period Register	I2CSCLLR	R/W	0	0	1	1	1	1	1	1
E4H	SCL High Period Register	I2CSCLHR	R/W	0	0	1	1	1	1	1	1
E5H	SDA Hold Time Register	I2CSDAHR	R/W	0	0	0	0	0	0	0	1
E6H	I2C Data Register	I2CDR	R/W	1	1	1	1	1	1	1	1
E9H	I2C Slave Address Register	I2CSAR	R/W	0	0	0	0	0	0	0	0
EAH	I2C Slave Address Register 1	I2CSAR1	R/W	0	0	0	0	0	0	0	0
E8H	Reset Flag Register	RSTFR	R/W	1	x	0	0	x	–	–	–
F0H	B Register	B	R/W	0	0	0	0	0	0	0	0
F8H	Interrupt Priority Register 1	IP1	R/W	–	–	0	0	0	0	0	0
FCH	USART Baud Rate Generation Register	UBAUD	R/W	1	1	1	1	1	1	1	1
FDH	USART Data Register	UDATA	R/W	0	0	0	0	0	0	0	0

Table 6. XSFR Map

Address	Function	Symbol	R/W	@Reset							
				7	6	5	4	3	2	1	0
1010H	Watch Dog Timer Clear Register	WDTC	R/W	0	0	0	0	0	0	0	0
1011H	Watch Dog Timer Status Register	WDTSR	R/W	0	0	0	0	0	0	0	0
1012H	Watch Dog Timer Count H Register	WDTCNTH	R	0	0	0	0	0	0	0	0
1013H	Watch Dog Timer Count L Register	WDTCNTL	R	0	0	0	0	0	0	0	0
1018H	USART Control Register 4	UCTRL4	R/W	–	–	–	0	0	0	0	0
1019H	USART Floating Point Counter	FPCR	R/W	0	0	0	0	0	0	0	0
101AH	Receiver Time Out Counter High Register	RTOCH	R	0	0	0	0	0	0	0	0
101BH	Receiver Time Out Counter Low Register	RTOCL	R	0	0	0	0	0	0	0	0
1020H	Flash Mode Register	FEMR	R/W	0	–	0	0	0	0	0	0
1021H	Flash Control Register	FECR	R/W	0	–	0	0	0	0	1	1
1022H	Flash Status Register	FESR	R/W	1	–	–	–	0	0	0	0
1023H	Flash Time Control Register	FETCR	R/W	0	0	0	0	0	0	0	0
1024H	Flash Address Middle Register 1	FEARM1	R/W	0	0	0	0	0	0	0	0
1025H	Flash Address Low Register 1	FEARL1	R/W	0	0	0	0	0	0	0	0
1028H	Flash Address High Register	FEARH	R/W	0	0	0	0	0	0	0	0
1029H	Flash Address Middle Register	FEARM	R/W	0	0	0	0	0	0	0	0
102AH	Flash Address Low Register	FEARL	R/W	0	0	0	0	0	0	0	0

5.4.3 Compiler compatible SFR

ACC (Accumulator Register): E0H

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

Initial value: 00H

ACC	Accumulator

B (B Register): F0H

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

Initial value: 00H

B	B Register

SP (Stack Pointer): 81H

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

Initial value: 07H

SP	Stack Pointer

DPL (Data Pointer Register Low): 82H

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

Initial value: 00H

DPL	Data Pointer Low

DPH (Data Pointer Register High): 83H

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

Initial value: 00H

DPH	Data Pointer High

DPL1 (Data Pointer Register Low 1): 84H

7	6	5	4	3	2	1	0
DPL1							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

DPL1	Data Pointer Low 1

DPH1 (Data Pointer Register High 1): 85H

7	6	5	4	3	2	1	0
DPH1							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

DPH1	Data Pointer High 1

PSW (Program Status Word Register): D0H

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

Initial value: 00H

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

EO (Extended Operation Register): A2H

7	6	5	4	3	2	1	0
–	–	–	TRAP_EN	–	DPSEL2	DPSEL1	DPSEL0
–	–	–	R/W	–	R/W	R/W	R/W

Initial value: 00H

TRAP_EN Select the Instruction **(Keep always '0')**.

0 Select MOVC @(DPTR++), A

1 Select Software TRAP Instruction

DPSEL[2:0] Select Banked Data Pointer Register

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

6 I/O ports

A96G174/A96S174 has 3 groups of I/O ports (P0 ~ P2). Each port can be easily configured by software as I/O pin, internal pull up and open-drain pin to meet various system configurations and design requirements. P0 includes a function that can generate interrupt signals according to state of a pin.

6.1 Port register

6.1.1 Data register (Px)

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

6.1.2 Direction register (PxIO)

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

6.1.3 Pull-up register selection register (PxPU)

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

6.1.4 Open-drain Selection Register (PxOD)

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

6.1.5 De-bounce Enable Register (PxDB)

All I/O Ports support debounce function. Debounce clocks of each ports are fx/1, fx/4, and fx/4096.

6.1.6 Port Function Selection Register (PxFSR)

These registers define alternative functions of ports. Please remember that these registers should be set properly for alternative port function. A reset clears the PxFSR register to '00H', which makes all pins to normal I/O ports.

6.1.7 Register Map

Table 7. Port Register Map

Name	Address	Direction	Default	Description
P0	80H	R/W	00H	P0 Data Register
P0IO	A1H	R/W	00H	P0 Direction Register
P0PU	ACH	R/W	00H	P0 Pull-up Resistor Selection Register
P0OD	91H	R/W	00H	P0 Open-drain Selection Register
P0DB	DEH	R/W	00H	P0 De-bounce Enable Register
P0FSRH	D3H	R/W	00H	P0 Function Selection High Register
P0FSRL	D2H	R/W	00H	P0 Function Selection Low Register
P1	88H	R/W	00H	P1 Data Register
P1IO	B1H	R/W	00H	P1 Direction Register
P1PU	ADH	R/W	00H	P1 Pull-up Resistor Selection Register
P1OD	92H	R/W	00H	P1 Open-drain Selection Register
P1DB	DFH	R/W	00H	P1/P5Debounce Enable Register
P1FSRH	D5H	R/W	00H	P1 Function Selection High Register
P1FSRL	D4H	R/W	00H	P1 Function Selection Low Register
PCI1	94H	R/W	00H	Pin Change Interrupt Register
P2	90H	R/W	00H	P2 Data Register
P2IO	B9H	R/W	00H	P2 Direction Register
P2PU	AEH	R/W	00H	P2 Pull-up Resistor Selection Register
P2OD	93H	R/W	00H	P2 Open-drain Selection Register
P2FSR	D6H	R/W	00H	P2 Function Selection Register

6.2 P0 port

6.2.1 P0 port description

P0 is an 8-bit I/O port. P0 control registers consist of P0 data register (P0), P0 direction register (P0IO), debounce enable register (P0DB), P0 pull-up resistor selection register (P0PU), and P0 open-drain selection register (P0OD). Refer to the port function selection registers for the P0 function selection.

6.2.2 Register description for P0

P0 (P0 Data Register): 80H

7	6	5	4	3	2	1	0
P07	P06	P05	P04	P03	P02	P01	P00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

P0[7:0] I/O Data

P0IO (P0 Direction Register): A1H

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

Initial value: 00H

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

0 Input

1 Output

NOTES:

EC0(P00) function possible when input

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

7	6	5	4	3	2	1	0
P07PU	P06PU	P05PU	P04PU	P03PU	P02PU	P01PU	P00PU
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

P0PU[7:0] Configure Pull-up Resistor of P0 Port

0 Disable

1 Enable

P0OD (P0 Open-drain Selection Register): 91H

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

Initial value: 00H

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

0 Push-pull output

1 Open-drain output

P0DB (P0 De-bounce Enable Register): DEH

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

Initial value: 00H

DBCLK[1:0] Configure De-bounce Clock of Port

DBCLK1	DBCLK0	Description
0	0	fx/1
0	1	fx/4
1	0	fx/4096
1	1	LSIRC (128kHz)

P07DB Configure De-bounce of P07 Port

0	Disable
1	Enable

P06DB Configure De-bounce of P06 Port

0	Disable
1	Enable

P05DB Configure De-bounce of P05 Port

0	Disable
1	Enable

P04DB Configure De-bounce of P04 Port

0	Disable
1	Enable

P03DB Configure De-bounce of P03Port

0	Disable
1	Enable

P02DB Configure De-bounce of P02 Port

0	Disable
1	Enable

NOTES:

1. If the same level is not detected on enabled pin three or four times in a row at the sampling clock, the signal is eliminated as noise.
2. A pulse level should be input for the duration of 3 clock or more to be actually detected as a valid edge.
3. The port de-bounce is automatically disabled at stop mode and recovered after stop mode release.

P0FSRH (Port 0 Function Selection High Register): D3H

7	6	5	4	3	2	1	0
P0FSRH7	P0FSRH6	P0FSRH5	P0FSRH4	P0FSRH3	P0FSRH2	P0FSRH1	P0FSRH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

P0FSRH[7:6] P07 Function Select

P0FSRH7	P0FSRH6	Description
0	0	I/O Port
0	1	RXD/MISO Function
1	0	T1O/PWM1O
1	1	AN9 Function

P0FSRH[5:4] P06 Function Select

P0FSRH5	P0FSRH4	Description
0	0	I/O Port
0	1	TXD/MOSI Function
1	0	PWM1OB
1	1	AN8 Function

P0FSRH[3:2] P05 Function Select

P0FSRH3	P0FSRH2	Description
0	0	I/O Port
0	1	SCK Function
1	0	Reserved
1	1	AN7 Function

P0FSRH[1:0] P04 Function Select

P0FSRH1	P0FSRH0	Description
0	0	I/O Port
0	1	Reserved
1	0	T2O/PWM2O
1	1	AN6 Function

P0FSRL (Port 0 Function Selection Low Register): D2H

7	6	5	4	3	2	1	0
P0FSRL7	P0FSRL6	P0FSRL5	P0FSRL4	P0FSRL3	P0FSRL2	P0FSRL1	P0FSRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

P0FSRL[7:6] P03 Function Select

P0FSRL7	P0FSRL6	Description
0	0	I/O Port
0	1	Reserved
1	0	T0O/PWM0O Function
1	1	AN5 Function

P0FSRL[5:4] P02 Function Select

P0FSRL5	P0FSRL4	Description
0	0	I/O Port(EINT0 function possible when input)
0	1	SCL Function
1	0	Reserved
1	1	Reserved

P0FSRL[3:2] P01 Function Select

P0FSRL3	P0FSRL2	Description
0	0	I/O Port
0	1	SDA Function
1	0	Reserved
1	1	AN4 Function

P0FSRL[1:0] P00 Function Select

P0FSRL1	P0FSRL0	Description
0	0	I/O Port(EC0 function possible when input)
0	1	SS Function
1	0	Reserved
1	1	AN3 Function

6.3 P1 port

6.3.1 P1 port description

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

6.3.2 Register description for P1

P1 (P1 Data Register): 88H

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

Initial value: 00H

P1[7:0] I/O Data

P1IO (P1 Direction Register): B1H

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

Initial value: 00H

P1IO[7:0] P1 Data I/O Direction

0 Input

1 Output

NOTE: EC1/EINT1 function possible when input

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

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

Initial value: 00H

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

0 Disable

1 Enable

P1OD (P1 Open-drain Selection Register): 92H

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

Initial value: 00H

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

0 Push-pull output

1 Open-drain output

P1DB (P1 De-bounce Enable Register): DFH

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

Initial value: 00H

P17DB	Configure De-bounce of P17 Port	
	0	Disable
	1	Enable
P16DB	Configure De-bounce of P16 Port	
	0	Disable
	1	Enable
P15DB	Configure De-bounce of P15 Port	
	0	Disable
	1	Enable
P14DB	Configure De-bounce of P14 Port	
	0	Disable
	1	Enable
P13DB	Configure De-bounce of P13 Port	
	0	Disable
	1	Enable
P12DB	Configure De-bounce of P12 Port	
	0	Disable
	1	Enable
P11DB	Configure De-bounce of P11 Port	
	0	Disable
	1	Enable
P10DB	Configure De-bounce of P10 Port	
	0	Disable
	1	Enable

NOTES:

1. If the same level is not detected on enabled pin three or four times in a row at the sampling clock, the signal is eliminated as noise.
2. A pulse level should be input for the duration of 3 clock or more to be actually detected as a valid edge.
3. The port de-bounce is automatically disabled at stop mode and recovered after stop mode release.
4. Refer to the port 0 de-bounce enable register (P0DB) for the de-bounce clock of port 1.

P1FSRH (Port 1 Function Selection High Register): D5H

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

Initial value: 00H

P1FSRH[7:6] P17 Function Select

P1FSRH7	P1FSRH6	Description
0	0	I/O Port
0	1	Reserved
1	0	PWM1OB Function
1	1	AN0 Function

P1FSRH[5:4] P16 Function Select

P1FSRH5	P1FSRH4	Description
0	0	I/O Port
0	1	Reserved
1	0	Reserved
1	1	AN14 Function

P1FSRH[3:2] P15 Function Select

P1FSRH3	P1FSRH2	Description
0	0	I/O Port(EINT2 function possible when input)
0	1	RXD/MISO Function
1	0	Reserved
1	1	AN13 Function

P1FSRH[1:0] P14 Function Select

P1FSRH1	P1FSRH0	Description
0	0	I/O Port
0	1	TXD/MOSI Function
1	0	Reserved
1	1	AN12 Function

P1FSRL (Port 1 Function Selection Low Register): D4H

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

Initial value: 00H

P1FSRL[7:6] P13 Function Select

P1FSRL7	P1FSRL6	Description
0	0	I/O Port(EC1/EINT1 function possible when input)
0	1	XCK Function
1	0	Reserved
1	1	AN11 Function

P1FSRL[5:4] P12 Function Select

P1FSRL5	P1FSRL4	Description
0	0	I/O Port
0	1	SS Function
1	0	T1O/PWM1O Function
1	1	Reserved

P1FSRL[3:2] P11 Function Select

P1FSRL3	P1FSRL2	Description
0	0	I/O Port
0	1	SDA Function
1	0	T2O/PWM2O Function
1	1	Reserved

P1FSRL[1:0] P10 Function Select

P1FSRL1	P1FSRL0	Description
0	0	I/O Port
0	1	reserved
1	0	Reserved
1	1	AN10 Function

PCI1 (Port Change Interrupt Register): 94H

7	6	5	4	3	2	1	0
PCI17	PCI16	PCI15	PCI14	PCI13	PCI12	PCI11	PCI10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

PCI[7:0] Select PCI interrupt enable or disable of P1x

0	Disable
1	Enable

6.4 P2 port

6.4.1 P2 port description

P2 is an 8-bit I/O port. P2 control registers consist of P2 data register (P2), P2 direction register (P2IO), P2 pull-up resistor selection register (P2PU) andP2 open-drain selection register (P2OD).Refer to the port function selection registers for the P2 function selection.

6.4.2 Register description for P2

P2 (P2 Data Register): 90H

7	6	5	4	3	2	1	0
-	-	-	-	-	P22	P21	P20
-	-	-	-	-	R/W	R/W	R/W

Initial value: 00H

P2[2:0]	I/O Data

P2IO (P2 Direction Register): B9H

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

Initial value: 00H

P2IO[2:0]	P2 Data I/O Direction	
	0	Input
	1	Output

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

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

Initial value: 00H

P2PU[2:0]	Configure Pull-up Resistor of P2 Port	
	0	Disable
	1	Enable

P2OD (P2 Open-drain Selection Register): 93H

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

Initial value: 00H

P2OD[2:0]	Configure Open-drain of P2 Port	
	0	Push-pull output
	1	Open-drain output

P2FSR (Port 2 Function Selection Register): D6H

7	6	5	4	3	2	1	0
-	-	-	-	P2FSR3	P2FSR2	P2FSR1	P2FSR0
-	-	-	-	R/W	R/W	R/W	R/W

Initial value: 00H

P2FSR[3:2] P21 Function Select

P2FSR3	P2FSR2	Description
0	0	I/O Port
0	1	reserved
1	0	Reserved
1	1	AN2 Function

P2FSR[1:0] P20 Function Select

P2FSR1	P2FSR0	Description
0	0	I/O Port
0	1	Reserved
1	0	T1O/PWM1O Function
1	1	AN1 Function

7 Interrupt controller

A96G174/A96S174 supports up to 14 interrupt sources. The interrupts have separate enable register bits associated with them, allowing software control. In addition, they have four levels of priority assigned to themselves.

A non-maskable interrupt source is always enabled with a higher priority than any other interrupt sources, and is not controllable by software.

Interrupt controller of A96G174/A96S174 has following features:

- Request receive from the 14 interrupt sources
- 2 groups of priority
- 4 levels of priority
- Multi Interrupt possibility
- A request of higher priority level is served first, when multiple requests of different priority levels are received simultaneously.
- Each interrupt source can be controlled by EA bit and each IEx bit
- Interrupt latency of 3 to 9 machine cycles in single interrupt system

A non-maskable interrupt is always enabled, while maskable interrupts are enabled through four pairs of interrupt enable registers (IE, IE1, IE2, and IE3). Each bit of IE, IE1, IE2, IE3 register individually enables/disables the corresponding interrupt source. Overall control is provided by bit 7 of IE (EA). When EA is set to '0', all interrupts are disabled: when EA is set to '1', interrupts are individually enabled or disabled through the other bits of the interrupt enable registers. The EA bit is always cleared to '0' jumping to an interrupt service vector and set to '1' executing the [RETI] instruction. The A96G174/A96S174 supports a four-level priority scheme. Each maskable interrupt is individually assigned to one of four priority levels according to IP and IP1.

Default interrupt mode is level-trigger mode basically, but if needed, it is possible to change to edge-trigger mode. Figure 15 shows the Interrupt Group Priority Level that is available for sharing interrupt priority. Priority of a group is set by two bits of interrupt priority registers (one bit from IP, another one from IP1). Interrupt service routine serves higher priority interrupt first. If two requests of different priority levels are received simultaneously, the request of higher priority level is served prior to the lower one.

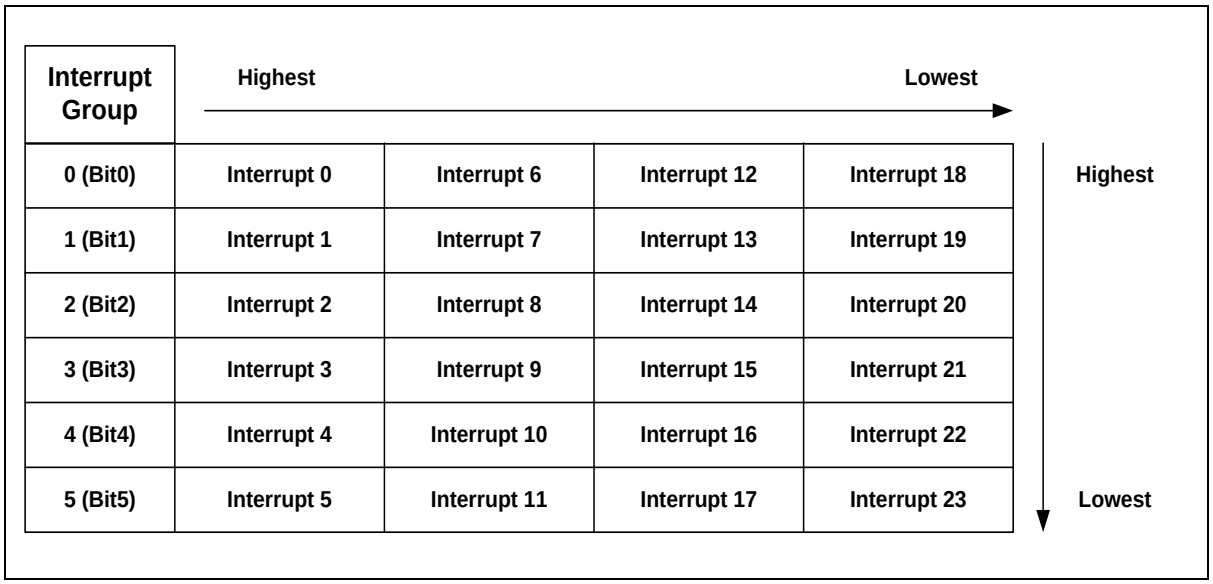


Figure 15. Interrupt Group Priority Level

7.1 External interrupt

The external interrupt on EINT0, EINT1 and EINT2 pins receive various interrupt request depending on the external interrupt polarity register (EIPOL) as shown in Figure 16. Also each external interrupt source has enable/disable bits. The External interrupt flag register (EIFLAG) provides the status of external interrupts.

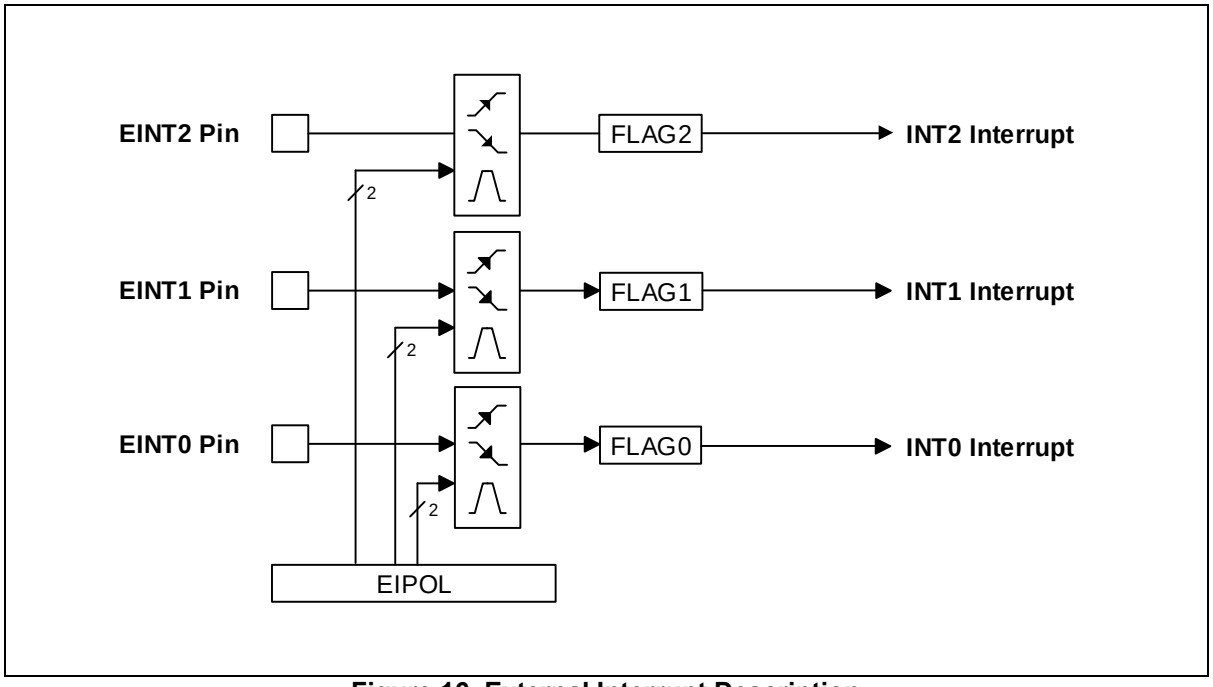


Figure 16. External Interrupt Description

7.2 Pin Change Interrupt

The pin change interrupt on P1 ports receive the both edge (Falling-edge and Rising-edge) interrupt request as shown in Figure 17. Also each pin change interrupt source had enable setting bits. The FLAG (flag register) register provides the status of ports change interrupts.

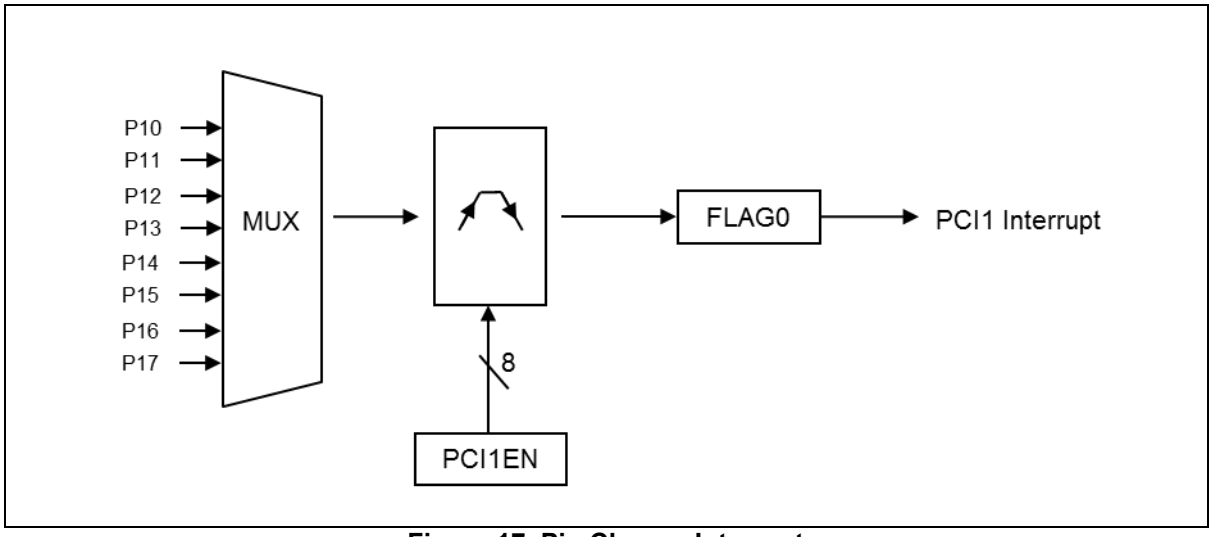
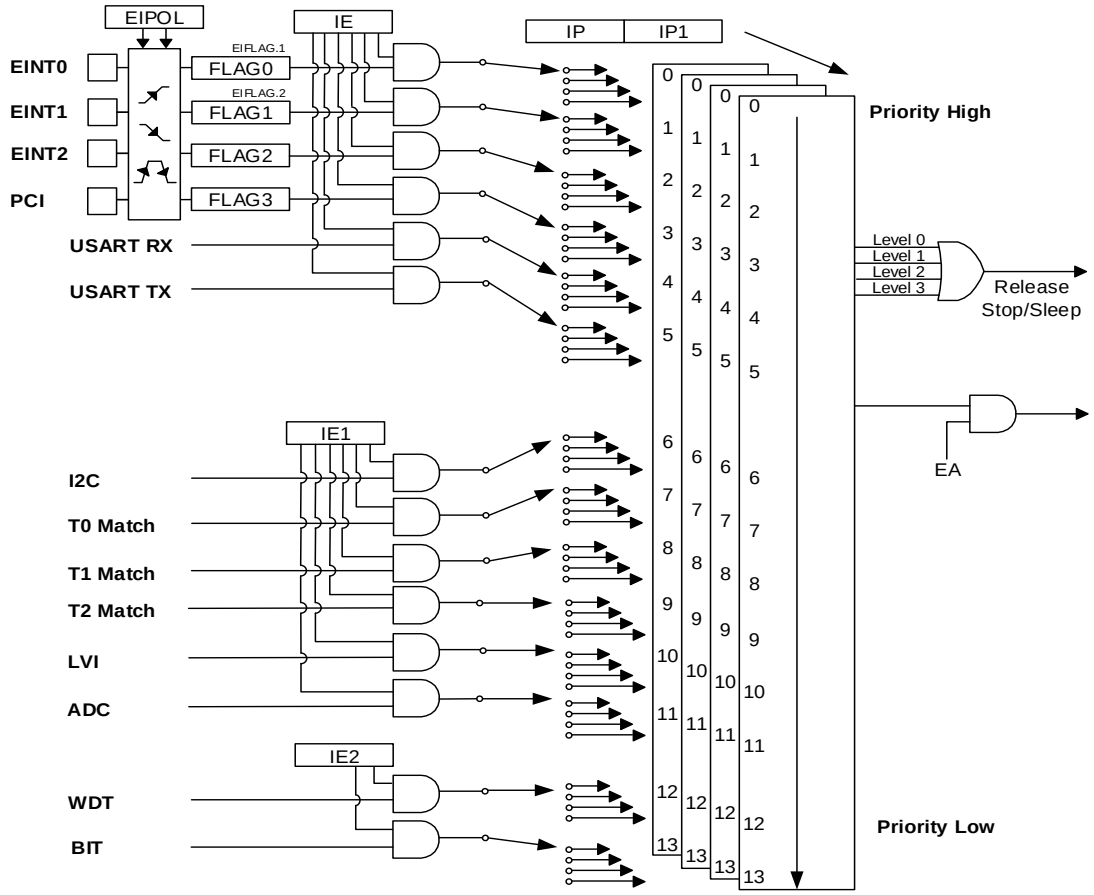


Figure 17. Pin Change Interrupt

7.3 Block diagram



NOTES:

1. The release signal for stop/idle mode may be generated by all interrupt sources which are enabled without reference to the priority level.
2. An interrupt request is delayed while data are written to IE, IE1, IE2, IE3, IP, IP1, and PCON register.

Figure 18. Interrupt Controller Block Diagram

7.4 Interrupt vector table

Interrupt controller of A96G174/A96S174 supports 14 interrupt sources as shown in Table 8. When interrupt is served, long call instruction (LCALL) is executed and program counter jumps to the vector address. All interrupt requests have their own priority order.

Table 8. Interrupt Vector Address Table

Interrupt source	Symbol	Interrupt enable bit	Priority	Mask	Vector address
Hardware Reset	RESETB	-	0	Non-Maskable	0000H
External Interrupt 0	INT0	IE.0	1	Maskable	0003H
External Interrupt 1	INT1	IE.1	2	Maskable	000BH
External Interrupt 2	INT2	IE.2	3	Maskable	0013H
PCI Interrupt	INT3	IE.3	4	Maskable	001BH
USART Rx Interrupt	INT4	IE.4	5	Maskable	0023H
USART Tx Interrupt	INT5	IE.5	6	Maskable	002BH
I2C Interrupt	INT6	IE1.0	7	Maskable	0033H
T0 Match Interrupt	INT7	IE1.1	8	Maskable	003BH
T1 Match Interrupt	INT8	IE1.2	9	Maskable	0043H
T2 Match Interrupt	INT9	IE1.3	10	Maskable	004BH
LVI Interrupt	INT10	IE1.4	11	Maskable	0053H
ADC Interrupt	INT11	IE1.5	12	Maskable	005BH
WDT Interrupt	INT12	IE2.0	13	Maskable	0063H
BIT Interrupt	INT13	IE2.1	14	Maskable	006BH

For maskable interrupt execution, EA bit must set '1' and specific interrupt must be enabled by writing '1' to associated bit in the IEx. If an interrupt request is received, the specific interrupt request flag is set to '1'. And it remains '1' until CPU accepts interrupt. If the interrupt is served, the interrupt request flag will be cleared automatically.

7.5 Interrupt sequence

An interrupt request is held until the interrupt is accepted or the interrupt latch is cleared to '0' by a reset or an instruction. Interrupt acceptance always generates at last cycle of the instruction. So instead of fetching the current instruction, CPU executes internally LCALL instruction and saves the PC at stack.

For the interrupt service routine, the interrupt controller gives the address of LJMP instruction to CPU. Since the end of the execution of current instruction, it needs 3 to 9 machine cycles to go to the interrupt service routine. The interrupt service task is terminated by the interrupt return instruction [RETI]. Once an interrupt request is generated, the following process in figure 18 is performed.

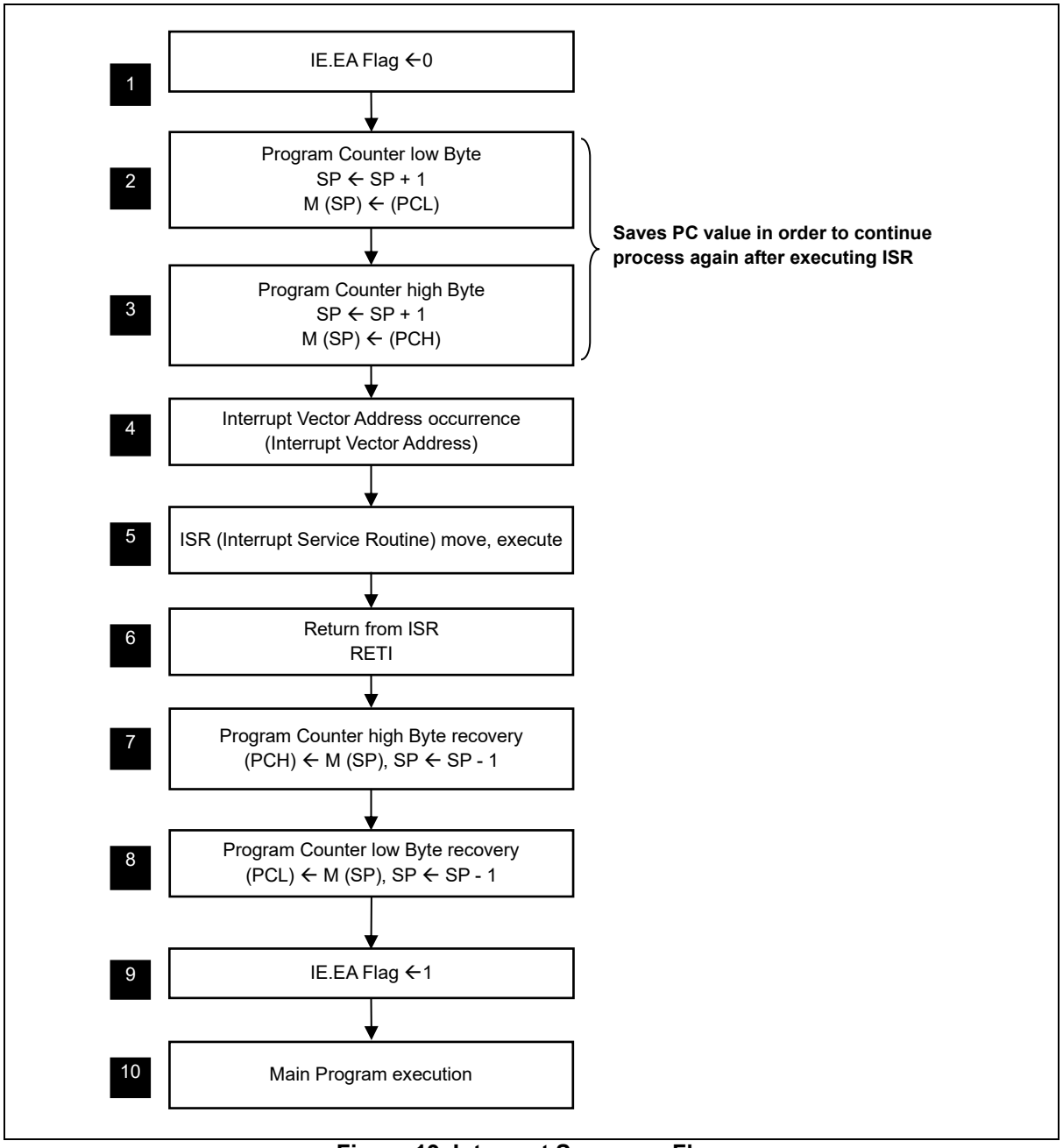


Figure 19. Interrupt Sequence Flow

7.6 Effective timing after controlling interrupt bit

Case A in Figure 20 shows the effective time after controlling Interrupt Enable Registers (IE, IE1, and IE2).

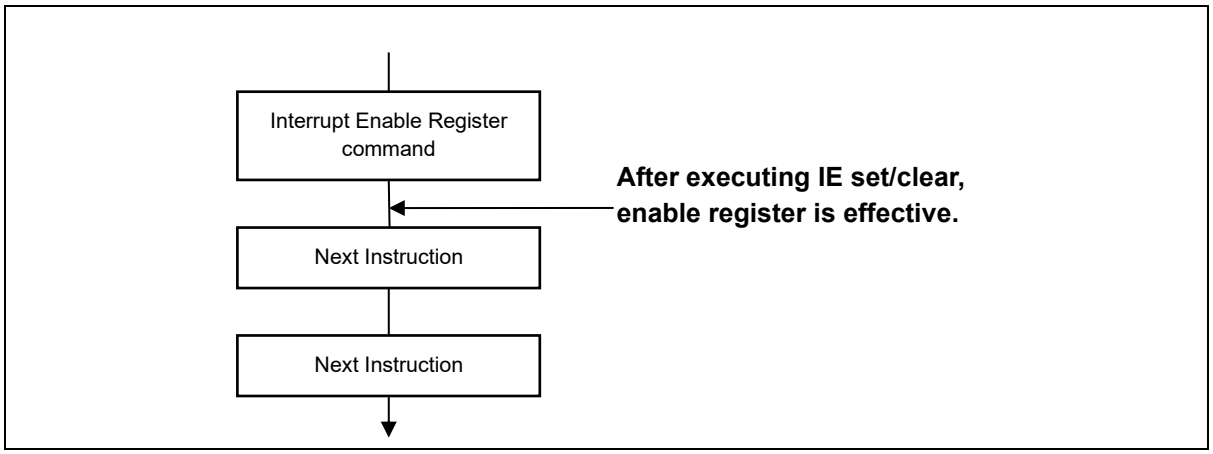


Figure 20. Effective Timing of Interrupt Enable Register

Case B in Figure 21 shows the effective time after controlling Interrupt Flag Registers.

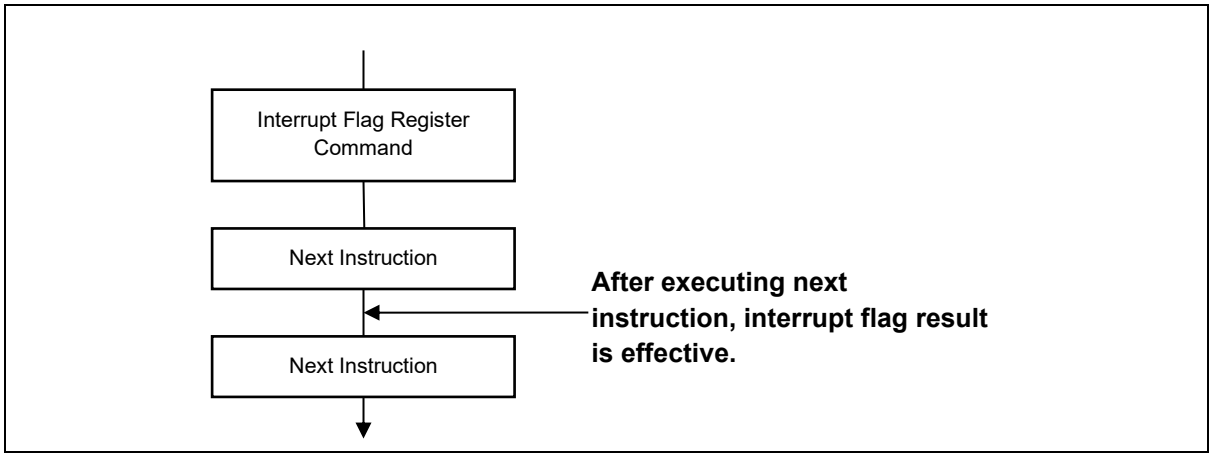


Figure 21. Effective Timing of Interrupt Flag Register

7.7 Multi-interrupt

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

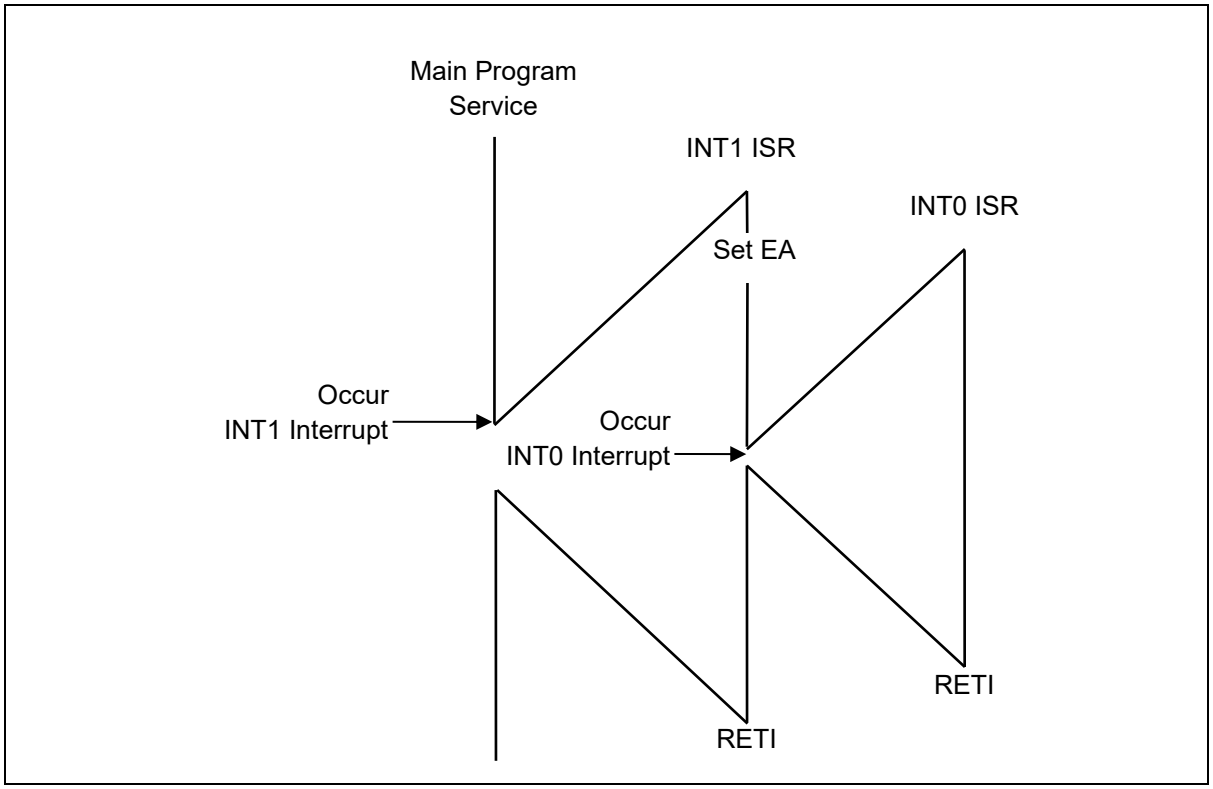


Figure 22. Effective Timing of Multi-Interrupt

Figure 22 shows an example of multi-interrupt processing. While INT1 is served, INT0 which has higher priority than INT1 is occurred. Then INT0 is served immediately and then the remaining part of INT1 service routine is executed. If the priority level of INT0 is same or lower than INT1, INT0 will be served after the INT1 service has completed.

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

7.8 Interrupt enable accept timing

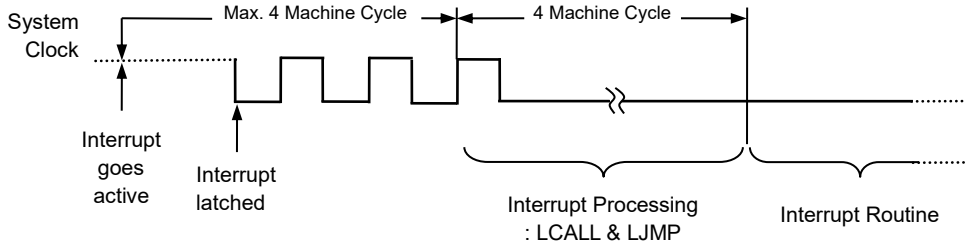


Figure 23. Interrupt Response Timing Diagram

7.9 Interrupt service routine address

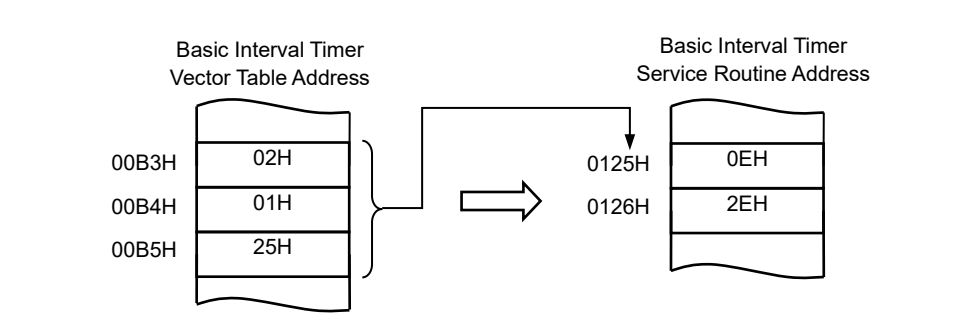


Figure 24. Correspondence between Vector Table Address and the Entry Address of ISR

7.10 Saving/restore general purpose registers

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

Interrupt_Processing:
          .
          .

          POP   ACC
          POP   B
          POP   DPH
          POP   DPL
          POP   PSW
          RETI
```

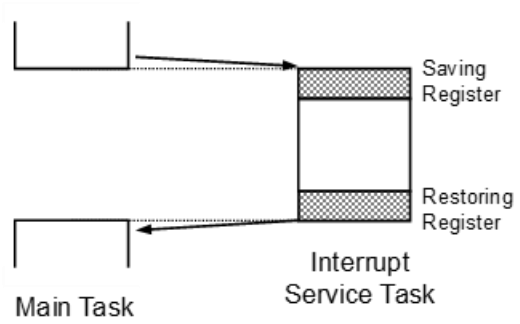
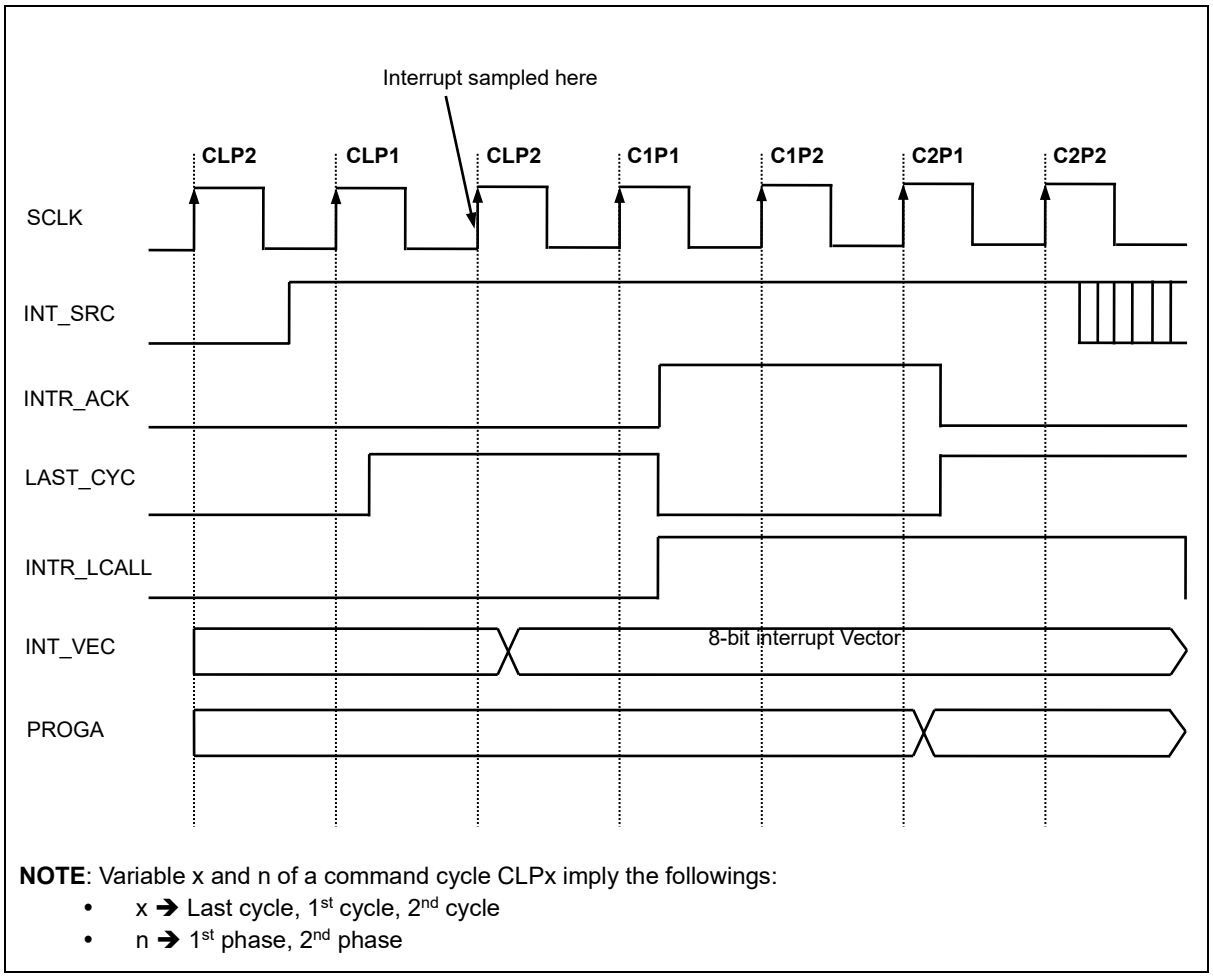


Figure 25. Saving/Restore Process Diagram and Sample Source

7.11 Interrupt timing



NOTE: Variable x and n of a command cycle CLPx imply the followings:

- x ➔ Last cycle, 1st cycle, 2nd cycle
- n ➔ 1st phase, 2nd phase

Figure 26. Timing Chart of Interrupt Acceptance and Interrupt Return Instruction

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

7.12 Interrupt register overview

7.12.1 Interrupt Enable Register (IE, IE1, and IE2)

Interrupt enable register consists of global interrupt control bit (EA) and peripheral interrupt control bits. Total 14 peripherals are able to control interrupt.

7.12.2 Interrupt Priority Register (IP and IP1)

14 interrupts are divided into 2 groups which have 6 interrupt sources respectively. A group can be assigned to 4 levels of interrupt priority using interrupt priority register. Level 3 is the highest priority, while level 0 is the lowest priority.

After a reset, IP and IP1 are cleared to '00H'. If interrupts have the same priority level, lower number interrupt is served first.

7.12.3 External Interrupt Flag Register (EIFLAG0 and EIFLAG1)

External Interrupt Flag 0 Register (EIFLAG0) and External Interrupt Flag 1 Register (EIFLAG1) are set to '1' when the external interrupt generating condition is satisfied. These flags are cleared when the interrupt service routine is executed. Alternatively, these flags can be cleared by writing '0' on to themselves.

7.12.4 External Interrupt Polarity Register (EIPOL0L, EIPOL0H, and EIPOL1)

External Interrupt Polarity0 high/low Register (EIPOL0H/L) and External Interrupt Polarity1 Register (EIPOL1) determines an edge type from rising edge, falling edge, and both edges of interrupt. Initially, default value is no interrupt at any edge.

7.12.5 Register map

Table 9. Interrupt Register Map

Name	Address	Direction	Default	Description
IE	A8H	R/W	00H	Interrupt Enable Register
IE1	A9H	R/W	00H	Interrupt Enable Register 1
IE2	AAH	R/W	00H	Interrupt Enable Register 2
IP	B8H	R/W	00H	Interrupt PriorityRegister
IP1	F8H	R/W	00H	Interrupt PriorityRegister 1
EIFLAG	C0H	R/W	00H	External Interrupt Flag Register
EIPOL	A4H	R/W	00H	External Interrupt Polarity Register

7.12.6 Interrupt register description

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 USART TX Interrupt	
	0	Disable
	1	Enable
INT4E	Enable or Disable USART RX Interrupt	
	0	Disable
	1	Enable
INT3E	Enable or Disable PCI Interrupt	
	0	Disable
	1	Enable
INT2E	Enable or Disable External Interrupt 2 (EINT2)	
	0	Disable
	1	Enable
INT1E	Enable or Disable External Interrupt 1(EINT1)	
	0	Disable
	1	Enable
INT0E	Enable or Disable External Interrupt 0 (EINT0)	
	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 ADC Interrupt	
	0	Disable
	1	Enable
INT10E	Enable or Disable LVI Interrupt	
	0	Disable
	1	Enable
INT9E	Enable or Disable T2 Interrupt	
	0	Disable
	1	Enable
INT8E	Enable or Disable T1 Interrupt	
	0	Disable
	1	Enable
INT7E	Enable or Disable T0 Interrupt	
	0	Disable
	1	Enable
INT6E	Enable or Disable I2C Interrupt	
	0	Disable
	1	Enable

IE2 (Interrupt Enable Register 2): AAH

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

Initial value: 00H

INT13E	Enable or Disable BIT Interrupt	
	0	Disable
	1	Enable
INT12E	Enable or Disable WDT Interrupt	
	0	Disable
	1	Enable

IP (Interrupt Priority Register): B8H

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

Initial value: 00H

IP1 (Interrupt Priority Register 1): F8H

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

Initial value: 00H

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

IP1x	IPx	Description
0	0	level 0 (lowest)
0	1	level 1
1	0	level 2
1	1	level 3 (highest)

EIFLAG0 (External Interrupt Flag0 Register): C0H

7	6	5	4	3	2	1	0
–	T0IFR	–	–	FLAG3	FLAG2	FLAG1	FLAG0
–	R/W	–	–	R/W	R/W	R/W	R/W

Initial value: 00H

T0IFR When T0 interrupt occurs, this bit becomes '1'. For clearing bit, write '0' to this bit or automatically clear by INT_ACK signal. Writing "1" has no effect.

0 T0 Interrupt no generation

1 T0 Interrupt generation

EIFLAG0[7:0] When an External Interrupt 0-7 is occurred, the flag becomes '1'.The flag is cleared only by writing '0' to the bit. So, the flag should be cleared by software.

0 External Interrupt0 ~ 7 not occurred

1 External Interrupt0 ~ 7 occurred

EIPOL (External Interrupt Polarity Register): A4H

7	6	5	4	3	2	1	0
–		POL2		POL1		POL0	
–	–	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

EIPOL0L[5:0] External interrupt (EINT0, EINT1, EINT2) polarity selection

POLn[1:0]		Description
0	0	No interrupt at any edge
0	1	Interrupt on rising edge
1	0	Interrupt on falling edge
1	1	Interrupt on both of rising and falling edge

Where n =0, 1, and 2

8 Clock generator

As shown in Figure 27, a clock generator produces basic clock pulses which provide a system clock for CPU and peripheral hardware.

Default system clock is 16MHz INT-RC Oscillator. To stabilize the system internally, 128kHz LOW INT-RC oscillator on POR is recommended.

Oscillators in the clock generator are introduced in the followings:

- Calibrated high internal RC oscillator (32MHz)
  - HSIRC OSC/2 (16MHz, default system clock)
  - HSIRC OSC/4 (8MHz)
  - HSIRC OSC/8 (4MHz)
  - HSIRC OSC/16 (2MHz)
  - HSIRC OSC/32 (1MHz)
  - HSIRC OSC/64 (0.5MHz)
- Internal LSIRC oscillator (128kHz)

8.1 Clock generator block diagram

In this section, a clock generator of A96G174/A96S174 is described in a block diagram.

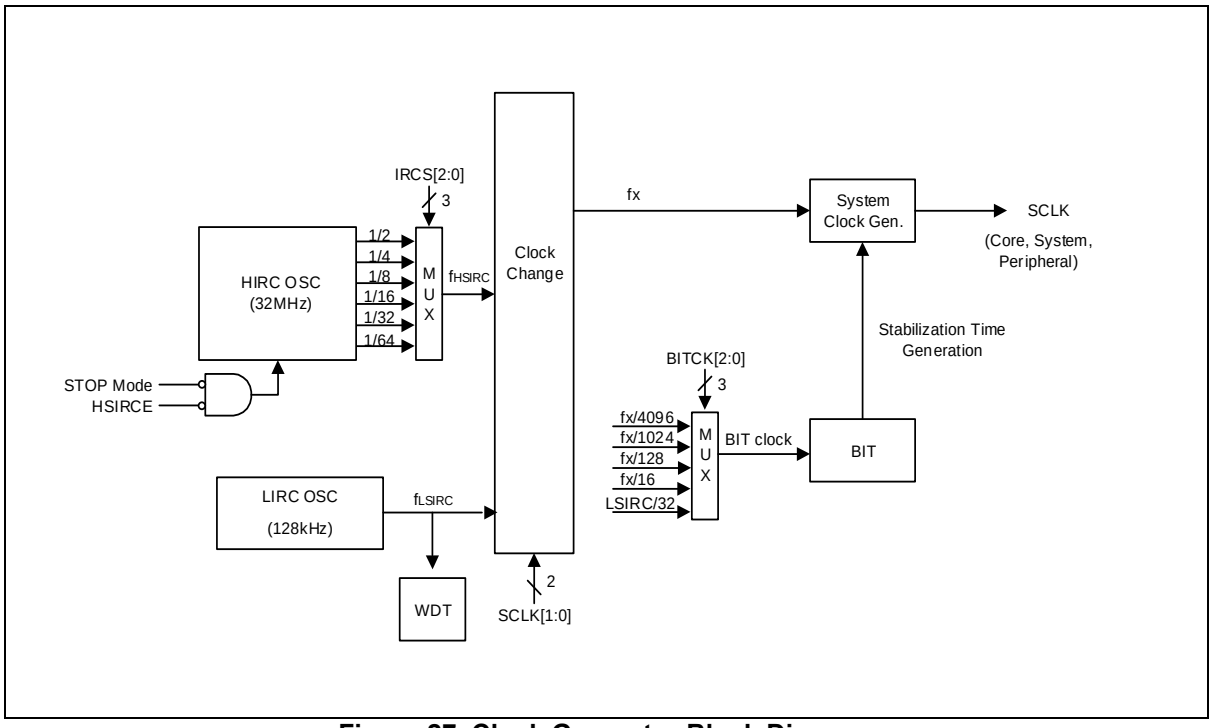


Figure 27. Clock Generator Block Diagram

8.2 Register map

Table 10. Clock Generator Register Map

Name	Address	Direction	Default	Description
SCCR	8AH	R/W	00H	System and Clock Control Register
OSCCR	C8H	R/W	28H	Oscillator Control Register

8.3 Register description

SCCR (System and Clock Control Register): 8AH

7	6	5	4	3	2	1	0
–	–	–	–	–	–	SCLK1	SCLK0
–	–	–	–	–	–	R/W	R/W

Initial value: 00H

SCLK [1:0] System Clock Selection Bit

SCLK1	SCLK0	Description
0	0	Internal 32MHz RC OSC (f_{HSIRC}) for system clock
0	1	Internal 128kHz RC OSC (f_{LSIRC}) for system clock
1	0	Reserved
1	1	Reserved

OSCCR (Oscillator Control Register): C8H

7	6	5	4	3	2	1	0
–	LSIRCE	IRCS2	IRCS1	IRCS0	HSIRCE	–	–
–	R/W	R/W	R/W	R/W	R/W	–	–

Initial value: 28H

LSIRCE Control the Operation of the Low Frequency (128kHz) internal RC Oscillator at Stop mode

0	Disable operation of LSIRC OSC
1	Enable operation of LSIRC OSC

IRCS[2:0] Internal RC Oscillator Post-divider Selection

IRCS2	IRCS1	IRCS0	Description
0	0	0	INT-RC/64 (0.5MHz)
0	0	1	INT-RC/32 (1MHz)
0	1	0	INT-RC/16 (2MHz)
0	1	1	INT-RC/8 (4MHz)
1	0	0	INT-RC/4 (8MHz)
1	0	1	INT-RC/2 (16MHz)
1	1	0	Test only
Other Values			reserved

HSIRCE Control the Operation of the High Frequency (32MHz) Internal RC Oscillator

0	Enable operation of HSIRC OSC
1	Disable operation of HSIRC OSC

9 Basic interval timer

A96G174/A96S174 has a free running 8-bit Basic Interval Timer (BIT). BIT generates the time base for watchdog timer counting, and provides a basic interval timer interrupt (BITIFR).

BIT of A96G174/A96S174 features the followings:

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

9.1 BIT block diagram

In this section, basic interval timer of A96G174/A96S174 is described in a block diagram.

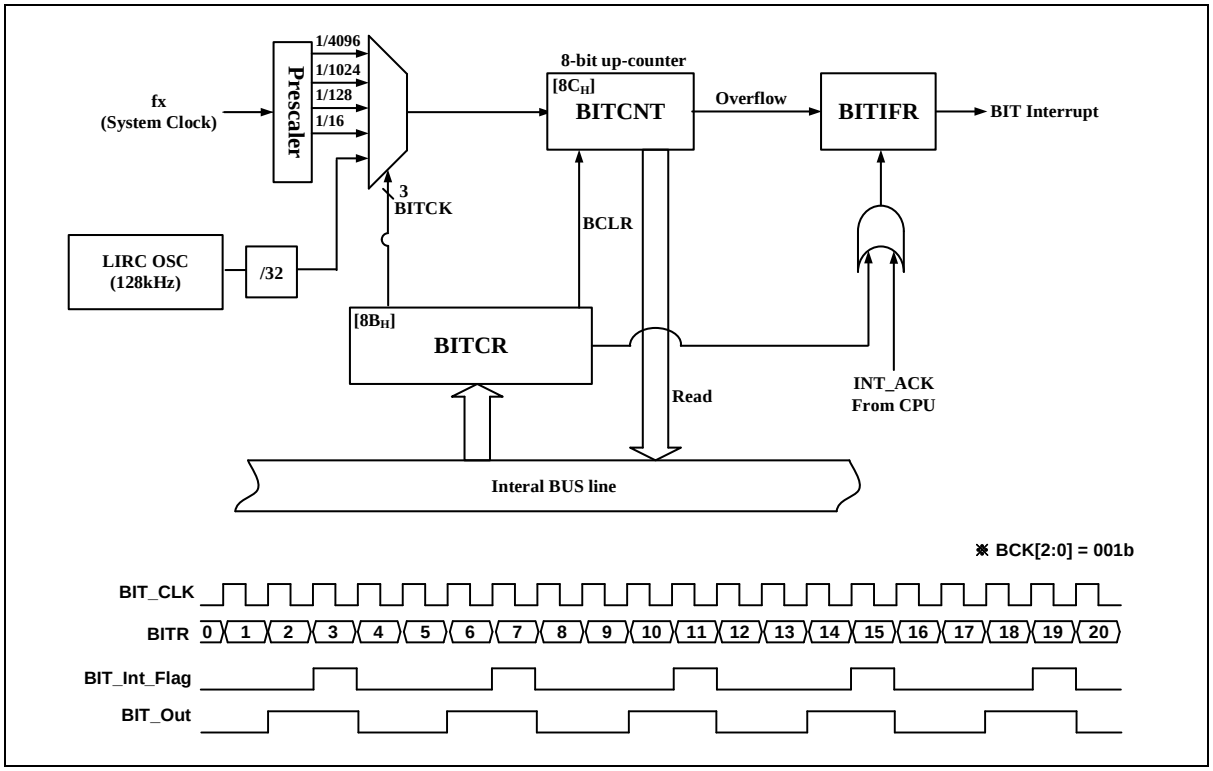


Figure 28. Basic Interval Timer Block Diagram

9.2 BIT register map

Table 11. Basic Interval Timer Register Map

Name	Address	Direction	Default	Description
BITCNT	8CH	R	00H	Basic Interval Timer Counter Register
BITCR	8BH	R/W	45H	Basic Interval Timer Control Register

9.3 BIT register description

BITCNT (Basic Interval Timer Counter Register): 8CH

7	6	5	4	3	2	1	0
BITCNT7	BITCNT6	BITCNT5	BITCNT4	BITCNT3	BITCNT2	BITCNT1	BITCNT0
R	R	R	R	R	R	R	R

Initial value: 00H

BITCNT[7:0] BIT Counter

BITCR (Basic Interval Timer Control Register): 8BH

7	6	5	4	3	2	1	0
BITIFR	BITCK2	BITCK1	BITCK0	BCLR	BCK2	BCK1	BCK0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 45H

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

0 BIT interrupt no generation

1 BIT interrupt generation

BITCK[2:0] Select BIT clock source

BITCK2	BITCK1	BITCK0	Description
0	0	0	fx/4096
0	0	1	fx/1024
0	1	0	fx/128
0	1	1	fx/16
1	Other Values		LSIRC/32 (Default)

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

0 Free Running

1 Clear Counter

BCK[2:0] Select BIT overflow period

BCK2	BCK1	BCK0	Description (fx=LSIRC 128k)
0	0	0	0.5ms (BIT Clock * 2)
0	0	1	1ms (BIT Clock * 4)
0	1	0	2ms (BIT Clock * 8)
0	1	1	4ms (BIT Clock * 16)
1	0	0	8ms (BIT Clock * 32)
1	0	1	16ms (BIT Clock * 64) (default)
1	1	0	32ms (BIT Clock * 128)
1	1	1	64ms (BIT Clock * 256)

10 Watchdog timer

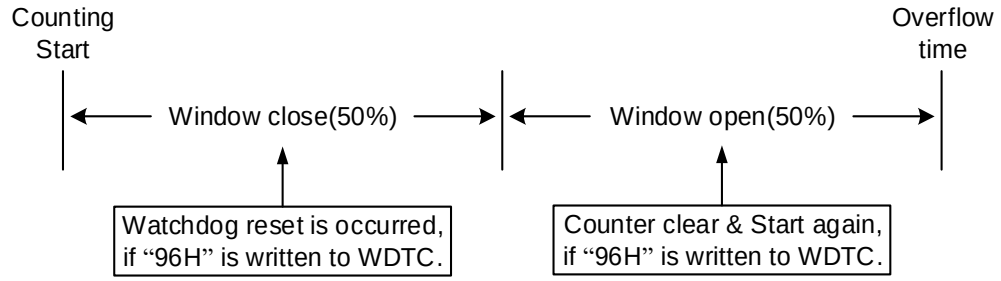
The watchdog timer rapidly detects the CPU malfunction such as endless looping caused by noise or something like that, and resumes the CPU to the normal state. The watchdog timer signal for malfunction detection can be used as either a CPU reset or an interrupt request. When the watchdog timer is not being used for malfunction detection, it can be used as a timer to generate an interrupt at fixed intervals. When 75% of the overflow time is reached, a watchdog interrupt can be generated. The overflow time of the watchdog timer can select by WDTOVF[2:0] of WDTCR. If an overflow occurs, an internal reset is generated. The WDTRC operation in the STOP/IDLE mode differs as follows depending on the setting value of WDTPDON. If WDTPDON = 0, the WDTRC operation stop in the STOP/IDLE mode and if WDTPDON = 1, the WDTRC operation in the STOP/IDLE mode. The watchdog timer operate on the 4kHz, based on clock 128kHz Ring oscillator clock.

Watchdog reset is occurred in the following cases:

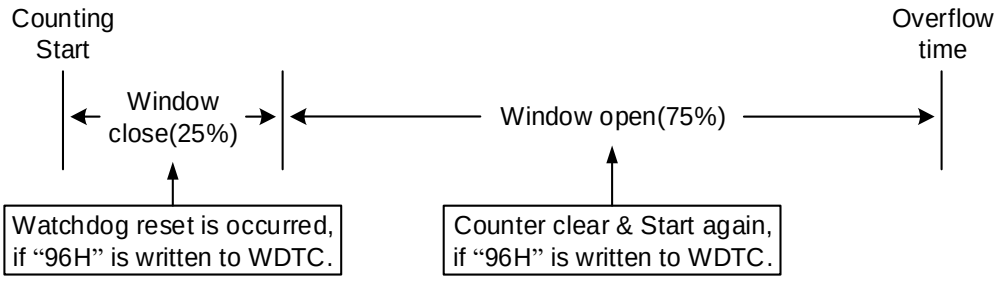
- When the watchdog timer counter overflows
- When the data except "96H" is written to the WDTC register
- When the data "96H" is written to the WDTC register during a window close period

10.1 Setting window open period of watchdog timer

1. WDT window open period is selected as 50%.



2. WDT window open period is selected as 75%.



3. WDT window open period is selected as 100%.

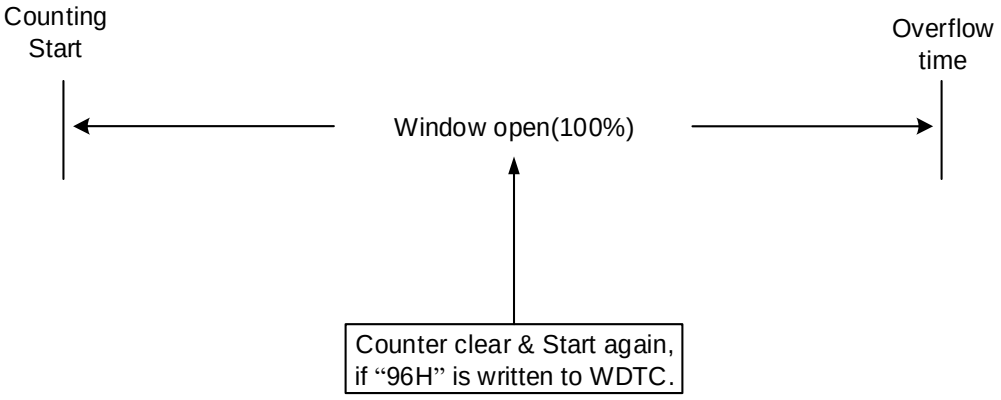


Figure 29. Watch Dog Timer Interrupt Timing Waveform

Table 12. Watch Dog Timer Register Map

Setting of window open period	Window close period	Window open period
50%, WINDOW[1:0]=00b & WDTPDON = 1	50%	50%
75%, WINDOW[1:0]=01b & WDTPDON = 1	25%	75%
100%, WINDOW[1:0]=10b & WDTPDON = 1	WDTCNT = "0000H"	100%
100%, WDTPDON = 0	WDTCNT = "0000H"	100%

10.2 WDT block diagram

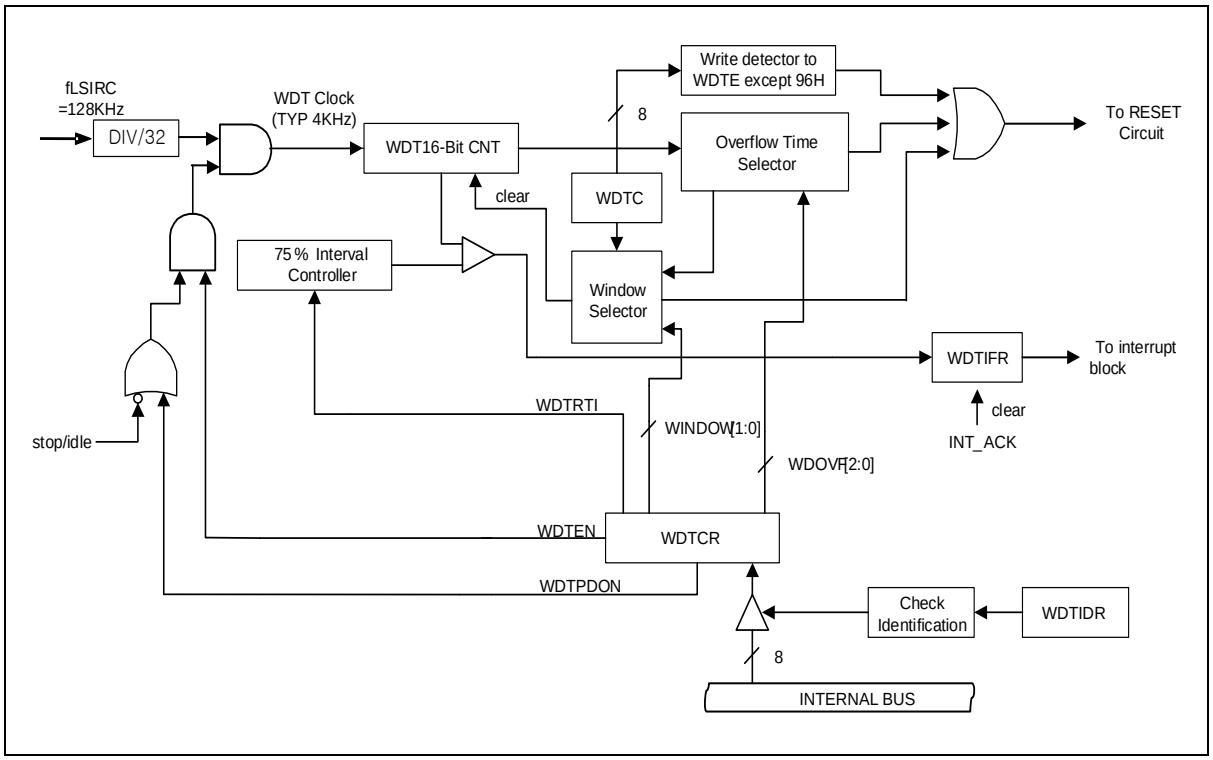


Figure 30. Watch Dog Timer Block Diagram

10.3 Register map

Table 13. Watchdog Timer Register Map

Name	Address	Direction	Default	Description
WDTC	1010H	R/W	00H	Watch Dog Timer Clear Register
WDTSR	1011H	R/W	00H	Watch Dog Timer Status Register
WDTIDR	8EH	W	00H	Watch Dog Timer Identification Register
WDTCR	8DH	R/W	07H	Watch Dog Timer Control Register
WDTCNTH	1012H	R	00H	Watch Dog Timer Count H Register
WDTCNTL	1013H	R	00H	Watch Dog Timer Count L Register

10.4 Register description

WDTCNTH (Watch Dog Timer Counter High Register: Read Case): 1012H

7	6	5	4	3	2	1	0
WDTCNT15	WDTCNT14	WDTCNT13	WDTCNT12	WDTCNT11	WDTCNT10	WDTCNT9	WDTCNT8
R	R	R	R	R	R	R	R

Initial value: 00H

WDTCNT[15:8] WDT Counter

WDTCNTL (Watch Dog Timer Counter Low Register: Read Case): 1013H

7	6	5	4	3	2	1	0
WDTCNT7	WDTCNT6	WDTCNT5	WDTCNT4	WDTCNT3	WDTCNT2	WDTCNT1	WDTCNT0
R	R	R	R	R	R	R	R

Initial value: 00H

WDTCNT[7:0] WDT Counter

WDTC (Watch Dog Timer Clear Register): 1010H

7	6	5	4	3	2	1	0
WDTC7	WDTC6	WDTC5	WDTC4	WDTC3	WDTC2	WDTC1	WDTC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

WDTC[7:0] WDT Counter clear

Others Reset occurs.

10010110 WDT counter clear and start again.

WDTSR (Watch Dog Timer Status Register): 1011H

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

Initial value: 00H

WSTATE Window Status

0 Close window

1 Open window

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

WDTIDR (Watch Dog Timer Identification Register: Write Case): 8EH

7	6	5	4	3	2	1	0
WDTID7	WDTID6	WDTID5	WDTID4	WDTID3	WDTID2	WDTID1	WDTID0
W	W	W	W	W	W	W	W

Initial value: 00H

WDTDR[7:0] WDT Identification for a WDTCR
Others No identification value.
01011001 Identification value for a WDTCR write.
(These bits are automatically cleared to logic '00H' immediately after WDTCR write.)

WDTCR (Watch Dog Timer Control Register): 8DH

7	6	5	4	3	2	1	0
WDTEN	WDTRTI	WDTPDON	WINDOW1	WINDOW0	WDOVF2	WDOVF1	WDOVF0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 07H

WDTEN Control WDT Operation
0 Disable(WDTRC Stop)
1 Enable

WDTRTI 3/4 Interval interrupt
0 Disable(WDT overflow reset used)
1 Enable

WDTPDON Operation on Stop/Idle Mode
0 WDTRC operation stop in Stop/Idle Mode
1 WDTRC operation in Stop/Idle Mode

WINDOW[1:0] Select WDT window open period

LOCKA1	LOCKA0	Description
0	0	50%
0	1	75%
1	0	100%
1	1	Not used

Note) The window open period is 100% when WDTPDON = 0, regardless the value of the WINDOW1 and WINDOW0 bits.

WDOVF[2:0] Select overflow time

WDOVF2	WDOVF1	WDOVF0	Description
0	0	0	$2^6/f_{WDT}$
0	0	1	$2^7/f_{WDT}$
0	1	0	$2^8/f_{WDT}$
0	1	1	$2^9/f_{WDT}$
1	0	0	$2^{11}/f_{WDT}$
1	0	1	$2^{13}/f_{WDT}$
1	1	0	$2^{14}/f_{WDT}$
1	1	1	$2^{16}/f_{WDT}$

11 Timer 0/1/2

11.1 Timer 0

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

Timer 0 operates in one of three modes introduced in the followings:

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

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

- TIMER0 clock source: f_X/2, 4, 8, 32, 128, 512, 2048 and EC0

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

Table 14. Timer 0 Operating Mode

T0EN	T0MS[1:0]	T0CK[2:0]	Timer 0
1	00	XXX	8-bit Timer/Counter Mode
1	01	XXX	8-bit PWM Mode
1	1X	XXX	8-bit Capture Mode

11.1.1 8-bit timer/counter mode

As shown in Figure 31, 8-bit timer/counter mode is selected by control register.

8-bit timer has counter and data registers. The counter register is increased by internal or external clock input. Timer 0 can use the input clock with one of 2, 4, 8, 32, 128, 512 and 2048 prescaler division rates (T0CK[2:0]). When both values of T0CNT and T0DR are identical to each other in timer 0, a match signal is generated and the interrupt of Timer 0 occurs.T0CNT value is automatically cleared by the match signal, and can be cleared by software (T0CC) too.

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

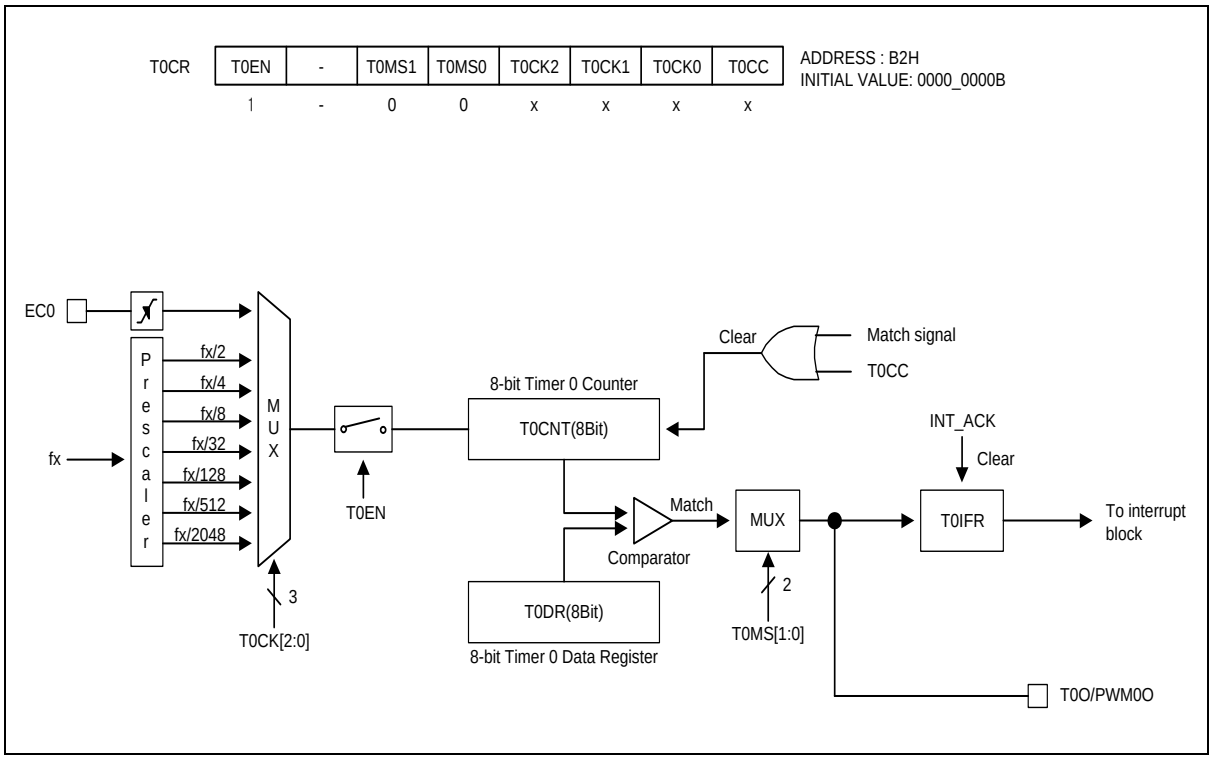


Figure 31. 8-bit Timer/Counter Mode for Timer 0

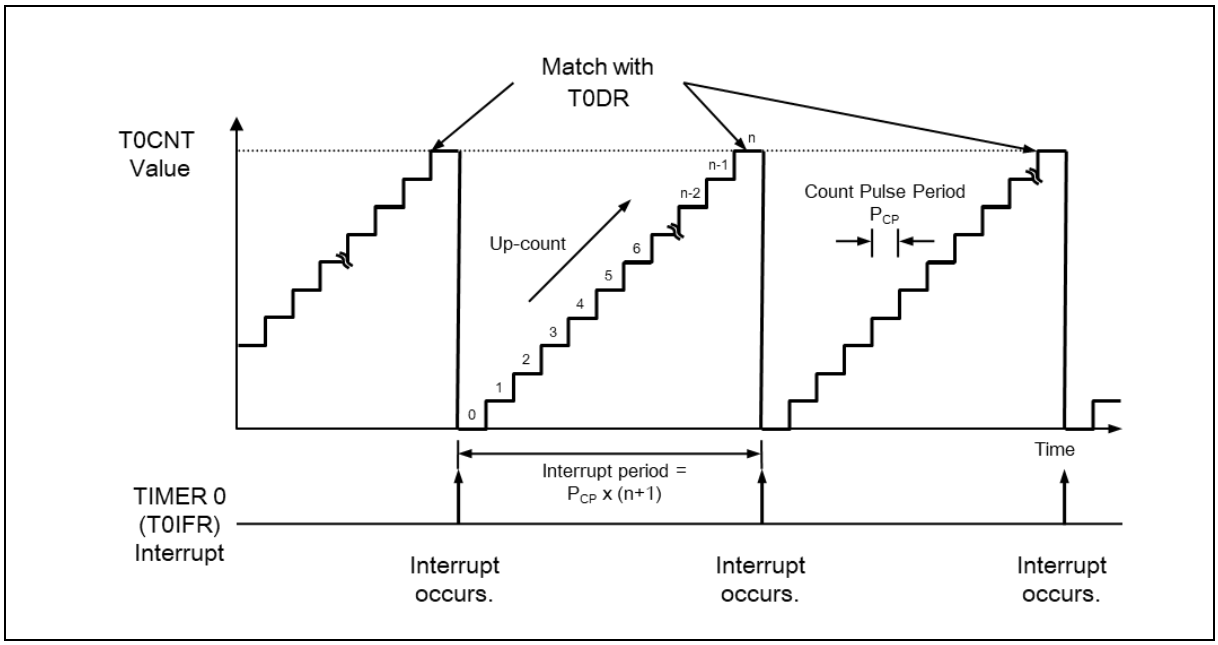


Figure 32. 8-bit Timer/Counter 0 Example

11.1.2 8-bit PWM mode

Timer 0 has a high speed PWM (Pulse Width Modulation) function. In PWM mode, T0O/PWM0O pin outputs up to 8-bit resolution PWM output. This pin should be configured as a PWM output by setting the T0O/PWM0O function by PxFSR bits.

In 8-bit timer/counter mode, a match signal is generated when the counter value is identical to the value of T0DR. When values of T0CNT and T0DR are identical to each other in timer 0, a match signal is generated and the interrupt of timer 0 occurs.

In PWM mode, the match signal does not clear the counter. Instead, it runs continuously, overflowing at "FFH". Then the counter continues incrementing from "00H". The timer 0 overflow interrupt is generated whenever a counter overflow occurs. T0CNT value is cleared by software (T0CC) bit.

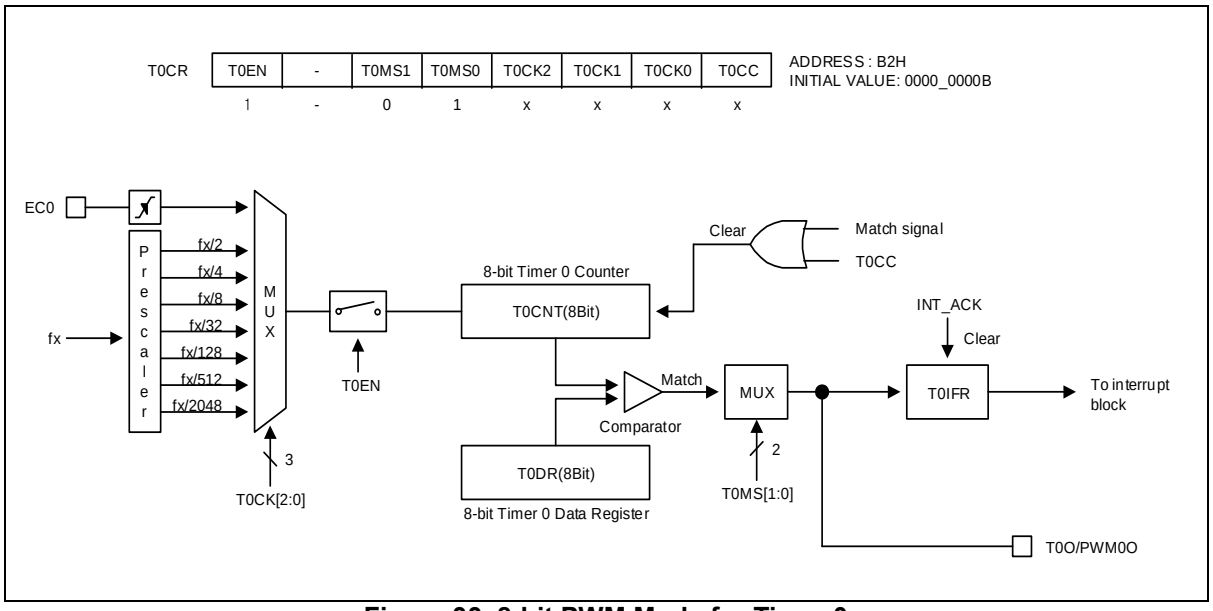


Figure 33. 8-bit PWM Mode for Timer 0

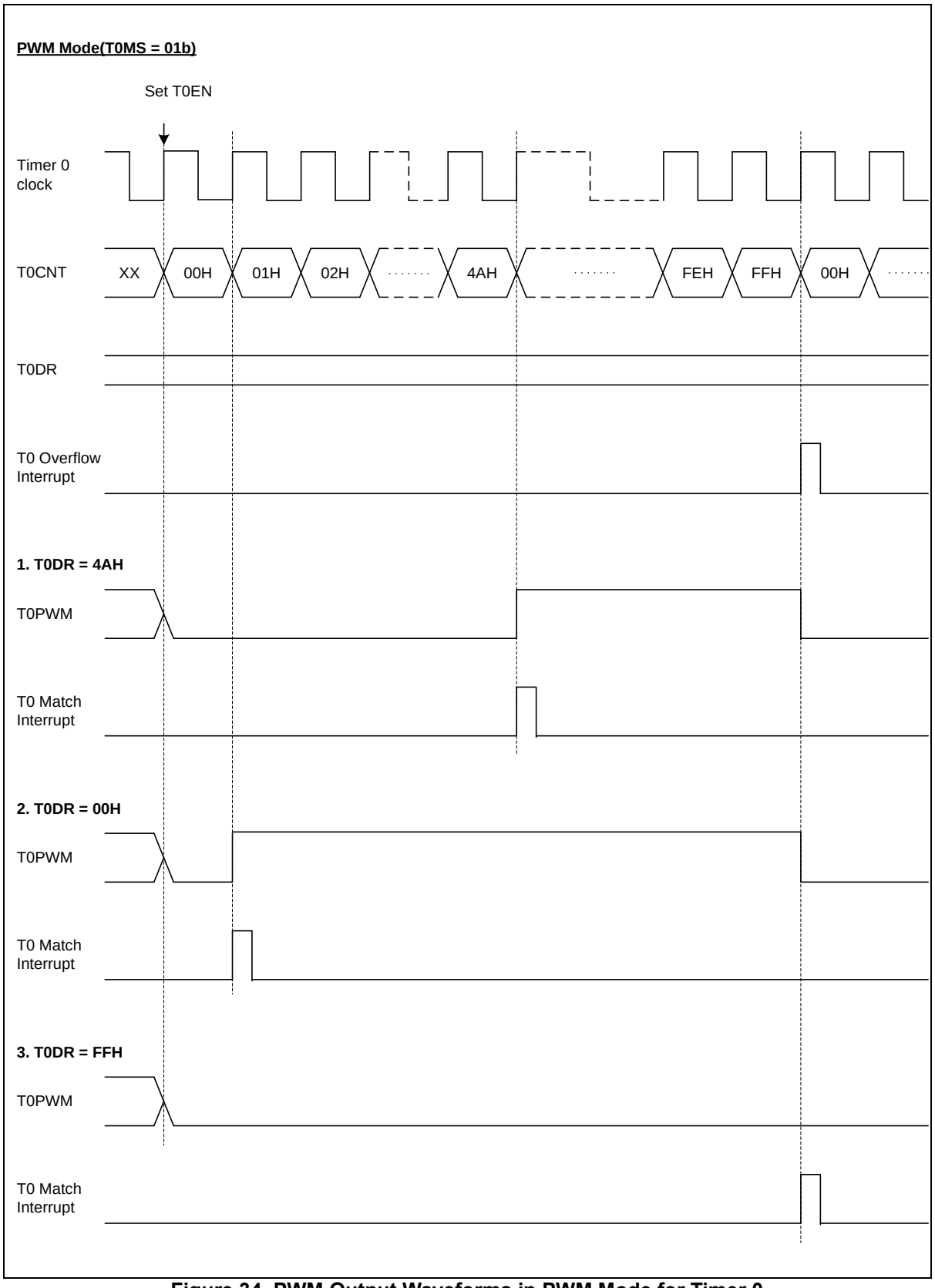


Figure 34. PWM Output Waveforms in PWM Mode for Timer 0

11.1.3 8-bit capture mode

Timer 0 capture mode is set by configuring T0MS[1:0] as '1x'. Clock source can use the internal/external clock. Basically, it has the same function as the 8-bit timer/counter mode has, and the interrupt occurs when T0CNT equals to T0DR. T0CNTvalue is automatically cleared by match signal and it can be also cleared by software (T0CC).

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

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

According to EIPOL registers setting, the external interrupt EINT0 function is chosen. Of course, the EINT0 pin must be set to an input port.

T0CDR and T0DR are in the same address. In the capture mode, reading operation readsT0CDR, not T0DR and writing operation will update T0DR.

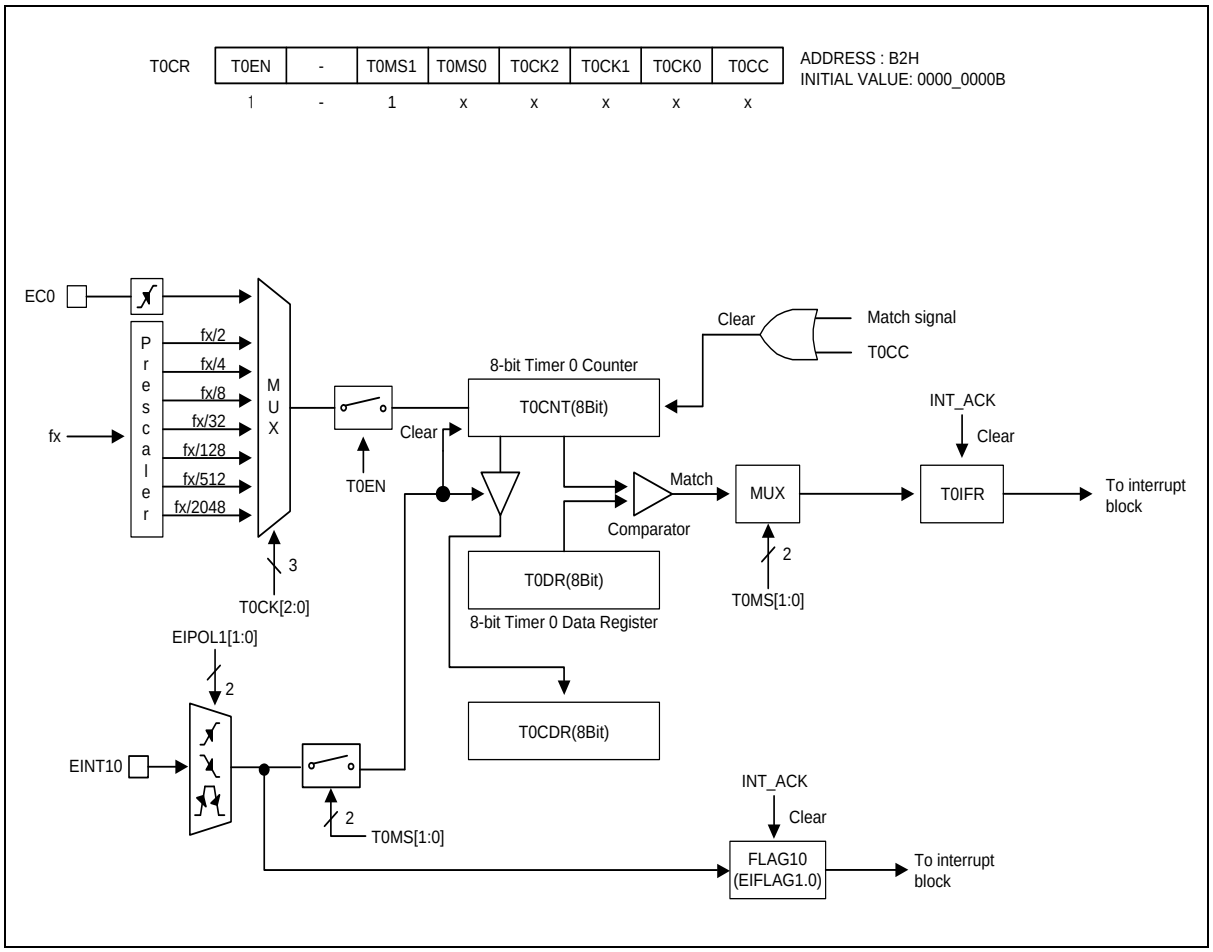


Figure 35. 8-bit Capture Mode for Timer 0

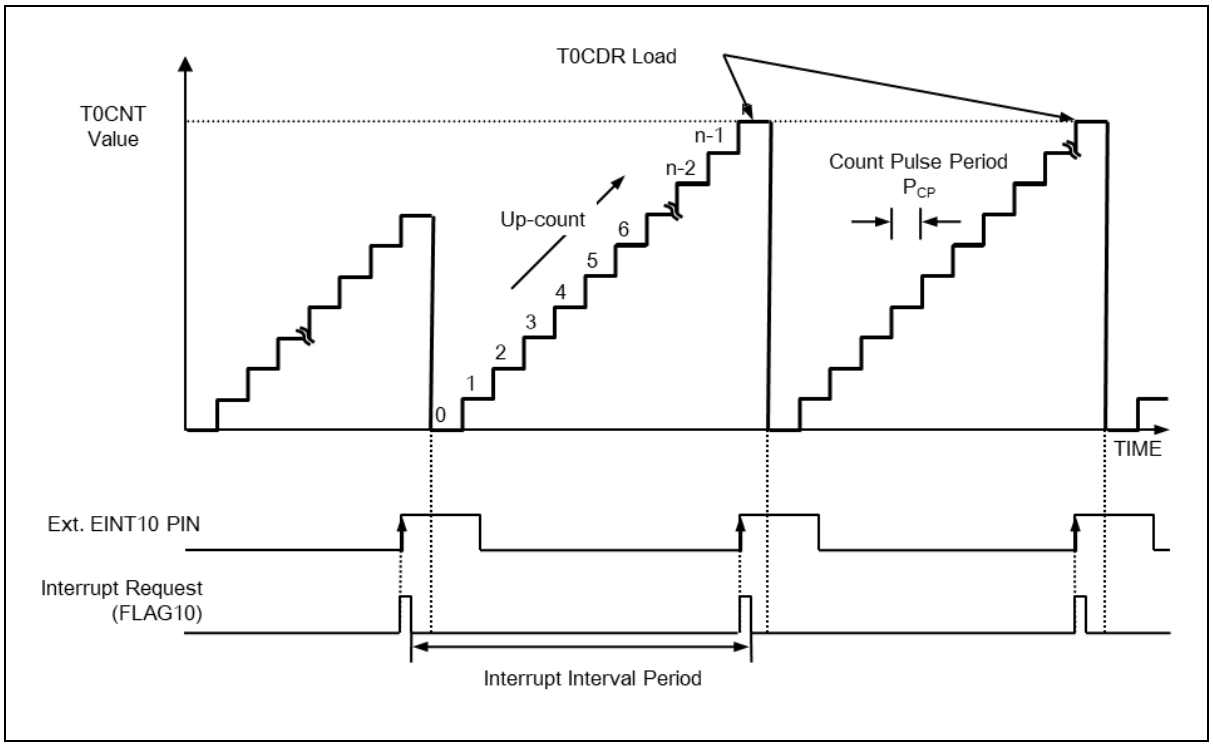


Figure 36. Input Capture Mode Operation for Timer 0

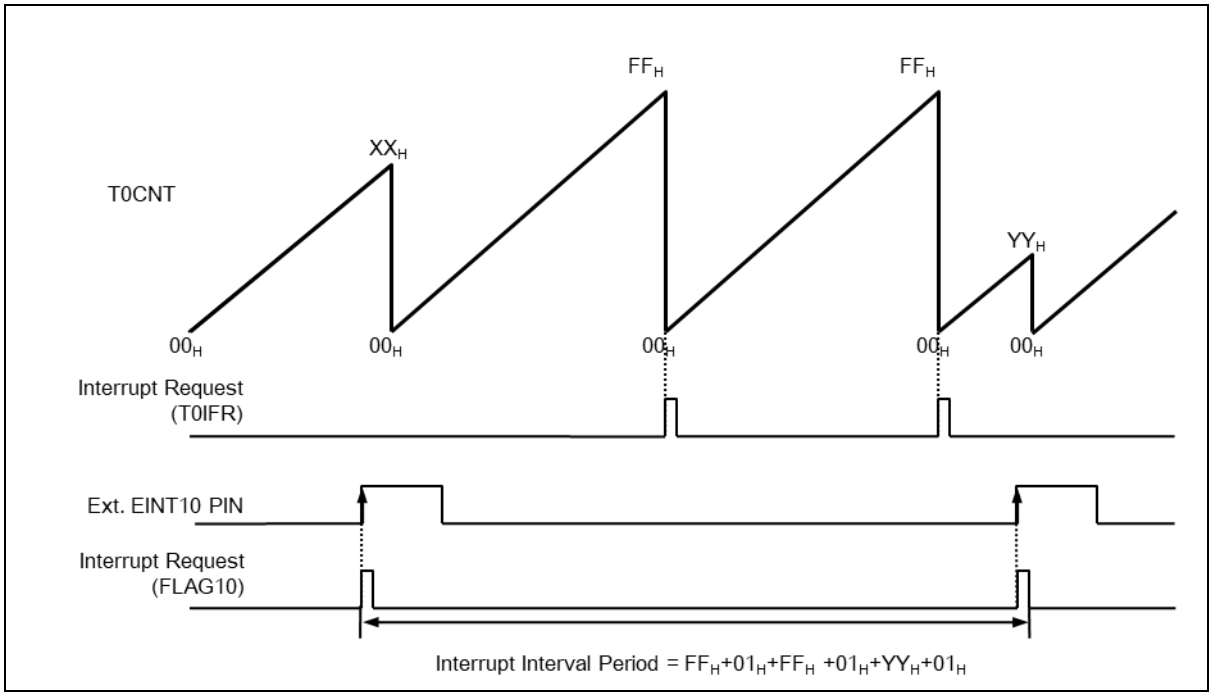


Figure 37. Express Timer Overflow in Capture Mode

11.1.4 Timer 0 block diagram

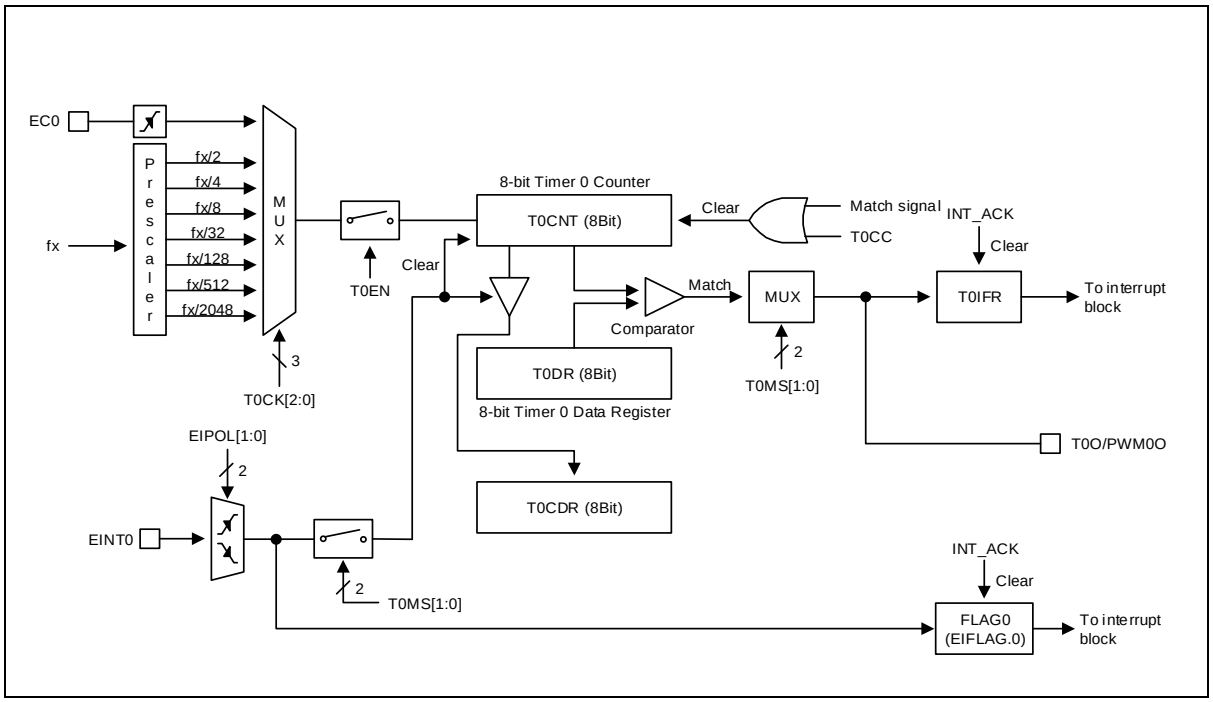


Figure 38. 8-bit Timer 0 Block Diagram

11.1.5 Register map

Table 15. Timer 0 Register Map

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

11.1.6 Register description

T0CNT (Timer 0 Counter Register): B3H

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

Initial value: 00H

T0CNT[7:0] T0 Counter

T0DR (Timer 0 Data Register): B4H

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

Initial value: FFH

T0DR[7:0] T0 Data

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

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

Initial value: 00H

T0CDR[7:0] T0 Capture Data

T0CR (Timer 0 Control Register): B2H

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

Initial value: 00H

T0EN	Control Timer 0			
	0	Timer 0 disable		
	1	Timer 0 enable		
T0MS[1:0]	Control Timer 0 Operation Mode			
	T0MS1	T0MS0	Description	
	0	0	Timer/counter mode	
	0	1	PWM mode	
	1	x	Capture mode	
T0CK[2:0]	Select Timer 0 clock source. fx is a system clock frequency			
	T0CK2	T0CK1	T0CK0	Description
	0	0	0	fx/2
	0	0	1	fx/4
	0	1	0	fx/8
	0	1	1	fx/32
	1	0	0	fx/128
	1	0	1	fx/512
	1	1	0	fx/2048
	1	1	1	External Clock (EC0)
T0CC	Clear timer 0 Counter			
	0	No effect		
	1	Clear the Timer 0 counter (When write, automatically cleared "0" after being cleared counter)		

NOTES:

1. Match Interrupt is generated in Capture mode.
2. Refer to the external interrupt flag 1 register (EIFLAG1) for the T0 interrupt flags.

11.2 Timer 1

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

Timer 1 operates in one of the following modes:

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

The timer/counter 1 uses an internal clock or an external clock (EC1) as an input clock source. The clock sources are introduced below, and one is selected by clock selection logic which is controlled by clock selection bits (T1CK[2:0]).

- TIMER 1 clock source: fX/1, 2, 4, 8, 64, 2048, HFO and EC1

In capture mode, the data is captured into input capture data register (T1BDRH/T1BDRL) by EINT11. Timer 1 results in the comparison between counter and data register through T1O port in timer/counter mode. In addition, Timer 1 outputs PWM waveform through PWM1Oport in the PPG mode.

Table 16. TIMER 1 Operating Modes

T1EN	P1FSRL[5:4]	T1MS[1:0]	T1CK[2:0]	Timer 1
1	11	00	XXX	16 Bit Timer/Counter Mode
1	00	01	XXX	16 Bit Capture Mode
1	11	10	XXX	16 Bit PPG Mode(one-shot mode)
1	11	11	XXX	16 Bit PPG Mode(repeat mode)

11.2.1 16-bit timer/counter mode

16-bit timer/counter mode is selected by control registers, and the 16-bit timer/counter has counter registers and data registers as shown in Figure 39. The counter register is increased by internal or external clock input. Timer 1 can use the input clock with one of 1, 2, 4, 8, 64, 2048 and Internal High Frequency Oscillator (HFO) prescaler division rates (T1CK[2:0]). When the value of T1CNTH, T1CNTL and the value of T1ADRH, T1ADRL are identical to each other in Timer 1, a match signal is generated and the interrupt of Timer1 occurs. The T1CNTH, T1CNTL value is automatically cleared by the match signal. It can be cleared by software (T1CC) too.

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

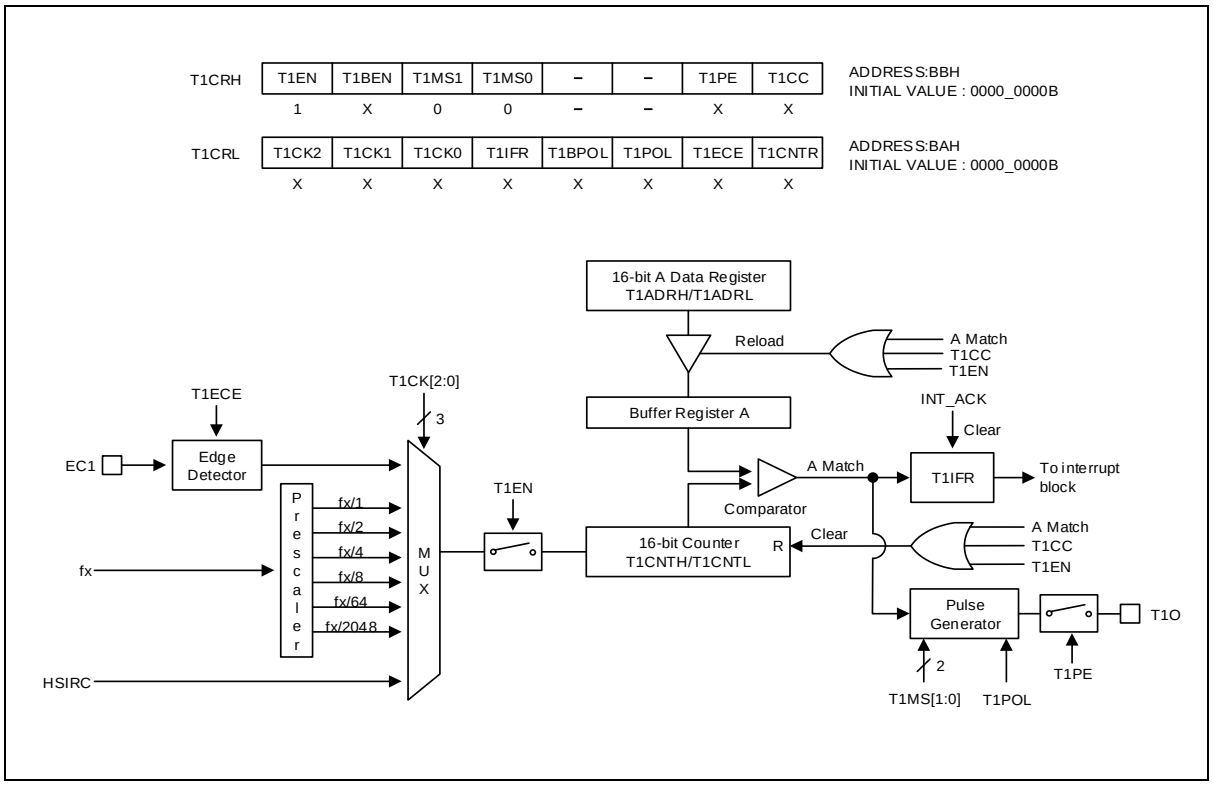


Figure 39. 16-bit Timer/Counter Mode of Timer 1

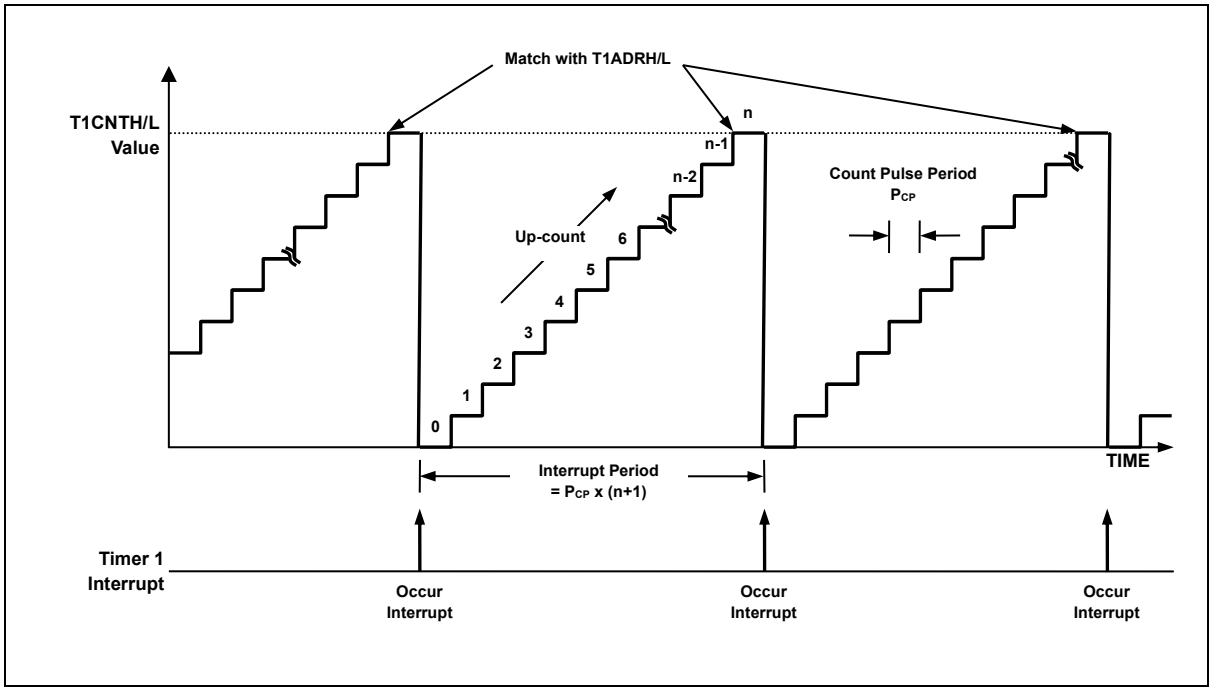


Figure 40. 16-bit Timer/Counter Mode Operation Example

11.2.2 16-bit capture mode

It uses an internal/external clock as a clock source. Basically, the 16-bit timer 1 capture mode has the same function as the 16-bit timer/counter mode, and the interrupt occurs when T1CNTH/T1CNTL is equal to T1ADRH/T1ADRL. The T1CNTH, T1CNTL values are automatically cleared by a match signal. It can be cleared by software (T1CC) too.

A timer interrupt in capture mode is very useful when the pulse width of captured signal is wider than the maximum period of timer. Capture result is loaded into T1BDRH/T1BDRL.

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

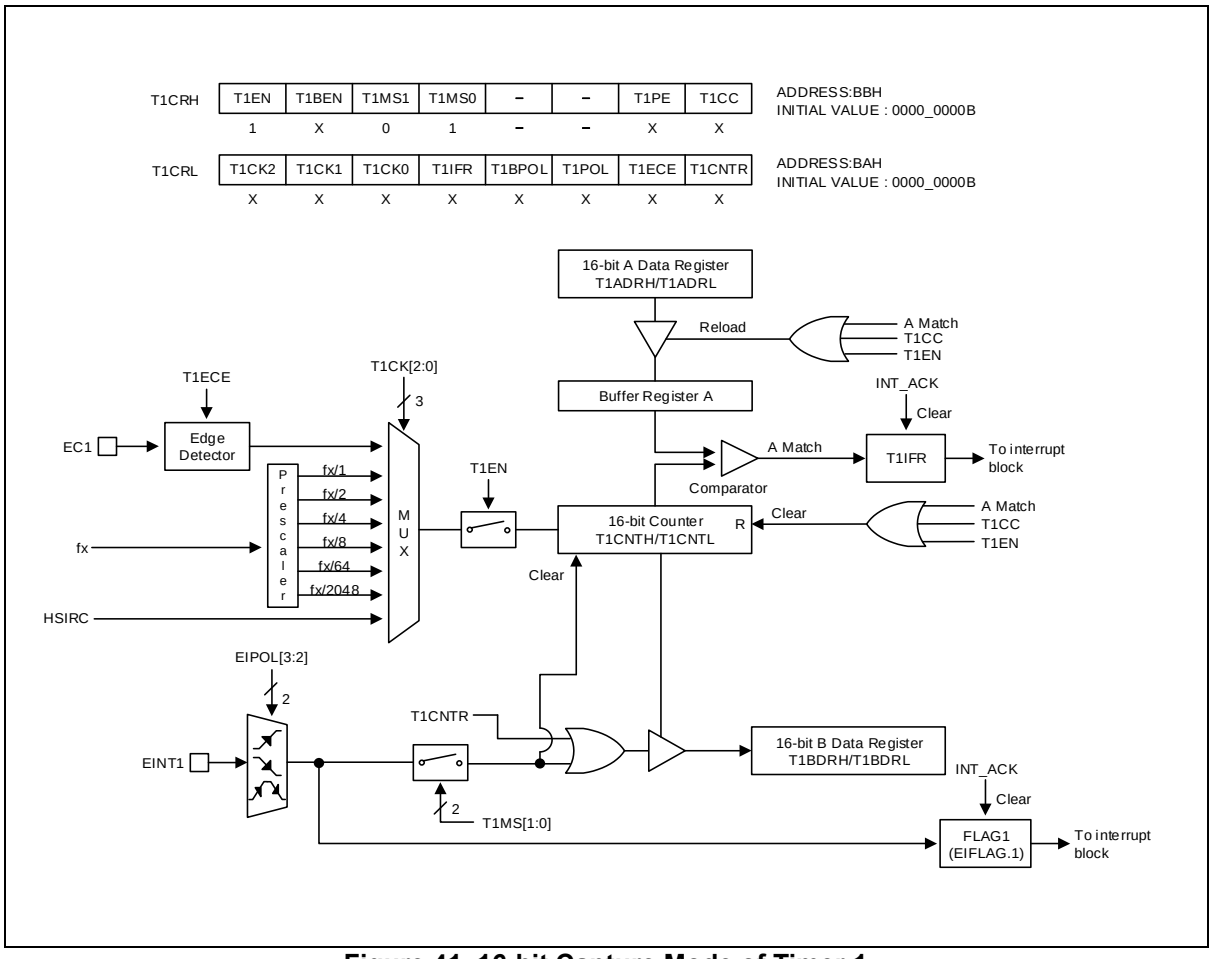


Figure 41. 16-bit Capture Mode of Timer 1

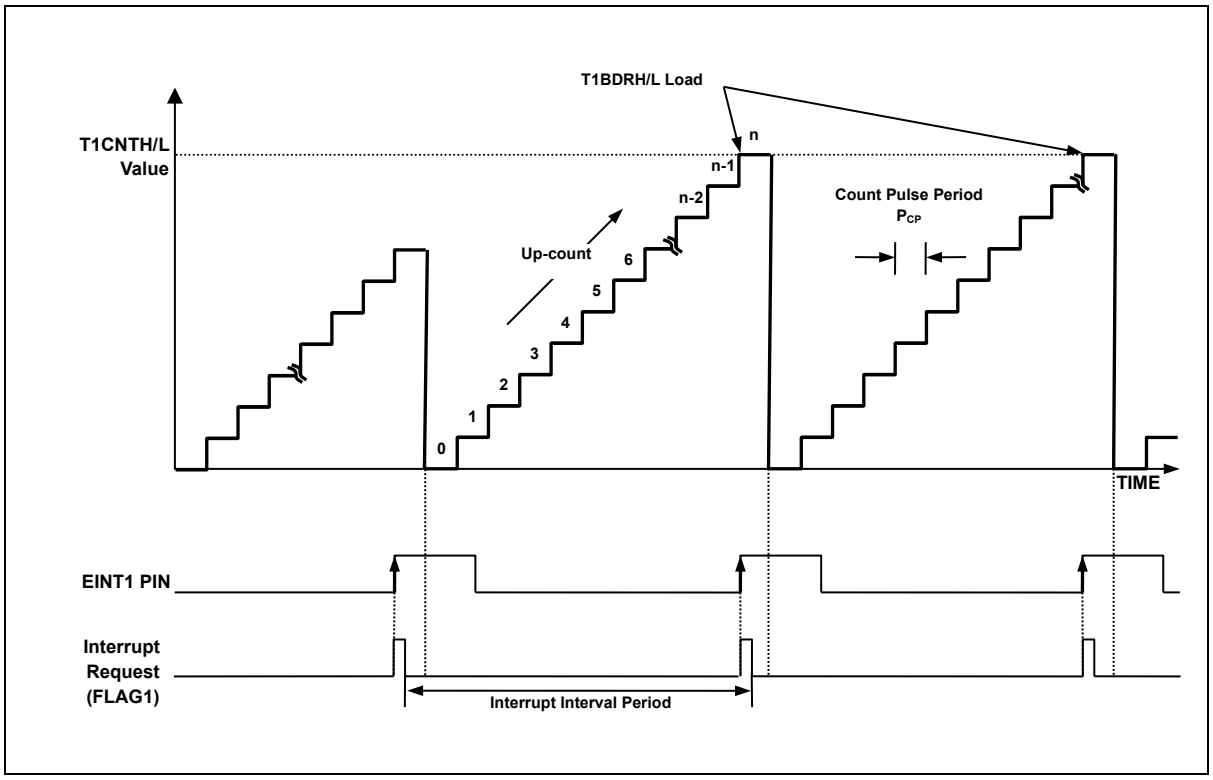


Figure 42. Input Capture Mode Operation for timer1

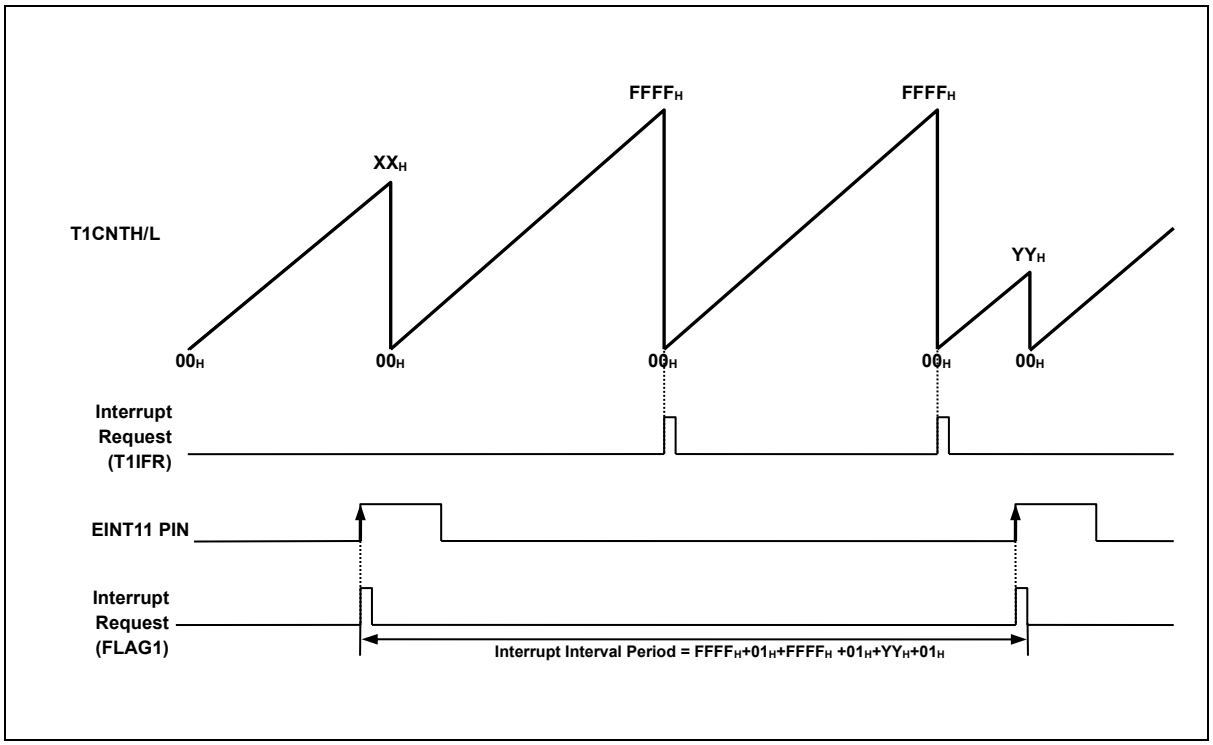
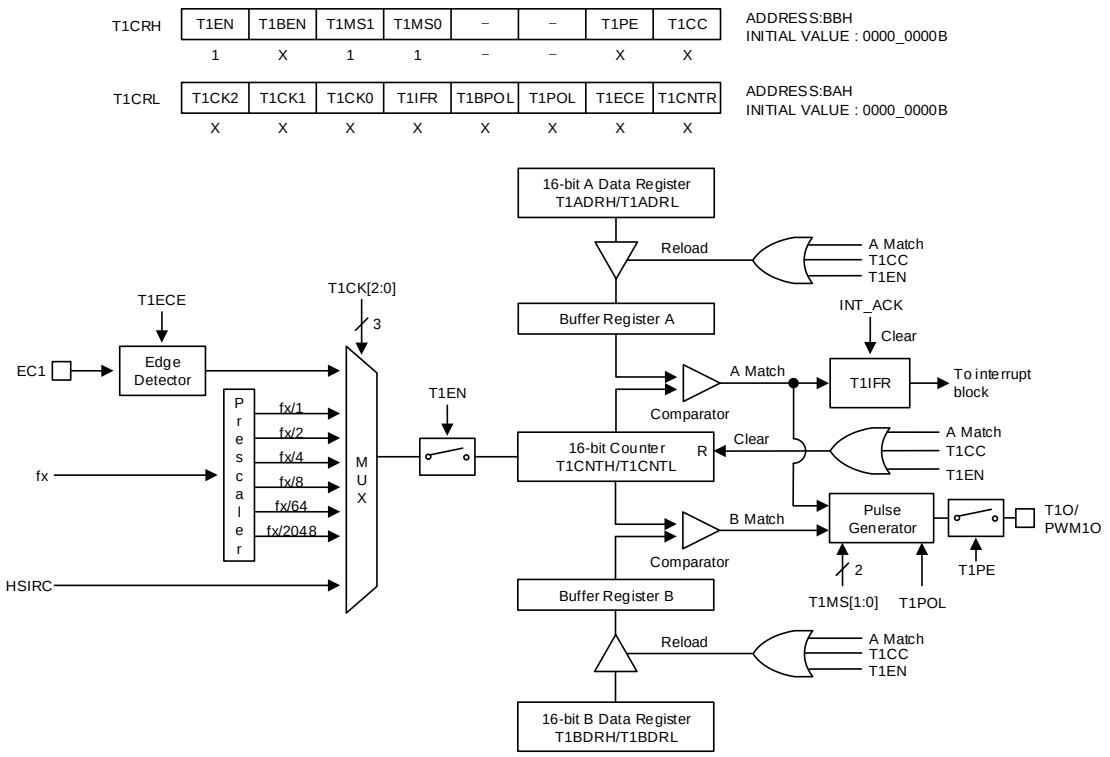


Figure 43. Express Timer Overflow in Capture Mode

11.2.3 16-bit PPG mode

TIMER 1 has a PPG (Programmable Pulse Generation) function. In PPG mode, T1O/PWM1O pin outputs up to 16-bit resolution PWM output. For this function, T1O/PWM1O pin must be configured as a PWM output by setting PxFSR. Period of the PWM output is determined by T1ADRH/T1ADRL, and duty of the PWM output is determined by T1BDRH/T1BDRL.



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

Figure 44. 16-bit PPG Mode of Timer 1

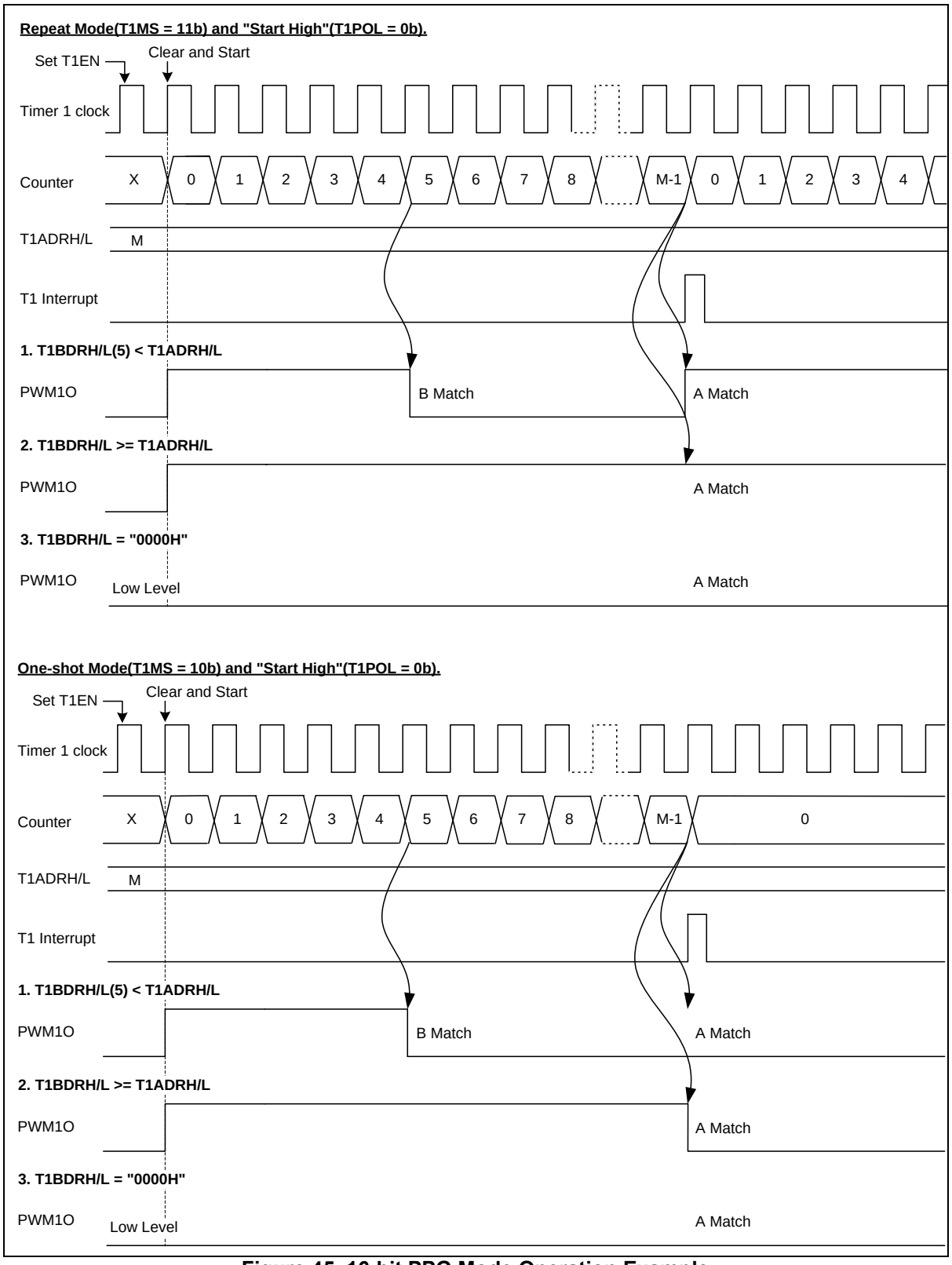
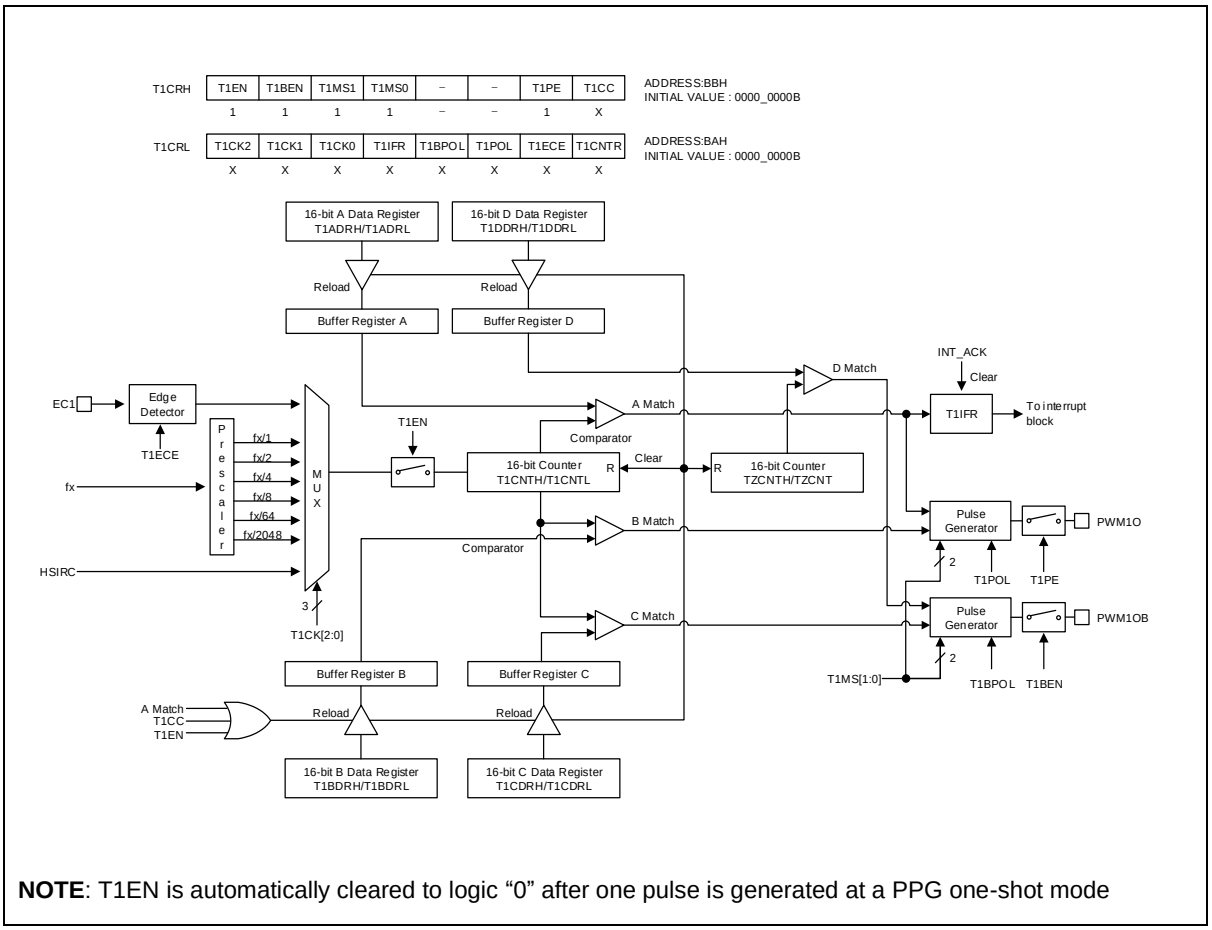


Figure 45. 16-bit PPG Mode Operation Example

11.2.4 16-bit Complementary PWM mode (Dead Time)

The timer 1 has a Complementary PWM function. The complementary PWM output function operates when T1BEN is set. In PPG mode, PWM1O/PWM1OB pin outputs up to 16-bit resolution complementary PWM output. This pin should be configured as a PWM output by setting PxFSRL.

The period of the PWM output is determined by the T1ADRH/T1ADRL. And the duty of the PWM output is determined by the T1BDRH/T1BDRL. The delay (dead time) of the complementary PWM output is determined by T1CDRH / T1CDRL. And the duty of the complementary PWM output is determined by T1DDRH / T1DDRL.



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

Figure 46. 16-bit Complementary PWM Mode for Timer1

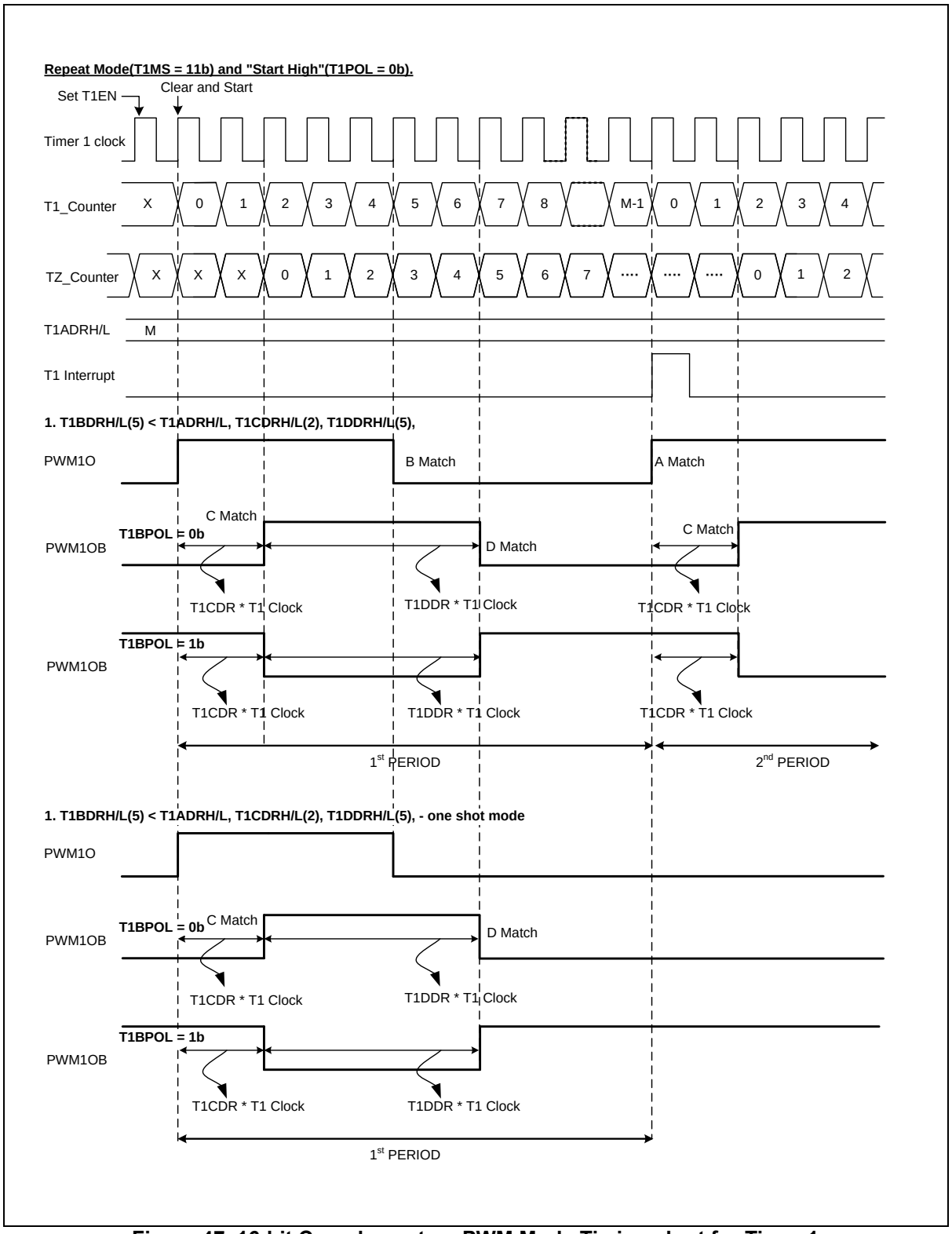


Figure 47. 16-bit Complementary PWM Mode Timing chart for Timer 1

11.2.5 16-bit timer 1 block diagram

In this section, a 16-bit timer 1 is described in a block diagram.

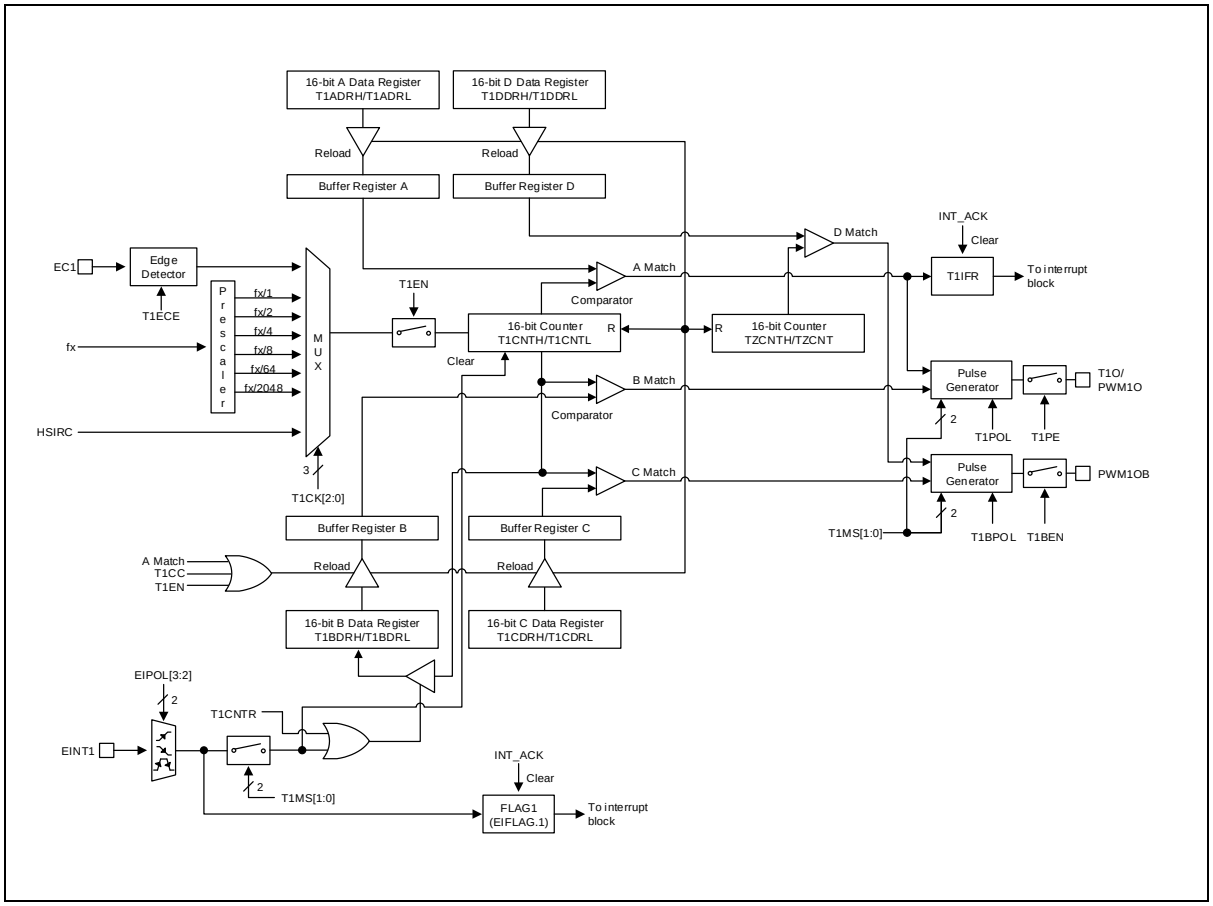


Figure 48. 16-bit Timer 1 Block Diagram

11.2.6 Register map

Table 17. TIMER 1 Register Map

Name	Address	Direction	Default	Description
T1ADRH	BDH	R/W	FFH	Timer 1 A Data High Register
T1ADRL	BCH	R/W	FFH	Timer 1 A Data Low Register
T1BDRH	BFH	R/W	FFH	Timer 1 B Data High Register
T1BDRL	BEH	R/W	FFH	Timer 1 B Data Low Register
T1CDRL	D9H	R/W	FFH	Timer 1 C Data Low Register
T1CDRH	DAH	R/W	FFH	Timer 1 C Data High Register
T1DDRL	DBH	R/W	FFH	Timer 1 D Data Low Register
T1DDRH	DCH	R/W	FFH	Timer 1 D Data High Register
T1CRL	BAH	R/W	00H	Timer 1 Control Low Register
T1CRH	BBH	R/W	00H	Timer 1 Control High Register

11.2.7 Register description

T1ADRH (Timer 1 A data High Register): BDH

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

Initial value: FFH

T1ADRH[7:0] T1 A Data High Byte

T1ADRL (Timer 1 A Data Low Register): BCH

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

Initial value: FFH

T1ADRL[7:0] T1 A Data Low Byte

NOTE: Do not write "0000H" in the T1ADRH/T1ADRL register when PPG mode

T1BDRH (Timer 1 B Data High Register): BFH

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

Initial value: FFH

T1BDRH[7:0] T1 B Data High Byte

T1BDRL (Timer 1 B Data Low Register): BEH

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

Initial value: FFH

T1BDRL[7:0] T1 B Data Low Byte

T1CDRH (Timer 1 C Data High Register): DAH

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

Initial value: FFH

T1CDRH[7:0] T1 C Data High Byte

T1CDRL (Timer 1 C Data Low Register): D9H

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

Initial value: FFH

T1CDRL[7:0] T1 C Data Low Byte

T1DDRH (Timer 1 D Data High Register): DCH

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

Initial value: FFH

T1DDRH[7:0]	T1 D Data High Byte

T1DDRL (Timer 1 D Data Low Register): DBH

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

Initial value: FFH

T1DDRL[7:0]	T1 D Data Low Byte

T1CRH (Timer 1ControlHigh Register): BBH

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

Initial value: 00H

T1EN Control Timer 1

0 Timer 1 disable

1 Timer 1 enable (Counter clear and start)

T1EN Control Complementary PWM

0 Complementary PWM disable

1 Complementary PWM enable

T1MS[1:0] Control Timer 1 Operation Mode

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

T1PE Control Timer 1 port output

0 Timer 1 output disable

1 Timer 1 output enable

T1CC Clear Timer 1 Counter

0 No effect

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

T1CRL (Timer 1ControlLow Register): BAH

7	6	5	4	3	2	1	0
T1CK2	T1CK1	T1CK0	T1IFR	T1BPOL	T1POL	T1ECE	T1CNTR
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

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

T1CK2	T1CK1	T1CK0	Description
0	0	0	fx/2048
0	0	1	fx/64
0	1	0	fx/8
0	1	1	fx/4
1	0	0	fx/2
1	0	1	fx/1
1	1	0	HSIRC Direct (32MHz)
1	1	1	External clock (EC1)

T1IFR When T1 Interrupt occurs, this bit becomes '1'. For clearing bit, write '0' to this bit or auto clear by INT_ACK signal. Writing "1" has no effect.

0 T1 Interrupt no generation

1 T1 Interrupt generation

T1BPOL PWM1OB Polarity Selection

0 Start High (PWM1OB is low level at disable)

1 Start Low (PWM1OB is high level at disable)

T1POL T1O/PWM1O Polarity Selection

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

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

T1ECE Timer 1 External Clock Edge Selection

0 External clock falling edge

1 External clock rising edge

T1CNTR Timer 1 Counter Read Control

0 No effect

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

11.3 Timer 2

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

Timer 2 operates in one of the following modes:

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

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

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

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

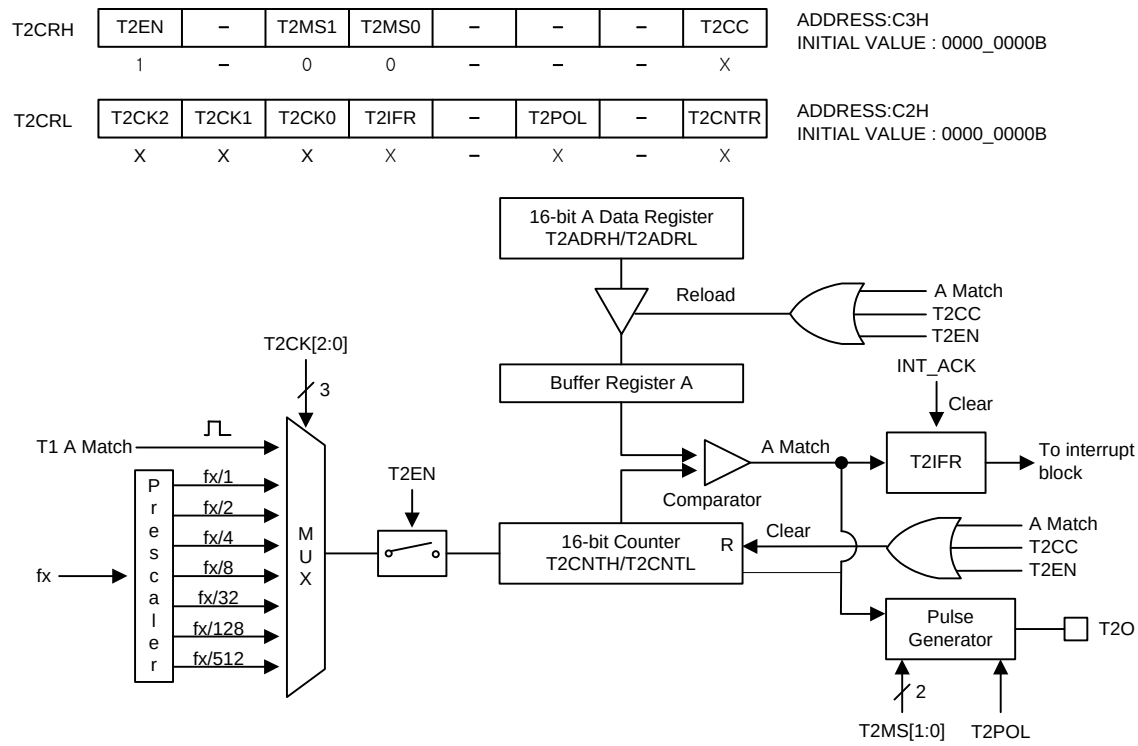
Table 18. TIMER 2 Operating Modes

T2EN	P1FSRL[3:2]	T2MS[1:0]	T2CK[2:0]	Timer 2
1	11	00	XXX	16 Bit Timer/Counter Mode
1	00	01	XXX	16 Bit Capture Mode
1	11	10	XXX	16 Bit PPG Mode(one-shot mode)
1	11	11	XXX	16 Bit PPG Mode(repeat mode)

11.3.1 16-bit timer/counter mode

16-bit timer/counter mode is selected by control registers, and the 16-bit timer/counter has counter registers and data registers as shown in Figure 49. The counter register is increased by internal or timer 1 A match clock input.

Timer 2 can use the input clock with one of 1, 2, 4, 8, 32, 128, 512 and T1 A Match prescaler division rates (T2CK[2:0]). When the values of T2CNTH/T2CNTL and T2ADRH/T2ADRL are identical to each other in timer 2, a match signal is generated and the interrupt of Timer 2 occurs. The T2CNTH/T2CNTL values are automatically cleared by the match signal. It can be cleared by software (T2CC) too.



NOTE: T1 A Match is a pulse for the timer 2 clock source if it is selected.

Figure 49. 16-bit Timer/Counter Mode of Timer 2

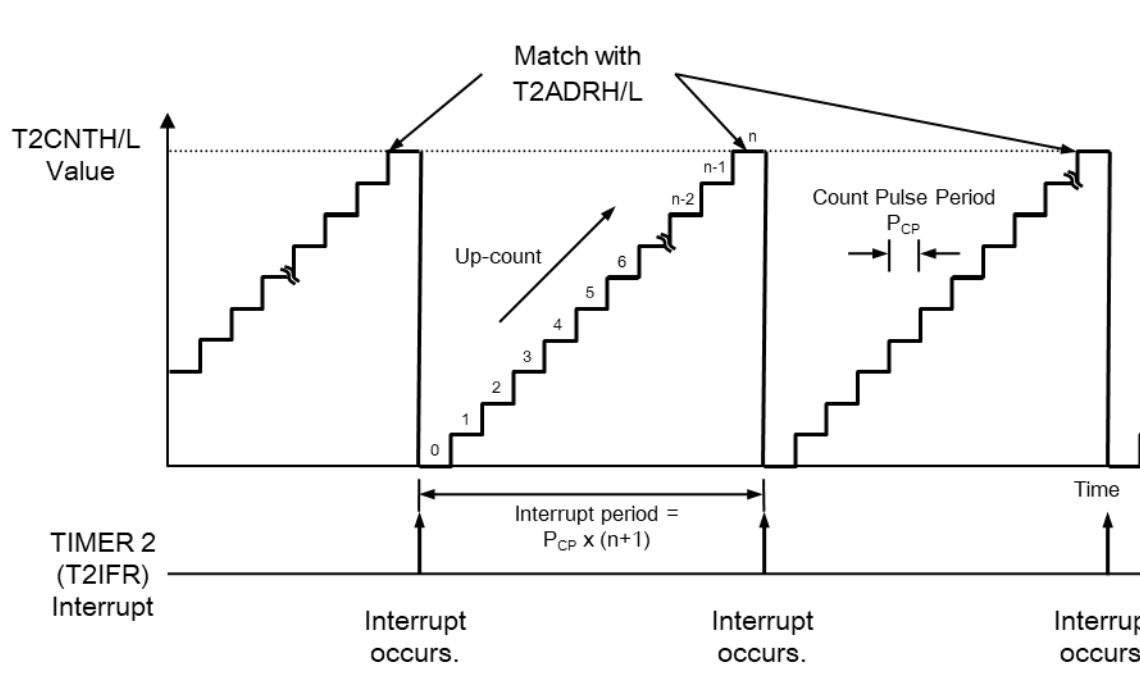


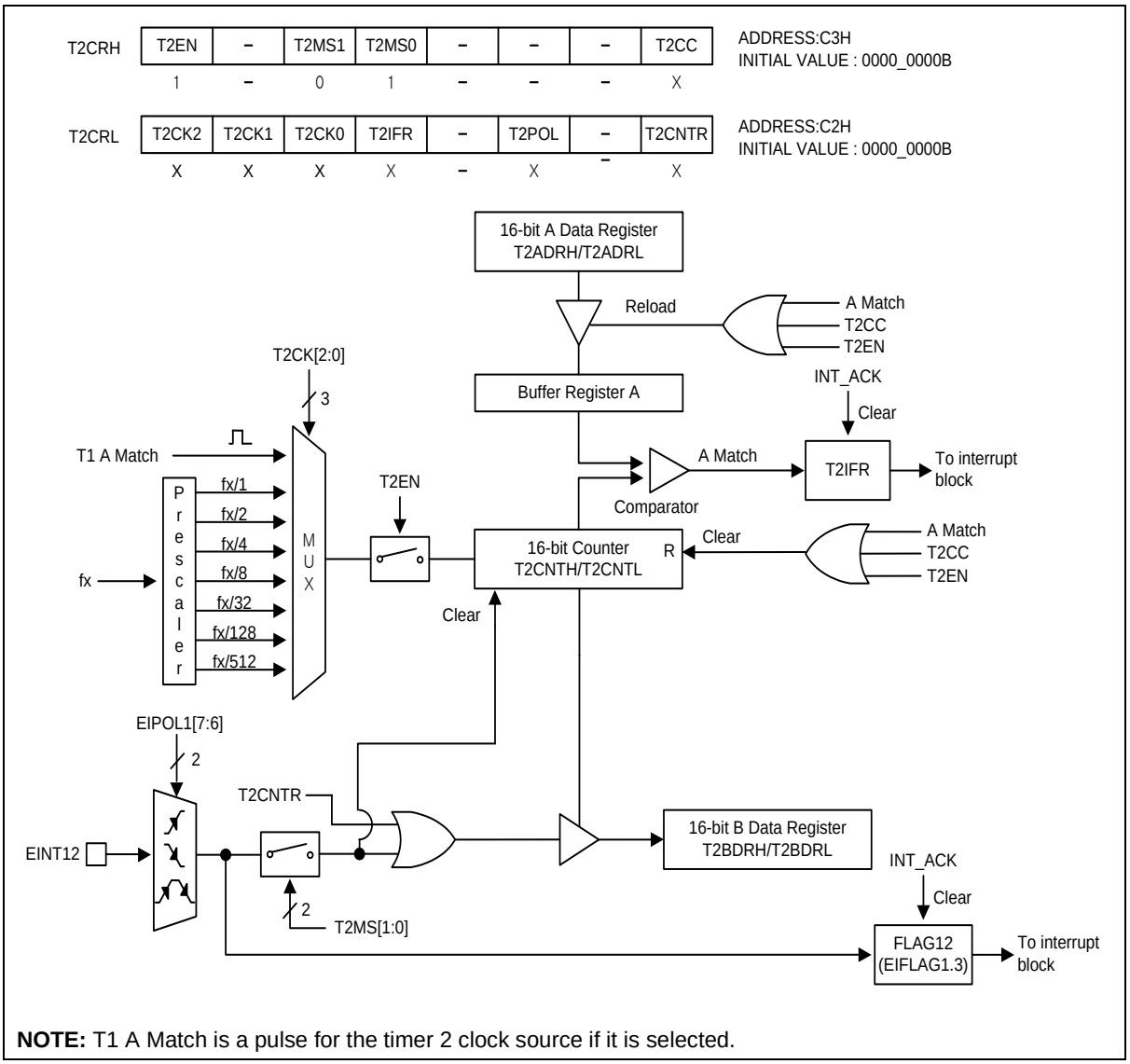
Figure 50. 16-bit Timer/Counter Mode Operation Example

11.3.2 16-bit capture mode

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

A timer interrupt in capture mode is very useful when the pulse width of captured signal is wider than the maximum period of timer. Capture result is loaded into T2BDRH/T2BDRL. In timer 2 capture mode, timer 2 output (T2O) waveform is not available.

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



NOTE: T1 A Match is a pulse for the timer 2 clock source if it is selected.

Figure 51. 16-bit Capture Mode of Timer 2

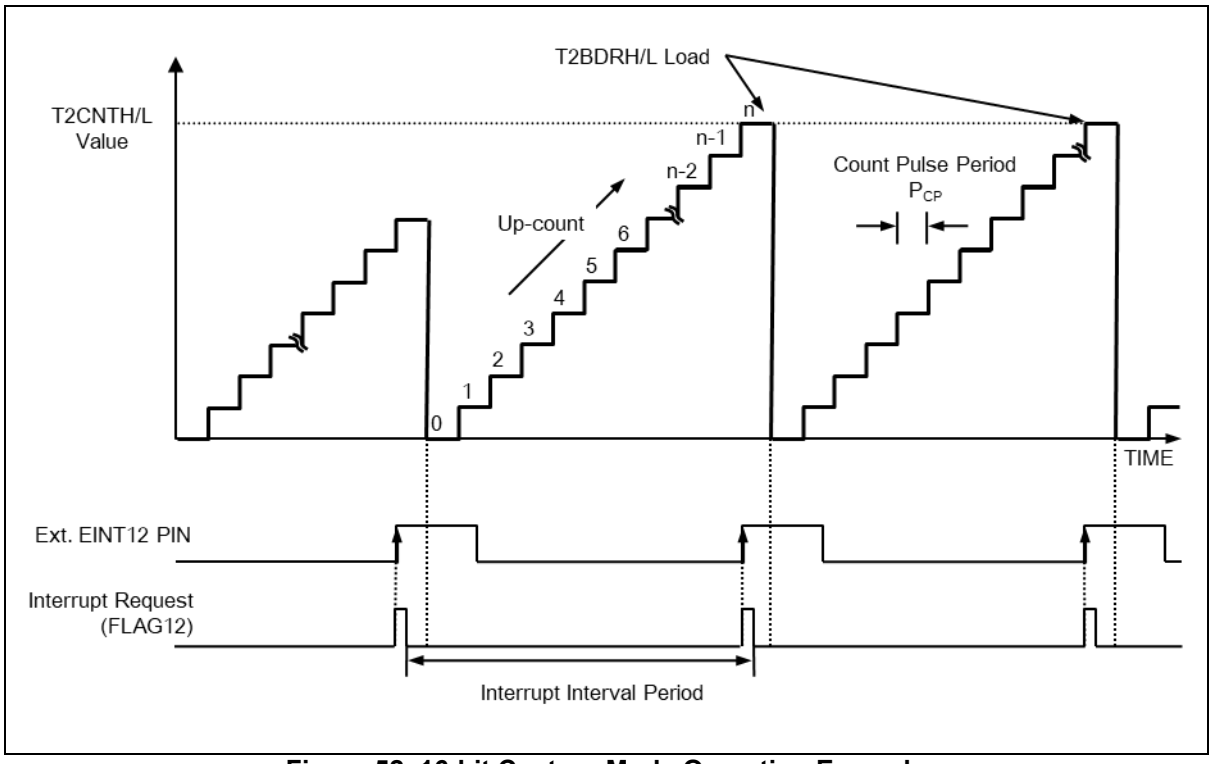


Figure 52. 16-bit Capture Mode Operation Example

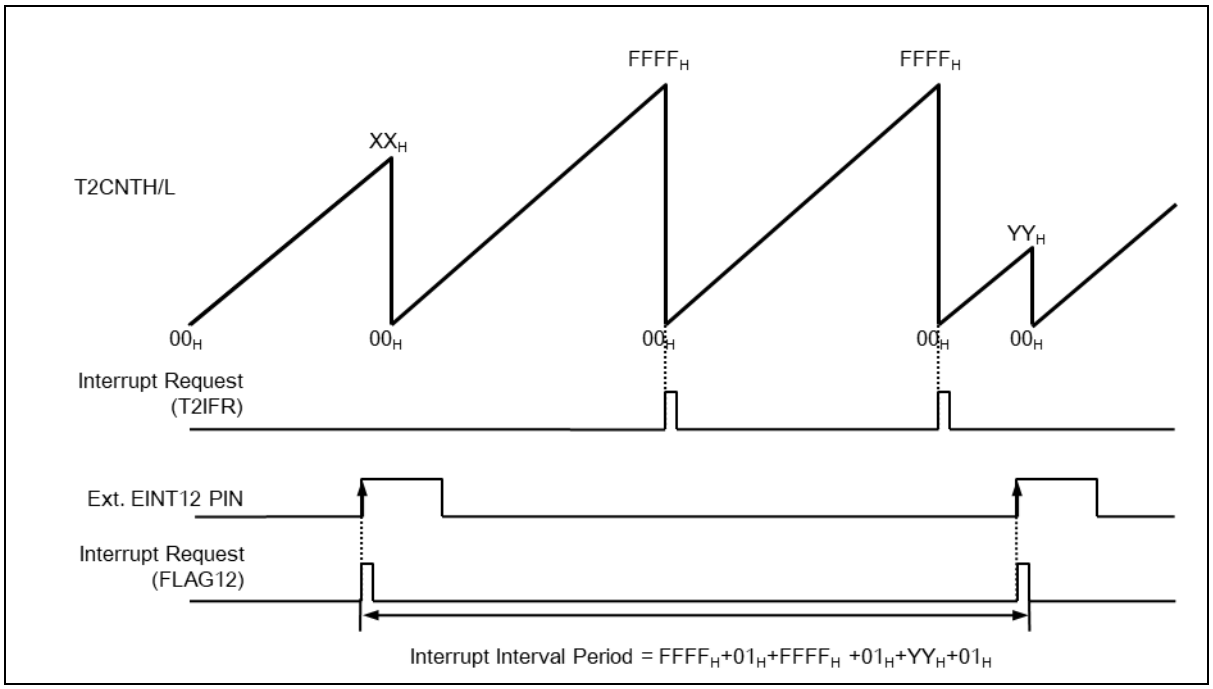
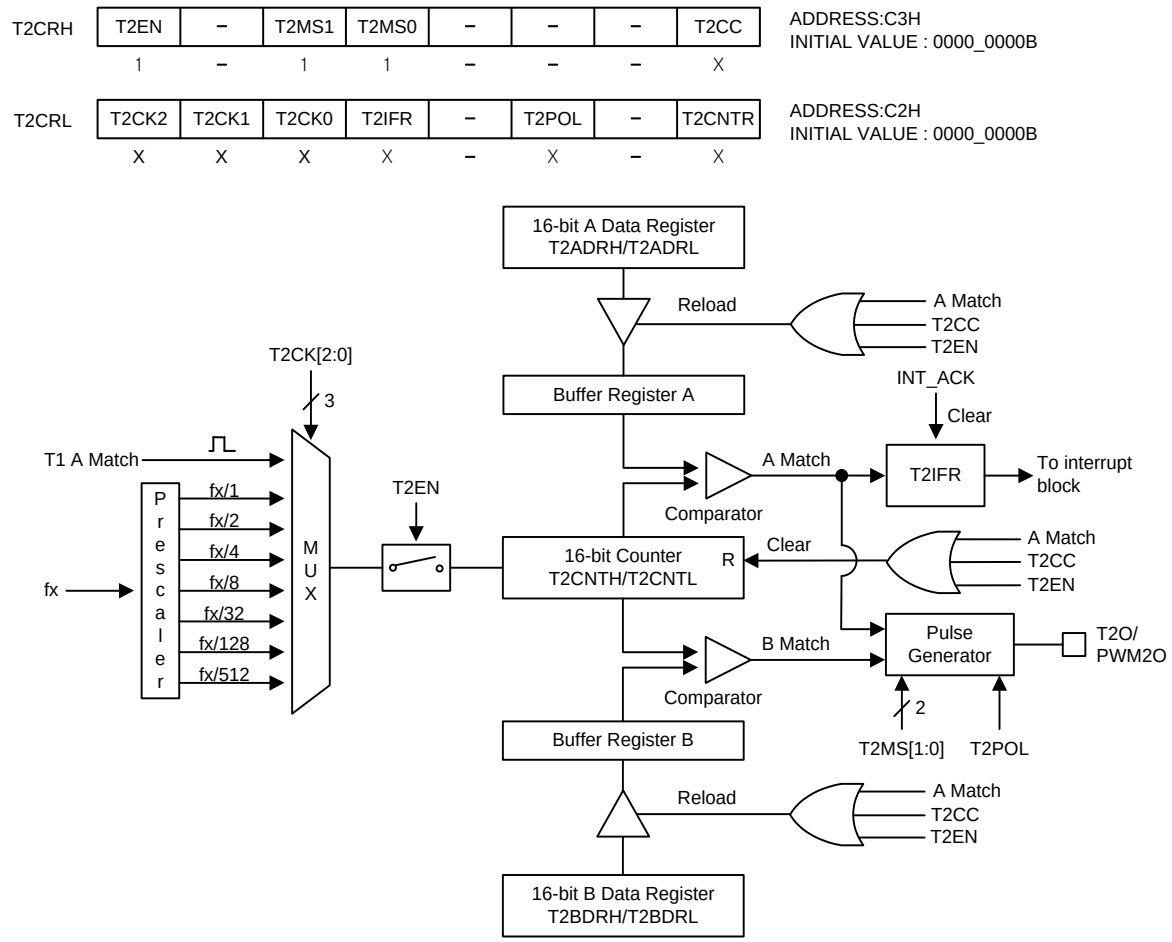


Figure 53. Express Timer Overflow in Capture Mode

11.3.3 16-bit PPG mode

TIMER 2 has a PPG (Programmable Pulse Generation) function. In PPG mode, T2O/PWM2O pin outputs up to 16-bit resolution PWM output. For this function, T2O/PWM2O pin must be configured as a PWM output by setting PxFSR. Period of the PWM output is determined by T2ADRH/T2ADRL, and duty of the PWM output is determined by T2BDRH/T2BDRL.



NOTES:

1. T2EN is automatically cleared to logic "0" after one pulse is generated at a PPG one-shot mode.
2. T1 A Match is a pulse for the timer 2 clock source if it is selected.

Figure 54. 16-bit PPG Mode of Timer 2

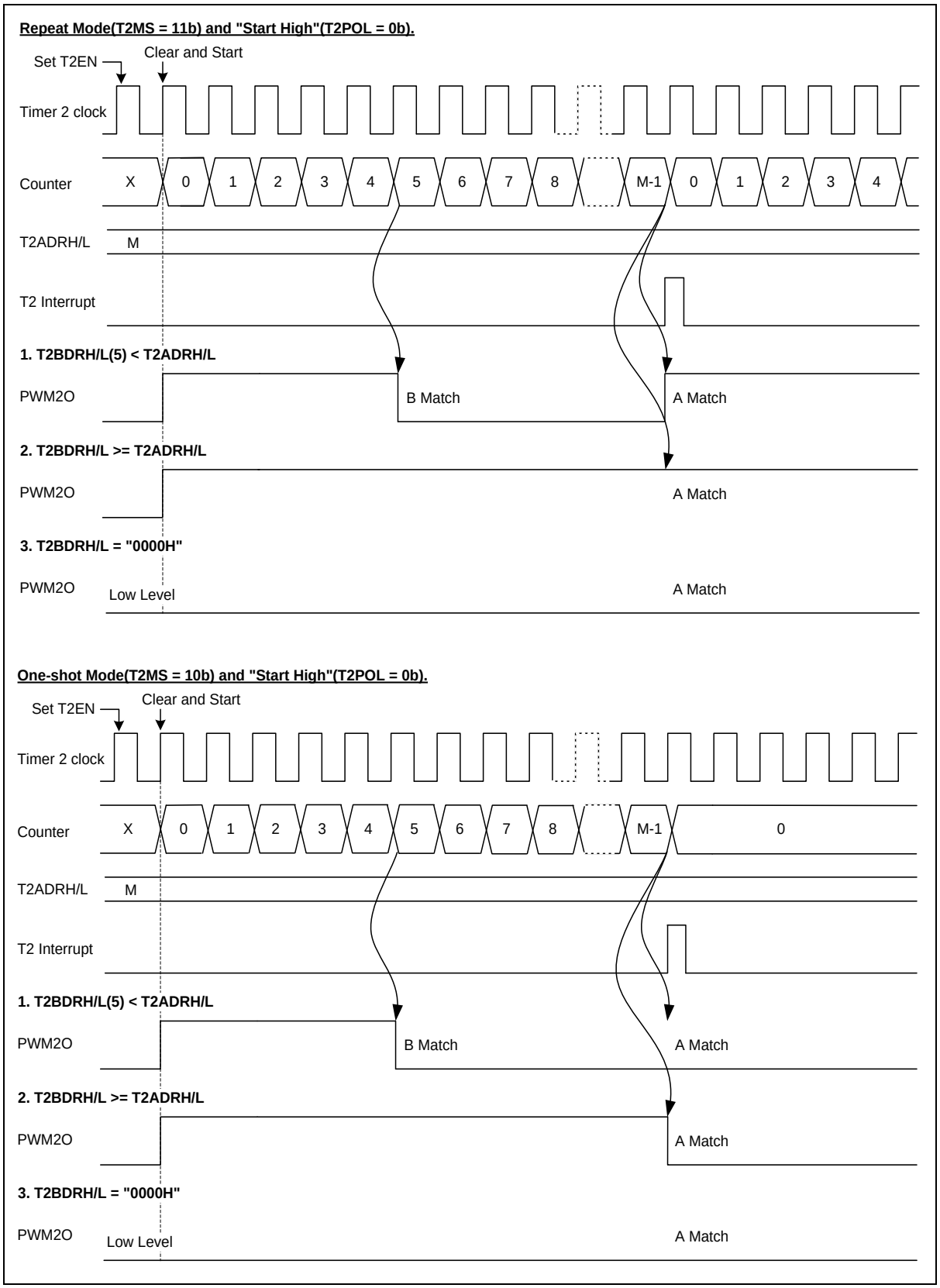


Figure 55. 16-bit PPG Mode Operation Example

11.3.4 16-bit timer 2 block diagram

In this section, a 16-bit timer 2 is described in a block diagram.

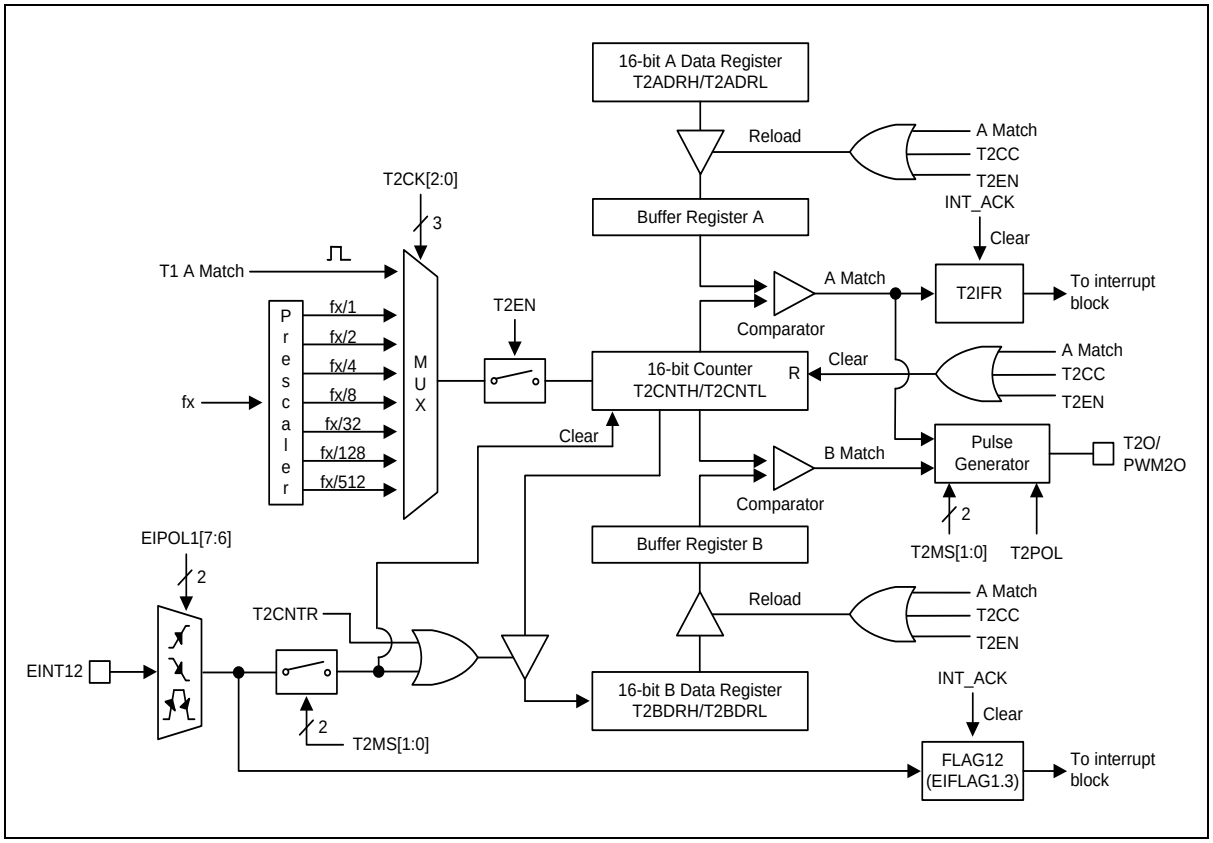


Figure 56. 16-bit Timer 2 Block Diagram

11.3.5 Register map

Table 19. TIMER 2 Register Map

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

11.3.6 Register description

T2ADRH (Timer 2 A data High Register): C5H

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

Initial value: FFH

T2ADRH[7:0] T2 A Data High Byte

T2ADRL (Timer 2 A Data Low Register): C4H

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

Initial value: FFH

T2ADRL[7:0] T2 A Data Low Byte

NOTE: Do not write "0000H" in the T2ADRH/T2ADRL register when PPG mode.

T2BDRH (Timer 2 B Data High Register): C7H

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

Initial value: FFH

T2BDRH[7:0] T2 B Data High Byte

T2BDRL (Timer 2 B Data Low Register): C6H

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

Initial value: FFH

T2BDRL[7:0] T2 B Data Low

T2CRH (Timer 2ControlHigh Register): C3H

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

Initial value: 00H

T2EN Control Timer 2

0 Timer 2 disable

1 Timer 2 enable (Counter clear and start)

T2MS[1:0] Control Timer 2Operation Mode

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

T2CC Clear Timer 2 Counter

0 No effect

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

T2CRL (Timer 2ControlLow Register): C2H

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

Initial value: 00H

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

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

T2IFR When T2 Match Interrupt occurs, this bit becomes '1'. For clearing bit, write '0' to this bit or auto clear by INT_ACK signal. Writing "1" has no effect.

0 T2interrupt no generation

1 T2 interrupt generation

T2POL T2O/PWM2O Polarity Selection

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

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

T2CNTR Timer 2 Counter Read Control

0 No effect

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

12 12-bit ADC

Analog-to-digital converter (ADC) of A96G174/A96S174 allows conversion of an analog input signal to corresponding 12-bit digital value. This A/D module has eight analog inputs. Output of the multiplexer becomes input into the converter which generates the result through successive approximation.

The A/D module has four registers which are the A/D converter control high register (ADCCRH), A/D converter control low register (ADCCRL), A/D converter data high register (ADCDRH), and A/D converter data low register (ADCDRL).

ADSEL[3:0] bits are used to select channels to be converted. To execute A/D conversion, TRIG[2:0] bits should be set to 'xxx'. Registers ADCDRH and ADCDRL contain the result of A/D conversion. When the conversion is completed, the result is loaded into ADCDRH and ADCDRL, A/D conversion status bit AFLAG is set to '1', and A/D interrupt is set. During the A/D conversion, AFLAG bit is read as '0'.

12.1 Conversion timing

A/D conversion process requires 4 steps (4 clock edges) to convert each bit and 12 clocks to set up A/D conversion. Therefore, total of 58 clocks are required to complete a 12-bit conversion: For example, when fxx/8 is selected for conversion clock with a 12MHz fxx clock frequency, one clock cycle is 0.66μs, and each bit conversion requires 4 clocks. The conversion rate is calculated as follows:

4 clocks/bit × 12 bits + set-up time = 60 clocks
ADC Conversion Time = ADCLK * 60 cycles

Please remember that the A/D converter requires at least 7.5us for conversion time, so the conversion time must be set bigger than 7.5us.

12.2 Block diagram

In this section, the 12-bit ADC is described in a block diagram, and an analog input pin and a power pin with capacitors respectively are introduced.

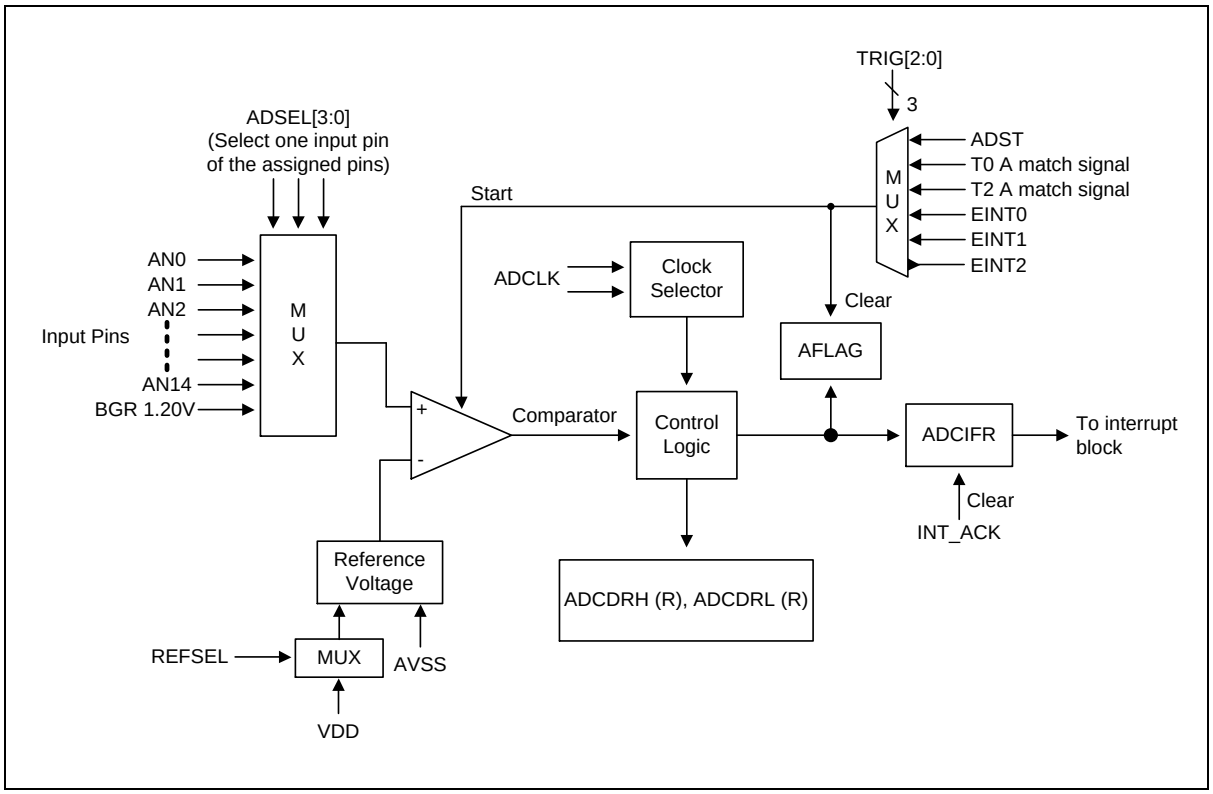


Figure 57. 12-bit ADC Block Diagram

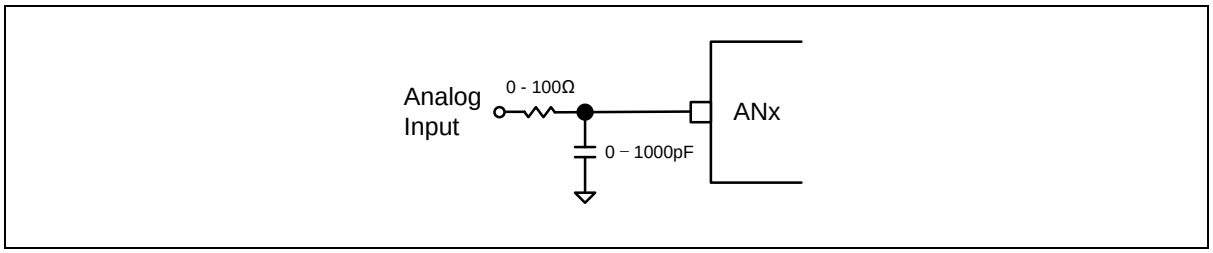


Figure 58. A/D Analog Input Pin with a Capacitor

12.3 ADC operation

In this section, control registers and align bits are introduced in Figure 59, and ADC operation flow sequence is introduced in Figure 60.

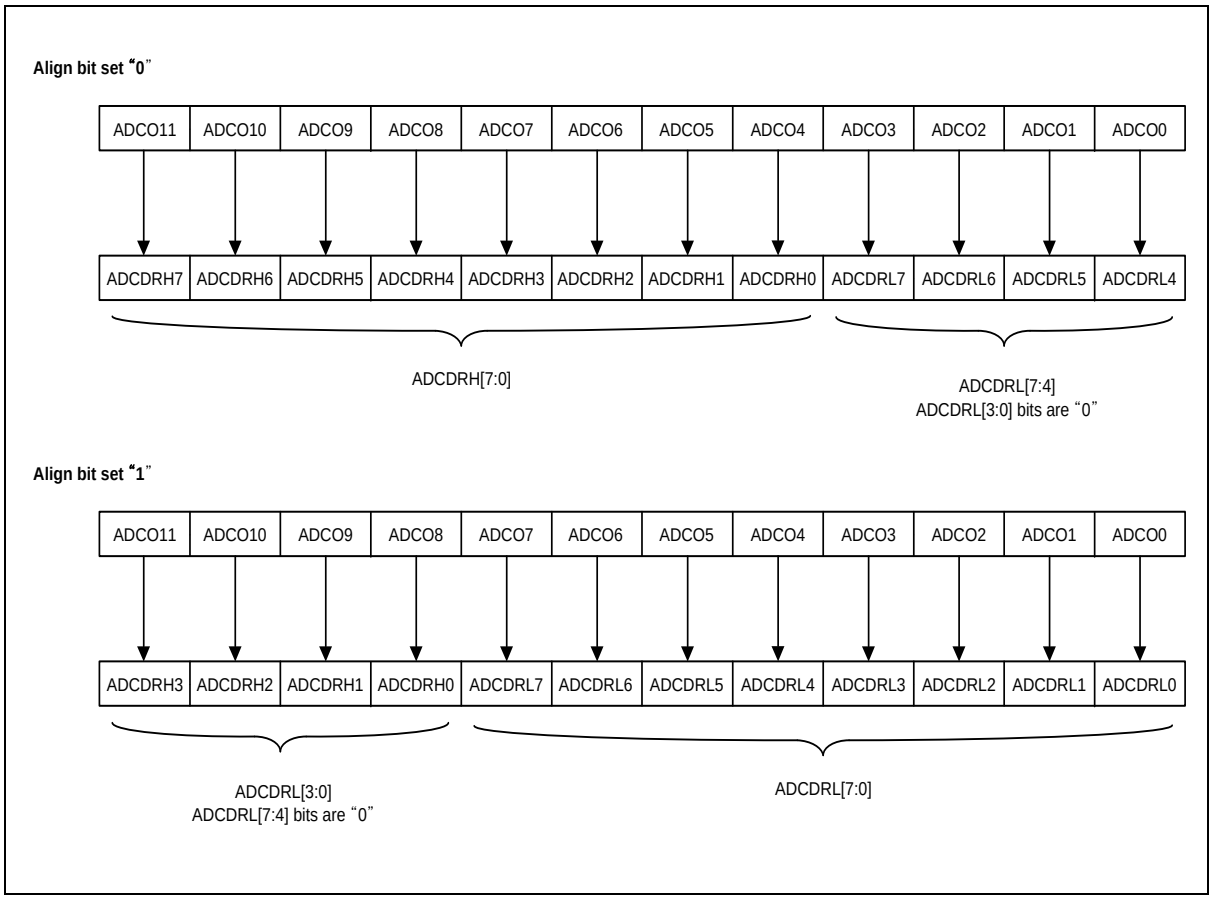


Figure 59. Control Registers and Align Bits

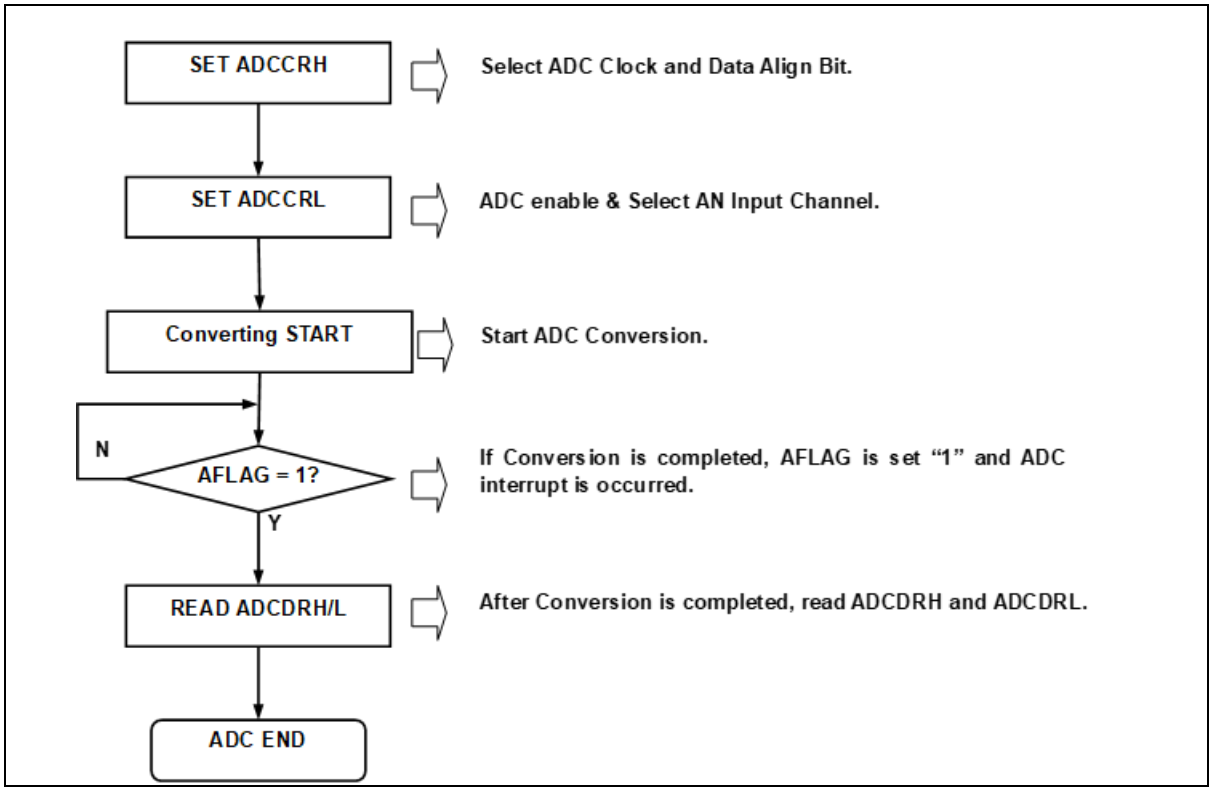


Figure 60. ADC Operation Flow Sequence

12.4 Register map

Table 20. ADC Register Map

Name	Address	Direction	Default	Description
ADCDRH	9FH	R	xxH	A/D Converter Data High Register
ADCDRL	9EH	R	xxH	A/D Converter Data Low Register
ADCCRH	9DH	R/W	01H	A/D Converter Control High Register
ADCCRL	9CH	R/W	00H	A/D Converter Control Low Register

12.5 Register description

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

7	6	5	4	3	2	1	0
ADDM11	ADDM10	ADDM9	ADDM8	ADDM7 ADDL11	ADDM6 ADDL10	ADDM5 ADDL9	ADDM4 ADDL8
R	R	R	R	R	R	R	R

Initial value: xxH

ADDM[11:4] MSB align, A/D Converter High Data (8-bit)

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

ADCDRL (A/D Converter Data Low Register): 9EH

7	6	5	4	3	2	1	0
ADDM3 ADDL7	ADDM2 ADDL6	ADDM1 ADDL5	ADDM0 ADDL4	ADDL3	ADDL2	ADDL1	ADDL0
R	R	R	R	R-	R	R	R

Initial value: xxH

ADDM[3:0]	MSB align, A/D Converter Low Data (4-bit)
ADDL[7:0]	LSB align, A/D Converter Low Data (8-bit)

ADCCRH (A/D Converter High Register): 9DH

7	6	5	4	3	2	1	0
ADCIFR	IREF	TRIG2	TRIG1	TRIG0	ALIGN	CKSEL1	CKSEL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 01H

ADCIFR — When ADC interrupt occurs, this bit becomes '1'. For clearing bit, write '0' to this bit or auto clear by INT_ACK signal. Writing "1" has no effect.

0	ADC Interrupt no generation
1	ADC Interrupt generation

IREF — Select internal voltage reference 1.55V input.

0	External input signal source select
1	Test only

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

TRIG2	TRIG1	TRIG0	Description
0	0	0	ADST
0	0	1	Timer 0 A match signal
0	1	0	Timer 2 A match signal
0	1	1	EXTINT0
1	0	0	EXTINT1
1	0	1	EXTINT2
Other Values			Not used

ALIGN — A/D Converter data align selection.

0	MSB align (ADCDRH[7:0], ADCDRL[7:4])
1	LSB align (ADCRDH[3:0], ADCDRL[7:0])

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

CKSEL1	CKSEL0	Description
0	0	fx/1
0	1	fx/2
1	0	fx/4
1	1	fx/8

NOTES:

1. fx : system clock
2. ADC clock should use below 8MHz

ADCCRL (A/D Converter Counter Low Register): 9CH

7	6	5	4	3	2	1	0
STBY	ADST	REFSEL	AFLAG	ADSEL3	ADSEL2	ADSEL1	ADSEL0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W

Initial value: 00H

STBY Control Operation of A/D
(The ADC module is automatically disabled at stop mode)
0 ADC module disable
1 ADC module enable

ADST Control A/D Conversion stop/start.
0 No effect
1 ADC Conversion Start and auto clear

REFSEL A/D Converter Reference Selection
0 Internal Reference (VDD)
1 Reserved

AFLAG A/D Converter Operation State (This bit is cleared to '0' when the STBY bit is set to '0' or when the CPU is at STOP mode)
0 During A/D Conversion
1 A/D Conversion finished

ADSEL[3:0] A/D Converter input selection

ADSEL3	ADSEL2	ADSEL1	ADSEL0	Description
0	0	0	0	AN0
0	0	0	1	AN1
0	0	1	0	AN2
0	0	1	1	AN3
0	1	0	0	AN4
0	1	0	1	AN5
0	1	1	0	AN6
0	1	1	1	AN7
1	0	0	0	AN8
1	0	0	1	AN9
1	0	1	0	AN10
1	0	1	1	AN11
1	1	0	0	AN12
1	1	0	1	AN13
1	1	1	0	AN14
1	1	1	1	AN15(BGR 1.20V)

13 I2C

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. Asynchronous mode (UART)

- Compatible with I²C bus standard
- Multi-master operation
- Up to 400kHz data transfer speed
- 7 bit address
- Support 2 slave addresses
- Both master and slave operation
- Bus busy detection

13.1 Block diagram

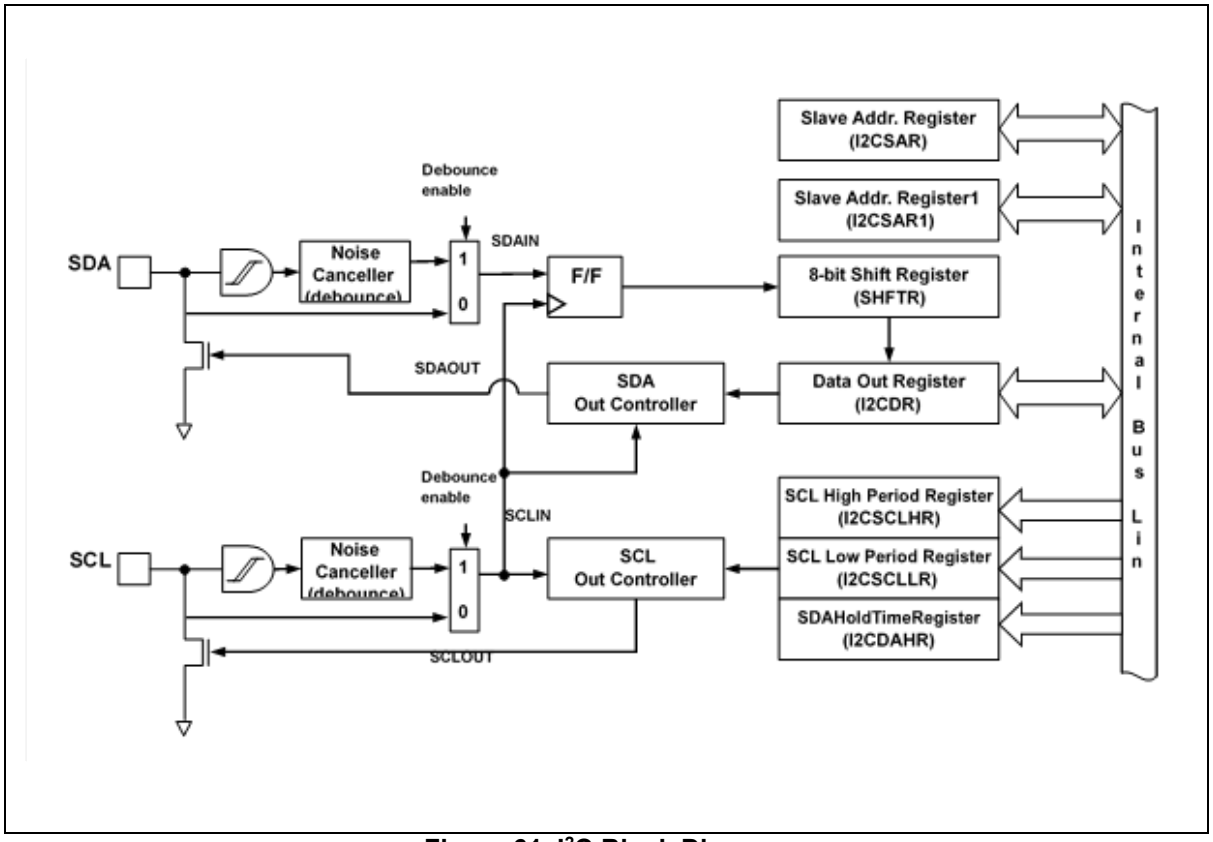


Figure 61. I²C Block Diagram

13.2 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.

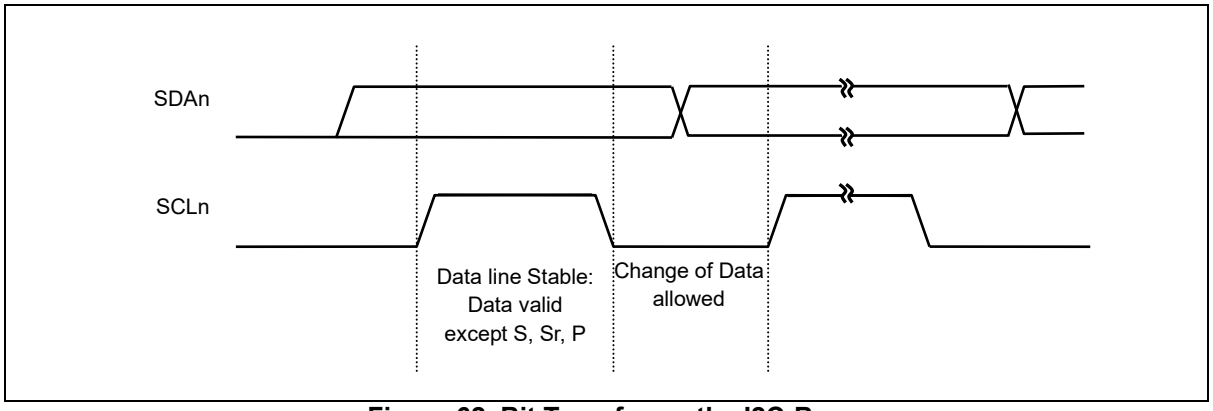


Figure 62. Bit Transfer on the I2C-Bus

13.3 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, i.e., 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.

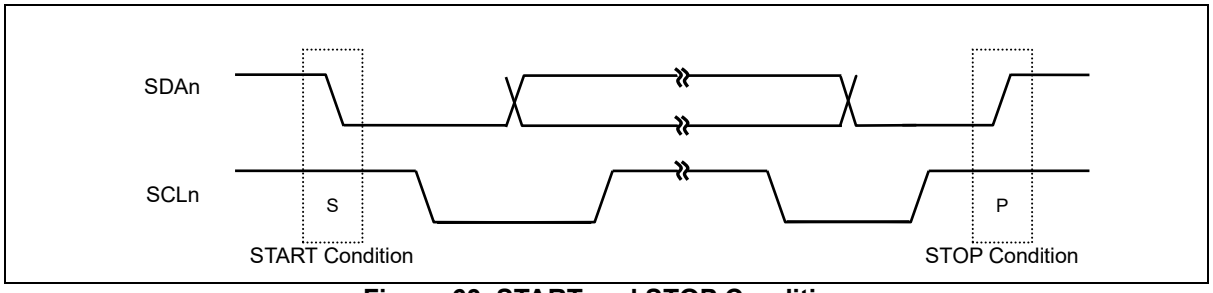


Figure 63. START and STOP Condition

13.4 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.

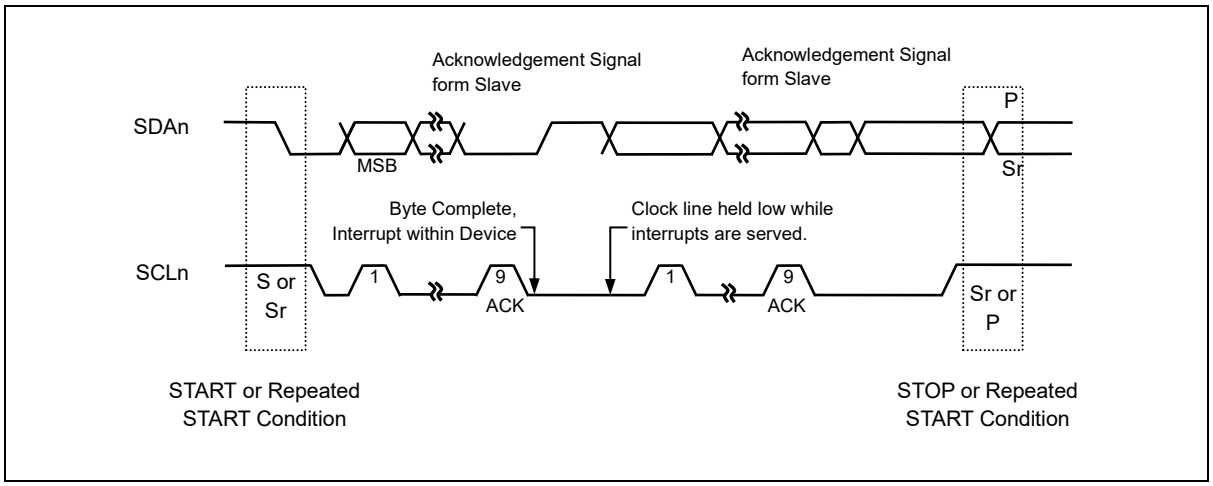


Figure 64. Data Transfer on the I2C-Bus

13.5 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.

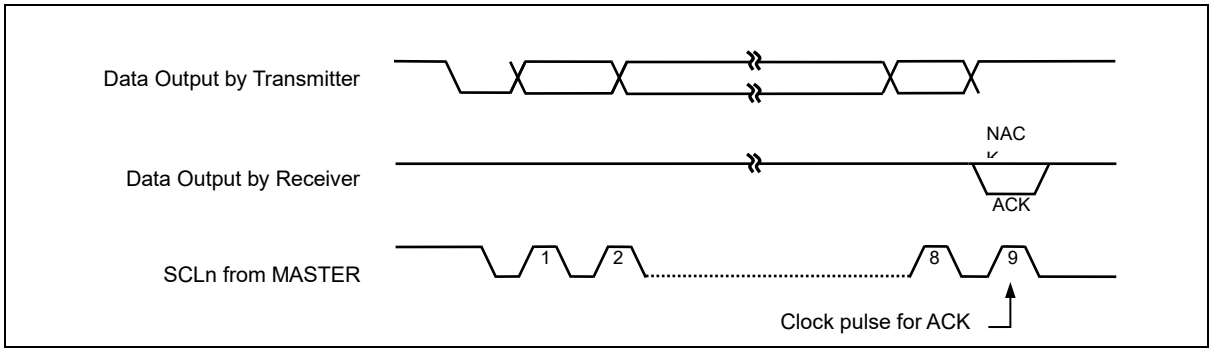


Figure 65. Acknowledge on the I2C-Bus

13.6 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 I2C bus. Its first stage is comparison of the address bits.

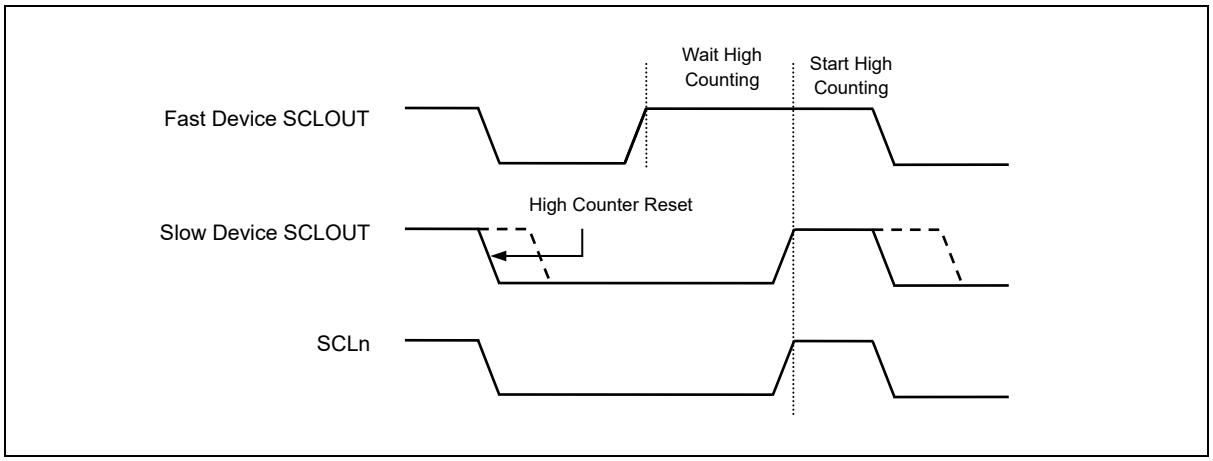


Figure 66. Clock Synchronization during Arbitration Procedure

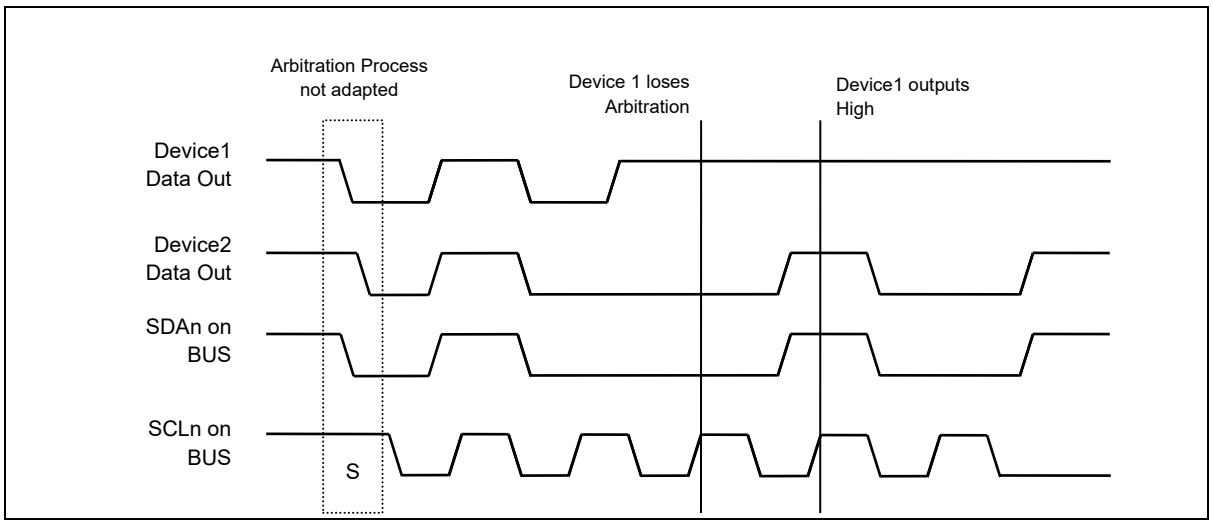


Figure 67. Arbitration Procedure of Two Masters

13.7 Block operation

The I2C block as peripheral design is independently operating with main CPU operation. The operation of I2C block does a byte unit of I2C frame. After finishing a byte operation (transmit/receive data and clock) on I2C bus system, I2C block generate I2C interrupt for next byte operation. The I2C Interrupt service manage I2C block with the SFR registers, data load/read register (I2CDR) from/to I2C bus system, block control register (I2CMR), the state register (I2CSR) contained operation result. An operation unit of I2C H/W block generates/ receive 9 SCL clock that are for 8 bits data and an ACK. I2C block send / receive ACK signal at 9th clock of SCL according to I2C specification.

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

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

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

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

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

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

13.7.1 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

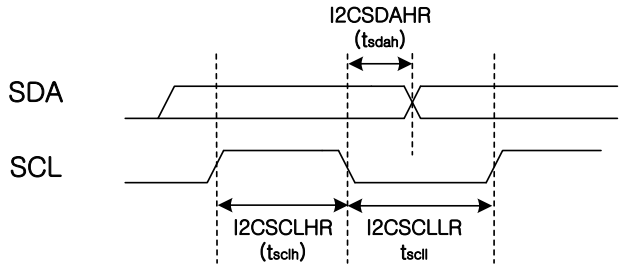
① I2C block will start operation (operation clock active) by setting IICEN bit on I2CMR register.

```
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

13.7.2 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.

13.7.3 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/$\overline{W}$) 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
```

13.7.4 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/$\overline{W}$) 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.

Example code of slave mode is introduced in the followings:

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;
```

13.8 Register map

Table 21. Register Map

Name	Address	Direction	Default	Description
I2CMR	E1H	R/W	00H	I2C Mode Control Register
I2CSR	E2H	R	00H	I2C Status Register
I2CSCLLR	E3H	R/W	3FH	SCL Low Period Register
I2CSCLHR	E4H	R/W	3FH	SCL High Period Register
I2CSDAHR	E5H	R/W	01H	SDA Hold Time Register
I2CDR	E6H	R/W	FFH	I2C Data Register
I2CSAR	E9H	R/W	00H	I2C Slave Address Register
I2CSAR1	EAH	R/W	00H	I2C Slave Address Register 1

13.9 I2C register description

I2CMR (I2C Mode Control Register): E1H

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): E2H

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

NOTES:
1. One of these bits can be a source of an interrupt.
2. When an I2C interrupt occurs except for STOP interrupt, the SCL line is hold LOW. To release SCL, write arbitrary value to I2CSR.
3. When I2CSR is written, the TEND, STOP, SSEL, LOST, RXACK bits are cleared.

I2CSCLLR (SCL Low Period Register): E3H

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 : tSCLK× (4 × SCLL + 1) where tSCLK is the period of SCLK

I2CSCLHR (SCL High Period Register): E4H

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: tSCLK× (4 × SCLH + 3) where tSCLK is the period of SCLK.

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

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

I2CSDAHR (SDA Hold Time Register): E5H

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 tSCLK× 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 tSCLK× (SDAH + 4). So, to insure normal operation in slave mode, the value tSCLK× (SDAH + 4) must be smaller than the period of SCL.

I2CDR (I2C Data Register): E6H

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): E9H

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): EAH

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

14 USART

Universal Synchronous and Asynchronous serial Receiver and Transmitter (USART) is a highly flexible serial communication device. USART of A96G174/A96S174 features the followings:

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

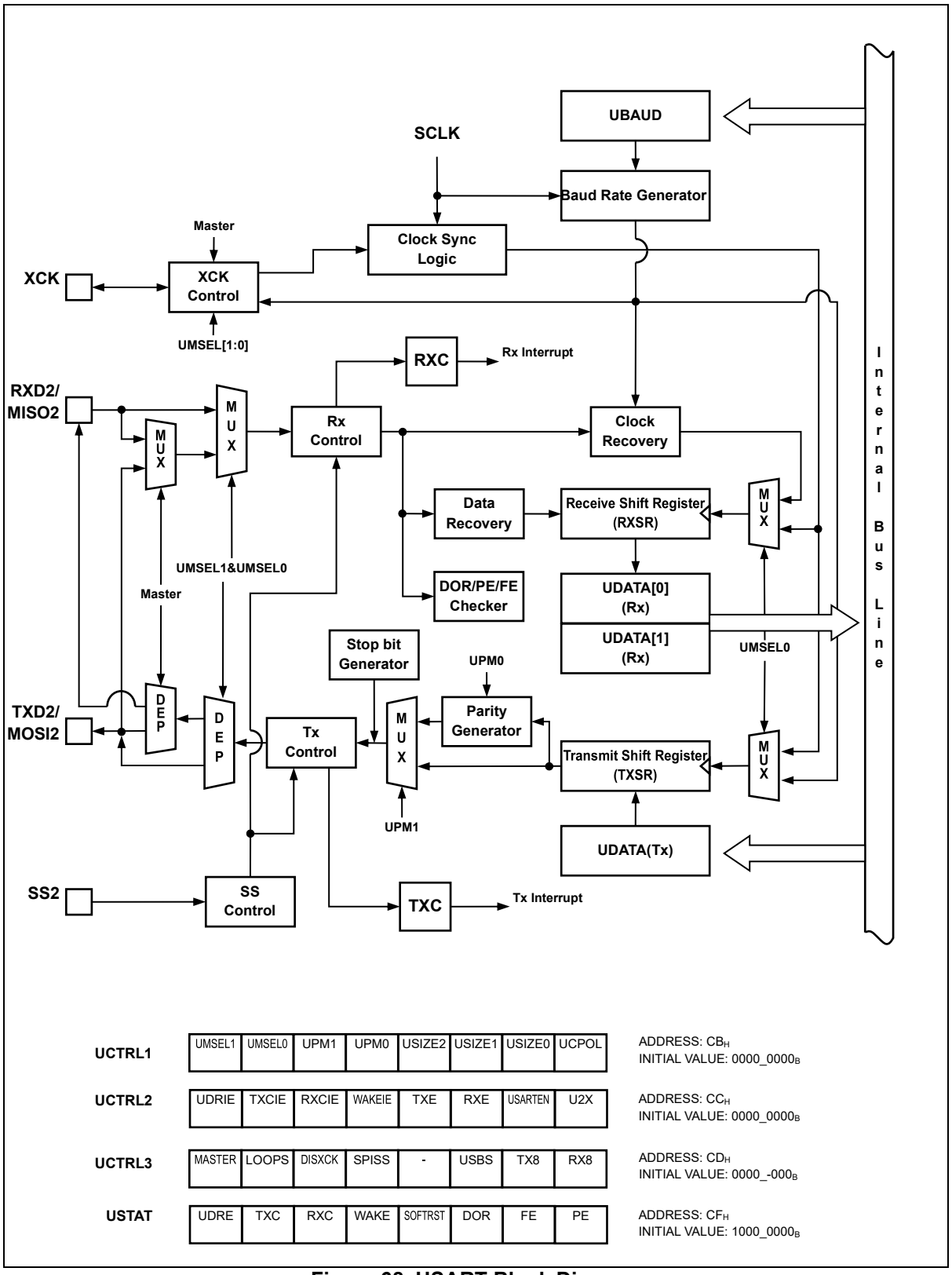
USART has three main parts such as a Clock Generator, Transmitter and Receiver.

Clock Generation logic consists of a synchronization logic for external clock input used by synchronous or SPI slave operation, and a baud rate generator for asynchronous or master (synchronous or SPI) operation.

Transmitter consists of a single write buffer, a serial shift register, parity generator and control logic for handling different serial frame formats. Write buffer allows a continuous transfer of data without any delay between frames.

Receiver is the most complex part of the USART module due to its clock and data recovery units. Recovery unit is used for asynchronous data reception. In addition to the recovery unit, the Receiver includes a parity checker, a shift register, a two level receive FIFO (UDATA) and control logic. The Receiver supports the same frame formats as the Transmitter and can detect Frame Error, Data OverRun and Parity Errors.

14.1 Block diagram



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

Figure 68. USART Block Diagram

14.2 Clock generation

Clock generation logic generates a base clock signal for the Transmitter and the Receiver. USART supports four modes of clock operation such as Normal Asynchronous mode, Double Speed Asynchronous mode, Master Synchronous mode, and Slave Synchronous mode.

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

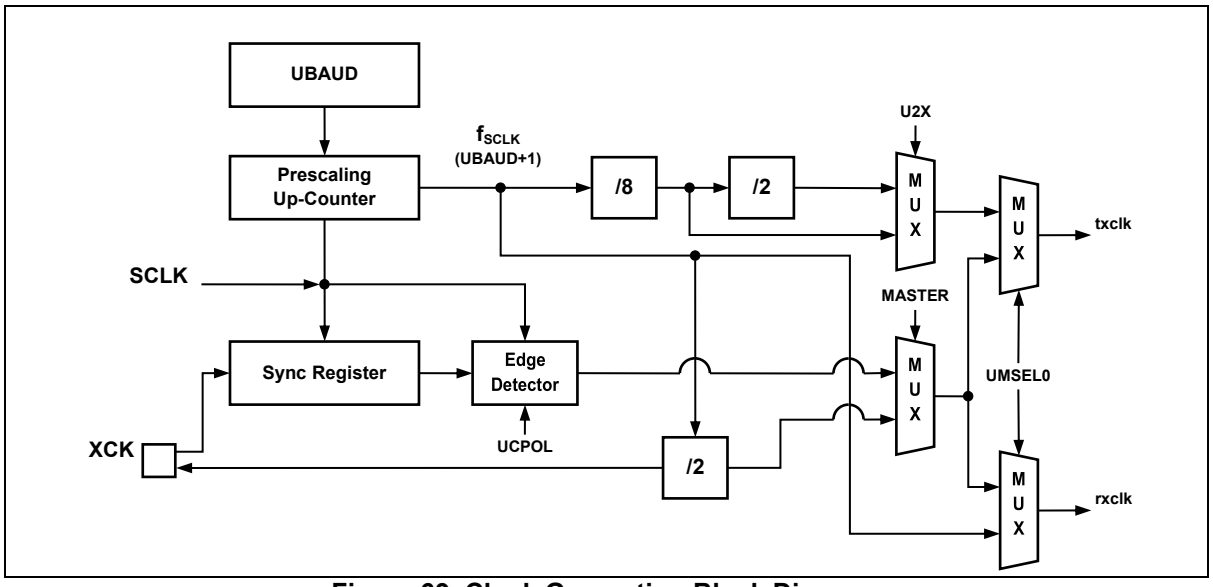


Figure 69. Clock Generation Block Diagram

Table 22 contains equations for calculating the baud rate (in bps).

Table 22. Equations for Calculating Baud Rate Register Setting

Operating mode	Equation for calculating baud rate
Asynchronous normal mode (U2X=0)	$\text{Baud Rate} = \frac{\text{fSCLK}}{16(\text{UBAUDx} + 1)}$
Asynchronous double speed mode (U2X=1)	$\text{Baud Rate} = \frac{\text{fSCLK}}{8(\text{UBAUDx} + 1)}$
Synchronous or SPI master mode	$\text{Baud Rate} = \frac{\text{fSCLK}}{2(\text{UBAUDx} + 1)}$

14.3 External clock (XCK)

External clocking is used by the synchronous or SPI slave modes of operation. External clock input from the XCK pin is sampled by a synchronization logic to remove meta-stability. The output from the synchronization logic must then pass through an edge detector before it can be used by the Transmitter and Receiver.

This process introduces a two CPU clock period delay and the maximum frequency of the external XCK pin is limited by the following equation:

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

NOTES:

1. fXCK is frequency of XCK.
2. fSCLK is frequency of main system clock (SCLK).

14.4 Synchronous mode operation

When synchronous mode or SPI mode is used, the XCK pin will be used as either clock input (slave) or clock output (master). The dependency between a clock edge and data sampling or data change is the same. The basic principle is that data input on RXD2 (MISO2 in SPI mode) pin is sampled at the opposite XCK clock edge of the edge in the data output on TXD2 (MOSI2 in SPI mode) pin is changed.

UCPOL bit in UCTRL1 register selects which XCK clock edge is used for data sampling and which is used for data change. As shown in Figure 70, when UCPOL is zero, data will be changed at XCK rising edge and sampled at XCK falling edge.

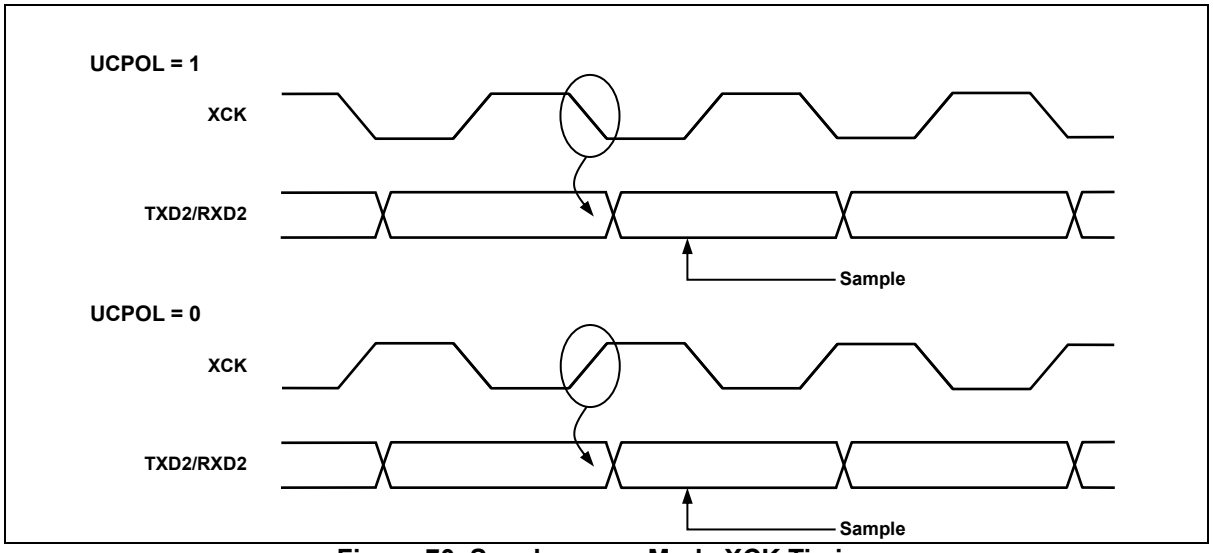


Figure 70. Synchronous Mode XCK Timing

14.5 Data format

A serial frame is defined to consist of one character of data bits with synchronization bits (start and stop bits), and optionally a parity bit for error checking.

USART supports all 30 combinations of the followings as a valid frame format.

- 1 start bit
- 5, 6, 7, 8 or 9 data bits
- no, even or odd parity bit
- 1 or 2 stop bits

A frame starts with the start bit followed by the least significant data bit (LSB). The next data bits, up to a total of nine, are succeeding, ending with the most significant bit (MSB). If enabled, the parity bit is inserted after the data bits, before the stop bits. A high to low transition on data pin is considered as start bit.

When a complete frame is transmitted, it can be directly followed by a new frame, or the communication line can be set to an idle state. The idle means high state of data pin. The next figure shows the possible combinations of the frame formats. Bits inside brackets are optional.

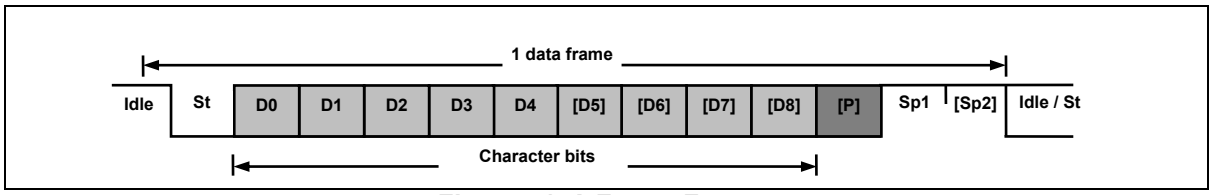


Figure 71. A Frame Format

Single data frame consists of the following bits

- Idle: No communication on communication line (TxD2/RxD2)
- St: Start bit (Low)
- Dn: Data bits (0~8)
- Parity bit ---------- Even parity, Odd parity, No parity
- Stop bit(s) ---------- 1 bit or 2 bits

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

14.6 Parity bit

Parity bit is calculated by doing an exclusive-or of all the data bits. If odd parity is used, the result of the exclusive-or is inverted. The parity bit is located between St + bits and first stop bit of a serial frame.

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

14.7 USART transmitter

USART Transmitter is enabled by setting the TXE bit in UCTRL1 register. When the Transmitter is enabled, normal port operation of TXD2 pin is overridden by serial output pin of the USART. Baud rate, operation mode and frame format must be setup once before doing any transmissions.

If synchronous or SPI operation is used, a clock on the XCK pin will be overridden and used as a transmission clock. If USART operates in SPI mode, SS2 pin is used as SS2 input pin in slave mode or can be configured as SS2 output pin in master mode. This can be done by setting SPISS bit in UCTRL3 register.

14.7.1 Sending Tx data

A data transmission is initiated by loading a transmit buffer (UDATA register I/O location) with data to be transmitted. The data written in the transmit buffer is moved to a shift register when the shift register is ready to send a new frame. The shift register is loaded with new data if it is in idle state or immediately after the last stop bit of the previous frame is transmitted.

When the shift register is loaded with new data, it will transfer one complete frame at the settings of control registers. If 9-bit characters are used in asynchronous or synchronous operation mode (USIZE[2:0] = 7), the ninth bit must be written to the TX8 bit in UCTRL3 register before loading a transmit buffer (UDATA register).

14.7.2 Transmitter flag and interrupt

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

UDRE flag indicates whether the transmit buffer is ready to be loaded with new data. This bit is set when the transmit buffer is empty and cleared when the transmit buffer contains transmission data which has not yet been moved into the shift register. And also this flag can be cleared by writing '0' to this bit field. Writing '1' to this field is not valid.

When Data Register Empty Interrupt Enable (UDRIE) bit in UCTRL2 register is set and Global Interrupt is enabled, USART Data Register Empty Interrupt is generated while UDRE flag is set.

Transmit Complete (TXC) flag bit is set when the entire frame in the transmit shift register has been shifted out and there are no more data in the transmit buffer. The TXC flag is automatically cleared when the Transmit Complete Interrupt service routine is executed, or it can be cleared by writing '0' to TXC bit in USTAT register.

When Transmit Complete Interrupt Enable (TXCIE) bit in UCTRL2 register is set and the Global Interrupt is enabled, USART Transmit Complete Interrupt is generated while TXC flag is set.

14.7.3 Parity generator

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

14.7.4 Disabling transmitter

Disabling the Transmitter by clearing TXE bit will not become effective until ongoing transmission is completed. When the Transmitter is disabled, the TXD2 pin is used as normal General Purpose I/O (GPIO) or primary function pin.

14.8 USART receiver

USART Receiver is enabled by setting the RXE bit in the UCTRL1 register. When the Receiver is enabled, normal pin operation of RXD2 pin is overridden by the USART as the serial input pin of the Receiver. Baud rate, mode of operation and frame format must be set before serial reception. If synchronous or SPI operation is used, a clock on the XCK pin will be used as a transfer clock. If the USART operates in SPI mode, SS2 pin is used as SS2 input pin in slave mode or can be configured as SS2 output pin in master mode. This can be done by setting SPISS bit in UCTRL3 register.

14.8.1 Receiving Rx data

When USART is in synchronous or asynchronous operation mode, the Receiver starts data reception when it detects a valid start bit (LOW) on RXD2 pin. Each bit following the start bit is sampled at pre-defined baud rate (asynchronous) or sampling edge of XCK (synchronous), and shifted into a receive shift register until the first stop bit of a frame is received.

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

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

14.8.2 Receiver flag and interrupt

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

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

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

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

Data Over Run (DOR) flag indicates data loss due to a receive buffer's full condition. The DOR occurs when the receive buffer is full, and another new data is present in the receive shift register which are to be stored into the receive buffer. After the DOR flag is set, all the incoming data are lost. To prevent data loss or clear this flag, read the receive buffer.

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

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

14.8.3 Parity checker

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

14.8.4 Disabling receiver

In contrast to Transmitter, disabling the Receiver by clearing RXE bit makes the Receiver inactive immediately. When the Receiver is disabled the Receiver flushes the receive buffer and the remaining data in the buffer is all reset. Function of USART is not overridden on the RXD2 pin, so the RXD2 pin becomes normal GPIO or primary function pin.

14.8.5 Asynchronous data reception

To receive asynchronous data frame, the USART includes a clock and data recovery unit. The Clock Recovery logic is used for synchronizing the internally generated baud rate clock to the incoming asynchronous serial frame on the RXD2 pin.

Data recovery logic samples incoming bits and low pass filters them, and this removes the noise of RXD2 pin.

Figure 72 describes sampling process of the start bit of an incoming frame. The sampling rate is 16 times the baud rate for normal mode, and 8 times the baud rate for Double Speed mode (U2X=1). The horizontal arrows show the synchronization variation due to the asynchronous sampling process. Note that larger time variation is shown when using the Double Speed mode.

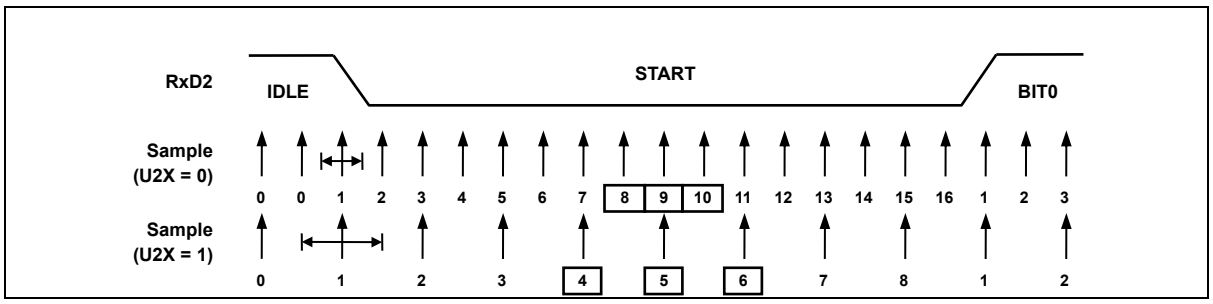


Figure 72. Start Bit Sampling

When the Receiver is enabled (RXE=1), the clock recovery logic tries to find a high to low transition on the RXD2 line, which is a start bit condition. After detecting the high to low transition on RXD2 line, the clock recovery logic uses the samples 8, 9, and 10 for Normal mode, and the samples 4, 5, and 6 for Double Speed mode to decide if a valid start bit is received. If more than 2 samples have logical low level, it is considered that a valid start bit is detected and the internally generated clock is synchronized to the incoming data frame. And the data recovery can begin. The synchronization process is repeated for each start bit.

As described above, when the Receiver clock is synchronized to the start bit, the data recovery can begin. Data recovery process is almost similar to the clock recovery process. The data recovery logic samples 16 times for each incoming bits for Normal mode and 8 times for Double Speed mode. It uses the samples 8, 9, and 10 to decide data value for Normal mode, and the samples 4, 5, and 6 for Double Speed mode.

If more than 2 samples have low levels, the received bit is considered to a logic 0. If more than 2 samples have high levels, the received bit is considered to a logic 1. The data recovery process is then repeated until a complete frame is received including the first stop bit. The decided bit value is stored in the receive shift register in order. Note that the Receiver only uses the first stop bit of a frame. Internally, after receiving the first stop bit, the Receiver is in idle state and waiting to find start bit.

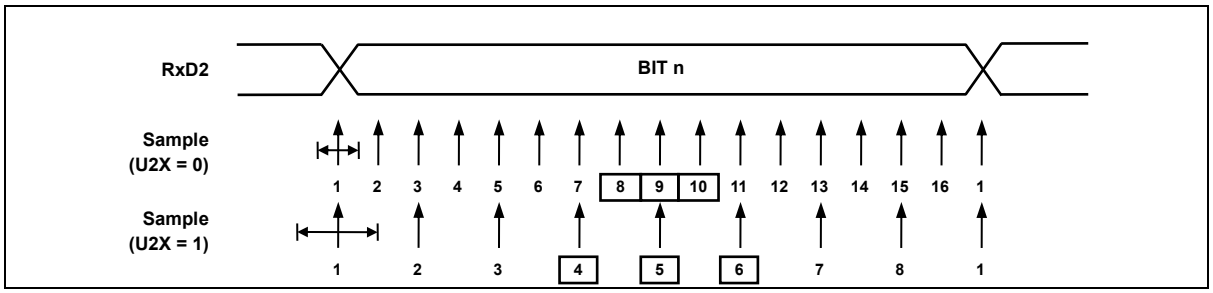


Figure 73. Sampling of Data and Parity Bit

A process for detecting stop bit is similar to the clock and data recovery process. That is, if 2 or more samples of 3 center values have high level, correct stop bit is detected. If not, a Frame Error flag will be set. After deciding whether a valid stop bit is received or not, the Receiver enters into idle state and monitors the RXD2 line to check a valid high to low transition is detected (start bit detection).

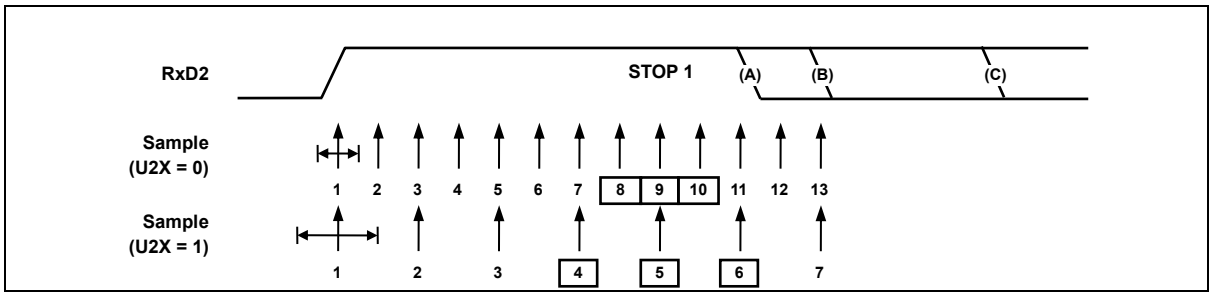


Figure 74. Stop Bit Sampling and Next Start Bit Sampling

14.9 SPI mode

The USART can be set to operate in industrial standard SPI compliant mode. The SPI mode has the following features.

- Full duplex, three-wire synchronous data transfer
- Master or Slave operation
- Supports all four SPI modes of operation (mode0, 1, 2, and 3)
- Selectable LSB first or MSB first data transfer
- Double buffered transmit and receive
- Programmable transmit bit rate

When SPI mode is enabled (UMSEL[1:0]=3), the Slave Select (SS2) pin becomes active low input in slave mode operation, or can be output in master mode operation if SPISS bit is set.

Note that during SPI mode of operation, the pin RXD2 is renamed as MISO2, and TXD2 is renamed as MOSI2 for compatibility to other SPI devices.

14.9.1 SPI clock formats and timing

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

Table 23. CPOL FunctionalityTable 23 shows four combinations of UCPOL and UCPHA for SPI mode 0, 1, 2, and 3.

Table 23. CPOL Functionality

SPI Mode	UCPOL	UCPHA	Leading edge	Trailing edge
0	0	0	Sample (Rising)	Setup (Falling)
1	0	1	Setup (Rising)	Sample (Falling)
2	1	0	Sample (Falling)	Setup (Rising)
3	1	1	Setup (Falling)	Sample (Rising)

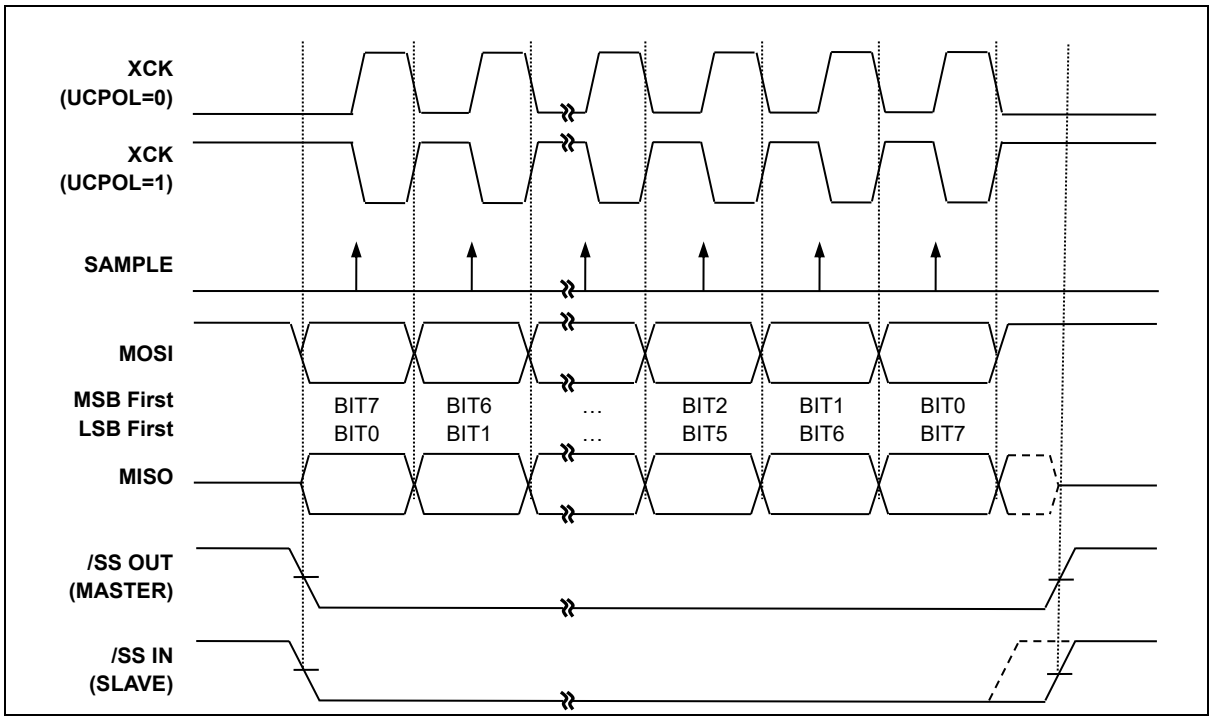


Figure 75. SPI Clock Formats when UCPHA = 0

When UCPHA=0, the slave begins to drive its MISO2 output with the first data bit value when SS goes to active low. The first XCK edge causes both the master and the slave to sample the data bit value on their MISO2 and MOSI inputs, respectively.

At the second XCK edge, the USART shifts the second data bit value out to the MOSI and MISO2 outputs of the master and slave, respectively. Unlike the case of UCPHA=1, when UCPHA=0, the slave's SS input must go to its inactive high level between transfers. This is because the slave can prepare the first data bit when it detects falling edge of SS input.

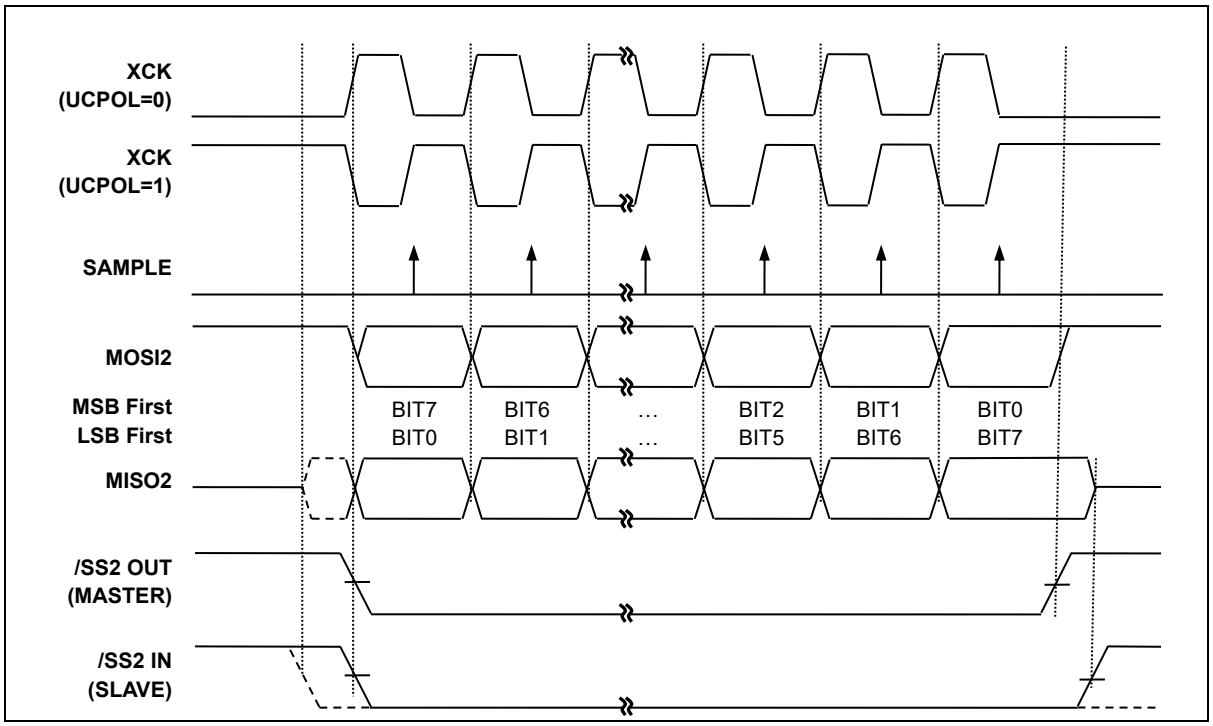


Figure 76. SPI Clock Formats when UCPHA = 1

When UCPHA=1, the slave begins to drive its MISO2 output when SS2 goes active low, but the data is not defined until the first XCK edge. The first XCK edge shifts the first bit of data from the shifter onto the MOSI2 output of the master and the MISO2 output of the slave.

The next XCK edge causes both the master and the slave to sample the data bit value on their MISO2 and MOSI2 inputs, respectively.

At the third XCK edge, the USART shifts the second data bit value out to the MOSI2 and MISO2 output of the master and slave respectively. When UCPHA=1, the slave's SS input is not required to go to its inactive high level between transfers.

Because an SPI logic reuses the USART resources, SPI mode of operation is similar to that of synchronous or asynchronous operation. An SPI transfer is initiated by checking for USART Data Register Empty flag (UDRE=1) and then by writing a byte of data to the UDATA Register.

In master mode of operation, even if transmission is not enabled (TXE=0), writing data to UDATA register is necessary because the clock XCK is generated from a transmitter block.

14.10 Receiver time out (RTO)

This USART system supports the time out function. This function generates an interrupt when stop bits are not in RX line during URTOC setting value. RTO Count stops in RXD signal live state and RTO Clear/Start is executed by stop bit recognition.

Example condition is listed in Table 24.

Table 24. Example Condition of RTO

Condition	• sysclk = 16MHz • Baud rate = 115,200 bps • Asynchronous Normal Mode (U2X = 0)
Baud rate	sysclk / 16 x (UBAUD + 1)
Calculated UBAUD	• (1000000 / Target Baud rate) – 1 = 7.68 ➔ **UBAUD = 8** • Error rate = 0.68 ➔ ➔
Real baud rate at sysclk 16Mhz	111,111 bps
1 bit time	9us
Maximum count time	9us * 65536(16bit count) = 589.8ms

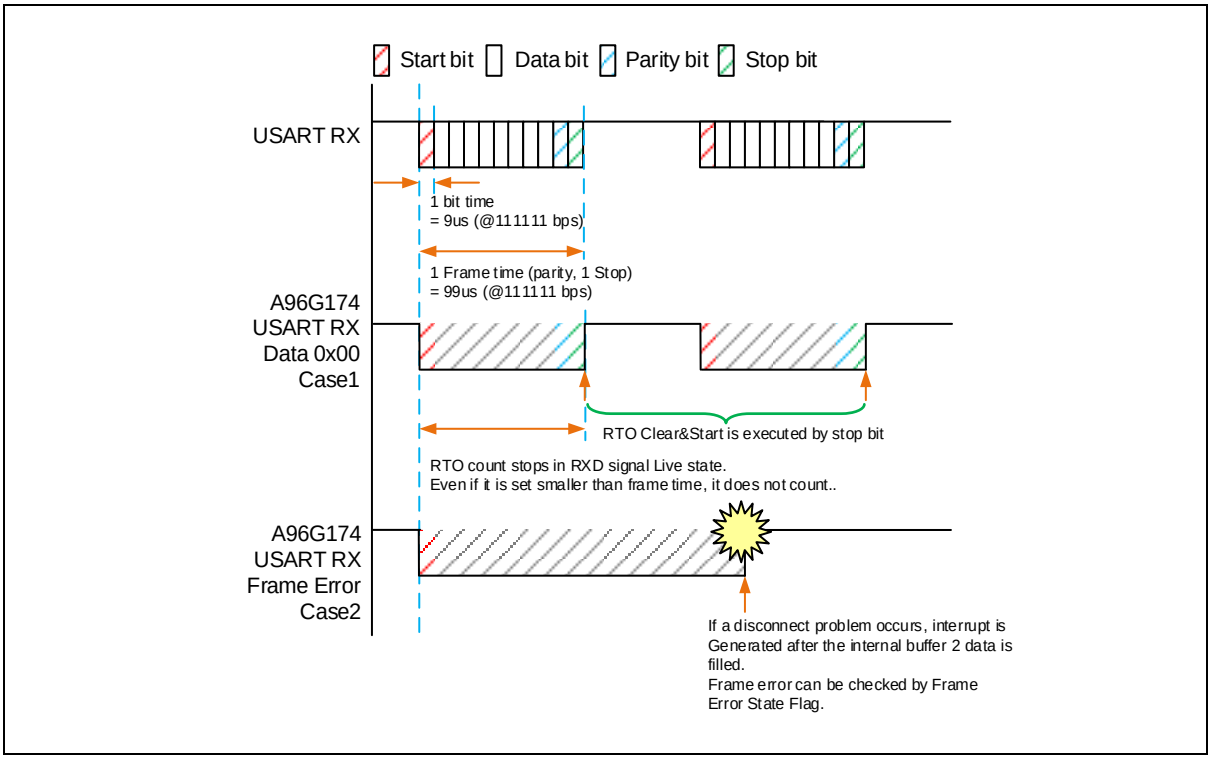


Figure 77. Example for RTO in USART

14.11 Register map

Table 25. USART Register Map

Name	Address	Direction	Default	Description
UCTRL1	CBH	R/W	00H	USART Control 1 Register
UCTRL2	CCH	R/W	00H	USART Control 2 Register
UCTRL3	CDH	R/W	00H	USART Control 3 Register
UCTRL4	1018H	R/W	00H	USART Control 4 Register
USTAT	CFH	R	80H	USART Status Register
UBAUD	FCH	R/W	FFH	USART Baud Rate Generation Register
UDATA	FDH	R/W	00H	USART Data Register
FPCR	1019H	R/W	00H	USART Floating Point Counter Register
RTOCH	101AH	R	00H	Receiver Time Out Counter High Register
RTOCL	101BH	R	00H	Receiver Time Out Counter Low Register

14.12 Register description

UCTRL1 (USART Control 1 Register) CBH

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

Initial value: 00_H

UMSEL[1:0] Selects operation mode of USART

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

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

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

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

USIZE2	USIZE1	USIZE0	Data length
0	0	0	5-bit
0	0	1	6-bit
0	1	0	7-bit
0	1	1	8-bit
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Reserved
1	1	1	9-bit

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

0	LSB First
1	MSB First

UCPOL Selects polarity of XCK in synchronous or SPI mode

0	TXD2 change @Rising Edge, RXD2 change @Falling Edge
1	TXD2 change @ Falling Edge, RXD2 change @ Rising Edge

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

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

UCTRL2 (USART Control 2 Register) CCH

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

Initial value: 00_H

UDRIE	Interrupt enable bit for USART Data Register Empty.	
	0	Interrupt from UDRE is inhibited (use polling)
	1	When UDRE is set, request an interrupt
TXCIE	Interrupt enable bit for Transmit Complete.	
	0	Interrupt from TXC is inhibited (use polling)
	1	When TXC is set, request an interrupt
RXCIE	Interrupt enable bit for Receive Complete	
	0	Interrupt from RXC is inhibited (use polling)
	1	When RXC is set, request an interrupt
WAKEIE	Interrupt enable bit for Asynchronous Wake in STOP mode. When device is in stop mode, if RXD2 goes to LOW level an interrupt can be requested to wake-up system.	
	0	Interrupt from Wake is inhibited
	1	When WAKE is set, request an interrupt NOTE) WAKEIE must set after USARTEN setting '1'.
TXE	Enables the transmitter unit.	
	0	Transmitter is disabled
	1	Transmitter is enabled
RXE	Enables the receiver unit.	
	0	Receiver is disabled
	1	Receiver is enabled
USARTEN	Activate USART module by supplying clock.	
	0	USART is disabled (clock is halted)
	1	USART is enabled
U2X	This bit only has effect for the asynchronous operation and selects receiver sampling rate.	
	0	Normal asynchronous operation
	1	Double Speed asynchronous operation

UCTRL3 (USART Control 3 Register) CDH

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

Initial value: 00_H

MASTER Selects master or slave in SPI or Synchronous mode operation and controls the direction of XCK pin.

0 Slave mode operation and XCK is input pin.

1 Master mode operation and XCK is output pin

LOOPS Controls the Loop Back mode of USART, for test mode

0 Normal operation

1 Loop Back mode

DISXCK In Synchronous mode of operation, selects the waveform of XCK output.

0 XCK is free-running while USART is enabled in synchronous master mode.

1 XCK is active while any frame is on transferring.

SPISS Controls the functionality of SS2 pin in master SPI mode.

0 SS2 pin is normal GPIO or other primary function

1 SS2 output to other slave device

USBS Selects the length of stop bit in Asynchronous or Synchronous mode of operation.

0 1 Stop bit

1 2 Stop bit

TX8 The ninth bit of data frame in Asynchronous or Synchronous mode of operation. Write this bit first before loading the UDATA register.

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

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

RX8 The ninth bit of data frame in Asynchronous or Synchronous mode of operation. Read this bit first before reading the receive buffer.

0 MSB (9^{th} bit) received is '0'

1 MSB (9^{th} bit) received is '1'

UCTRL4 (USART Control 4 Register) 1018H

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

Initial value: 00_H

RTOEN Enable receiver time out.
- 0 Disable
- 1 Enable

RTO_FLAG This bit is set when RTO count overflows. This flag can generate an RTO interrupt. Writing '0' to this bit position will clear RTO_FLAG.
- 0 RTO count dose not overflow.
- 1 RTO count overflow.

FPCREN Enable baud rate compensation
- 0 Disable
- 1 Enable

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

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

USTAT (USART Status Register) CFH

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

Initial value: 80_H

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

0 Transmit buffer is not empty.

1 Transmit buffer is empty.

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

0 Transmission is ongoing.

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

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

0 There is no data unread in the receive buffer

1 There are more than 1 data in the receive buffer

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

0 No WAKE interrupt is generated.

1 WAKE interrupt is generated.

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

0 No operation

1 Reset USART

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

0 No Data OverRun

1 Data OverRun detected

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

0 No Frame Error

1 Frame Error detected

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

0 No Parity Error

1 Parity Error detected

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

UBAUD (USART Baud-Rate Generation Register) FCH

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

Initial value: FF_H

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

UDATA (USART Data Register) FDH

7	6	5	4	3	2	1	0
UDATA7	UDATA6	UDATA 5	UDATA 4	UDATA 3	UDATA 2	UDATA 1	UDATA 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00_H

UDATA [7:0]	The USART Transmit Buffer and Receive Buffer share the same I/O address with this DATA register. The Transmit Data Buffer is the destination for data written to the UDATA register. Reading the UDATA register returns the contents of the Receive Buffer. Write this register only when the UDRE flag is set. In SPI or synchronous master mode, write this register even if TX is not enabled to generate clock, XCK.

FPCR (USART Floating Point Register) 1019H

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

Initial value: 00_H

FPCR [7:0]	USART Floating Point Counter 8-bit floating point counter

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

Example1

- Condition : sysclk = 16MHz, Baud rate = 9600 bps, Asynchronous Normal Mode (U2X = 0)
- Baud rate = sysclk / 16 x (UBAUD + 1)
- Calculated UBAUD = (1000000 / Target Baud rate) – 1 = 103.17, Error rate = 0.17 ⇒ **UBAUD = 104**
- UCTRL4 = 0x04, Enable baud rate Compensation
- Calculated FPCR = (UBAUD - Calculated UBAUD) x 256 = (104 – 103.17) x 256 = 212.48 ⇒ **FPCR = 213**

Example2

- Condition : sysclk = 16MHz, Baud rate = 115,200 bps, Asynchronous Normal Mode (U2X = 0)
- Baud rate = sysclk / 16 x (UBAUD + 1)
- Calculated UBAUD = (1000000 / Target Baud rate) – 1 = 7.68, Error rate = 0.68 ⇒ **UBAUD = 8**
- UCTRL4 = 0x04, Enable baud rate Compensation
- Calculated FPCR = (UBAUD - Calculated UBAUD) x 256 = (8 – 7.68) x 256 = 81.92 ⇒ **FPCR = 82**

RTOCH (Receiver Time Out Counter High Register) 101AH

7	6	5	4	3	2	1	0
RTOCH7	RTOCH6	RTOCH5	RTOCH4	RTOCH3	RTOCH2	RTOCH1	RTOCH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00_H

RTOCL (Receiver Time Out Counter Low Register) 101BH

7	6	5	4	3	2	1	0
RTOCL7	RTOCL6	RTOCL5	RTOCL4	RTOCL3	RTOCL2	RTOCL1	RTOCL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00_H

14.13 Baud rate settings (example)

Table 26. Examples of UBAUD Settings for Commonly Used Oscillator Frequencies

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

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

Table 26. Examples of UBAUD Settings for Commonly Used Oscillator Frequencies (continued)

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

14.14 0% error baud rate

USART system of A96G174/A96S174 supports floating point counter logic for 0% error of baud rate. By using 8-bit floating point counter logic, cumulative error to below the decimal point can be removed.

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

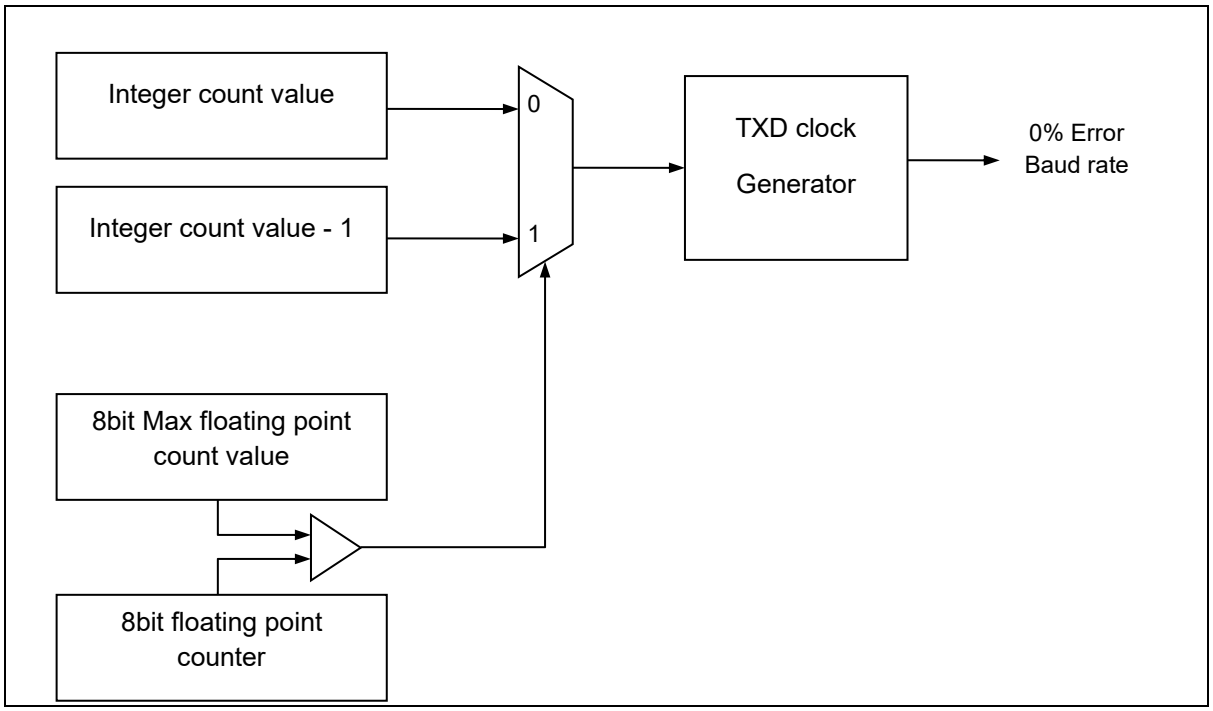


Figure 78. 0% Error Baud Rate Block Diagram

15 Power down operation

A96G174/A96S174 has two power-down modes to minimize the power consumption of the device. In power down mode, power consumption is reduced considerably. A96G174/A96S174 provides three kinds of power saving functions such as Main-IDLE mode, Sub-IDLE mode and STOP mode. During one of these three modes, program will be stopped.

15.1 Peripheral operation in IDLE/ STOP mode

Table 27 shows operation status of each peripheral in IDLE mode and STOP mode.

Table 27. Peripheral Operation Status during Power-down Mode

Peripheral	IDLE mode	STOP mode
CPU	ALL CPU operations are disabled.	ALL CPU operations are disabled.
RAM	Retains.	Retains.
Basic Interval Timer	Operates continuously.	Stops (can be operated with WDTRC OSC).
Watch Dog Timer	Operates continuously.	Stops (can be operated with WDTRC OSC).
Timer0~2	Operates continuously.	Halts (only when the event counter mode is enabled, timer operates normally).
ADC	Operates continuously.	Stops.
Internal OSC (32MHz)	Oscillates.	Stops when the system clock (fx) is fHSIRC.
WDTRC OSC (128kHz)	Can be operated with setting value.	Can be operated programmable.
I/O Port	Retains.	Retains.
Control Register	Retains.	Retains.
Address Data Bus	Retains.	Retains.
Release Method	By RESET All Interrupts	By RESET Timer Interrupt (EC0, EC1, EC3) External Interrupt USART by RX, WT (sub clock), WDT USI0/1 by RX, I2C(Slave mode)

15.2 IDLE mode

Power control register is set to '01h' to enter into IDLE mode. In IDLE mode, internal oscillation circuits remain active. Oscillation continues and peripherals are operated normally, but CPU stops.

It is released by reset or an interrupt. To be released by an interrupt, the interrupt should be enabled before IDLE mode. If using a reset, because the device is initialized, registers become to have reset values.

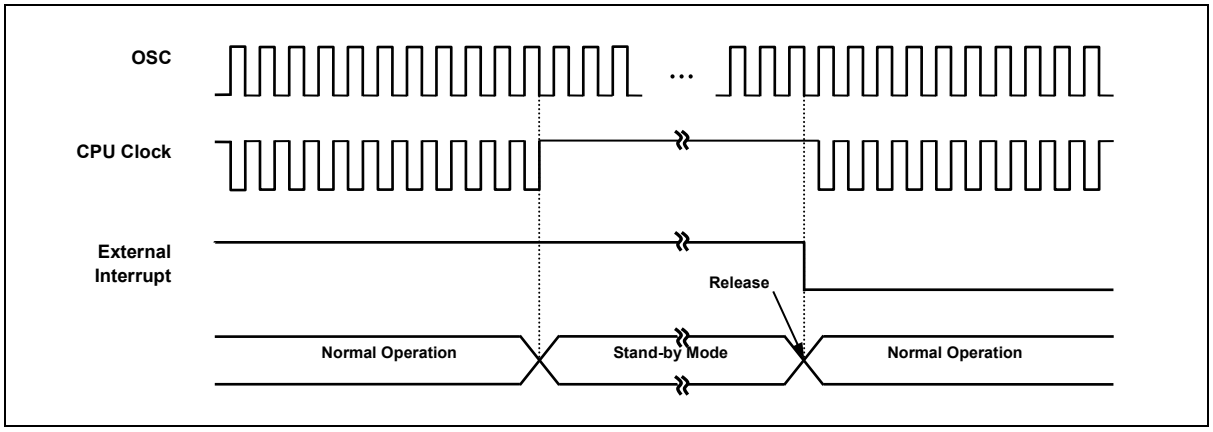


Figure 79. IDLE Mode Release Timing by an External Interrupt

15.3 STOP mode

Power control register is set to '03H' to enter into STOP mode. In STOP mode, the selected oscillator, system clock and peripheral clock is stopped, but watch timer can be continued to operate with sub clock.

With the clock frozen, all functions are stopped, but the on-chip RAM and control registers are held. For example, If the internal RC oscillator (fIRC) is selected for the system clock and the sub clock (fSUB) is oscillated, the internal RC oscillator stops oscillation and the sub clock is continuously oscillated in stop mode. At that time, the watch timer can be operated with the sub clock.

Sources to exit from STOP mode is hardware reset and interrupts. The hardware reset re-defines all control registers. When awaking from STOP mode, enough oscillation stabilization time is required to normal operation. Figure 80 shows the timing diagram.

As shown in the Figure 80, when released from STOP mode, the basic interval timer is activated on wake-up. Therefore, before STOP instruction, a user must set relevant prescale divide ratio to have long enough time. This guarantees that an oscillator has started and stabilized.

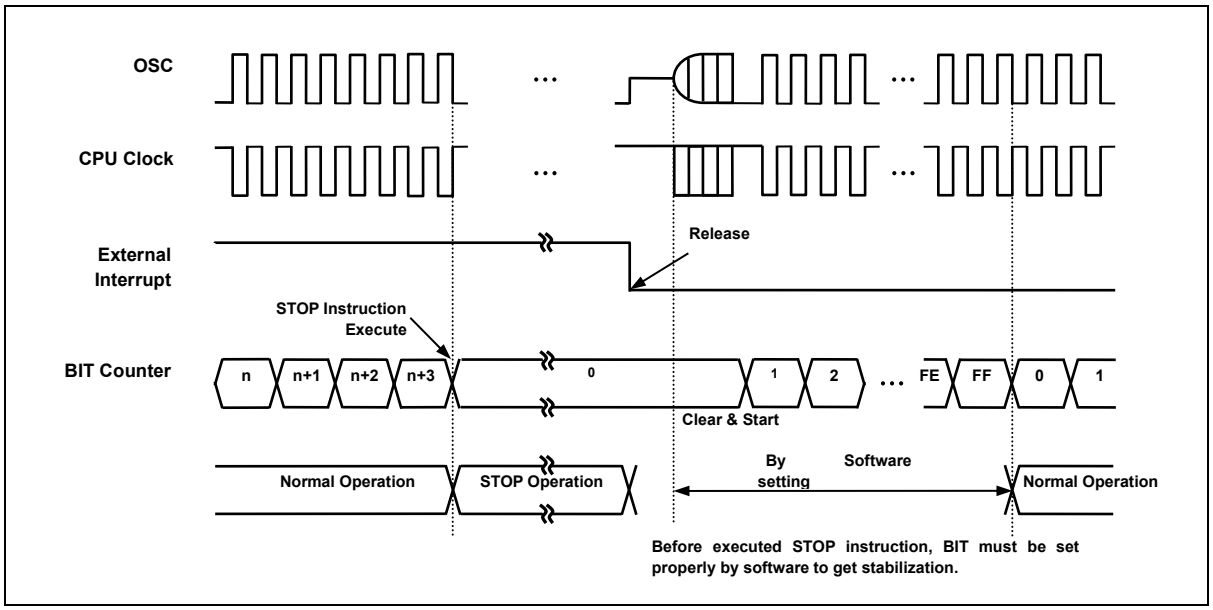


Figure 80. STOP Mode Release Timing by External Interrupt

15.4 Released operation of STOP mode

After STOP mode is released, operation begins according to content of related interrupt register just before STOP mode starts (refer to Figure 81). If the global interrupt Enable Flag (IE.EA)is set to `1`, the STOP mode is released by a certain interrupt of which interrupt enable flag = `1` and the CPU jumps to the relevant interrupt service routine. Even if the IE.EA bit is cleared to '0', the STOP mode is released by the interrupt of which the interrupt enable flag is set to '1'.

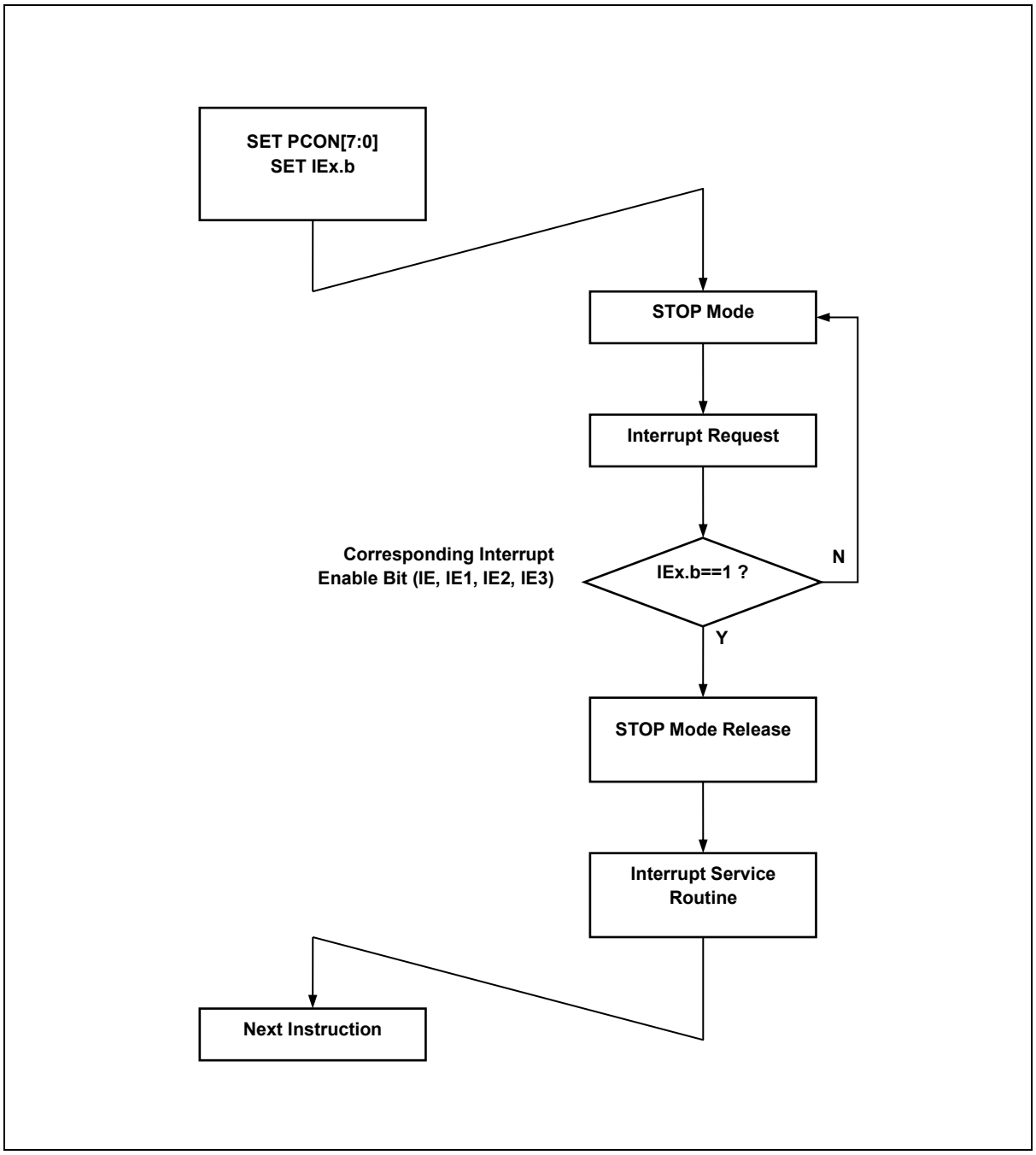


Figure 81. STOP Mode Release Flow

15.5 Register map

Table 28. Power-down Operation Register Map

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

15.6 Register description

PCON (Power Control Register): 87H

7	6	5	4	3	2	1	0
PCON7	–	–	–	PCON3	PCON2	PCON1	PCON0
R/W	–	–	–	R/W	R/W	R/W	R/W

Initial value: 00H

PCON[7:0]	Power Control
01H	IDLE mode enable
03H	STOP mode enable
Other Values	Normal operation

NOTES:

1. To enter into IDLE mode, PCON must be set to '01H'.
2. To enter into STOP mode, PCON must be set to '03H'.
3. The PCON register is automatically cleared by a release signal in STOP/IDLE mode.

Three or more NOP instructions must immediately follow the instruction that make the device enter into STOP/IDLE mode. Refer to the following examples.

```
Example 1                          Example 2
MOV   PCON, #01H   ; IDLE mode     MOV   PCON, #03H   ; STOP mode
NOP                                NOP
NOP                                NOP
NOP                                NOP
.                                  .
.                                  .
.                                  .
```

16 Reset

Table 29 shows hardware setting values of main peripherals.

Table 29. Hardware Setting Values in Reset State

On Chip Hardware	Initial Value
Program Counter (PC)	0000h
Accumulator	00h
Stack Pointer (SP)	07h
Peripheral Clock	On
Control Register	Refer to the Peripheral Registers

A96G174/A96S174 has five types of reset sources as shown in the followings:

- External RESETB
- Power ON RESET (POR)
- WDT Overflow Reset (In the case of WDTEN = `1`)
- Low Voltage Reset (In the case of LVREN = `0 `)
- OCD Reset

16.1 Reset block diagram

In this section, reset unit is described in a block diagram.

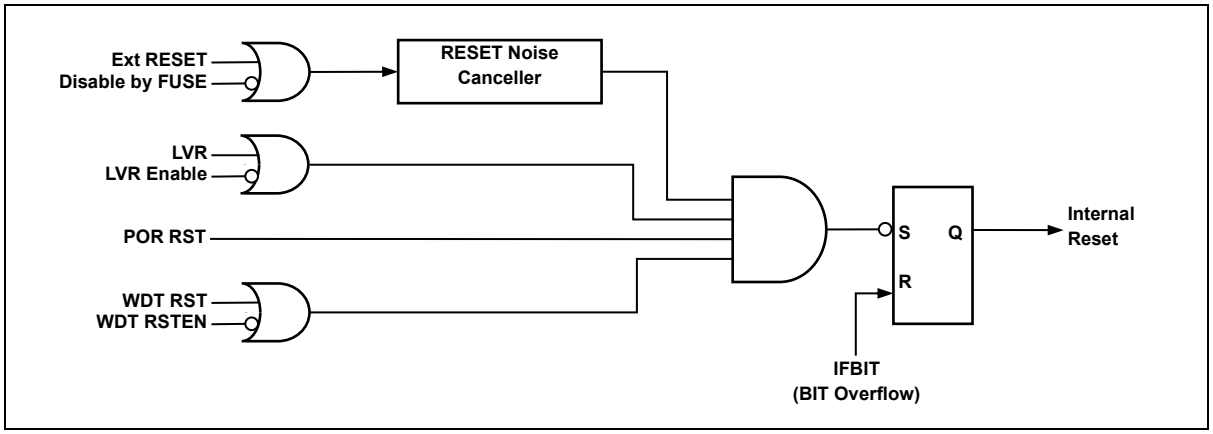


Figure 82. Reset Block Diagram

16.2 Power on reset

When rising device power, POR (Power On Reset) has a function to reset a device. If POR is used, it executes the device RESET function instead of the RESET IC or the RESET circuits.

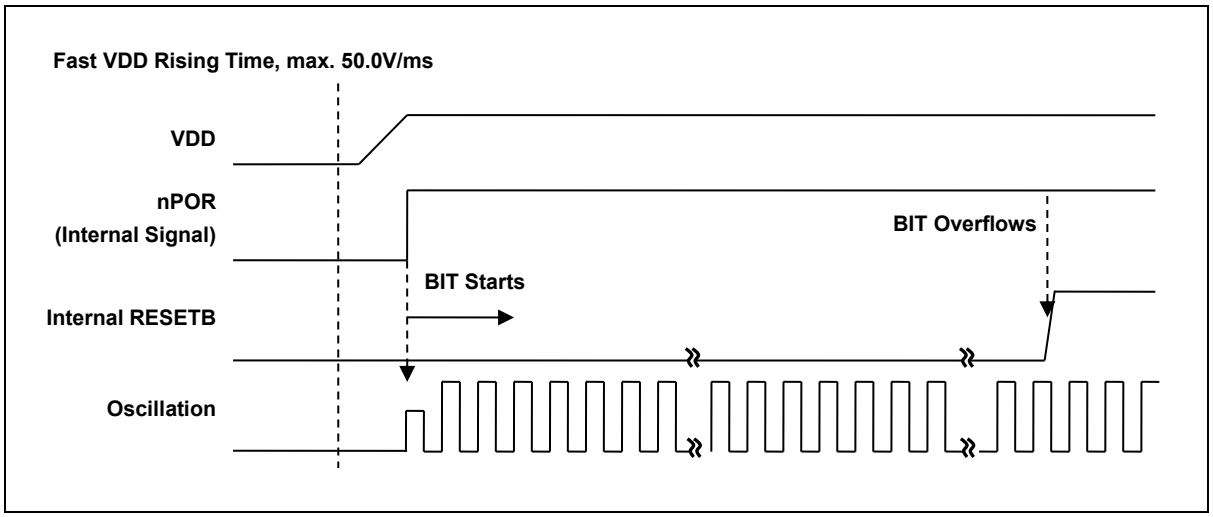


Figure 83. Fast VDD Rising Time

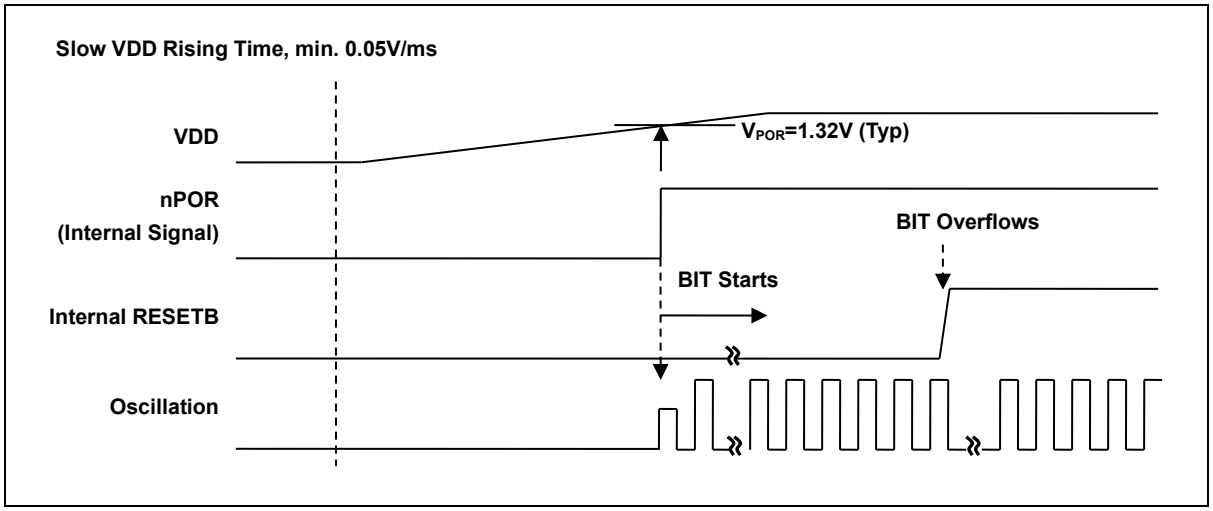


Figure 84. Internal RESET Release Timing On Power-Up

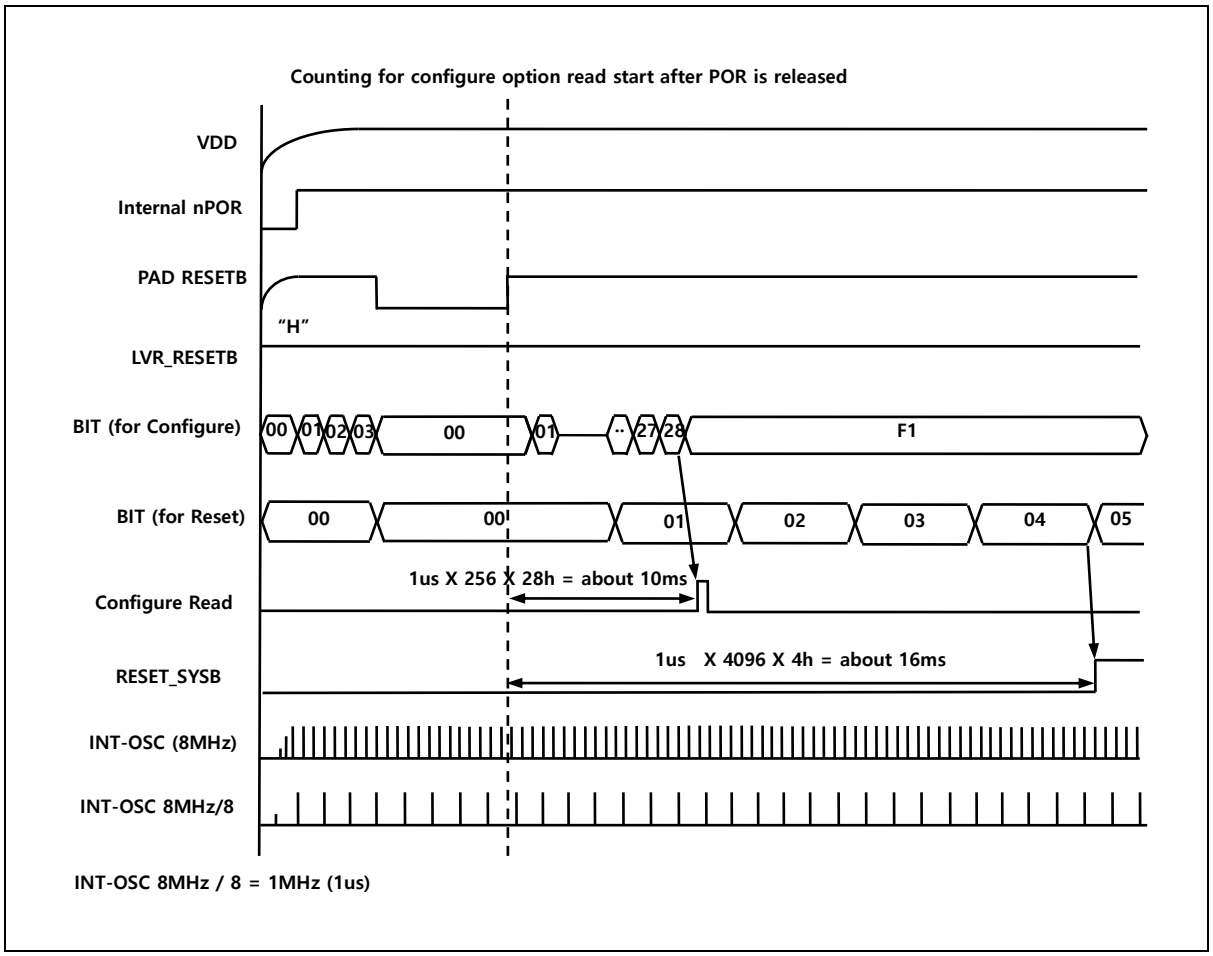


Figure 85. Configuration Timing when Power-on

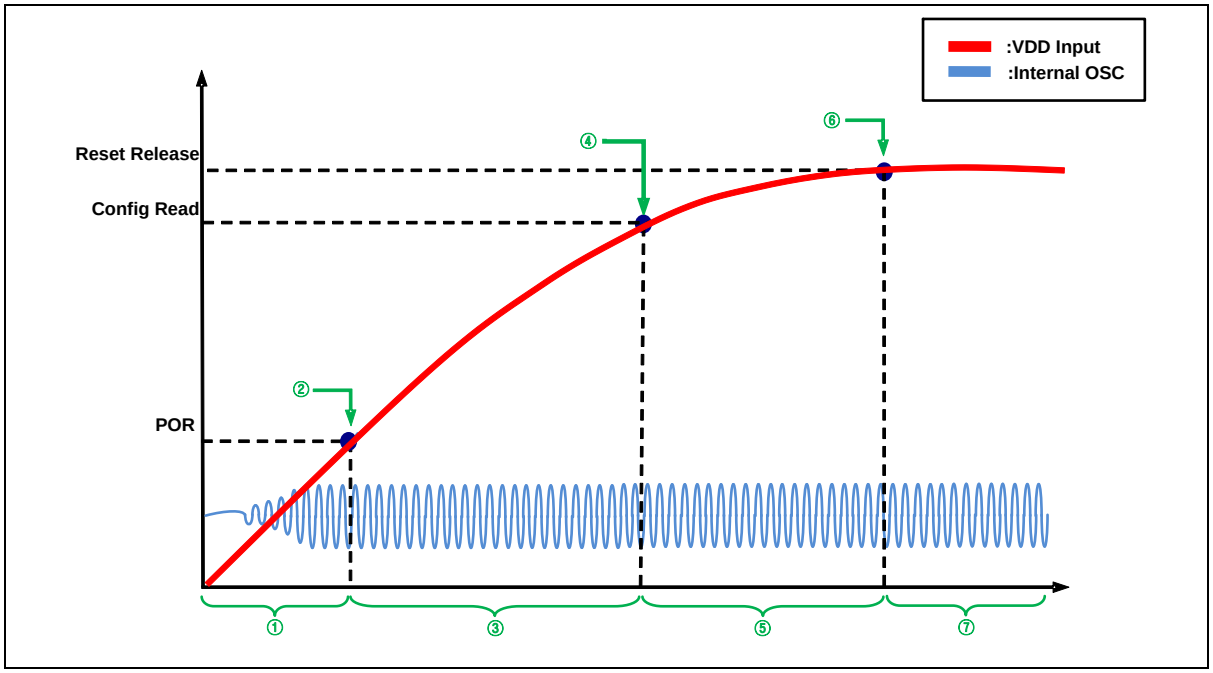


Figure 86. Boot Process Waveform

Table 30. Boot Process Description

Process	Description	Remarks
①	• No Operation • LSIRC (128kHz) ON	0.7V to 0.9V
②	1st POR level Detection	About 1.1V to 1.3V
③	• (LSIRC 128kHz/32)x32h Delay section (=10ms) • VDD input voltage must rise over than flash operating voltage for Configure option read	Slew Rate >= 0.025V/ms
④	Configure option read point	• About 1.6V to 1.8V • Configure 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	

16.3 External RESETB input

External RESETB is input to a Schmitt trigger. If the RESETB pin is held with low for at least 50us over within the operating voltage range and stable oscillation, it is applied and the internal state is initialized. After reset state becomes '1', it needs 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.

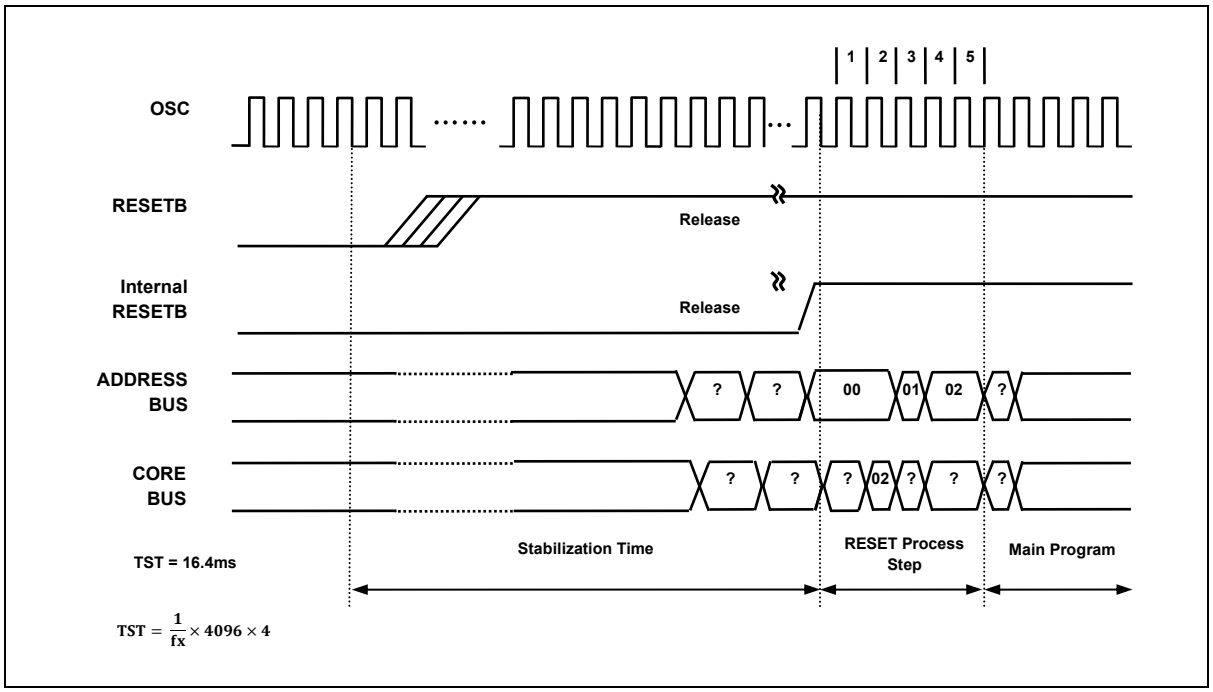


Figure 87. Timing Diagram after RESET

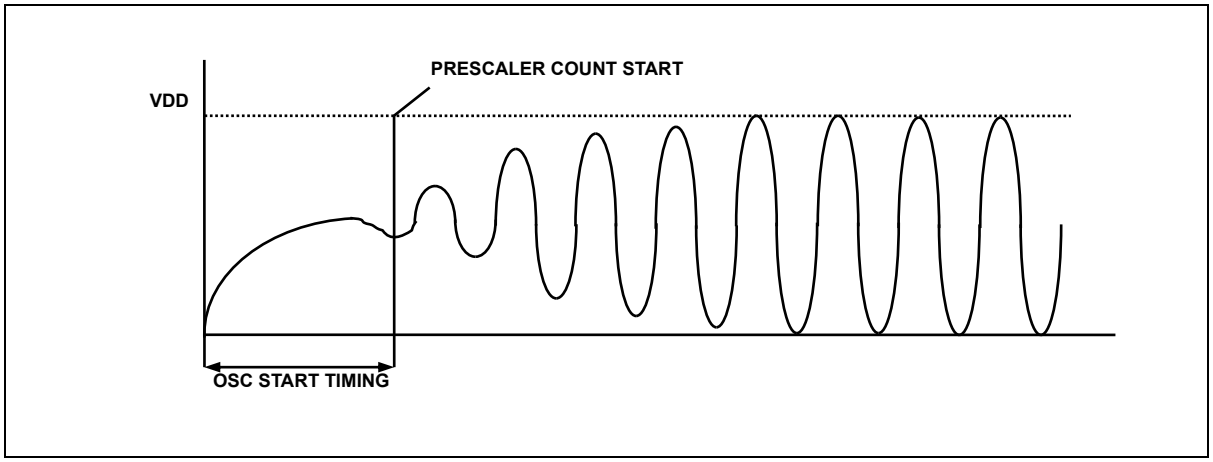


Figure 88. Oscillator generating waveform example

As shown Figure 88, the stable generating time is not included in the start-up time. The RESETB pin has a pull-up register by hardware

16.4 Low voltage reset process

A96G174/A96S174 has an On-chip brown-out detection circuit (BOD) for monitoring VDD level during operation by comparing it to a fixed trigger level. Trigger level for the BOD can be selected by configuring LVRVS[2:0] bits to be 1.61V, 1.77V, 2.13V, 2.46V, 3.56V.

In the STOP mode, this will contribute significantly to the total current consumption. So to minimize the current consumption, LVREN bit is set to off by software.

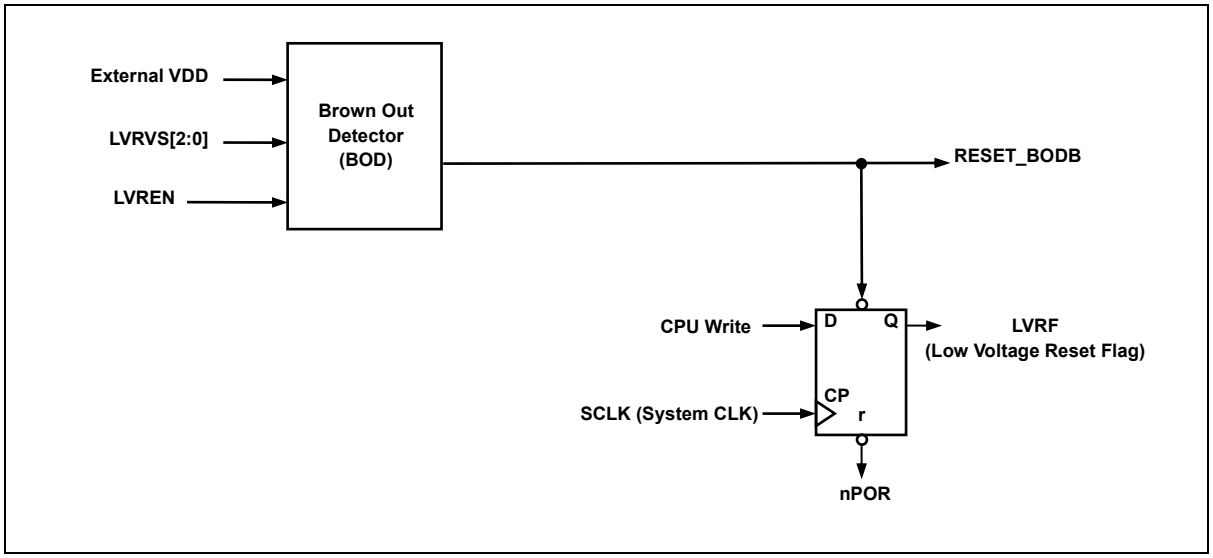


Figure 89. Block Diagram of LVR

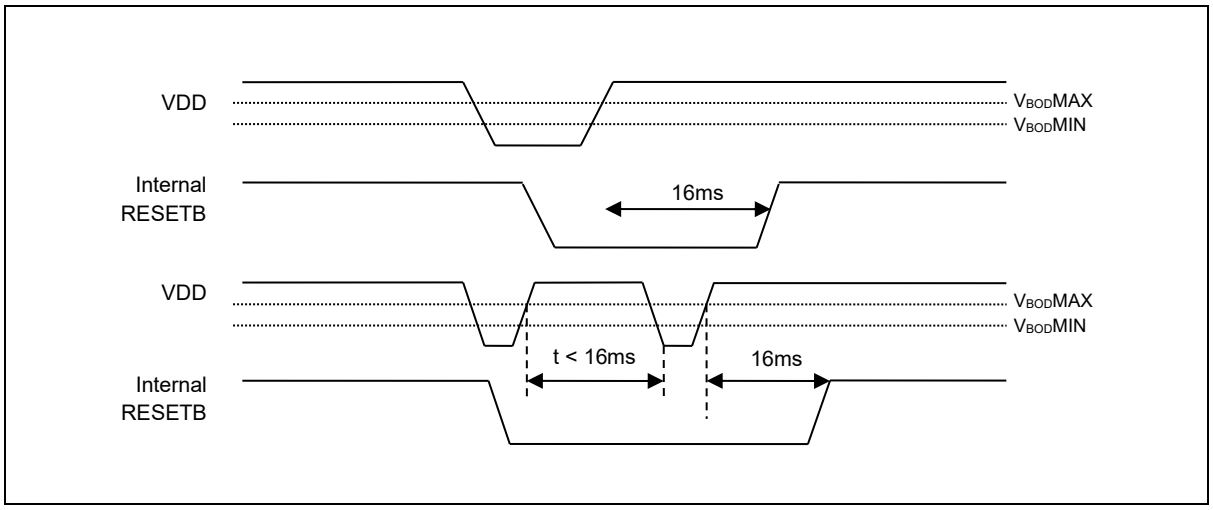


Figure 90. Internal Reset at Power Fail Situation

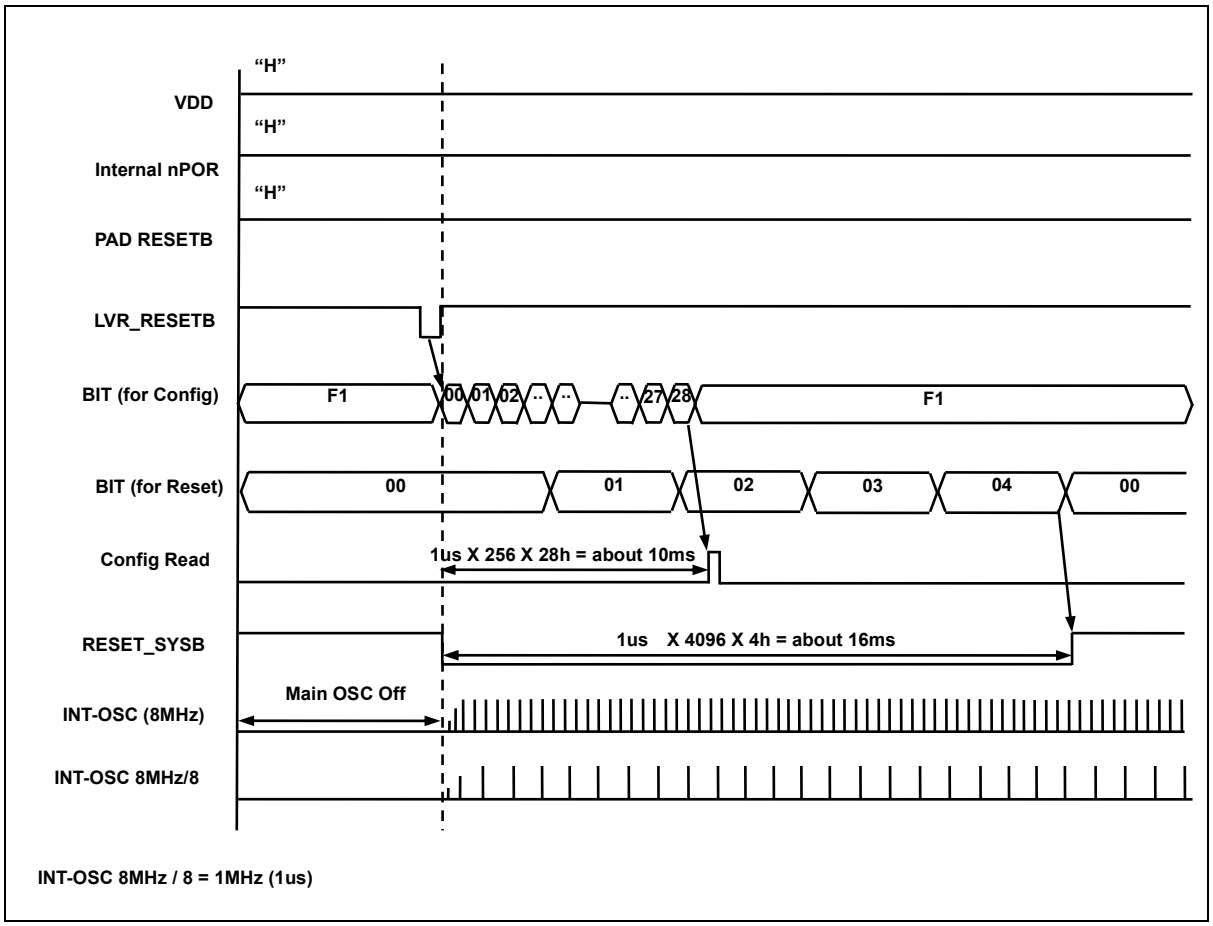


Figure 91. Configuration Timing When LVR RESET

16.5 LVI block diagram

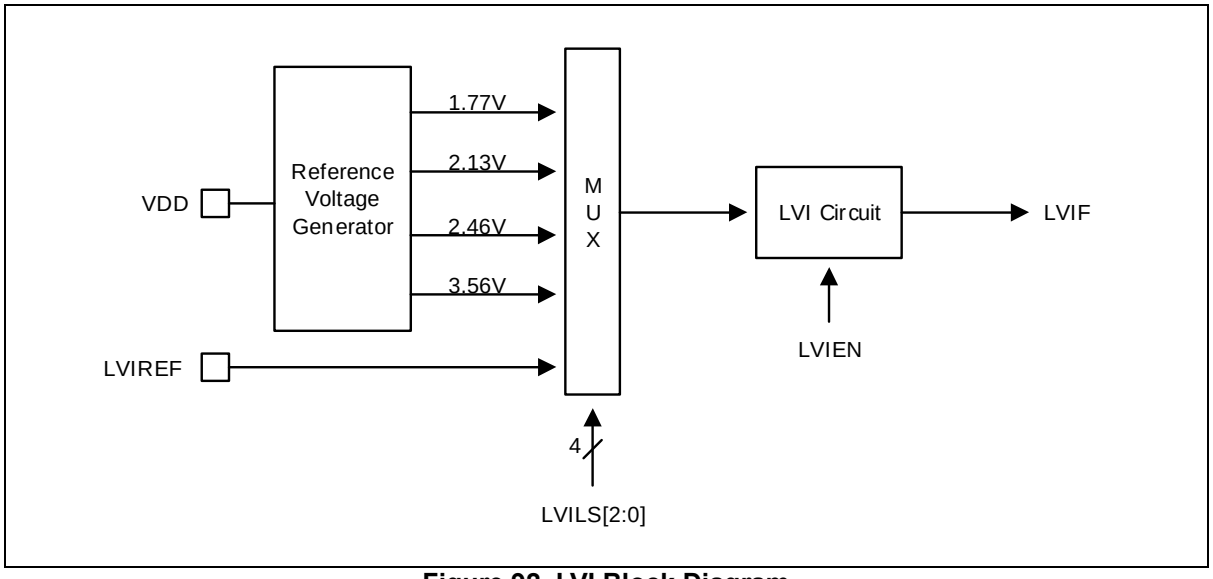


Figure 92. LVI Block Diagram

16.6 Register Map

Table 31. Reset Operation Register Map

Name	Address	Direction	Default	Description
RSTFR	E8H	R/W	80H	Reset Flag Register
LVRCR	D8H	R/W	00H	Low Voltage Reset Control Register
LVICR	86H	R/W	00H	Low Voltage Indicator Control Register

16.7 Reset Operation Register Description

RSTFR (Reset Flag Register): E8H

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

Initial value: 80H

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

EXTRF External Reset (RESETB) flag bit. The bit is reset by writing '0' to this bit or by Power-On Reset.
- 0 No detection
- 1 Detection

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

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

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

NOTES:

1. When the Power-On Reset occurs, the PORF bit is only set to "1", the other flag (WDTRF) bits are all cleared to "0".
2. When the Power-On Reset occurs, the EXTRF bit is unknown, at that time, the EXTRF bit can be set to "1" when External Reset (RESETB) occurs.
3. When the Power-On Reset occurs, the LVRF bit is unknown, at that time, the LVRF bit can be set to "1" when LVR Reset occurs.
4. When a reset except the POR occurs, the corresponding flag bit is only set to "1", the other flag bits are kept in the previous values.

LVRCR (Low Voltage Reset Control Register): D8H

7	6	5	4	3	2	1	0
–	–	–	–	LVRVS2	LVRVS1	LVRVS0	LVREN
–	–	–	–	R/W	R/W	R/W	R/W

Initial value: 00H

LVRVS[2:0] LVR Voltage Select

LVRVS2	LVRVS1	LVRVS0	Description
0	0	0	1.61V
0	0	1	1.77V
0	1	0	2.13V
0	1	1	2.46V
1	0	0	3.56V
Other Values			Reserved

LVREN LVR Operation

0	LVR Enable
1	LVR Disable

NOTES:

1. The LVRVS[2:0] bits are cleared by a power-on reset but are retained by other reset signals.
2. The LVRVS[2:0] bits should be set to '0000b' while LVREN bit is "1".

LVICR (Low Voltage Indicator Control Register): 86H

7	6	5	4	3	2	1	0
–	–	LVIF	LVIEN	–	–	LVILS1	LVILS0
–	–	R/W	R/W	–	–	R/W	R/W

Initial value: 00H

LVIF Low Voltage Indicator Flag Bit

0	No detection
1	Detection

LVIEN LVI Enable/Disable

0	Disable
1	Enable

LVIVS[1:0] LVI Level Select

LVIVS1	LVIVS0	Description
0	0	1.77V
0	1	2.13V
1	0	2.46V
1	1	3.56V

17 Memory programming

A96G174/A96S174 has flash memory to which a program can be written, erased, and overwritten while mounted on the board. Serial ISP mode is supported.

Flash of A96G174/A96S174 features the followings:

- Flash Size : 8Kbytes
- Single power supply program and erase
- Command interface for fast program and erase operation
- Up to 30,000 program/erase cycles at typical voltage and temperature for flash memory
- Security feature

17.1 Flash control and status registers

Registers controlling 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.

17.1.1 Register map

Table 32. Flash Control and Status Register Map

Name	Address	Dir	Default	Description
FEMR	1020H	R/W	00H	Flash Mode Register
FECR	1021H	R/W	03H	Flash Control Register
FESR	1022H	R/W	80H	Flash Status Register
FETCR	1023H	R/W	00H	Flash Time Control Register
FEARL1	1025H	R/W	00H	Flash Address Low Register 1
FEARM1	1024H	R/W	00H	Flash Address Middle Register 1
FEARL	102AH	R/W	00H	Flash Address Low Register
FEARM	1029H	R/W	00H	Flash Address Middle Register
FEARH	1028H	R/W	00H	Flash Address High Register

17.1.2 Register description

FEMR (Flash Mode Register): 1020H

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): 1021H

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

Initial value: 03H

AEF — Enable flash bulk erase mode

0	Disable bulk erase mode of Flash memory
1	Enable bulk erase mode of Flash memory

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

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

WRITE — Start to program or erase of Flash. It is cleared automatically after 1 clock

0	No operation
1	Start to program or erase of Flash

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

0	No operation
1	Start auto-verify of Flash (Checksum or CRC16)

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

0	Reset Flash control logic
1	No operation (default)

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

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

NOTE: 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): 1022H

7	6	5	4	3	2	1	0
PEVBSY	REMAPSI	REMAP	-	ROMINT	WMODE	EMODE	VMODE
R	R/W	R/W	-	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
REMAPSI	Remapping for check the serial ID.
	0 No operation
	1 Remapping OTP area to FFC0~FFFF.
REMAP	Remapping OTP.
	0 No operation
	1 Remapping OTP all area.
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

FEARL1 (Flash address low Register 1): 1025H

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

Initial value: 00H

ARL1[7:0]	Flash address low 1

FEARM1 (Flash address middle Register 1): 1024H

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

Initial value: 00H

ARM1[7:0]	Flash address middle 1

FEARL (Flash address low Register): 102AH

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): 1029H

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): 1028H

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

NOTES:

1. 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.
2. EARs are write-only register. Reading these registers returns 24-bit checksum result.
3. When calculating flash checksum, the lower 4 bits of start address are calculated as 0x0000 and the lower 4 bits of end address as 0x1111 for protection.
4. This device can support internal Checksum calculation, device verification time will be decreased dramatically.
5. Checksum cannot detect error address or error bit, but it is quite good feature in mass product programming.
6. Device data read out time takes few seconds. The execution time per byte is 4~5ms based on 16MHz.

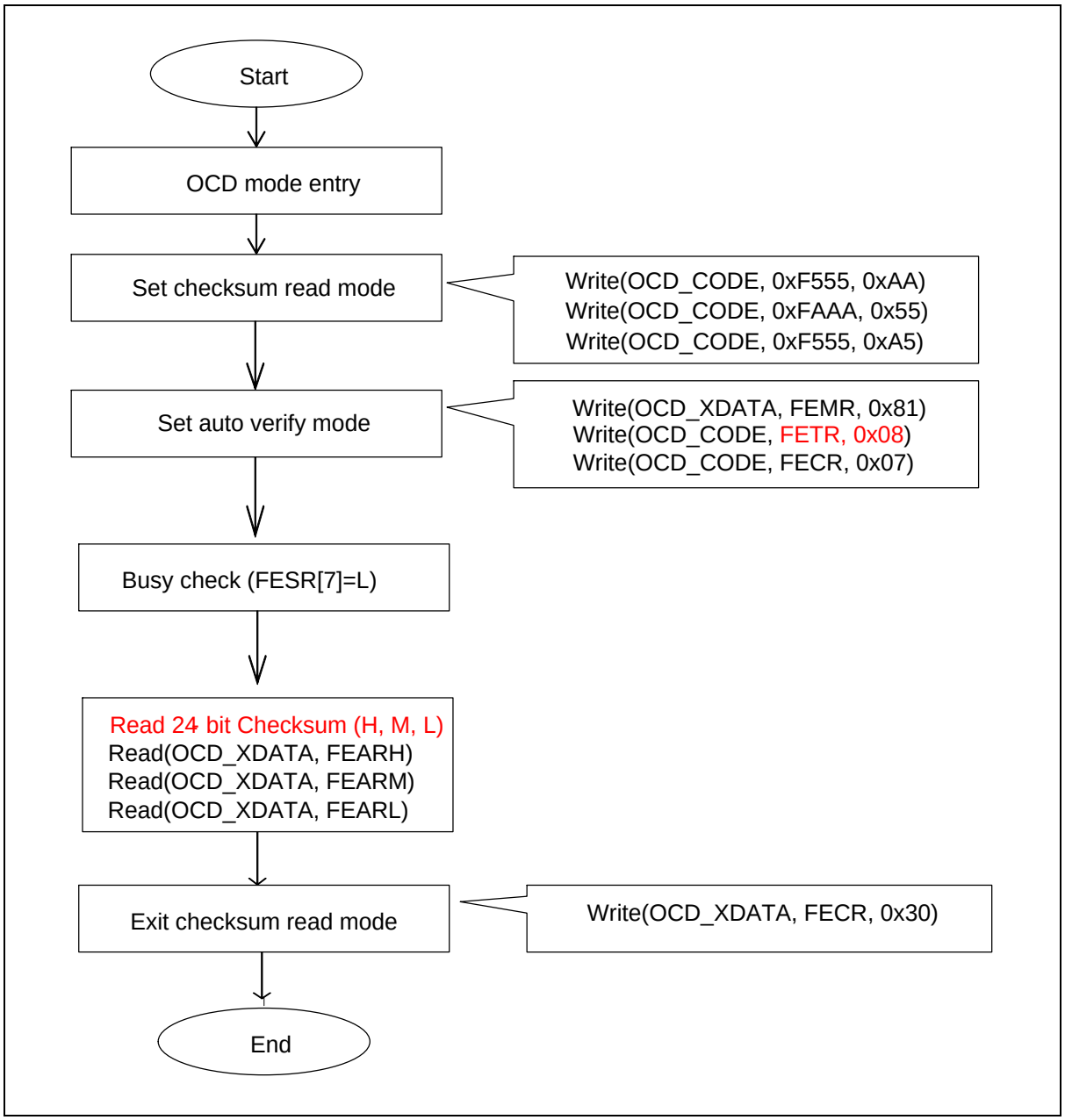


Figure 93. Read Device Internal Checksum (Full Size)

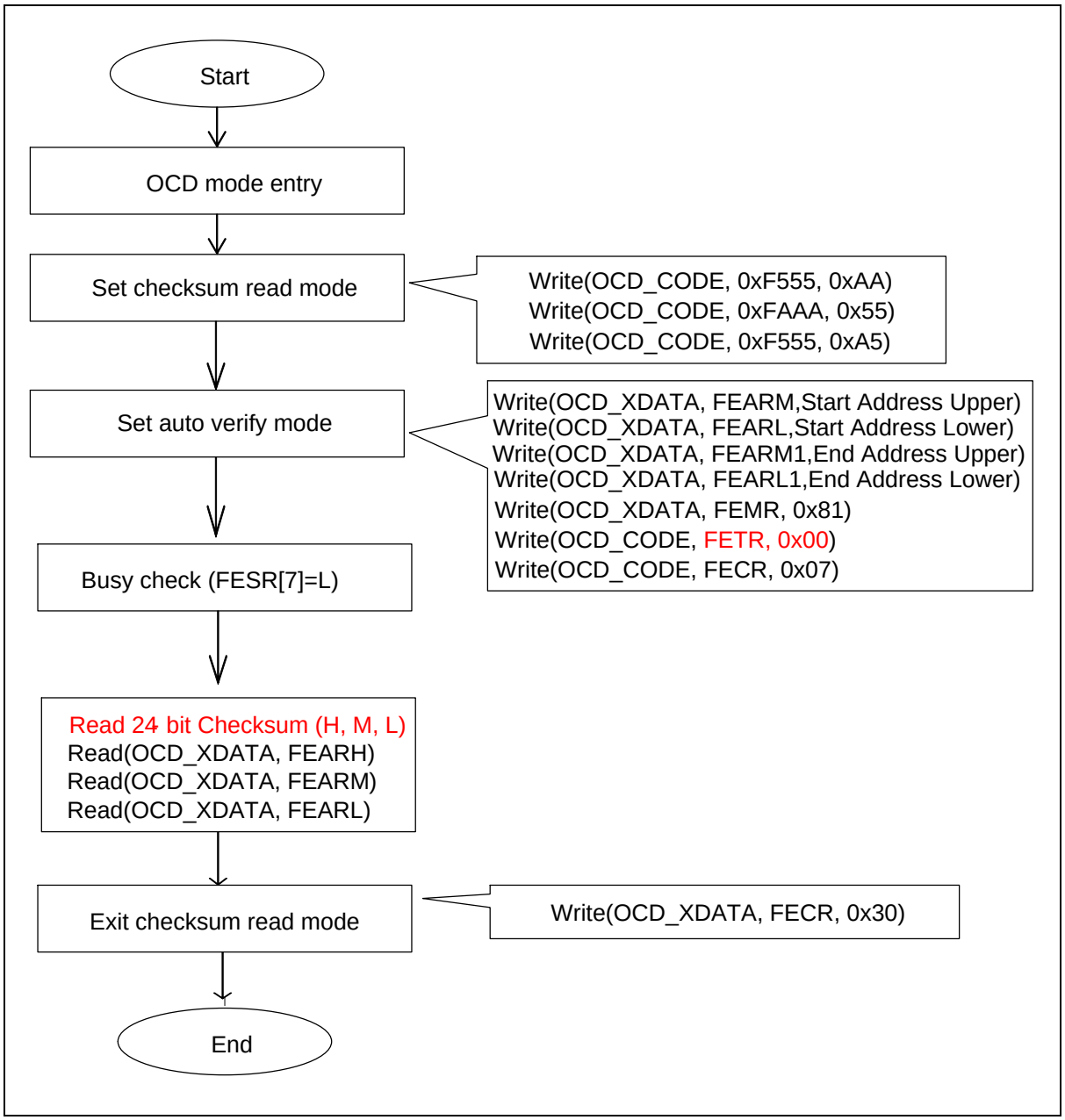


Figure 94. Read Device Internal Checksum (User Define Size)

FETCR (Flash Time control Register): 1023H

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_{LSIRC}=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 INTRC/256 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_{LSIRC})
- For bulk erase, Tbe = (TCON+1) * 4 * (f_{LSIRC})
- Recommended bulk erase time : FETCR = 57h
- Recommended program / page erase time : FETCR = AFh

Table 33. Program and Erase Time

	Min	Typ	Max	Unit
Program/erase time	2.4	2.5	2.6	Ms

17.2 Memory map

17.2.1 Flash memory map

Program memory uses 8K bytes of flash memory. It is read by byte and written by byte or page. One page is 32-bytes

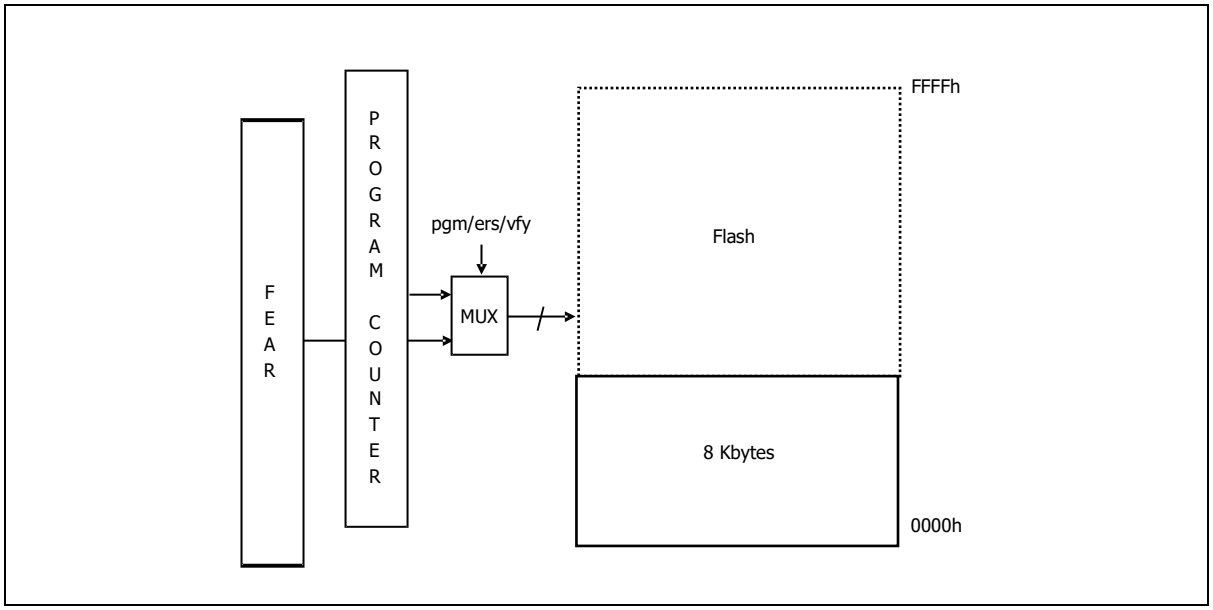


Figure 95. Flash Memory Map

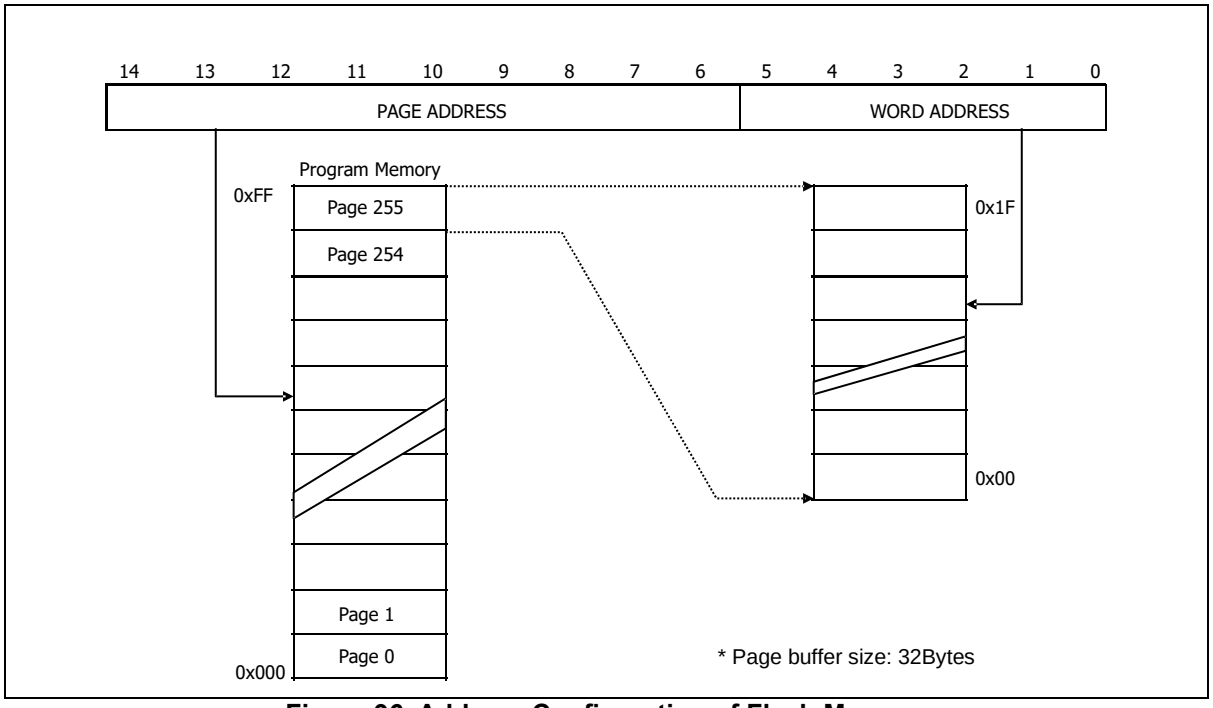


Figure 96. Address Configuration of Flash Memory

17.3 Serial in-system program mode

Serial in-system program uses the interface of debugger which uses two wires. Refer to chapter 18 Development tools in details about debugger.

17.3.1 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

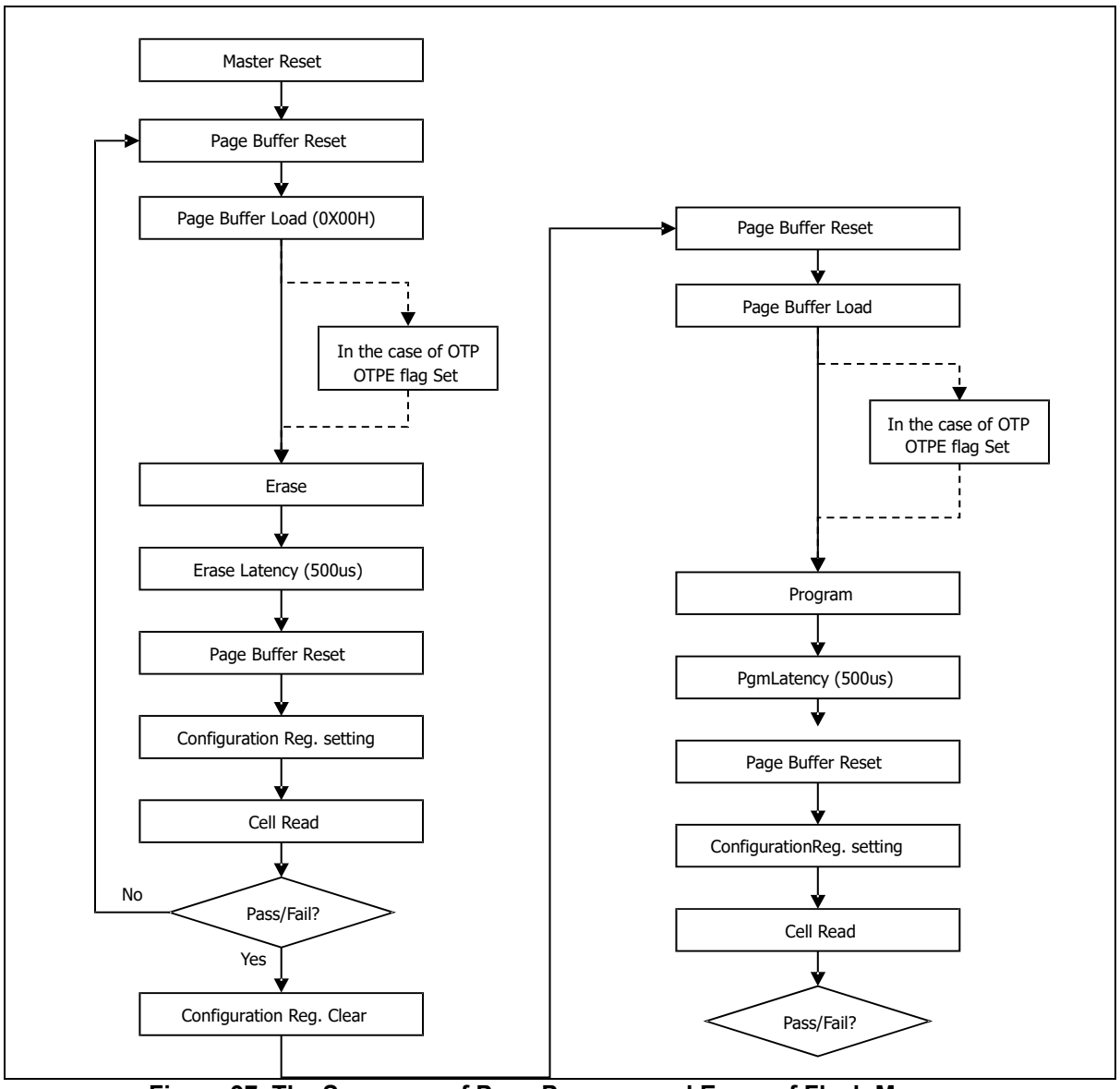


Figure 97. The Sequence of Page Program and Erase of Flash Memory

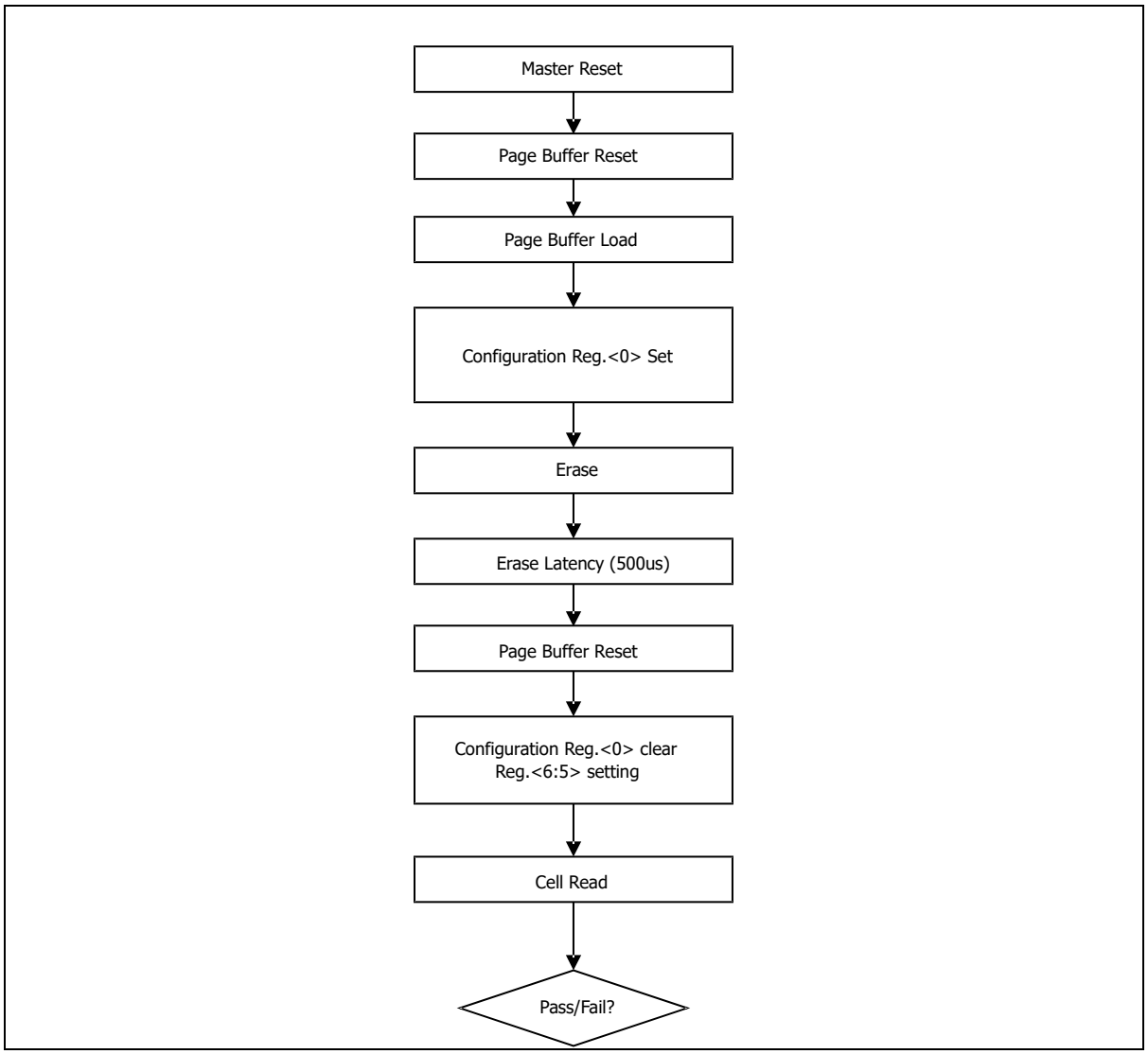


Figure 98. The Sequence of Bulk Erase of Flash Memory

Flash read

① Enter OCD (=ISP) mode.

② Set ENBDM bit of BCR.

③ Enable debug and Request debug mode.

④ Read data from Flash.

Enable program mode

① Enter OCD(=ISP) mode.[NOTE1]

② Set ENBDM bit of BCR.

③ Enable debug and Request debug mode.

④ Enter program/erase mode sequence.[NOTE2]

A. Write 0xAA to 0xF555.

B. Write 0x55 to 0xFAAA.

C. Write 0xA5 to 0xF555.

NOTES:
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.

Flash write mode

① Enable program mode.

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

③ Select page buffer. FEMR:1000_1001

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

⑤ Set write mode. FEMR:1010_0001

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

⑦ Set FETCR.

⑧ Start program. FECR:0000_1011

⑨ Insert one NOP operation

⑩ Read FESR until PEVBSY is 1.

⑪ Repeat ② to ⑧ until all pages are written.

Flash page erase mode

① Enable program mode.

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

③ Select page buffer. FEMR:1000_1001

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

⑤ Set erase mode. FEMR:1001_0001

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

⑦ Set FETCR.

⑧ Start erase. FECR:0000_1011

⑨ Insert one NOP operation

⑩ Read FESR until PEVBSY is 1.

⑪ Repeat ② to ⑧ until all pages are erased

Flash bulk erase mode

① Enable program mode.

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

③ Select page buffer. FEMR:1000_1001

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

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).

⑥ Set FETCR

⑦ Start bulk erase. FECR:1000_1011

⑧ Insert one NOP operation

⑨ Read FESR until PEVBSY is 1.

Flash OTP area read mode

① Enter OCD (=ISP) mode.

② Set ENBDM bit of BCR.

③ Enable debug and Request debug mode.

④ Select OTP area. FEMR:1000_0101

⑤ Read data from Flash.

Flash OTP area write mode

① Enable program mode.

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

③ Select page buffer. FEMR:1000_1001

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

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

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

⑦ Set FETCR.

⑧ Start program. FECR:0000_1011

⑨ Insert one NOP operation

⑩ Read FESR until PEVBSY is 1.

Flash OTP area erase mode

① Enable program mode.

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

③ Select page buffer. FEMR:1000_1001

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

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

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

⑦ Set FETCR.

⑧ Start erase. FECR:0000_1011

⑨ Insert one NOP operation

⑩ Read FESR until PEVBSY is 1.

Flash program verify mode

① Enable program mode.

② Set program verify mode. FEMR:1010_0011

③ Read data from Flash.

OTP program verify mode

① Enable program mode.

② Set program verify mode. FEMR:1010_0111

③ Read data from Flash.

Flash erase verify mode

① Enable program mode.

② Set erase verify mode. FEMR:1001_0011

③ Read data from Flash

Flash page buffer read

① Enable program mode.

② Select page buffer. FEMR:1000_1001

③ Read data from Flash.

Summary of flash program/erase mode

Table 34. 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.

17.4 Mode entrance method of ISP mode

17.4.1 Mode entrance method for ISP

Table 35. Mode entrance method for ISP

Target mode	DSDA	DSCL	DSDA
OCD(ISP)	'hC	'hC	'hC

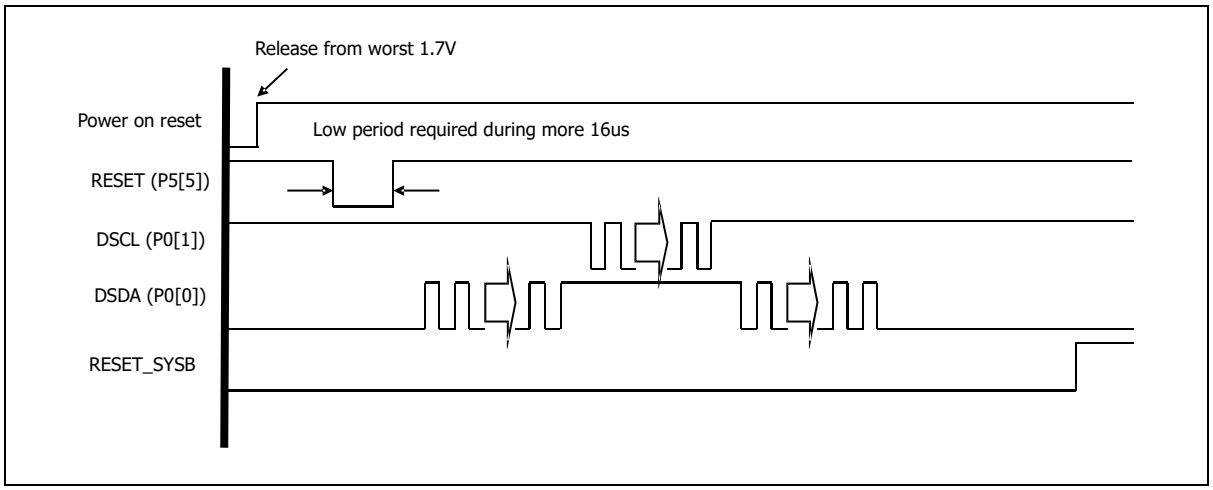


Figure 99. ISP Mode

17.5 Security

A96G174/A96S174 provides Lock bits which can be left un-programmed ("0") or can be programmed ("1") to obtain the additional features listed in Table 36. The Lock bit can only be erased to "0" with the bulk erase command and a value of more than 0x40 at FETCR.

Table 36. 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

NOTES:

1. LOCKF: Lock bit of Flash memory
2. R: Read
3. W: Write
4. PE: Page erase
5. BE: Bulk Erase
6. O: Operation is possible.
7. X: Operation is impossible.

17.6 Configure option

For the configure option control, corresponding data should be written in the configure option area (003EH to 003FH) by programmer (writer tools).

CONFIGURE OPTION 2: ROM Address 0001H

7	6	5	4	3	2	1	0
R_P	HL	–	VAPEN	–	–	–	RSTS

Initial value: 00H

R_P Code Read Protection
- 0 Disable
- 1 Enable

HL Code Write Protection
- 0 Disable
- 1 Enable

VAPEN Vector area (00H~FFH) Protection
- 0 Disable Protection
- 1 Enable Protection

RSTS Select RESETB pin
- 0 Disable RESETB pin(P02-A96G174 / P16-A96S174)
- 1 Enable RESETB pin

NOTE: Code write protection and Vector area protection are disabled at OCD Mode.

CONFIGURE OPTION 1: ROM Address 0000H (A96G174/A96S174 32K Series)

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

Initial value: 00H

PAEN Enable Specific Area Write Protection

0 Disable Protection

1 Enable Protection

PASS [2:0] Select Specific Area for Write Protection

NOTE: When PAEN = '1', it is applied.

PASS2	PASS1	PASS0	Description
0	0	0	Address 0100H – 07FFH
0	0	1	Address 0100H – 0FFFH
0	1	0	Address 0100H – 17FFH
0	1	1	Address 0100H – 1BFFH
1	0	0	Address 0100H – 1DFFH
1	0	1	Address 0100H – 1EFFH
1	1	0	Address 0100H – 1FFFH
1	1	1	Reserved

Note: Specific area write protection are disabled at OCD Mode.

17.6.1 How to write the configure option in user program

To reapply the configure value to be written by programmer (writer tool)

```
// RINGOSC Code Address       : 0x2045
// BGR3 Code Address          : 0x2046
#define CODEROM        ((unsigned char volatile code *) 0)
#define RINGOSC        *(volatile unsigned char xdata *) 0x2F55
#define BGR3           *(volatile unsigned char xdata *) 0x2F56
void main(void)
{
        FESR |= 0x20;                 // Remap Enable
        RINGOSC=CODEROM[0x2045];      // RINGOSC : Code Address 0x2045
        BGR3=CODEROM[0x2046];         // BGR3 : Code Address 0x2046
        FESR &= ~(0x20);              // Remap Disable
        …
}
```

18 Development tools

This chapter introduces a wide range of development tools for microcontrollers. ABOV offers software tools, debuggers, and programmers to help a user in generating right results to match target applications. ABOV supports entire developer ecosystem of the customers.

18.1 Compiler

ABOV semiconductor does not provide any compiler for the A96G174/A96S174. Regarding the compilers, it is recommended to consult with your compiler provider.

Since A96G174/A96S174 has the Mentor 8051 as a core, and ROM is smaller than 64Kbytes in size, a developer can use any standard 8051 compiler from other providers.

18.2 Core and debug tool information

ABOV′s microcontroller uses OCD (On-Chip Debugger) interface for debugging, which is ABOV′s own interface. The OCD not only monitors and controls the core, but also supports the read and write operations of external memory and devices. In addition, it supports memory monitoring and break functions.

Debug interfaces such as OCD interfaces enable microcontrollers to write to internal programmable memory, allowing them to support ISP (In-System Program) that makes possible to write as a single chip or as an embedded chip in the system. Table 37 provides information of the core and debug emulation interface.

Table 37. Information of Core and Debug Emulation Interfaces

	Description	Remark
Device Name	A9xXxxx	
Series	94/ 95/ 96/ 97 series	
Core	M8051/ CM8051	
Extended Stack Pointer	Yes/ no	94, 97 series only
Debug Interface	OCD 1/ OCD 2	
Number of Break Point	4/ 8	
Real-time Monitoring	Yes/ no	OCD 2 only
Run Flag Port	Yes/ no	OCD 2 option

NOTES:

1. The A96G174/A96S174 has the 96 series core and OCD 1 interface.
2. The A96G174/A96S174 can be operated with OCD II dongle too, because the OCD II dongle includes all of OCD1 functions.
3. The 95 series core is the old version of 96 series core.

18.2.1 Feature of 94/96/97 series core

ABOV's 8-bit microcontroller contains the M8051/CM8051 core that is an improved version of the 8051. The M8051/CM8051 core is compatible with the 8051, and reduces time of operation cycles. It makes development easier by providing the OCD debug function.

ABOV's 8-bit microcontroller has a core of the 94-series, 96-series, or 97-series that is basically compatible with the 8051 series at the instruction set level. The cores in each series use different Debug Interfaces, as shown in Table 38.

Table 38. Core and Debug Interface by Series

	Core	Debug Interface
96 Series	M8051	OCD 1
97 Series	M8051	OCD 2
94 Series	CM8051	OCD 2

Features of each series are compared in Table 39.

Table 39. Feature Comparison Chart By Series and Cores

	96 Series	**97 Series**	**94 Series**
CPU Core	M8051	M8051	CM8051
Cycle Compatible with MCS51	1/6	1/6	No
OCD Function	OCD 1	OCD 2	OCD 2
Program BUS	8-bit		
Data Bus	8-bit IRAM/ XRAM separated		8-bit single SRAM
EA Auto Clear NOTE1	Yes	Yes	Yes
EA=0, Idle/ Stope Mode Wake up	Yes	Yes	Yes
Interrupt Priority NOTE2	6 group x 4 level	Interrupt x 4 level	Interrupt x 2 level
Nested Interrupt Priority	4 level	4 level	Interrupt x 2 level (max. 4 times)
SFR BUS (read/ write)	Two ports	Two ports	Single port
Stack Extension	X	O	O
Register	SRAM		
Register Bank	4		
CPU/ Flash Clock Ratio	x 1		

Table 39. Feature Comparison Chart By Series and Cores (continued)

	96 Series	97 Series	94 Series
Pipeline	No	No	2-stage (IF + ID/ EX)
DHRY Stone Score (I8051: 1.00)	6.0	6.0	8.4
Average Instruction Set Exe. Cycle Compare with i8051	x 6.0	x 6.0	x 6.4
Power Consumption/ DHRY (@synthesis)	52.27uA/ MHz	52.27uA/ MHz	30.19 uA/ MHz

NOTE:

1. EA means that All Interrupt Enable bit or Disable bit (Standard 8051).
2. Group: When a programmer selects a specific interrupt (e.g. Interrupt1),
   Whole interrupts: 0, 6, 12, and 18 have higher priorities.
3. The A96G174/A96S174 has the 96 series core and OCD 1 interface.
4. The A96G174/A96S174 can be operated with the OCD II dongle too, because the OCD II dongle includes all functions of the OCD1.

ABOV′s 8-bit microcontroller maintains binary compatibility with 8051 cores; however, the cores and series have differences in performances, core functionalities, and debug interfaces.

You can see the differences between each series in the following sections.

18.2.2 OCD type of 94/96/97 series core

Cores of the 96-series use the OCD 1 for debug interfaces, while cores of the 94-series and 97-series use the OCD 2 for debug interfaces. The OCD 1 and OCD 2 use the same method on the Hardware, however, the protocols are incompatible with each other.

In the OCD 2, it is able to measure the emulation time through the "Run Flag" pin.

Table 40. OCD Type of Each Series

	96-Series	97-Series	94-Series	Remark
OCD type	OCD 1	OCD 2	OCD 2	

In Table 41, debug interfaces of the OCD 1 and OCD 2 are compared.

Table 41. Comparison of OCD 1 and OCD 2

	Value	Description
OCD 1	Break point MAX.8	PC break only
OCD 2	Break point MAX.12	With RAM break • Code, XDATA, IDATA • 1/8/16/32bit compare
	Real-time monitoring	Code, XDATA, IDATA
	Frequency output	Examine CPU frequency
	Run Flag port	Option for run time measurement

96 Series – OCD 1

The 96 series supports basic operation of debug interfaces such as Run, Stop, Step, Break point, register reading/ writing, Memory reading/ writing, and SFR reading/ writing..

94 Series and 97 Series – OCD 2

The 94 series and 97 series support the features listed below, as well as the features of the OCD 1 (however, their protocol is incompatible with the OCD1):

- RTM support: CODE, XDATA, IDATA are updated during the Run Time (Real Time Monitoring available).
- Run Flag support: Emulation Time can be measured in OCD mode (using the Run Flag port).
- RAM Break support: IDATA, SFR, and XDATA break are added.

18.2.3 Interrupt priority of 94/96/97 series core

In the M8051, users can set interrupt priorities by group. The 96-series microcontroller with the basic M8051 core only supports interrupt priorities in group units. In the 94-series or 97-series microcontroller, users set interrupt priorities to have more functionalities than existing features, and can set individual priority for each interrupt source.

Table 42. Interrupt Priorities in Groups and Levels

Series	96-Series	97-Series	94-Series	Remark
Interrupt Priority	6 Grouped 4 Level	Fully 4 Level	Fully 4 Level	**96 Series:** IP/IP (Interrupt Priority Register) **94, 97 Series:** IPxL/IPxH (Interrupt Priority Register)

96 Series

- The priority by group is available only with IP/IP1 settings.
  - With the IP/IP1 settings, users can set the interrupt priorities in group units.
  - The interrupt priority in group units (4 interrupts in a group) can be changed to the level between 0 and 3 according to the value of the IP/IP1.

94, 97 Series

- The individual interrupt priority can be set by setting IPxL/IPxH (x = 0 to x = 3).
- The individual interrupt priority can be changed to the level between 0 and 3 according to value of the IPxL/IPxH (x = 0 to x = 3).

18.2.4 Extended stack pointer of 94/96/97 series core

The M8051 uses IRAM area for Stack Pointer. However, 94-series and 97-series microcontrollers use both IRAM area and XRAM area for the Stack Pointer by configuring additional registers.

The XSP and XSPCR registers are involved in this functionality as described below:

- By configuring the XSP/XSPCR register, you can use the XRAM area for the Stack Pointer.
  - The XSPCR decides whether to use XRAM for the Stack Pointer.
    - If XSPCR = '0', the IRAM is available for the Stack Pointer.
    - If XSPCR = '1', the XRAM is available for the Stack Pointer.
  - The XSP decides a position of XRAM Stack Pointer.
    - This is valid only if XSPCR='1'.

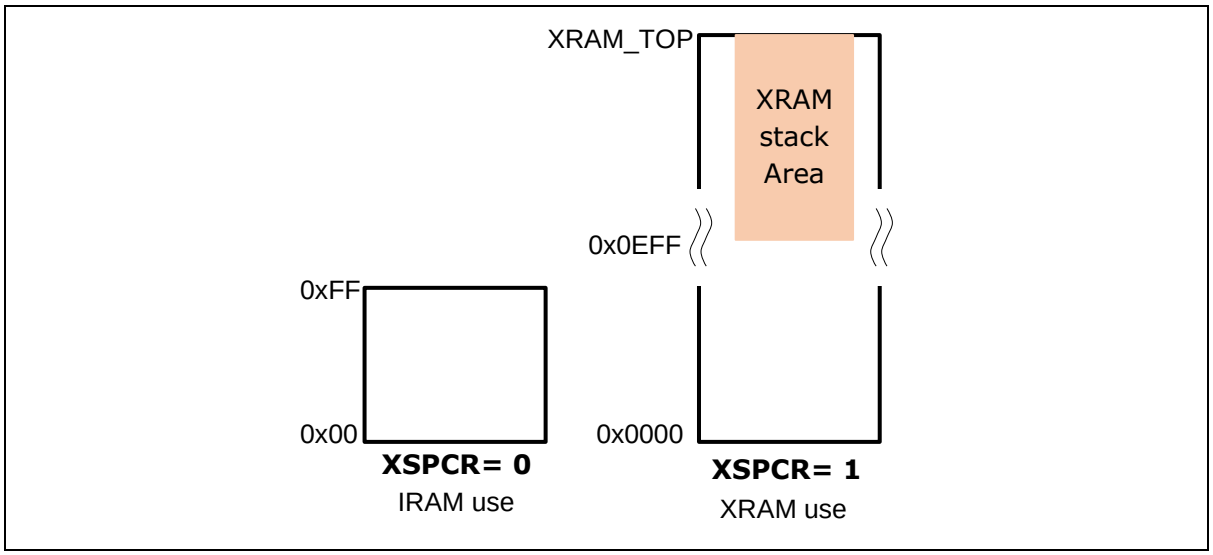


Figure 100. Configuration of the Extended Stack Pointer

STACK_POINTER = {XSP[7:0],SP[7:0]} = XRAM_TOP – STACK_SIZE

Ex) If only 256bytes of XRAM will be used for stack,

- XRAM_TOP = 4K(0x0FFF)
- STACK_SIZE = 256byte(0x0100)
- XSPCR= 1, XSP= 0x0E
- SP=0xFF setting
- Stack Pointer Position = 0x0FFF - 0x0100= 0x0EFF

18.3 OCD (On-chip debugger) emulator and debugger

Microcontrollers with 8051 cores have an OCD (On-Chip Debugger), a debug emulation block. The OCD is connected to a target microcontroller using two lines such as DSCL and DSDA. The DSCL is used for clock signal and the DSDA is used for bi-directional data.

The two lines work for the core management and control by doing register management and execution, step execution, break point and watch functions. In addition, they control and monitor the internal memory and the device by reading and writing.

Table 43. Debug Feature by Series

Series name	96-series	97-series	94-series
OCD function	OCD 1	OCD 2	OCD 2
Max. number of breakpoints	8	8	4
Saving stack in XRAM	No	Yes	Yes
Real time monitoring	No	Yes	Yes
Run flag support	No	Yes	Yes

The OCD 2 applied to the 94-series and 97-series provides the RTM (Real Time Monitoring) function that monitors internal memory and I/O status without stopping the debugging. In addition, the OCD 2 provides the breakpoint function (RAM Break Function) for the IDATA, SFR, and XDATA.

The following functions have been extended from the OCD 2:

- Emulation Time can be measured in OCD mode (using the Run Flag port)
- CODE, XDATA, and IDATA are updated during the Run Time (Real Time Monitoring available).
- IDATA, SFR, and XDATA break are added (RAM Break support).

Figure 101 shows the standard 10-pin connector of OCD 1 and OCD 2.

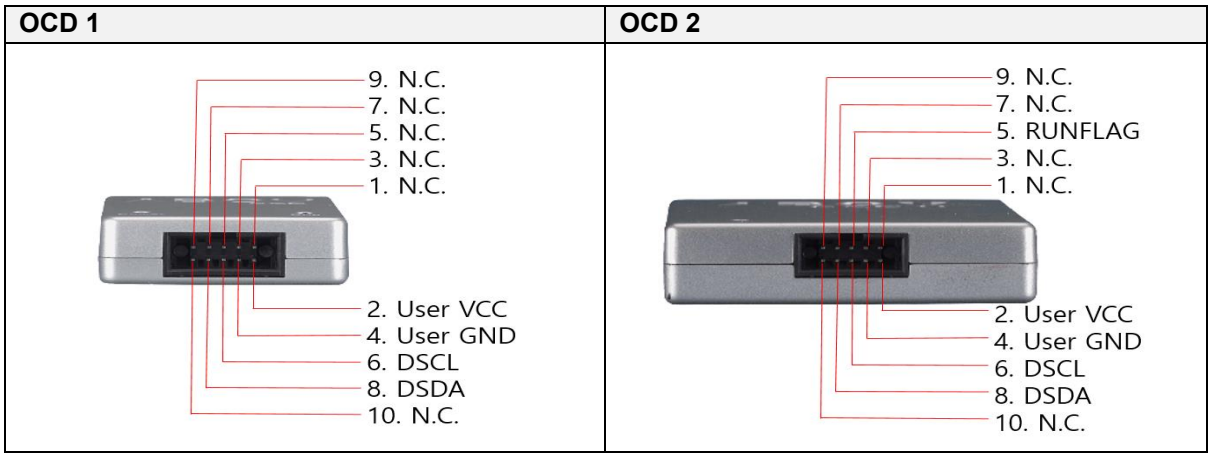


Figure 101. OCD 1 and OCD 2 Connector Pin Diagram

Table 44 introduces pins used for the OCD 1 and OCD 2.

Table 44. OCD 1 and OCD 2 Pin Description

Pin name	Microcontroller function in Debug Mode	
	I/O	Description
DSCL	I	Serial clock pin. Input only pin.
DSDA	I/O	• Serial data pin. • Output port when reading and input port when programming. • IT can be assigned as input/push-pull output port.
VDD,VSS	__	Logic power supply pin.

The OCD emulator supports ABOV's 8051 series MCU emulation. The OCD uses two wires that are interfaces between PC and MCU, which is attached to user's system. The OCD can read or change the value of MCU's internal memory and I/O peripherals. In addition, the OCD controls MCU's internal debugging logic. This means that the OCD controls emulation, step run, monitoring and many more functions regarding debugging.

The OCD debugger program runs underneath MS operating system such as MS-Windows NT/ 2000/ XP/ Vista (32-bit). If you want to see more details, please visit ABOV's website (www.abovsemi.com), and download debugger S/W and OCD debugger manuals.

- Connection:
  - DSCL (A96G174/A96S174 P01 port)
  - DSDA (A96G174/A96S174 P00 port)

Figure 102 shows pinouts of OCD connector.

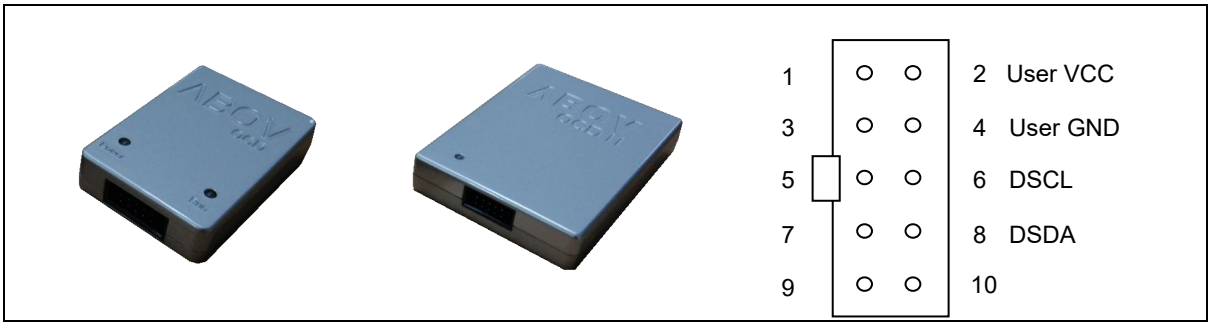


Figure 102. Debugger (OCD1/OCD2) and Pinouts

18.3.1 On-chip debug system

A96G174/A96S174 supports On-chip debug (OCD) system. We recommend developing and debugging program with A96G1xx series. The OCD system of the A96G174/A96S174 can be used for programming the non-volatile memories and on-chip debugging.

In this section, you can find detailed descriptions for programming via the OCD interface. Table 45 introduces features of the OCD.

Table 45. OCD Features

Two wire external interface	• 1 for serial clock input • 1 for bi-directional serial data bus
Debugger accesses	• All internal peripherals • Internal data RAM • Program Counter • Flash memory and data EEPROM memory
Extensive On-Chip Debugging supports for Break Conditions	• Break instruction • Single step break • Program memory break points on single address • Programming of Flash, EEPROM, Fuses, and Lock bits through the two-wire interface • On-Chip Debugging supported by Dr. Choice®
Operating frequency	The maximum frequency of a target MCU.

Figure 103 shows a block diagram of the OCD interface and the On-chip Debug system.

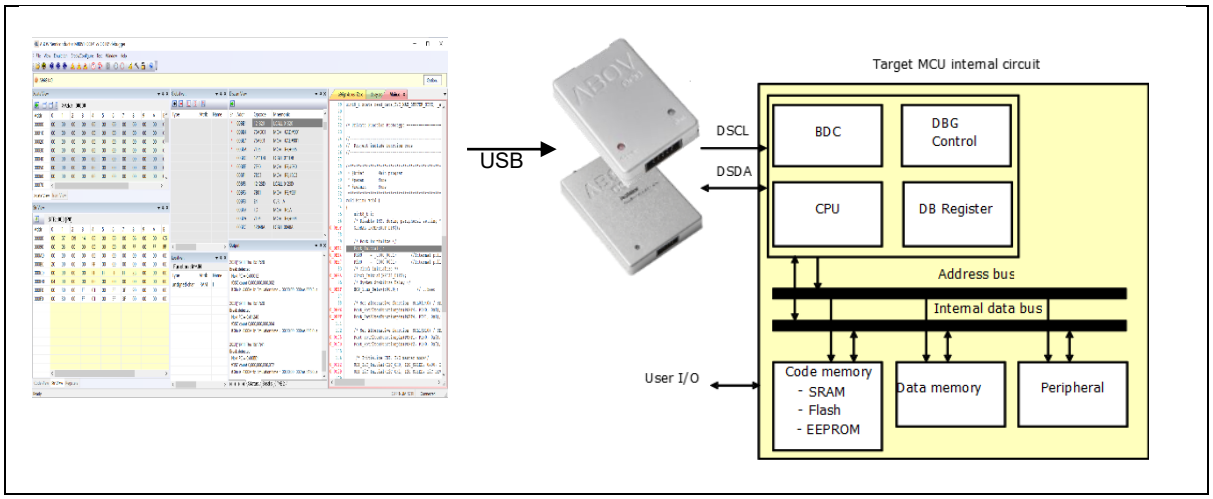


Figure 103. On-Chip Debugging System in Block Diagram

18.3.2 Entering debug mode

While communicating through the OCD, you can enter the microcontroller into DEBUG mode by applying power to it. This means that the microcontroller enters DEBUG mode when you place specific signals to the DSCL and DSDA at the moment of initialization when the microcontroller is powered on. This requires that you can control power of the microcontroller (VCC or VDD) and need to be careful to place capacitive loads such as large capacity condensers on a power pin.

Please remember that the microcontroller can enter DEBUG mode only when power is applied, and it cannot enter DEBUG mode once the OCD is run.

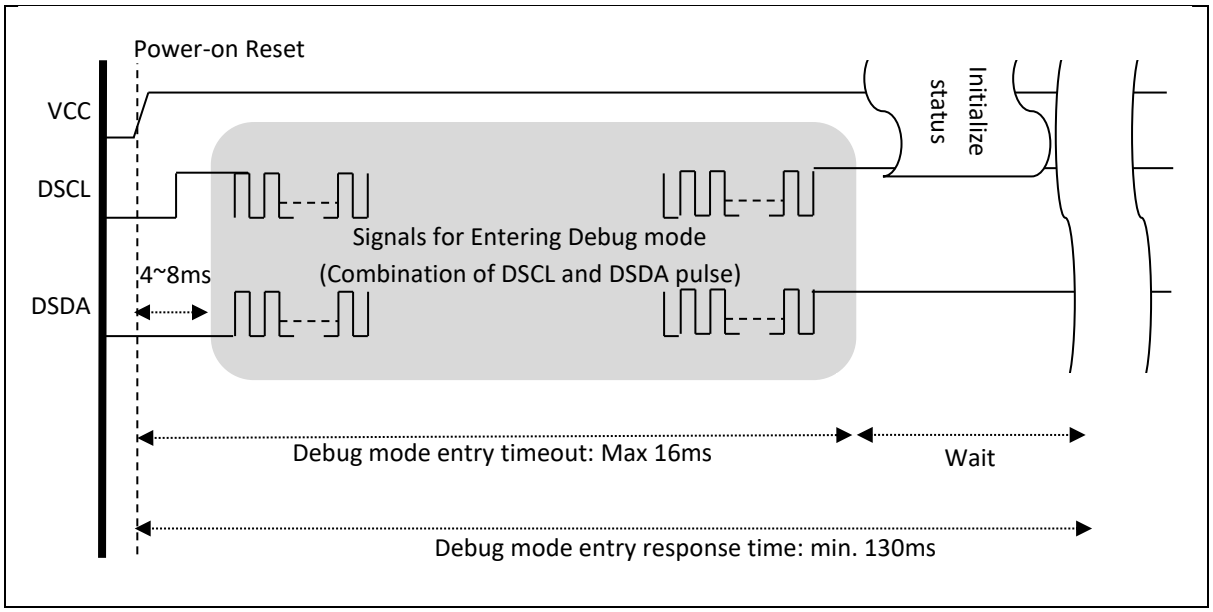


Figure 104. Timing Diagram of Debug Mode Entry

18.3.3 Two-wire communication protocol

For the OCD interface, the semi-duplex communication protocol is used through separate two wires, the DSCL and DSDA. The DSCL is used for serial clock signal and the DSDA is used for bi-directional serial address and data.

A unit packet of transmission data is 10-bit long and consists of a byte of data, 1-bit of parity, and 1-bit of acknowledge. A parity check bit and a receive acknowledge bit are transmitted to guarantee stability of the data communication. A communication packet includes a start bit and an end bit to indicate the start and end of the communication.

More detailed information of this communication protocol is listed below:

Basic transmission packet

- A 10-bit packet transmission using two-pin interface.
- A 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 a command which has error again.
- Background debugger command is composed of a bundle of packet.
- Start and stop conditions notify start and stop of the background debugger command respectively.

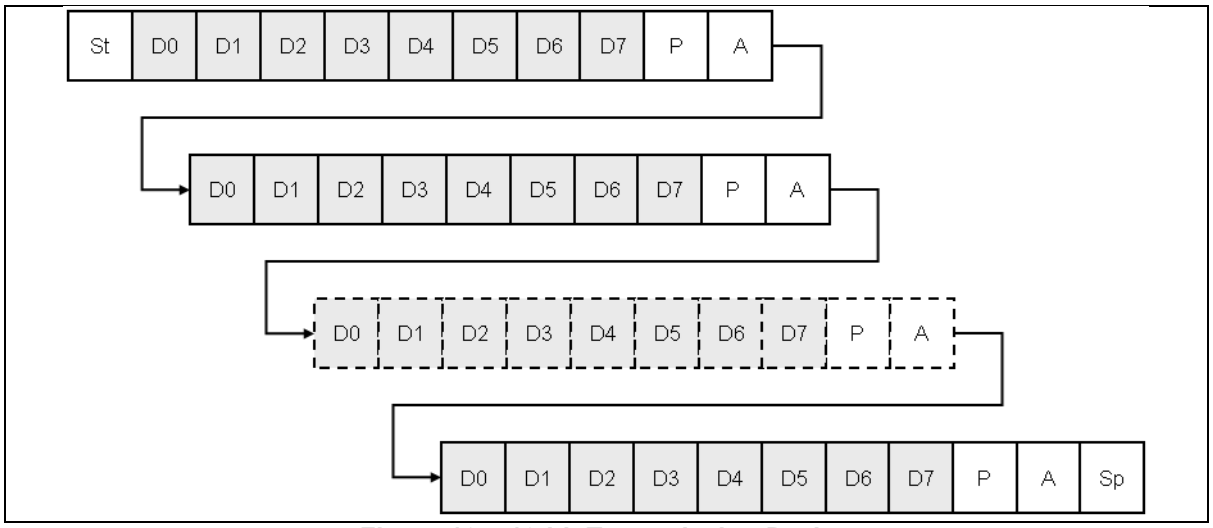


Figure 105. 10-bit Transmission Packet

Packet transmission timing

Figure 106 shows a timing diagram of a packet transmission using the OCD communication protocol.

A start bit in the figure means start of a packet and is valid when the DSDA falls from 'H' to 'L' while External Host maintains the DSCL to 'H'. After the valid start bit, communication data is transferred and received between a Host and a microcontroller.

An end bit means end of the data transmission and is valid when the DSDA changes from 'L' to 'H' while a Debugger maintains the DSCL to 'H'. Next, the microcontroller places the bus in a wait state and processes the received data.

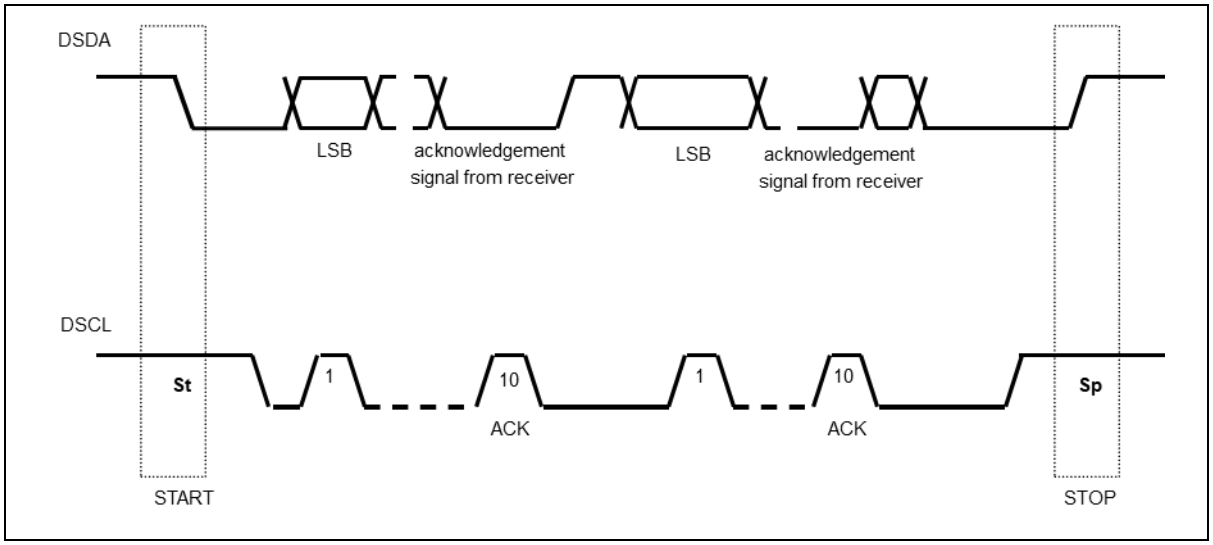


Figure 106. Data Transfer on OCD

Figure 107 shows a timing diagram of each bit based on state of the DSCL clock and the DSDA data. Similar to I2C signal, the DSDA data is allowed to change when the DSCL is 'L'. If the data changes when the DSCL is 'H', the change means 'START' or 'STOP'.

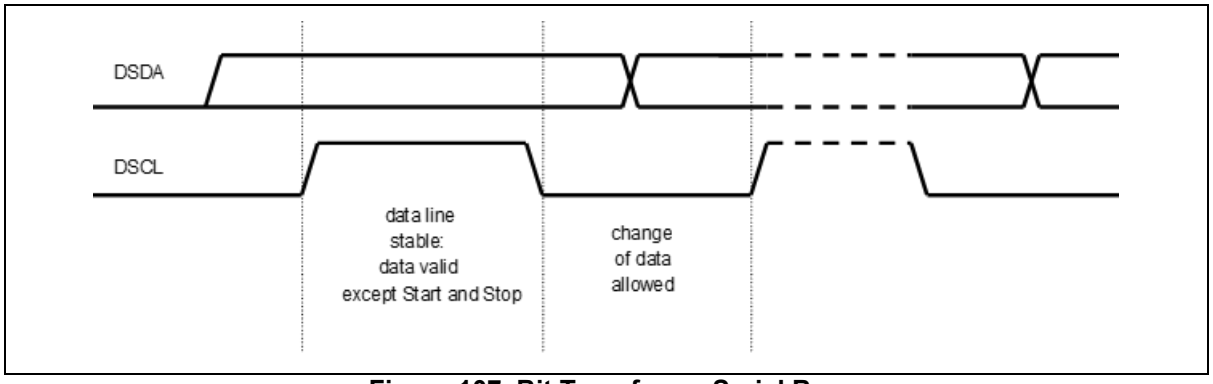


Figure 107. Bit Transfer on Serial Bus

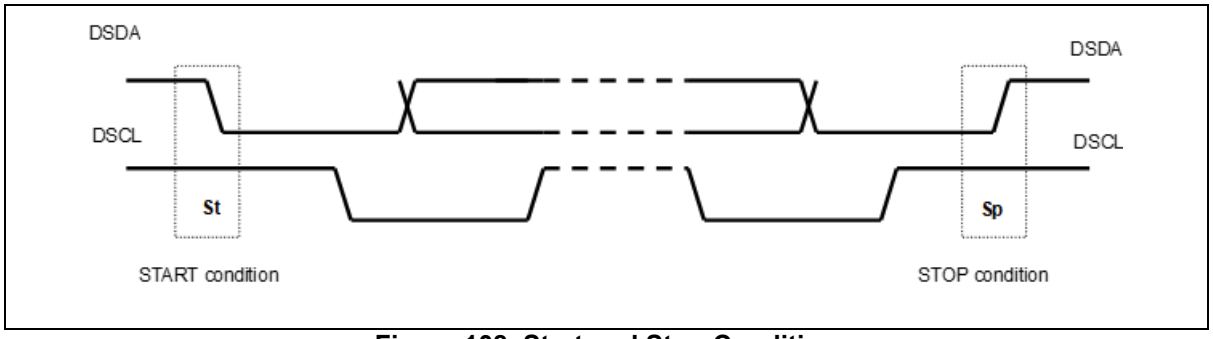


Figure 108. Start and Stop Condition

During the OCD communication, each data byte is transferred in accompany with a parity bit. When data is transferred in succession, a receiver returns the acknowledge bit to inform that it received.

As shown in Figure 109, when transferring data, a receiver outputs the DSDA to 'L' to inform the normal reception of data. If a receiver outputs DSDA to 'H', it means error reception of data.

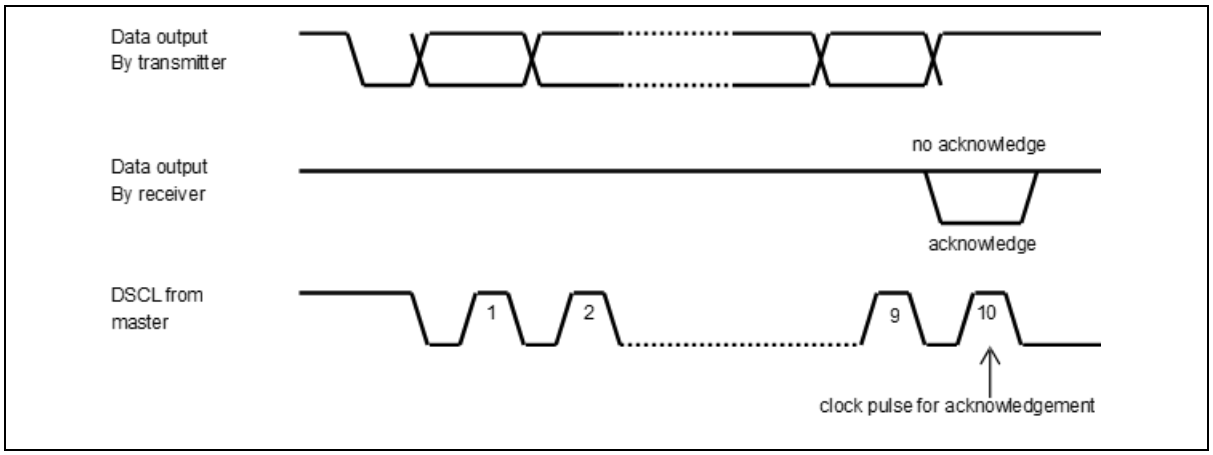


Figure 109. Acknowledge on Serial Bus

While the Host Debugger executes data communications, if a microcontroller needs communication delay or process delay, it can request communication delay to the Host Debugger.

Figure 110 shows timing diagrams where a microcontroller requests communication delay to the Host Debugger. If the microcontroller requests timing delay of the DSCL signal that the Host Debugger outputs, the microcontroller maintains the DSCL signal to 'L' to delay the clock change although the Host Debugger changes the DSCL to 'H'.

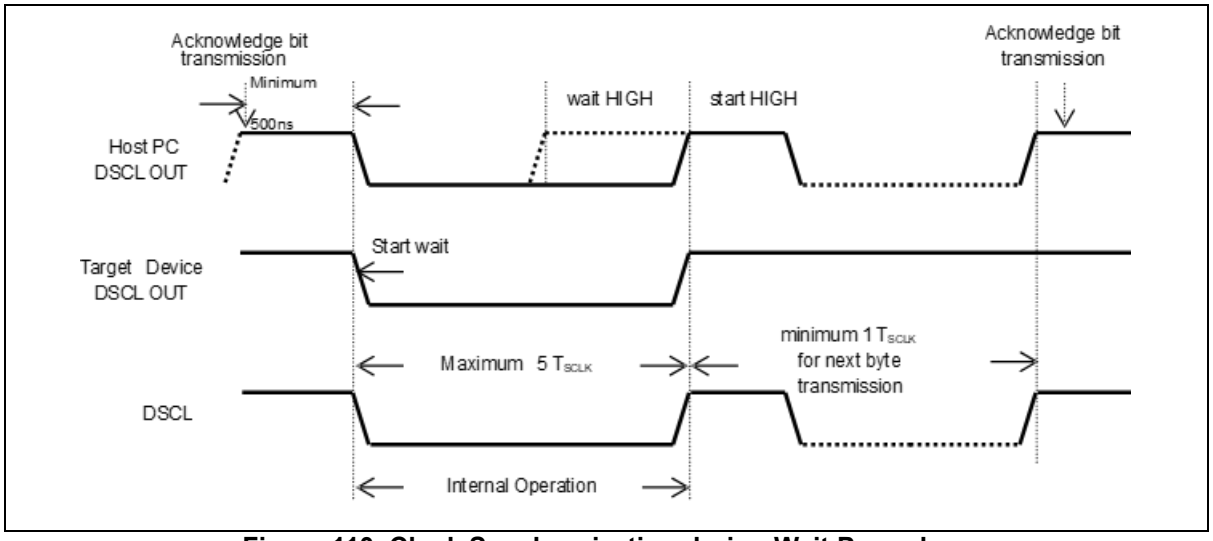


Figure 110. Clock Synchronization during Wait Procedure

18.4 Programmers

18.4.1 E-PGM+

E-PGM+ USB is a single programmer. You can program A96G174/A96S174 directly using the E-PGM+.

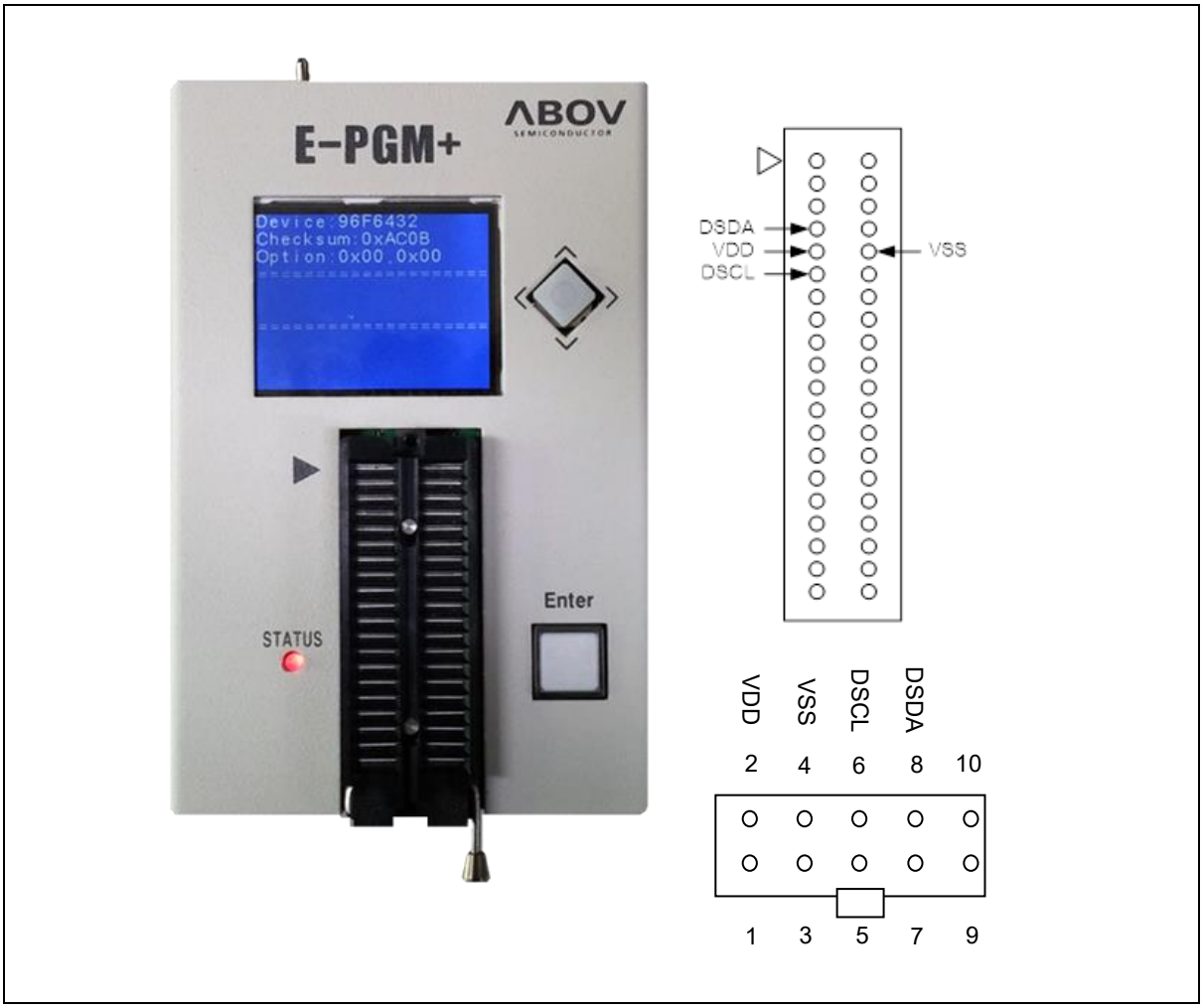


Figure 111. E-PGM+ (Single Writer) and Pinouts

18.4.2 OCD emulator

OCD emulator allows users to write code on the device too, since OCD debugger supports In System Programming (ISP). It doesn't require additional H/W, except developer's target system.

18.4.3 Gang programmer

E-Gang4 and E-Gang6 allow users to program multiple devices simultaneously. They can be run not only in PC controlled mode but also in standalone mode without the PC control.

USB interface is available, and it is easy to connect to the handler.

Figure 112. E-Gang4 and E-Gang6 (for Mass Production)

18.5 Flash programming

Program memory for A96G174/A96S174 is a Flash type. This Flash ROM is accessed through four pins such as DSCL, DSDA, VDD, and VSS in serial data format. For detailed information about the Flash memory programming, please refer to **17. Memory programming.**

Table 46 introduces corresponding pins and I/O status.

Table 46. Pins for Flash Programming

Pin name	Main chip pin name	During programming	
		I/O	Description
DSCL	P01	I	Serial clock pin. Input only pin.
DSDA	P00	I/O	Serial data pin. Output port when reading and input port when programming. Can be assigned as input/push-pull output port.
VDD, VSS	VDD, VSS	—	Logic power supply pin.

18.5.1 On-board programming

Microcontrollers need only four signal lines including VDD and VSS pins, to program the Flash ROM using serial protocol. Therefore, on-board programming is possible if the programming signal lines are considered at the time the PCB of application board is designed.

18.6 Connection of transmission

OCD's two-wire communication interfaces use the Open-Drain Method (Wire-AND Bi-Directional I/O).

Normally, it is recommended to place a resister greater than 4.7kΩ for the DSCL and DSDA respectively. The capacitive load is recommended to be less than 100pF. Outside these ranges, because the communication may not be accomplished, the connection to Debug mode is not guaranteed.

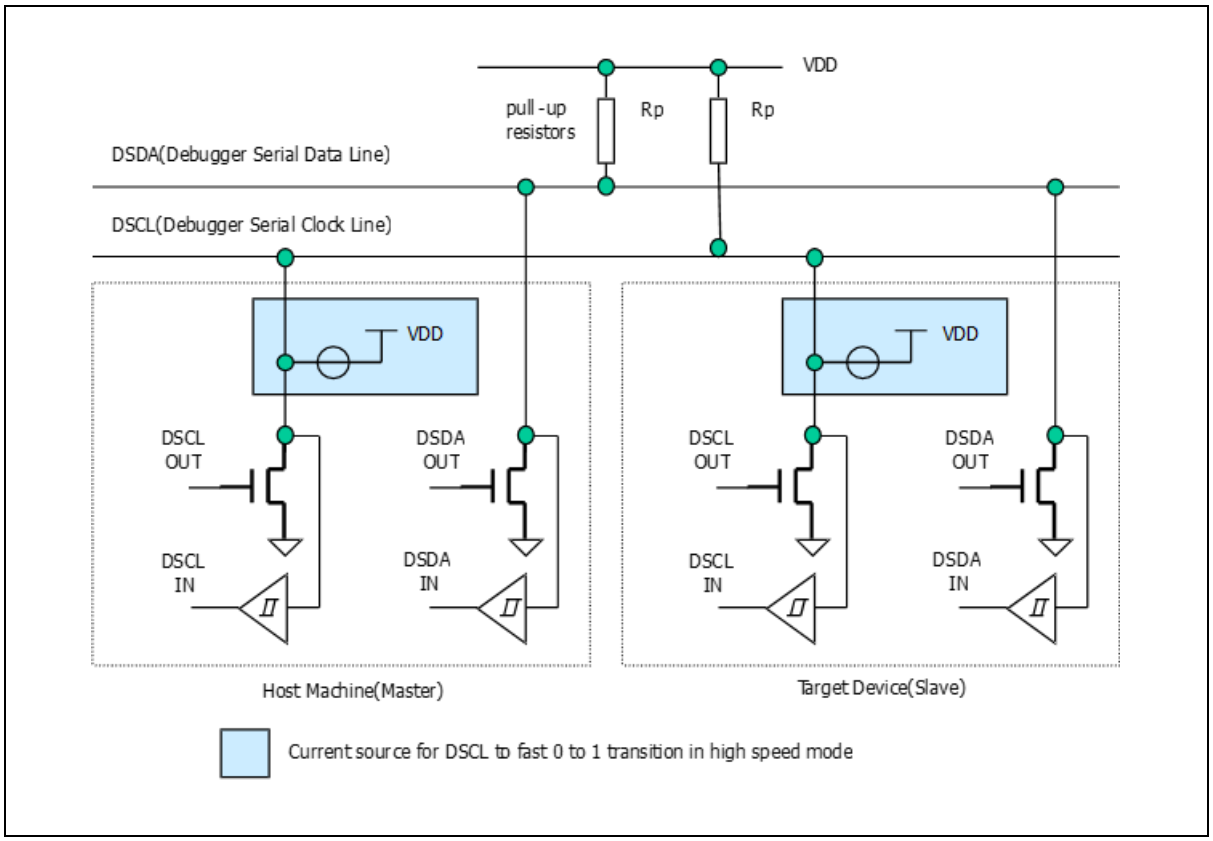


Figure 113. Connection of Transmission

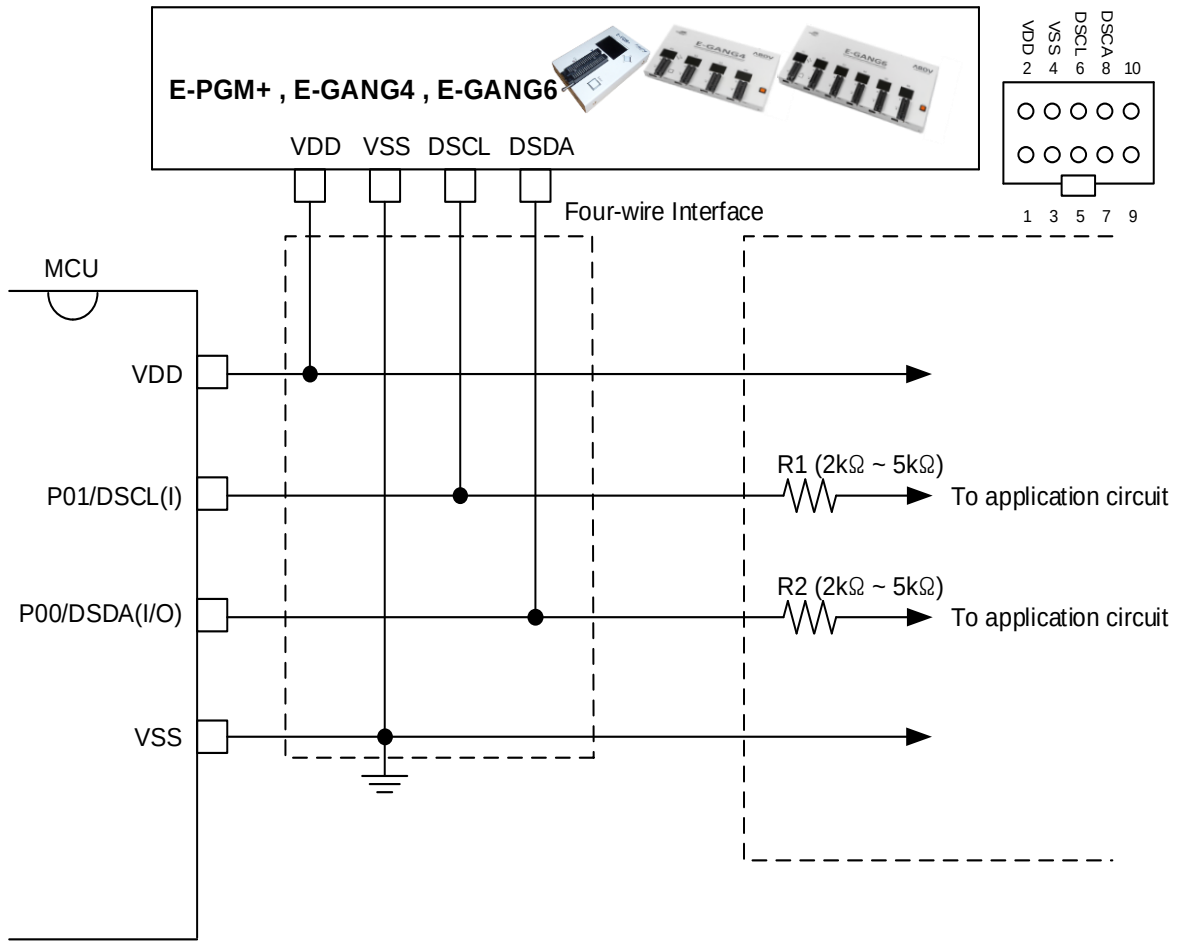
18.7 Circuit design guide

To program Flash memory, programming tools require 4 signal lines, DSCL, DSDA, VDD, and VSS. When designing a PCB circuit, you should consider these 4 signal lines for on-board programming. In addition, you need to be careful when designing the related circuit of these signal pins, because rising/falling timing of the DSCL and DSDA is very important for proper programming.

When you use the OCD pins exclusively or share them with other functions, it needs to be careful, too.

Figure 114 shows an example circuit where the OCD pins (DSCL and DSDA) are shared with other functions. They must be connected when debugging or executing In System Program (ISP).

Normally, the OCD pins are connected to outside to execute the predefined functions. Even when they are connected for debugging or executing ISP directly, the OCD pins are shared with other functions by using resistors as shown in Figure 114. By doing this, the OCD pins process the normal external signals and execute the OCD functions first when they are shared.



NOTES:

1. In on-board programming mode, very high-speed signal will be provided to pin DSCL and DSDA. And it will cause some damages to the application circuits connected to DSCL or DSDA port if the application circuit is designed as high-speed response such as relay control circuit. If possible, the I/O configuration of DSDA, DSCL pins had better be set to input mode.
2. The value of R1 and R2 is recommended value. It varies with circuit of system.

Figure 114. PCB Design Guide for On-Board Programming

Appendix

Instruction table

Instructions are either1, 2 or 3bytes 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.

Table 47. Instruction Table

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

Table 47. Instruction Table (continued)

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

Table 47. Instruction Table (continued)

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

Table 47. Instruction Table (continued)

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

Table 47. Instruction Table (continued)

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, entries such as E8-EF indicate continuous blocks of hex opcodes used for 8 different registers. 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.

CJNE instructions use abbreviation of #d for immediate data; other instructions use #data as an abbreviation.

Revision history

Date	Revision	Description
2019.12.10	1.00	First creation
2020.02.04	1.01	Added the disclaimer and modified the distributor.
2020.04.02	1.02	Revise "Ordering information"
2020.04.06	1.03	Revise "Device Numbering Nomenclature"
2020.05.22	1.04	Revise "Low voltage reset and low voltage indicator characteristics"
2020.06.08	1.05	Deleted AVREF at Figure 56 12-bit ADC Block Diagram. Deleted Figure 58. A/D Power (AVREF) Pin with a Capacitor. Changed the maximum analog input voltage to VDD at Table 39 A/D Converter Characteristics. Updated Basic Timer Block Diagram at Figure 27. Added the description of V_{LVD}/V_{LVI} at LVR and LVI Characteristics. Extended maximum operating temperature up to 105°C as well as 85°C.
2020.07.20	1.06	Enhanced the minimum ADC operation voltage to 2.2V at Table 39 A/D Converter Characteristics. Deleted Analog Reference Voltage item at Table 39 A/D Converter Characteristics. Added the note of "Guaranteed by design" at Table 40. Recommended ADC Resolution. Enhanced the tolerance of High-Speed Internal RC within "-40°C to +85°C" to ±2.5% at Table 44. High Internal RC Oscillator Characteristics. Corrected the conditions of Supply Current at Table 46. DC Characteristics. Updated the table and figures for USART characteristics in 17 Electrical characteristics. Corrected the minimum A/D Conversion time to 7.5us at Table 39. A/D Converter Characteristics.
2020.08.31	1.07	Advanced Flash Endurance times from 10,000 to 30,000.
2020.09.28	1.08	Updated the initial value in Table 5. SFR Map. Added the description of REMAP bit of FESR in 16.1.2 Register description. Added the example of reapplying the configure bytes by user program in 16.6 Configure option. Updated a typo in Table 42. Power-on Reset Characteristics, Figure 82. Fast VDD Rising Time and Figure 83. Internal RESET Release Timing On Power-Up.
2021.04.26	1.09	Changed the figures of Package Outline Drawing in 19 Package information. Updated High Speed Internal RC Oscillator Tolerance at Table 44. High Internal RC Oscillator Characteristics. Corrected the frequency unit from KHz to kHz.
2022.04.01	1.10	Updated the Table 41. BGR Characteristics at Electrical characteristics.
2022.04.27	1.11	Added 4 Central Processing Unit(CPU). Updated 18 Development tools chapter Removed Electrical characteristics, Package information, Ordering information.
2022.10.14	1.12	Revised the font of this document Fixed the typing error of P12 at Figure 3. A96S174 20TSSOP pin assignment.

2023.04.04	1.13	Fixed RESETB Input low width 10us->50us in 16.3 External RESETB description. Fixed Figure 5,6. A96G(S)174 20QFN Pin Assignment.

Korea

Regional Office, Seoul
R&D, Marketing & Sales
8th Fl., 330, Yeongdong-daero,
Gangnam-gu, Seoul,
06177, Korea

Tel: +82-2-2193-2200
Fax: +82-2-508-6903
http://www.abovsemi.com/

HQ, Ochang
R&D, QA, and Test Center
93, Gangni 1-gil, Ochang-eup,
Cheongwon-gun,
Chungcheongbuk-do,
28126, Korea

Tel: +82-43-219-5200
Fax: +82-43-217-3534
http://www.abovsemi.com/

Domestic Sales Manager
Tel: +82-2-2193-2206
Fax: +82-2-508-6903
Email: sales_kr@abov.co.kr

Global Sales Manager
Tel: +82-2-2193-2281
Fax: +82-2-508-6903
Email: sales_gl@abov.co.kr

China Sales Manager
Tel: +86-755-8287-2205
Fax: +86-755-8287-2204
Email: sales_cn@abov.co.kr