A96G150
User's Manual

16 MHz 8-bit A96G150 Microcontroller
64 Kbyte Flash memory, 2Kbyte EEPROM, 12-bit ADC,
6 Timers, USART, USI, High Current Port

Version 1.01

Introduction

This user's manual targets application developers who use A96G150 for their specific needs. It provides complete information of how to use A96G150 device. Standard functions and blocks including corresponding register information of A96G150 are introduced in each chapter, while instruction set is in Appendix.

A96G150 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: 64Kbytes of FLASH, 256bytes of IRAM, 2304bytes of XRAM, 2Kbytes of Data EEPROM, general purpose I/O, basic interval timer, watchdog timer, 8/16-bit timer/counter, 16-bit PPG output, 8-bit PWM output, 16-bit PWM output, watch timer, buzzer driving port, USI, 12-bit A/D converter, on-chip POR, LVR, LVI, on-chip oscillator and clock circuitry.

As a field proven best seller, A96G150 has been sold more than 3 billion units up to now, and introduces rich features such as excellent noise immunity, code optimization, cost effectiveness, and so on.

Reference document

- A96G150 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

A96G150 is an advanced CMOS 8-bit microcontroller with 64Kbytes 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 A96G150 and peripheral counts are introduced.

Table 1. A96G150 Device Features and Peripheral Counts

Peripherals		Description
Core	CPU	8-bit CISC core (M8051, 2 clocks per cycle)
	Interrupt	Up to 23 peripheral interrupts supported. • EINT40 to 47, EINT0, EINT1, EINT2, EINT3 (5) • Timer (0/1/2/3/4/5) (6) • WDT (1) • BIT (1) • WT (1) • USART (Rx, CRC)/Tx (2) • USI 2-ch. *Rx/Tx/I2C (6) • ADC (1) • LVI (1)
Memory	ROM (FLASH) capacity	• 64 KB FLASH with self-read and write capability • In-system programming (ISP) • Endurance: 30,000 times
	IRAM	256 bytes
	XRAM	2304 bytes
	EEPROM	• 2 KB • Endurance: 300,000 times at room temperature • Retention: 10 years
Programmable pulse generation		• Pulse generation (by T1/T2/T3/T4/T5) • 8-bit PWM (by T0)
Buzzer		8-bit × 1-ch
Minimum instruction execution time		• 125 ns (@ 16 MHz main clock) • 61 us (@ 32.768 kHz sub clock)
Power down mode		• STOP mode • IDLE mode

Table 1. A96G150 Device Features and Peripheral Counts (continued)

Peripherals		Description
General Purpose I/O (GPIO)		• Normal I/O: 42 ports • High sink current port: 8 ports P3[7:0]
Reset	Power on reset	Reset release level: 1.2 V
	Low voltage reset	• 16 levels detect • 1.61 / 1.68 / 1.77 / 1.88 / 2.00 / 2.13 / 2.28 / 2.46 / 2.68 / 2.81 / 3.06 / 3.21 / 3.56 / 3.73 / 3.91 / 4.25V
Low voltage indicator		• 13 levels detect • 1.88 / 2.00 / 2.13 / 2.28 / 2.46 / 2.68 / 2.81 / 3.06 / 3.21 / 3.56 / 3.73 / 3.91 / 4.25V
Watch Timer (WT)		3.91 ms / 0.25 s / 0.5 s / 1 s / 1 min interval at 32.768 kHz
Timer/counter		• Basic interval timer (BIT) 8-bit x 1-ch. • Watchdog timer (WDT) 8-bit x 1-ch. • 8-bit x 1-ch (T0), 16-bit x 5-ch (T1 / T2 / T3 / T4 / T5)
Communication function	USART2	• 8-bit USART x 1-ch or 8-bit SPI x 1-ch • Receiver timer out (RTO) • 0% error baud rate
	USI0/1	• USART + SPI + I2C • 8-bit USART x 2-ch or 8-bit SPI x 2-ch or I2C x 2-ch
12-bit A/D converter		15 input channels
Oscillator type		• 4 MHz to 12 MHz crystal or ceramic for main clock • 32.768 kHz Crystal for sub clock
LCD Driver		• 24 segments and 8 common terminals • Internal or external resistor bias • 4 Internal Resistors Selectable • 1/3, 1/4, 1/5, 1/6 and 1/8 duty selectable • Resistor Bias and 16-step contrast control
CRC		• CRC16 • Polynomial representations Normal : 0x8C81 $f_{(x)} = 1 + x^7 + x^{10} + x^{11} + x^{15} + x^{16}$

Table 1. A96G150 Device Features and Peripheral Counts (continued)

Peripherals	Description
Internal RC oscillator	• HSI 32MHz ±2.0% (T_A=-40~ +85°C) • HSI 32MHz ±3.0% (T_A=-40~ +105°C) • LSI 128kHz ±20% (T_A= -40~ +85°C) • LSI 128kHz ±30% (T_A= -40~ +105°C)
Operating voltage and frequency	• LVR (<1.8V) to 5.5V @ 32.768KHz with crystal • 2.4V to 5.5V @ 4MHz to 12MHz with crystal • LVR (<1.8V) to 5.5V @ 0.5MHz to 16MHz with internal RC
Operating temperature	-40℃ to +85℃, -40℃ to +105℃
Package	• Pb-free packages • 44 LQFP 1010

1.2 A96G150 block diagram

In this section, A96G150 device with peripherals are described in a block diagram.

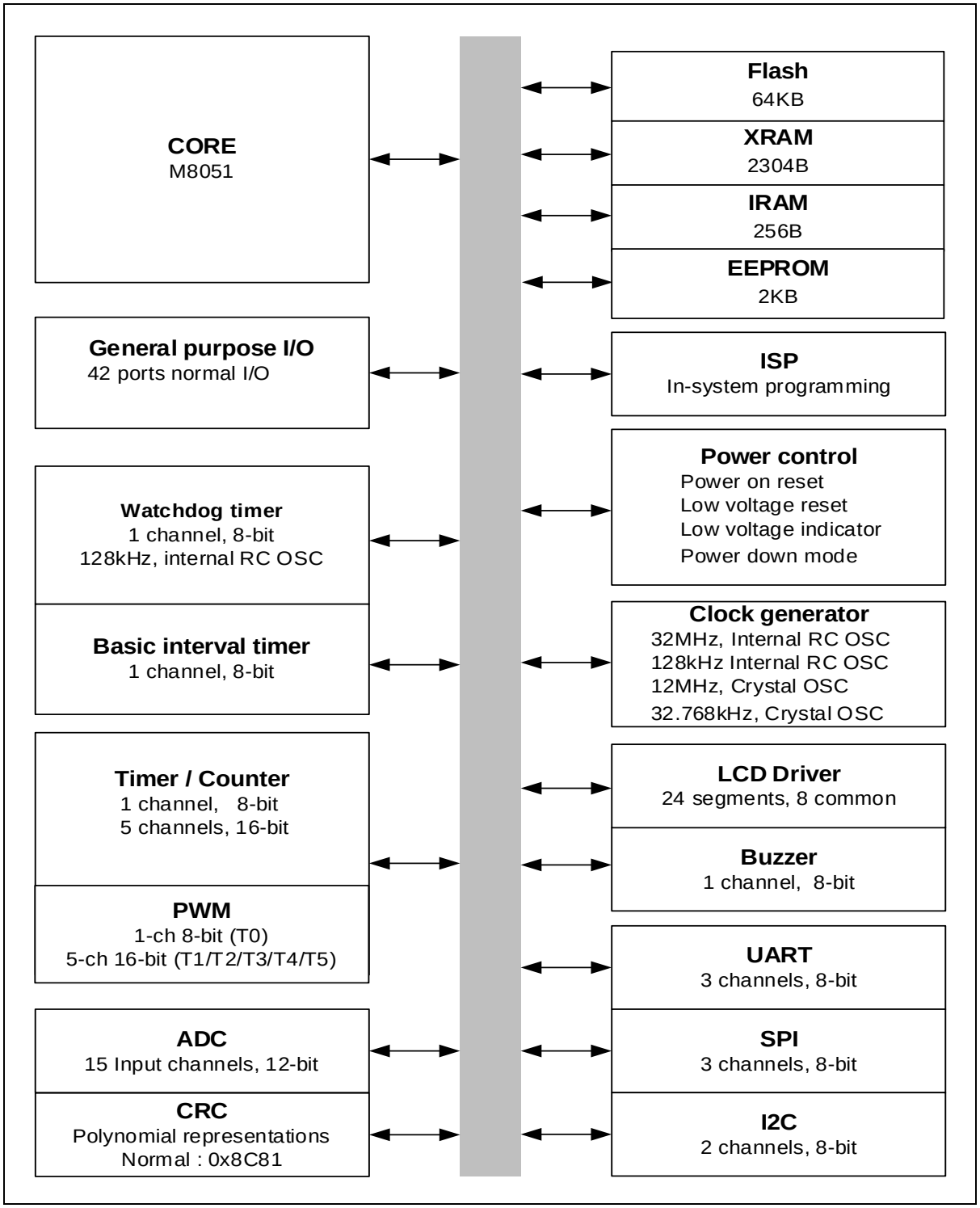
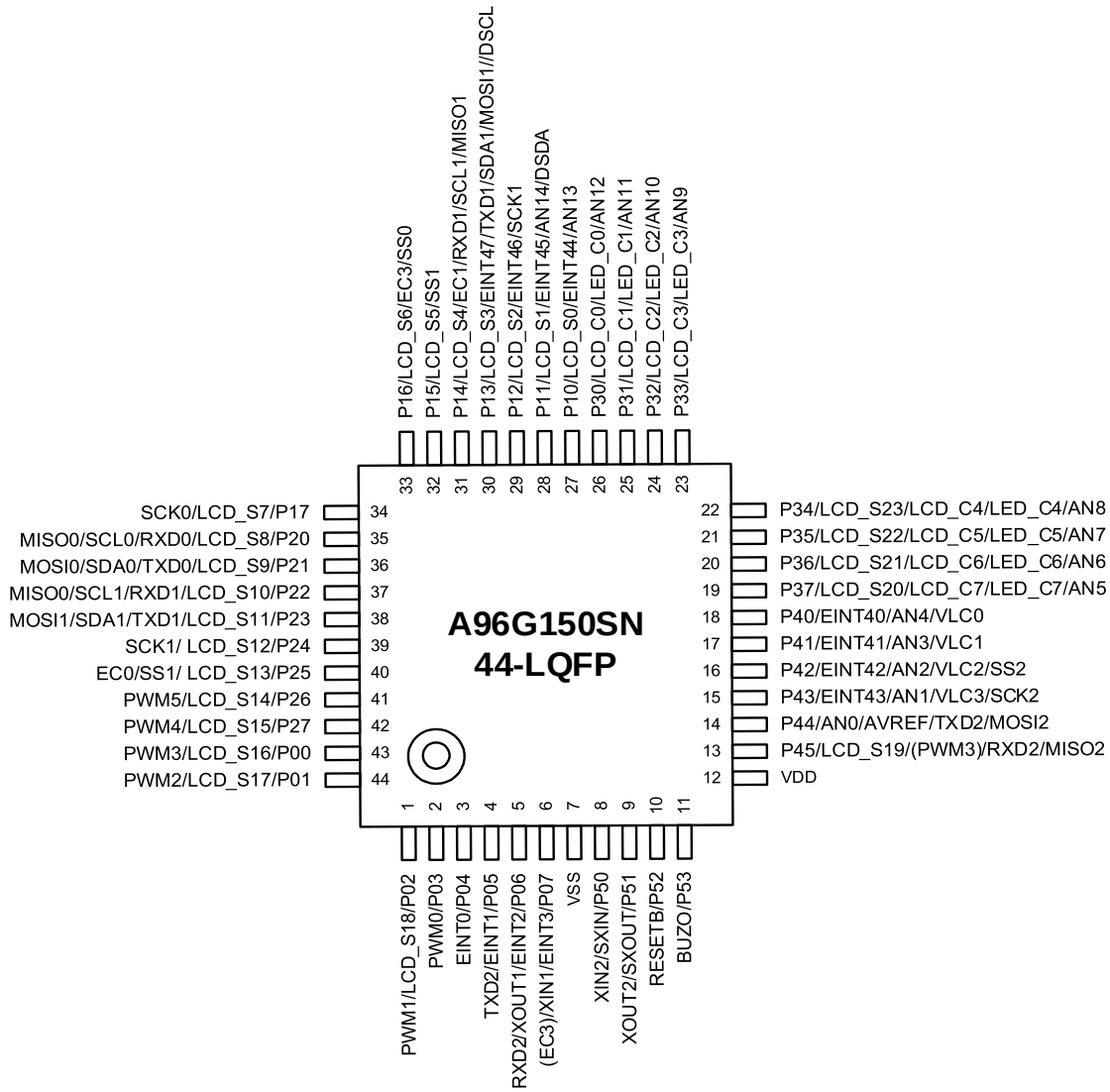


Figure 1. A96G150 Block Diagram

2 Pinouts and pin description

In this chapter, A96G150 device pinouts and pin descriptions are introduced.

2.1 Pinouts



NOTE: The programmer (E-PGM+, E-Gang4/6) uses P1[3], P1[1] pin as DSCL, DSDA.

Figure 2. A96G150 44LQFP-1010 Pin Assignment

2.2 Pin description

Table 2. Normal Pin Description

Pin no.	Pin name	I/O(1)	Description	Remark
44				
43	P00*	IOUS	Port 0 bit 0 Input/output	
	LCD_S16	O	LCD Segment Signal 16 Output	
	T3O	O	Timer 3 interval output	
	PWM3O	O	Timer 3 PWM output	
44	P01*	IOUS	Port 0 bit 1 Input/output	
	LCD_S17	O	LCD Segment Signal 17 Output	
	T2O	O	Timer 2 interval output	
	PWM2O	O	Timer 2 PWM output	
1	P02*	IOUS	Port 0 bit 2 Input/output	
	LCD_S18	O	LCD Segment Signal 18 Output	
	T1O	O	Timer 1 interval output	
	PWM1O	O	Timer 1 PWM output	
2	P03*	IOUS	Port 0 bit 3 Input/output	
	T0O	O	Timer 0 interval output	
	PWM0O	O	Timer 0 PWM output	
3	P04*	IOUS	Port 0 bit 4 Input/output	
	EINT0	I	External interrupt input ch-0	
4	P05*	IOUS	Port 0 bit 5 Input/output	
	EINT1	I	External interrupt input ch-1	
	TXD2	O	USART2 data transmit	
5	P06*	IOUS	Port 0 bit 6 Input/output	
	EINT2	I	External interrupt input ch-2	
	XOUT1	O	Main Oscillator Output ch-1	
	RXD2	I	USART2 data receive	
6	P07*	IOUS	Port 0 bit 7 Input/output	
	EINT3	I	External interrupt input ch-3	
	XIN1	I	Main Oscillator Input ch-1	
	EC3	I	Timer 3(Event Capture) input	

Table 2. Normal Pin Description (continued)

Pin no.	Pin name	I/O(1)	Description	Remark
44				
27	P10*	IOUS	Port 1 bit 0 Input/output	
	LCD_S0	O	LCD Segment Signal 0 Output	
	EINT44	I	External interrupt input ch-44	
	AN13	IA	ADC input ch-13	
28	P11*	IOUS	Port 1 bit 1 Input/output	
	LCD_S1	O	LCD Segment Signal 1 Output	
	EINT45	I	External interrupt input ch-45	
	AN14	IA	ADC input ch-14	
	DSDA	IOU	OCD debugger data input/output	Pull-up
29	P12*	IOUS	Port 1 bit 2 Input/output	
	LCD_S2	O	LCD Segment Signal 2 Output	
	EINT46	I	External interrupt input ch-46	
	SCK1	IO	USART1 clock signal	
30	P13*	IOUS	Port 1 bit 3 Input/output	
	LCD_S3	O	LCD Segment Signal 3 Output	
	EINT47	I	External interrupt input ch-47	
	TXD1	O	USART1 data transmit	
	SDA1	IO	I2C1 data signal	
	MOSI1	IO	USART1 SPI MOSI	
	DSCL	IOU	OCD debugger clock	Pull-up
31	P14*	IOUS	Port 1 bit 4 Input/output	
	LCD_S4	O	LCD Segment Signal 4 Output	
	EC1	I	Timer 1(Event Capture) input	
	RXD1	I	USART1 data receive	
	SCL1	IO	I2C1 clock signal	
	MISO1	IO	USART1 SPI MISO	
32	P15*	IOUS	Port 1 bit 5 Input/output	
	LCD_S5	O	LCD Segment Signal 5 Output	
	SS1	IO	USART1 slave select signal	
33	P16*	IOUS	Port 1 bit 6 Input/output	
	LCD_S6	O	LCD Segment Signal 6 Output	
	EC3	I	Timer 3(Event Capture) input	
	SS0	IO	USART0 slave select signal	
34	P17*	IOUS	Port 1 bit 7 Input/output	
	LCD_S7	O	LCD Segment Signal 7 Output	
	SCK0	IO	USART0 clock signal	

Table 2. Normal Pin Description (continued)

Pin no.	Pin name	I/O(1)	Description	Remark
44				
35	P20*	IOUS	Port 2 bit 0 Input/output	
	LCD_S8	O	LCD Segment Signal 8 Output	
	RXD0	I	USART0 data receive	
	SCL0	IO	I2C0 clock signal	
	MISO0	IO	USART0 SPI MISO	
36	P21*	IOUS	Port 2 bit 1 Input/output	
	LCD_S9	O	LCD Segment Signal 9 Output	
	TXD0	O	USART0 data transmit	
	SDA0	IO	I2C0 data signal	
	MOSI0	IO	USART0 SPI MOSI	
37	P22*	IOUS	Port 2 bit 2 Input/output	
	LCD_S10	O	LCD Segment Signal 10 Output	
	RXD1	I	USART1 data receive	
	SCL1	IO	I2C1 clock signal	
	MISO1	IO	USART1 SPI MISO	
38	P23*	IOU	Port 2 bit 3 Input /output	
	LCD_S11	O	LCD Segment Signal 11 Output	
	TXD1	O	USART1 data transmit	
	SDA1	IO	I2C1 data signal	
	MOSI1	IO	USART1 SPI MOSI	
39	P24*	IOU	Port 2 bit 4 Input /output	
	LCD_S12	O	LCD Segment Signal 12 Output	
	SCK1	IO	USART1 clock signal	
40	P25*	IOU	Port 2 bit 5 Input /output	
	LCD_S13	O	LCD Segment Signal 13 Output	
	EC0	I	Timer 0(Event Capture) input	
	SS1	IO	USART1 slave select signal	
41	P26*	IOU	Port 2 bit 6 Input /output	
	LCD_S14	O	LCD Segment Signal 14 Output	
	T5O	O	Timer 5 interval output	
	PWM5O	O	Timer 5 PWM output	
42	P27*	IOU	Port 2 bit 7 Input /output	
	LCD_S15	O	LCD Segment Signal 15 Output	
	T4O	O	Timer 4 interval output	
	PWM4O	O	Timer 4 PWM output	

Table 2. Normal Pin Description (continued)

Pin no.	Pin name	I/O(1)	Description	Remark
44				
26	P30*	IOUS	Port 3 bit 0 Input /output	
	LED_C0	O	High sink current ports	
	LCD_C0	O	LCD Common Signal 0 Output	
	AN12	IA	ADC input ch-12	
25	P31*	IOUS	Port 3 bit 1 Input /output	
	LED_C1	O	High sink current ports	
	LCD_C1	O	LCD Common Signal 1 Output	
	AN11	IA	ADC input ch-11	
24	P32*	IOUS	Port 3 bit 2 Input /output	
	LED_C2	O	High sink current ports	
	LCD_C2	O	LCD Common Signal 2 Output	
	AN10	IA	ADC input ch-10	
23	P33*	IOUS	Port 3 bit 3 Input /output	
	LED_C3	O	High sink current ports	
	LCD_C3	O	LCD Common Signal 3 Output	
	AN9	IA	ADC input ch-9	
22	P34*	IOUS	Port 3 bit 4 Input /output	
	LED_C4	O	High sink current ports	
	LCD_C4	O	LCD Common Signal 4 Output /	
	LCD_S23	O	LCD Segment Signal 23 Output	
	AN8	IA	ADC input ch-8	
21	P35*	IOUS	Port 3 bit 5 Input /output	
	LED_C5	O	High sink current ports	
	LCD_C5	O	LCD Common Signal 5 Output	
	LCD_S22	O	LCD Segment Signal 22 Output	
	AN7	IA	ADC input ch-7	
20	P36*	IOUS	Port 3 bit 6 Input/output	
	LED_C6	O	High sink current ports	
	LCD_C6	O	LCD Common Signal 6 Output	
	LCD_S21	O	LCD Segment Signal 21 Output	
	AN6	IA	ADC input ch-6	
19	P37*	IOUS	Port 3 bit 7 Input/output	
	LED_C7	O	High sink current ports	
	LCD_C7	O	LCD Common Signal 7 Output	
	LCD_S20	O	LCD Segment Signal 20 Output	
	AN5	IA	ADC input ch-5	

Table 2. Normal Pin Description (continued)

Pin no.	Pin name	I/O(1)	Description	Remark
44				
18	P40*	IOUS	Port 4 bit 0 Input/output	
	EINT40	I	External interrupt input ch-40	
	AN4	IA	ADC input ch-4	
	VLC0	IA	External LCD Voltage bias 0	
17	P41*	IOUS	Port 4 bit 1 Input/output	
	EINT41	I	External interrupt input ch-41	
	AN3	IA	ADC input ch-3	
	VLC1	IA	External LCD Voltage bias 1	
16	P42*	IOUS	Port 4 bit 2 Input/output	
	EINT42	I	External interrupt input ch-42	
	AN2	IA	ADC input ch-2	
	VLC2	IA	External LCD Voltage bias 2	
	SS2	IO	USART2 slave select signal	
15	P43*	IOUS	Port 4 bit 3 Input/output	
	EINT43	I	External interrupt input ch-43	
	AN1	IA	ADC input ch-1	
	VLC3	IA	External LCD Voltage bias 3	
	XCK2	IO	USART2 clock signal	
14	P44*	IOUC	Port 4 bit 4 Input/output	
	AN0	IA	ADC input ch-0	
	AVREF	P	A/D converter reference voltage	
	MOSI2	IO	USART2 SPI MOSI	
	TXD2	O	USART2 data transmit	
13	P45*	IOUC	Port 4 bit 5 Input/output	
	LCD_S19	O	LCD Segment Signal 19 Output	
	T3O	O	Timer 3 interval output	
	PWM3O	O	Timer 3 PWM output	
	MISO2	IO	USART2 SPI MISO	
	RXD2	I	USART2 data receive	

Table 2. Normal Pin Description (continued)

Pin no.	Pin name	I/O(1)	Description	Remark
44				
8	P50*	IOUS	Port 5 bit 0 Input/output	
	XIN2	I	Main Oscillator Input ch-2	
	SXIN	I	Sub Oscillator Input	
9	P51*	IOUS	Port 5 bit 1 Input/output	
	XOUT2	O	Main Oscillator Output ch-2	
	SXOUT	O	Sub Oscillator Output	
10	P52*	IOUS	Port 5 bit 2 Input/output	
	RESETB	IU	Reset pin	Pull-up
11	P53*	IOUS	Port 5 bit 3 Input/output	
	BUZO	O	Buzzer output	

NOTES:

1. The P52/RESETB pin is configured as one of the P52 and RESETB pin by the “CONFIGURE OPTION.”
2. If the P11/LCD_S1/EINT45/AN14/DSDA and P13/LCD_S3/EINT47/DSCL pins are connected to the programmer during power-on reset, the pins are automatically configured as In-system programming pins.
3. The P11/LCD_S1/EINT45/AN14/DSDA and P13/LCD_S3/EINT47/DSCL pins are configured as inputs with internal pull-up resistor only during the reset or power-on reset.
4. The P50/XIN2/SXIN, P51/XOUT2/SXOUT, P06/RXD2/XOUT/EINT2, and P07/XIN/EINT3 pins are configured as a function pin by software control.
5. (1) I=Input, O=Output, U=Pull-up, D=Pull-down, S=Schmitt-Trigger Input Type, C=CMOS Input Type, A=Analog, P=Power
6. The * means ‘Selected pin function after reset condition

3 Port structures

In this chapter, two port structures are introduced in Figure 3 and Figure 4 regarding general purpose I/O port and external interrupt I/O port respectively.

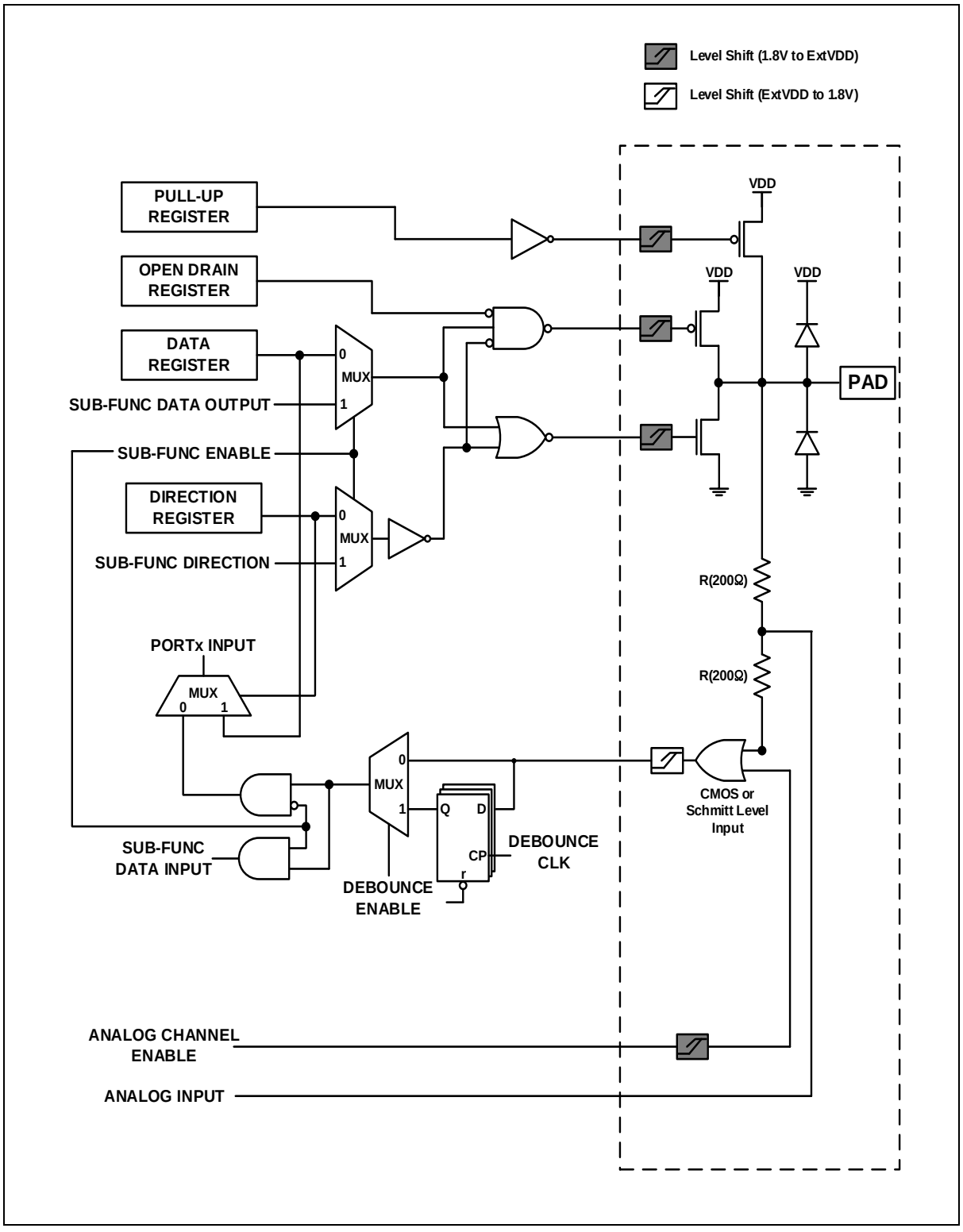


Figure 3. General Purpose I/O Port

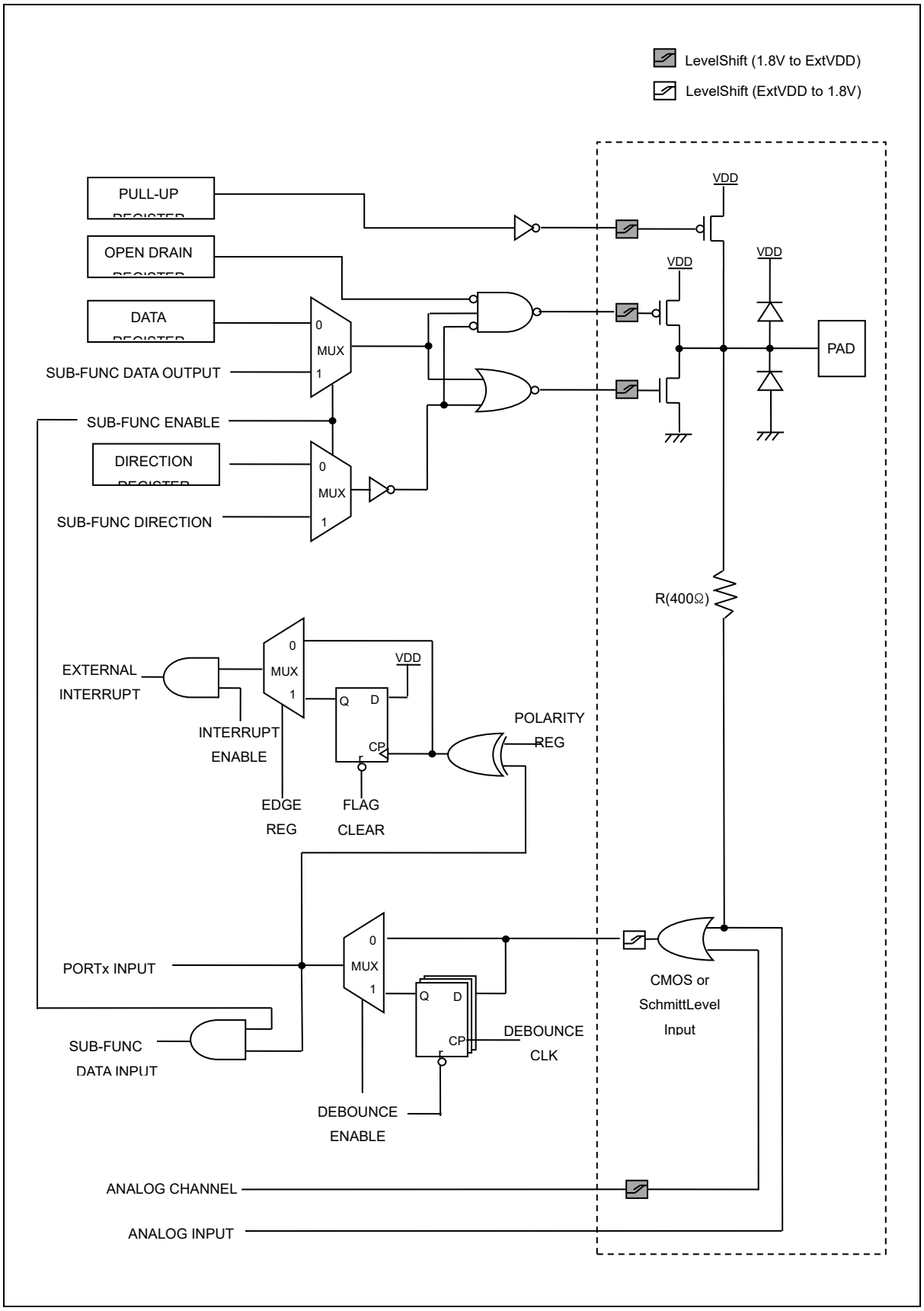


Figure 4. External Interrupt I/O Port

4 Central Processing Unit (CPU)

This is based on Mentor Graphics M8051EW core, and it improves code efficiency and performance.

4.1 Architecture and registers

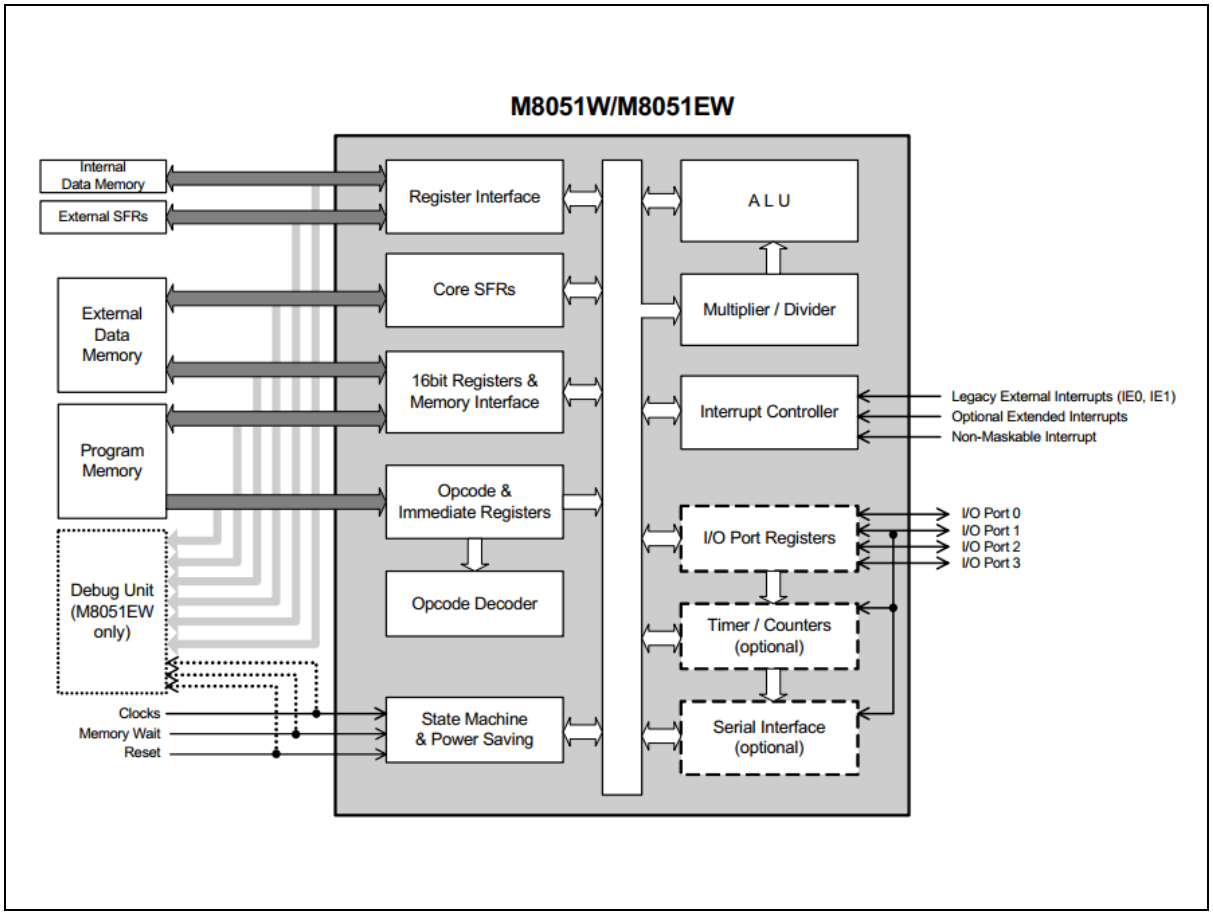


Figure 5. M8051EW Architecture

- Two clocks per machine cycle architecture:
  This allows the device either to run at up to six times the speed at the same power consumption or to use one sixth of the power when running at the standard speed. All instructions have zero-wait-state execution times that are exactly 1/6 those of the standard part.

- Debug support (OCD and OCD II):
  The M8051EW offers a Debug Mode together with a set of dedicated Debug signals which may be used by external debug hardware of 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 1 Mbyte of external Data Memory, accessible by a choice of interfaces
  - Up to 256 bytes of Internal Data Memory
  - Up to 1 Mbyte of RAM or ROM Program Memory, accessible by a choice of interfaces
- Support for synchronous and asynchronous Program, External Data & Internal Data Memory
- Wait states support for slow Program and External Data Memory
- 16-bit Data Memory address is generated through the DPTR(Data Point register).
- 16-bit program counter - capable of addressing up to Flash size in Each device
- A single data pointer, two memory-mapped data pointers, or 2, 4 or 8 banked data pointers
- Support 2 or 4 level of priority scheme – Up to 24 maskable Interrupt sources
- External Special Function Register(SFR) are memory mapped into Direct Memory between addresses 80 hex and FF hex

4.2 Addressing

- 6 addressing modes
- Direct addressing
  - In Direct Addressing, the operand is specified by an 8-bit address field. Only internal data and SFRs may be accessed using this mode.
- Indirect addressing
  - In Indirect Addressing, the operand is specified by an address 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.
- Register addressing
  - In Register Addressing, 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 bits 3 and 4 of the PSW.
- Register specific addressing
  - 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.
- Immediate DATA
  - 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.
    For Example) MOV A, #100
    loads the Accumulator with the decimal number 100. The same number could be specified in hex digits as 64H.
- Indexed addressing
  - Only Program Memory may be addressed using Indexed Addressing. 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

If you need an Instruction table, please refer to Appendix.

- 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 implement ADD, ADDC (Add with Carry), SUBB (Subtract with Borrow), INC (Increment) and DEC (Decrement) functions, which may 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 implement ANL (AND Logical), ORL (OR Logical), and XRL (Exclusive-OR Logical) functions, which again may 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 may 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 may be moved from the accumulator to or from an 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 to –128 to +127bytes), Long jumps (LJMP) are absolute 16-bit jumps and Absolute jumps (AJMP) are absolute 11-bit jumps (ex. within a 2Kbyte memory page). The last type is an Indexed jump, JMP @A+DPTR, which jumps to a location contained in the DPTR register, 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 –128 to +127 bytes.
- 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 the operands and each addressable bit may be set, cleared or tested in a jump instruction.
- Flag :
  Certain instructions affect one or more of the flags generated by the ALU

5 Memory organization

A96G150 addresses two separate address 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.

A96G150 provides on-chip 64Kbytes 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 2304bytes and internal EEPROM is 2Kbytes.

5.1 Program memory

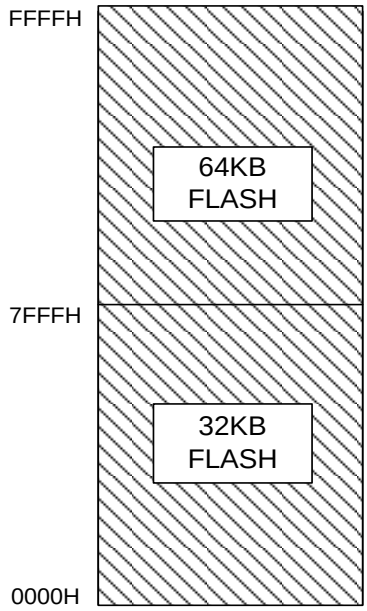
A 16-bit program counter is capable of addressing up to 64Kbytes, and A96G150 has just 64Kbytes program memory space.

Figure 6 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 64Kbytes includes the Interrupt Vector Region.

Figure 6. 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 7, 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 8. 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.

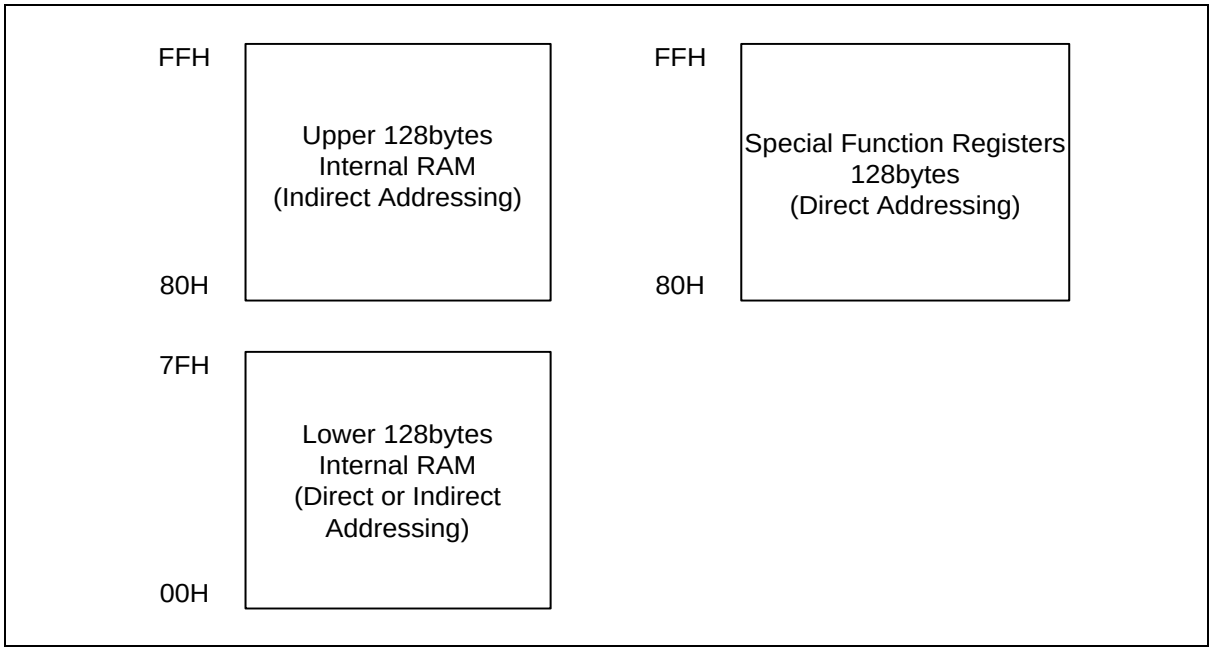


Figure 7. Data Memory Map

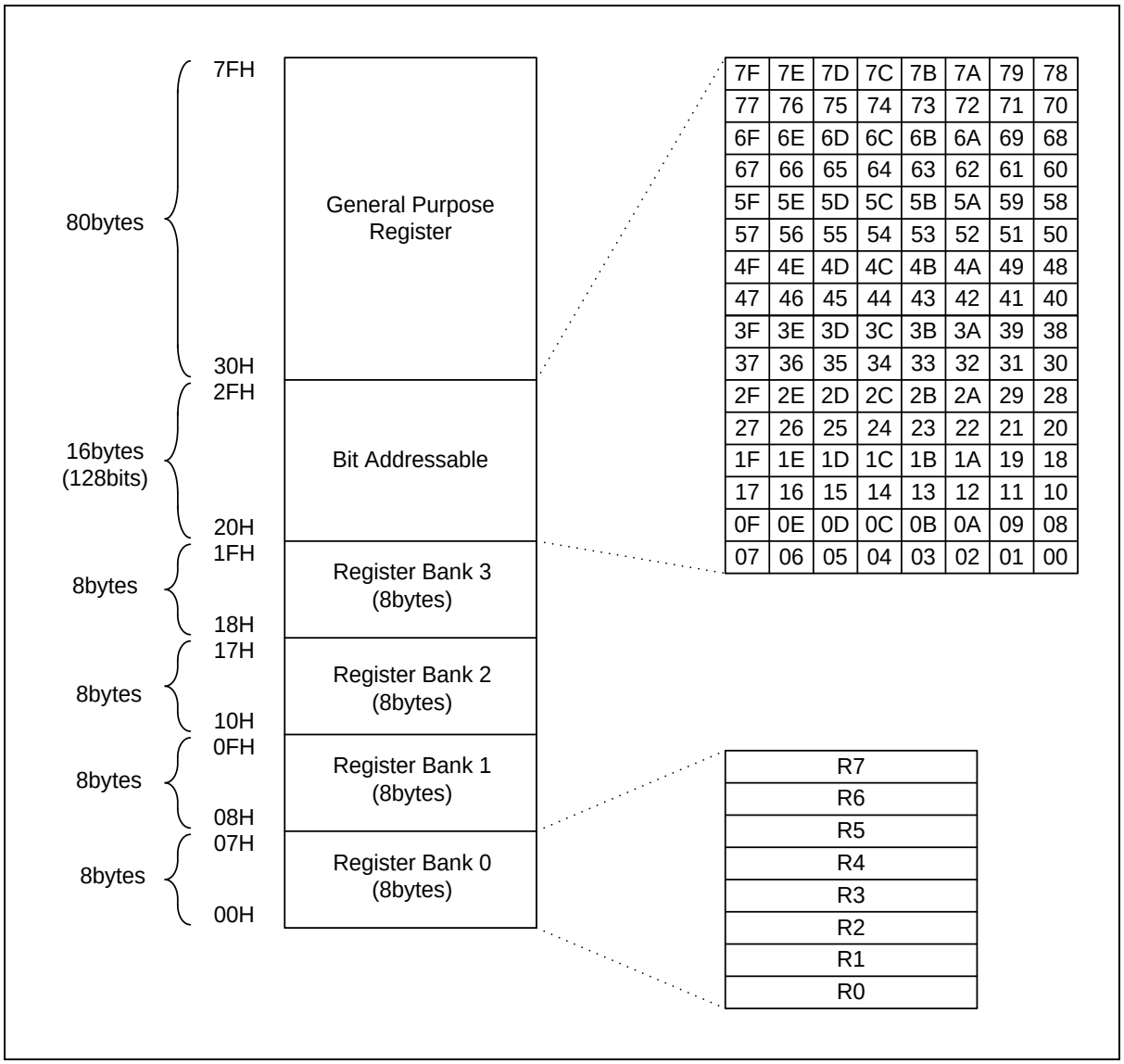


Figure 8. Lower 128bytes of RAM

5.3 EEPROM data memory and external data memory

A96G150 has 2048bytes of EEPROM, 2304bytes 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.

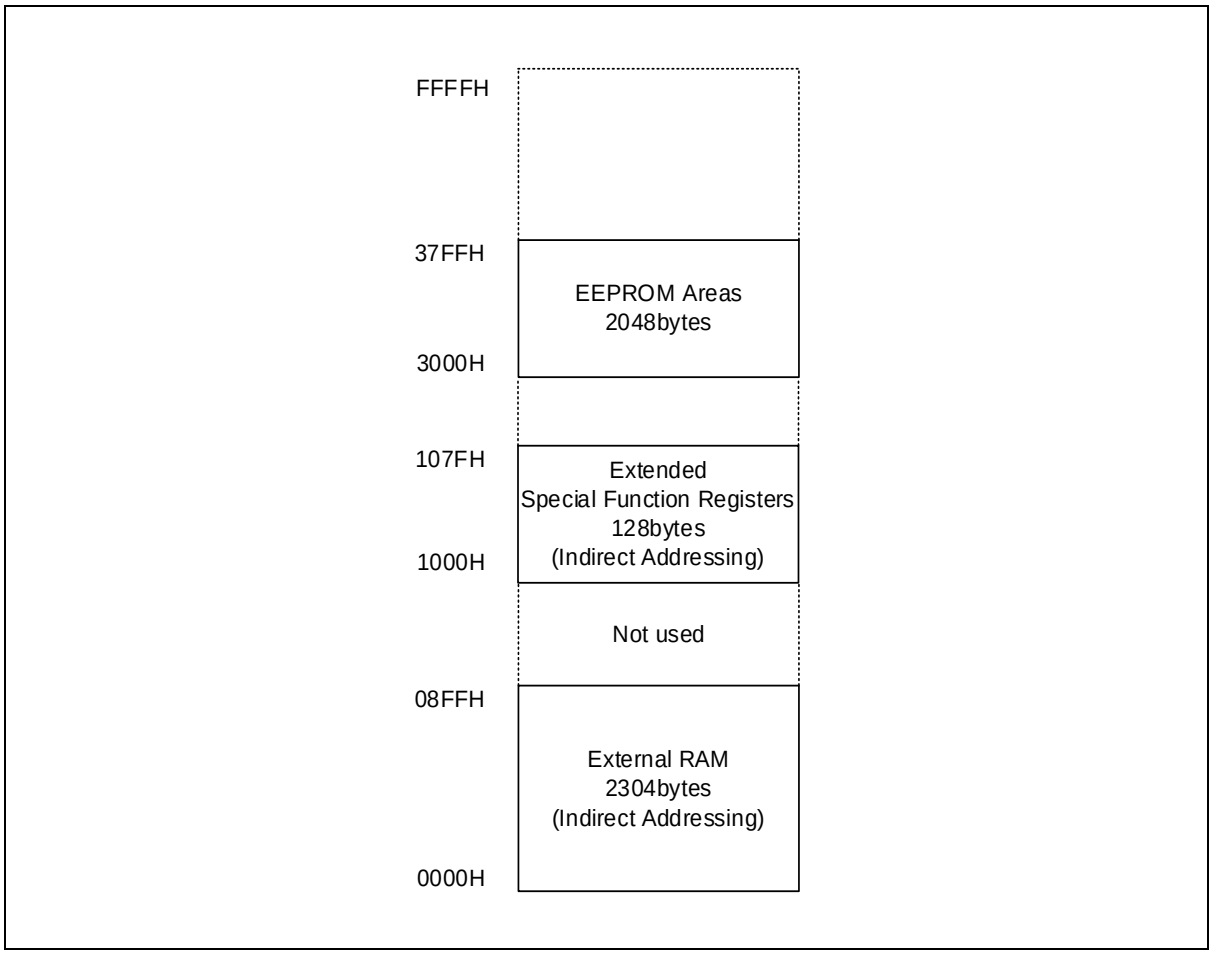


Figure 9. XDATA Memory Area

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	P4FSRL	P4FSRH	–	UBAUD	UDATA	–	P5FSR
0F0H	B	USI1ST1	USI1ST2	USI1BD	USI1SDHR	USI1DR	USI1SCLR	USI1SCHR
0E8H	RSTFR	USI1CR1	USI1CR2	USI1CR3	USI1CR4	USI1SAR	P3FSRL	P3FSRH
0E0H	ACC	USI0ST1	USI0ST2	USI0BD	USI0SDHR	USI0DR	USI0SCLR	USI0SCHR
0D8H	LVRCR	USI0CR1	USI0CR2	USI0CR3	USI0CR4	USI0SAR	P0DB	P14DB
0D0H	PSW	P5IO	P0FSRL	P0FSRH	P1FSRL	P1FSRH	P2FSRL	P2FSRH
0C8H	OSCCR	P4IO	–	UCTRL1	UCTRL2	UCTRL3	–	USTAT
0C0H	EIFLAG0	P3IO	T2CRL	T2CRH	T2ADRL	T2ADRH	T2BDRL	T2BDRH
0B8H	IP	P2IO	T1CRL	T1CRH	T1ADRL	T1ADRH	T1BDRL	T1BDRH
0B0H	P5	P1IO	T0CR	T0CNT	T0DR/ T0CDR	P1SSR	–	–
0A8H	IE	IE1	IE2	IE3	P0PU	P1PU	P2PU	P3PU
0A0H	P4	P0IO	EO	P4PU	EIPOL0L	EIPOL0H	EIFLAG1	EIPOL1
98H	P3	–	–	–	ADCCRL	ADCCRH	ADCDRL	ADCDRH
90H	P2	P0OD	P1OD	P2OD	P4OD	P5PU	WTCR	BUZCR
88H	P1	WTDR/ WTCNT	SCCR	BITCR	BITCNT	WDTCR	WDTDR/ WDTCNT	BUZDR
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	—	CRC_ADDR_START_H	CRC_ADDR_START_M	CRC_ADDR_START_L	CRC_ADDR_END_H	CRC_ADDR_END_M	CRC_ADDR_END_L	—
1070H	CRC_CON	—	CRC_H	CRC_L	CRC_MNT_H	CRC_MNT_L	—	—
1068H	—	—	—	—	—	—	—	—
1060H	LCD_DR16	LCD_DR17	LCD_DR18	LCD_DR19	LCD_DR20	LCD_DR21	LCD_DR22	LCD_DR23
1058H	LCD_DR8	LCD_DR9	LCD_DR10	LCD_DR11	LCD_DR12	LCD_DR13	LCD_DR14	LCD_DR15
1050H	LCD_DR0	LCD_DR1	LCD_DR2	LCD_DR3	LCD_DR4	LCD_DR5	LCD_DR6	LCD_DR7
1048H	LCDCR	LCDBCCRH	LCDBCCRL	LCDBSSRH	LCDBSSRL	—	—	—
1040H	—	—	—	—	—	—	—	—
1038H	XTFLSR	—	—	—	—	—	—	—
1030H	—	—	—	—	—	—	—	—
1028H	FEARH	FEARM	FEARL	FEDR	FETR	—	—	—
1020H	FEMR	FECR	FESR	FETCR	FEARM1	FEARL1	—	—
1018H	UCTRL4	FPCR	RTOCH	RTOCL	—	—	—	—
1010H	T5CRH	T5CRL	T5ADRH	T5ADRL	T5BDRH	T5BDRL	—	—
1008H	T4CRH	T4CRL	T4ADRH	T4ADRL	T4BDRH	T4BDRL	—	—
1000H	T3CRH	T3CRL	T3ADRH	T3ADRL	T3BDRH	T3BDRL	—	—

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
89H	Watch Timer Data Register	WTDR	W	0	1	1	1	1	1	1	1
	Watch Timer Counter Register	WTCNT	R	–	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
8EH	Watch Dog Timer Data Register	WDTDR	W	1	1	1	1	1	1	1	1
	Watch Dog Timer Counter Register	WDTCNT	R	0	0	0	0	0	0	0	0
8FH	BUZZER Data Register	BUZDR	R/W	1	1	1	1	1	1	1	1
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
95H	P5 Pull-up Resistor Selection Register	P5PU	R/W	–	–	0	0	0	0	0	0
96H	Watch Timer Control Register	WTCR	R/W	0	–	–	0	0	0	0	0
97H	BUZZER Control Register	BUZCR	R/W	–	–	–	–	–	0	0	0
98H	P3 Data Register	P3	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
A0H	P4 Data Register	P4	R/W	0	0	0	0	0	0	0	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
A3H	P4 Pull-up Resistor Selection Register	P4PU	R/W	0	0	0	0	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
ABH	Interrupt Enable Register 3	IE3	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
AFH	P3 Pull-up Resistor Selection Register	P3PU	R/W	0	0	0	0	0	0	0	0
B0H	P5 Data Register	P5	R/W	–	–	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
B5H	P1 SEG Option Selection Register	P1SSR	R/W	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

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
C1H	P3 Direction Register	P3IO	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
C9H	P4 Direction Register	P4IO	R/W	0	0	0	0	0	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
D1H	P5 Direction Register	P5IO	R/W	–	–	–	–	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 Low Register	P2FSRL	R/W	0	0	0	0	0	0	0	0
D7H	P2 Function Selection High Register	P2FSRH	R/W	0	0	0	0	0	0	0	0
D8H	Low Voltage Reset Control Register	LVRCR	R/W	–	–	–	0	0	0	0	0
D9H	USI0 Control Register 1	USI0CR1	R/W	0	0	0	0	0	0	0	0
DAH	USI0 Control Register 2	USI0CR2	R/W	0	0	0	0	0	0	0	0
DBH	USI0 Control Register 3	USI0CR3	R/W	0	0	0	0	0	0	0	0
DCH	USI0 Control Register 4	USI0CR4	R/W	0	–	0	0	0	0	0	0
DDH	USI0 Slave Address Register	USI0SAR	R/W	0	0	0	0	0	0	0	0
DEH	P0 De-bounce Enable Register	P0DB	R/W	0	0	0	0	0	0	0	0
DFH	P1/P4 De-bounce Enable Register	P14DB	R/W	0	0	0	0	0	0	0	0

Table 5. SFR Map (continued)

Address	Function	Symbol	R/W	@Reset							
				7	6	5	4	3	2	1	0
E0H	Accumulator Register	ACC	R/W	0	0	0	0	0	0	0	0
E1H	USI0 Status Register 1	USI0ST1	R/W	0	0	0	0	–	0	0	0
E2H	USI0 Status Register 2	USI0ST2	R	0	0	0	0	0	0	0	0
E3H	USI0 Baud Rate Generation Register	USI0BD	R/W	1	1	1	1	1	1	1	1
E4H	USI0 SDA Hold Time Register	USI0SHDR	R/W	0	0	0	0	0	0	0	1
E5H	USI0 Data Register	USI0DR	R/W	0	0	0	0	0	0	0	0
E6H	USI0 SCL Low Period Register	USI0SCLR	R/W	0	0	1	1	1	1	1	1
E7H	USI0 SCL High Period Register	USI0SCHR	R/W	0	0	1	1	1	1	1	1
E8H	Reset Flag Register	RSTFR	R/W	1	x	0	0	x	–	–	–
E9H	USI1 Control Register 1	USI1CR1	R/W	0	0	0	0	0	0	0	0
EAH	USI1 Control Register 2	USI1CR2	R/W	0	0	0	0	0	0	0	0
EBH	USI1 Control Register 3	USI1CR3	R/W	0	0	0	0	0	0	0	0
ECH	USI1 Control Register 4	USI1CR4	R/W	0	–	–	0	0	–	0	0
EDH	USI1 Slave Address Register	USI1SAR	R/W	0	0	0	0	0	0	0	0
EEH	P3 Function Selection Low Register	P3FSRL	R/W	0	0	0	0	0	0	0	0
EFH	P3 Function Selection High Register	P3FSRH	R/W	0	0	0	0	0	0	0	0
F0H	B Register	B	R/W	0	0	0	0	0	0	0	0
F1H	USI1 Status Register 1	USI1ST1	R/W	0	0	0	0	–	0	0	0
F2H	USI1 Status Register 2	USI1ST2	R	0	0	0	0	0	0	0	0
F3H	USI1 Baud Rate Generation Register	USI1BD	R/W	1	1	1	1	1	1	1	1
F4H	USI1 SDA Hold Time Register	USI1SHDR	R/W	0	0	0	0	0	0	0	1
F5H	USI1 Data Register	USI1DR	R/W	0	0	0	0	0	0	0	0
F6H	USI1 SCL Low Period Register	USI1SCLR	R/W	0	0	1	1	1	1	1	1
F7H	USI1 SCL High Period Register	USI1SCHR	R/W	0	0	1	1	1	1	1	1
F8H	Interrupt Priority Register 1	IP1	R/W	–	–	0	0	0	0	0	0
F9H	P4 Function Selection Low Register	P4FSRL	R/W	0	0	0	0	0	0	0	0
FAH	P4 Function Selection High Register	P4FSRH	R/W	0	0	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
FFH	P5 Function Selection Register	P5FSR	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
1000H	Timer 3 Control High Register	T3CRH	R/W	0	–	0	0	–	–	–	0
1001H	Timer 3 Control Low Register	T3CRL	R/W	0	0	0	0	–	0	0	0
1002H	Timer 3 A Data High Register	T3ADRH	R/W	1	1	1	1	1	1	1	1
1003H	Timer 3 A Data Low Register	T3ADRL	R/W	1	1	1	1	1	1	1	1
1004H	Timer 3 B Data High Register	T3BDRH	R/W	1	1	1	1	1	1	1	1
1005H	Timer 3 B Data Low Register	T3BDRL	R/W	1	1	1	1	1	1	1	1
1008H	Timer 4 Control High Register	T4CRH	R/W	0	–	0	0	–	–	–	0
1009H	Timer 4 Control Low Register	T4CRL	R/W	0	0	0	0	–	0	–	0
100AH	Timer 4 A Data High Register	T4ADRH	R/W	1	1	1	1	1	1	1	1
100BH	Timer 4 A Data Low Register	T4ADRL	R/W	1	1	1	1	1	1	1	1
100CH	Timer 4 B Data High Register	T4BDRH	R/W	1	1	1	1	1	1	1	1
100DH	Timer 4 B Data Low Register	T4BDRL	R/W	1	1	1	1	1	1	1	1
1010H	Timer 5 Control High Register	T5CRH	R/W	0	–	0	0	–	–	–	0
1011H	Timer 5 Control Low Register	T5CRL	R/W	0	0	0	0	–	0	–	0
1012H	Timer 5 A Data High Register	T5ADRH	R/W	1	1	1	1	1	1	1	1
1013H	Timer 5 A Data Low Register	T5ADRL	R/W	1	1	1	1	1	1	1	1
1014H	Timer 5 B Data High Register	T5BDRH	R/W	1	1	1	1	1	1	1	1
1015H	Timer 5 B Data Low Register	T5BDRL	R/W	1	1	1	1	1	1	1	1
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
1038H	Main Crystal OSC Filter Selection Register	XTFLSR	R/W	–	–	0	0	0	0	0	0

Table 6. XSFR Map (continued)

Address	Function	Symbol	R/W	@Reset							
				7	6	5	4	3	2	1	0
1048H	LCD Driver Control Register	LCDCR	R/W	0	0	0	0	0	0	0	0
1049H	LCD Automatic Bias and Contrast Control High Register	LCDBCCRH	R/W	0	0	–	0	–	0	0	0
104AH	LCD Automatic Bias and Contrast Control Low Register	LCDBCCRL	R/W	–	–	0	–	0	0	0	0
104BH	LCD source Selection High Register	LCDBSSRH	R/W	0	–	–	–	0	–	0	0
104CH	LCD source Selection Low Register	LCDBSSRL	R/W	0	0	0	0	0	0	0	0
1050H	LCD Display Data Register 0	LCDDR0	R/W	0	0	0	0	0	0	0	0
1051H	LCD Display Data Register 1	LCDDR1	R/W	0	0	0	0	0	0	0	0
1052H	LCD Display Data Register 2	LCDDR2	R/W	0	0	0	0	0	0	0	0
1053H	LCD Display Data Register 3	LCDDR3	R/W	0	0	0	0	0	0	0	0
1054H	LCD Display Data Register 4	LCDDR4	R/W	0	0	0	0	0	0	0	0
1055H	LCD Display Data Register 5	LCDDR5	R/W	0	0	0	0	0	0	0	0
1056H	LCD Display Data Register 6	LCDDR6	R/W	0	0	0	0	0	0	0	0
1057H	LCD Display Data Register 7	LCDDR7	R/W	0	0	0	0	0	0	0	0
1058H	LCD Display Data Register 8	LCDDR8	R/W	0	0	0	0	0	0	0	0
1059H	LCD Display Data Register 9	LCDDR9	R/W	0	0	0	0	0	0	0	0
105AH	LCD Display Data Register 10	LCDDR10	R/W	0	0	0	0	0	0	0	0
105BH	LCD Display Data Register 11	LCDDR11	R/W	0	0	0	0	0	0	0	0
105CH	LCD Display Data Register 12	LCDDR12	R/W	0	0	0	0	0	0	0	0
105DH	LCD Display Data Register 13	LCDDR13	R/W	0	0	0	0	0	0	0	0
105EH	LCD Display Data Register 14	LCDDR14	R/W	0	0	0	0	0	0	0	0
105FH	LCD Display Data Register 15	LCDDR15	R/W	0	0	0	0	0	0	0	0
1060H	LCD Display Data Register 16	LCDDR16	R/W	0	0	0	0	0	0	0	0
1061H	LCD Display Data Register 17	LCDDR17	R/W	0	0	0	0	0	0	0	0
1062H	LCD Display Data Register 18	LCDDR18	R/W	0	0	0	0	0	0	0	0
1063H	LCD Display Data Register 19	LCDDR19	R/W	0	0	0	0	0	0	0	0
1064H	LCD Display Data Register 20	LCDDR20	R/W	0	0	0	0	0	0	0	0
1065H	LCD Display Data Register 21	LCDDR21	R/W	0	0	0	0	0	0	0	0
1066H	LCD Display Data Register 22	LCDDR22	R/W	0	0	0	0	0	0	0	0
1067H	LCD Display Data Register 23	LCDDR23	R/W	0	0	0	0	0	0	0	0

Table 6. XSFR Map (continued)

Address	Function	Symbol	R/W	@Reset							
				7	6	5	4	3	2	1	0
1070H	CRC Control Register	CRC_CON	R/W	0	0	0	0	0	0	0	0
1072H	CRC High Register	CRC_H	R/W	0	0	0	0	0	0	0	0
1073H	CRC Low Register	CRC_L	R/W	0	0	0	0	0	0	0	0
1074H	CRC Monitor High Register	CRC_MNT_H	R/W	0	0	0	0	0	0	0	0
1075H	CRC Monitor Low Register	CRC_MNT_L	R/W	0	0	0	0	0	0	0	0
1079H	CRC Start Address High Register	CRC_ADDR_START_H	R/W	–	–	–	–	–	–	–	0
107AH	CRC Start Address Middle Register	CRC_ADDR_START_M	R/W	0	0	0	0	0	0	0	0
107BH	CRC Start Address Low Register	CRC_ADDR_START_L	R/W	0	0	0	0	0	0	0	0
107CH	CRC End Address High Register	CRC_ADDR_END_H	R/W	–	–	–	–	–	–	–	0
107DH	CRC End Address Middle Register	CRC_ADDR_END_M	R/W	0	0	0	0	0	0	0	0
107EH	CRC End Address Low Register	CRC_ADDR_END_L	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

A96G150 has ten groups of I/O ports (P0 ~ P5). Each port can be easily configured by software as I/O pin, internal pull up and open-drain pin to meet various system configurations and design requirements. 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)

P0[7:4], P1[3:0], P4[3:0] support debounce function. Debounce clocks of each ports are fx/1, fx/4, fx/4096 and LSIRC(128kHz).

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
P14DB	DFH	R/W	00H	P1/P4Debounce Enable Register
P1FSRH	D5H	R/W	00H	P1 Function Selection High Register
P1FSRL	D4H	R/W	00H	P1 Function Selection Low Register
P1SSR	B5H	R/W	00H	P1 SEG option Selection 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
P2FSRH	D7H	R/W	00H	P2 Function Selection High Register
P2FSRL	D6H	R/W	00H	P2 Function Selection Low Register
P3	98H	R/W	00H	P3 Data Register
P3IO	C1H	R/W	00H	P3 Direction Register
P3PU	AFH	R/W	00H	P3 Pull-up Resistor Selection Register
P3FSRH	EFH	R/W	00H	P3 Function Selection High Register
P3FSRL	EEH	R/W	00H	P3 Function Selection Low Register
P4	A0H	R/W	00H	P4 Data Register
P4IO	C9H	R/W	00H	P4 Direction Register
P4PU	A3H	R/W	00H	P4 Pull-up Resistor Selection Register
P4OD	94H	R/W	00H	P4 Open-drain Selection Register
P4FSRH	FAH	R/W	00H	P4 Function Selection High Register
P4FSRL	F9H	R/W	00H	P4 Function Selection Low Register
P5	B0H	R/W	00H	P5 Data Register
P5IO	D1H	R/W	00H	P5 Direction Register
P5PU	95H	R/W	00H	P5 Pull-up Resistor Selection Register
P5FSR	FFH	R/W	00H	P5 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:

EINT0 to EINT3 function possible when input.

EC3 (P07) possible when P0FSRH[7:6] = ‘10’.

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	-	-
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	

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 (EINT3 function possible when input)
0	1	XIN1 Function
1	0	EC3 Function
1	1	reserved

P0FSRH[5:4] P06 Function Select

P0FSRH5	P0FSRH4	Description
0	0	I/O Port (EINT2 function possible when input)
0	1	XOUT1 Function
1	0	reserved
1	1	RXD2 Function

P0FSRH[3:2] P05 Function Select

P0FSRH3	P0FSRH2	Description
0	0	I/O Port (EINT1 function possible when input)
0	1	reserved
1	0	reserved
1	1	TXD2 Function

P0FSRH[1:0] P04 Function Select

P0FSRH1	P0FSRH0	Description
0	0	I/O Port (EINT0 function possible when input)
0	1	reserved
1	0	reserved
1	1	reserved

NOTE: When EC3 function is used on the P07 other than the P16, the P16 must be set as an Output or used for other functions except EC3, because it overlaps the P07 (EC3 function) if the P16 is used as an Input.

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	reserved

P0FSRL[5:4] P02 Function Select

P0FSRL5	P0FSRL4	Description
0	0	I/O Port
0	1	LCD_S18 Function
1	0	T1O/PWM1O Function
1	1	reserved

P0FSRL[3:2] P01 Function Select

P0FSRL3	P0FSRL2	Description
0	0	I/O Port
0	1	LCD_S17 Function
1	0	T2O/PWM2O Function
1	1	reserved

P0FSRL[1:0] P00 Function Select

P0FSRL1	P0FSRL0	Description
0	0	I/O Port
0	1	LCD_S16 Function
1	0	T3O/PWM3O Function
1	1	reserved

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 (P14DB), P1 pull-up resistor selection register (P1PU), andP1 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 (P14), EC3 (P16) / EINT44 to EINT47 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

P14DB (P1/P4 De-bounce Enable Register): DFH

7	6	5	4	3	2	1	0
P43DB	P42DB	P41DB	P40DB	P13DB	P12DB	P11DB	P10DB
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

P43DB	Configure De-bounce of P43 Port	
	0	Disable
	1	Enable
P42DB	Configure De-bounce of P42 Port	
	0	Disable
	1	Enable
P41DB	Configure De-bounce of P41 Port	
	0	Disable
	1	Enable
P40DB	Configure De-bounce of P40 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 and port 4.

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	LCD_S7 Function
1	0	reserved
1	1	SCK0 Function

P1FSRH[5:4] P16 Function Select

P1FSRH5	P1FSRH4	Description
0	0	I/O Port (EC3 function possible when input)
0	1	LCD_S6 Function
1	0	reserved
1	1	SS0 Function

P1FSRH[3:2] P15 Function Select

P1FSRH3	P1FSRH2	Description
0	0	I/O Port
0	1	LCD_S5 Function
1	0	reserved
1	1	SS1 Function

P1FSRH[1:0] P14 Function Select

P1FSRH1	P1FSRH0	Description
0	0	I/O Port (EC1 function possible when input)
0	1	LCD_S4 Function
1	0	reserved
1	1	RXD1/SCL1/MISO1 Function

NOTE: When EC3 function is used on the P07 other than the P16, the P16 must be set as an Output or used for other functions except EC3, because it overlaps the P07 (EC3 function) if the P16 is used as an Input.

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 (EINT47 function possible when input)
0	1	LCD_S3 Function
1	0	reserved
1	1	TXD1/SDA1/MOSI1 Function

P1FSRL[5:4] P12 Function Select

P1FSRL5	P1FSRL4	Description
0	0	I/O Port (EINT46 function possible when input)
0	1	LCD_S2 Function
1	0	reserved
1	1	SCK1 Function

P1FSRL[3:2] P11 Function Select

P1FSRL3	P1FSRL2	Description
0	0	I/O Port (EINT45 function possible when input)
0	1	LCD_S1 Function
1	0	AN14 Function
1	1	reserved

P1FSRL[1:0] P10 Function Select

P1FSRL1	P1FSRL0	Description
0	0	I/O Port (EINT44 function possible when input)
0	1	LCD_S0 Function
1	0	AN13 Function
1	1	reserved

P1SSR (P1 SEG option Selection Register): B5H

7	6	5	4	3	2	1	0
P17SSR	P16SSR	P15SSR	P14SSR	P13SSR	P12SSR	P11SSR	P10SSR
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

P1SSR[7:0] Seg option enable of P1 Port

0	Normal GPIO current
1	Seg option current 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
P27	P26	P25	P24	P23	P22	P21	P20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

P2[7:0]	I/O Data

P2IO (P2 Direction Register): B9H

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

Initial value: 00H

P2IO[7:0]	P2 Data I/O Direction	
	0	Input
	1	Output
	NOTE: EC0 (P25) function possible when input.	

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

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

Initial value: 00H

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

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

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

Initial value: 00H

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

P2FSRH (Port 2 Function Selection High Register): D7H

7	6	5	4	3	2	1	0
P2FSRH7	P2FSRH6	P2FSRH5	P2FSRH4	P2FSRH3	P2FSRH2	P2FSRH1	P2FSRH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

P2FSRH[7:6] P27 Function Select

P2FSRH7	P2FSRH6	Description
0	0	I/O Port
0	1	LCD_S15 Function
1	0	T4O/PWM4O Function
1	1	reserved

P2FSRH[5:4] P26 Function Select

P2FSRH5	P2FSRH4	Description
0	0	I/O Port
0	1	LCD_S14 Function
1	0	T5O/PWM5O Function
1	1	reserved

P2FSRH[3:2] P25 Function Select

P2FSRH3	P2FSRH2	Description
0	0	I/O Port (EC0 function possible when input)
0	1	LCD_S13 Function
1	0	reserved
1	1	SS1 Function

P2FSRH[1:0] P24 Function Select

P2FSRH1	P2FSRH0	Description
0	0	I/O Port
0	1	LCD_S12 Function
1	0	reserved
1	1	SCK1 Function

P2FSRL (Port 2 Function Selection Low Register): D6H

7	6	5	4	3	2	1	0
P2FSRL7	P2FSRL6	P2FSRL5	P2FSRL4	P2FSRL3	P2FSRL2	P2FSRL1	P2FSRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

P2FSRL[7:6] P23 Function Select

P2FSRL7	P2FSRL6	Description
0	0	I/O Port
0	1	LCD_S11 Function
1	0	reserved
1	1	TXD1/SDA1/MOSI1 Function

P2FSRL[5:4] P22 Function Select

P2FSRL5	P2FSRL4	Description
0	0	I/O Port
0	1	LCD_S10 Function
1	0	reserved
1	1	RXD1/SCL1/MISO1 Function

P2FSRL[3:2] P21 Function Select

P2FSRL3	P2FSRL2	Description
0	0	I/O Port
0	1	LCD_S9 Function
1	0	reserved
1	1	TXD0/SDA0/MOSI0 Function

P2FSRL[1:0] P20 Function Select

P2FSRL1	P2FSRL0	Description
0	0	I/O Port
0	1	LCD_S8 Function
1	0	reserved
1	1	RXD0/SCL0/MISO0 Function

6.5 P3 port

6.5.1 P3 port description

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

6.5.2 Register description for P3

P3 (P3 Data Register): 98H

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

Initial value: 00H

P3[7:0]	I/O Data

P3IO (P3 Direction Register): C1H

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

Initial value: 00H

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

P3PU (P3 Pull-up Resistor Selection Register): AFH

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

Initial value: 00H

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

P3FSRH (Port 3 Function Selection High Register): EFH

7	6	5	4	3	2	1	0
P3FSRH7	P3FSRH6	P3FSRH5	P3FSRH4	P3FSRH3	P3FSRH2	P3FSRH1	P3FSRH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

P3FSRH[7:6] P37 Function Select

P3FSRH7	P3FSRH6	Description
0	0	I/O Port
0	1	LCD_C7/LCD_S20 Function
1	0	AN5 Function
1	1	LED_C7 Function

P3FSRH[5:4] P36 Function Select

P3FSRH5	P3FSRH4	Description
0	0	I/O Port
0	1	LCD_C6/LCD_S21 Function
1	0	AN6 Function
1	1	LED_C6 Function

P3FSRH[3:2] P35 Function Select

P3FSRH3	P3FSRH2	Description
0	0	I/O Port
0	1	LCD_C5/LCD_S22 Function
1	0	AN7 Function
1	1	LED_C5 Function

P3FSRH[1:0] P34 Function Select

P3FSRH1	P3FSRH0	Description
0	0	I/O Port
0	1	LCD_C4/LCD_S23 Function
1	0	AN8 Function
1	1	LED_C4 Function

P3FSRL (Port 3 Function Selection Low Register): EEH

7	6	5	4	3	2	1	0
P3FSRL7	P3FSRL6	P3FSRL5	P3FSRL4	P3FSRL3	P3FSRL2	P3FSRL1	P3FSRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

P3FSRL[7:6] P33 Function Select

P3FSRL7	P3FSRL6	Description
0	0	I/O Port
0	1	LCD_C3 Function
1	0	AN9 Function
1	1	LED_C3 Function

P3FSRL[5:4] P32 Function Select

P3FSRL5	P3FSRL4	Description
0	0	I/O Port
0	1	LCD_C2 Function
1	0	AN10 Function
1	1	LED_C2 Function

P3FSRL[3:2] P31 Function Select

P3FSRL3	P3FSRL2	Description
0	0	I/O Port
0	1	LCD_C1 Function
1	0	AN11 Function
1	1	LED_C1 Function

P3FSRL[1:0] P30 Function Select

P3FSRL1	P3FSRL0	Description
0	0	I/O Port
0	1	LCD_C0 Function
1	0	AN12 Function
1	1	LED_C0 Function

NOTE: LED_Cn function is used for high current driving and it can drive about 160mA (Typ.) @VDD=5V, V_{OL}= 1.5V.

6.6 P4 port

6.6.1 P4 port description

P4 is a 6-bit I/O port. P4 control registers consist of P4 data register (P4), P4 direction register (P4IO), P4 pull-up resistor selection register (P4PU) andP4 open-drain selection register (P4OD). Refer to the port function selection registers for the P4 function selection.

6.6.2 Register description for P4

P4 (P4 Data Register): A0H

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

Initial value: 00H

P4[5:0] I/O Data

P4IO (P4 Direction Register): C9H

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

Initial value: 00H

P4IO[5:0] P4 Data I/O Direction

0 Input

1 Output

NOTE: EINT40 to EINT43 function possible when input.

P4PU (P4 Pull-up Resistor Selection Register): A3H

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

Initial value: 00H

P4PU[5:0] Configure Pull-up Resistor of P4 Port

0 Disable

1 Enable

P4OD (P4 Open-drain Selection Register): 94H

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

Initial value: 00H

P4OD[5:0] Configure Open-drain of P4 Port

0 Push-pull output

1 Open-drain output

P4FSRH (Port 4 Function Selection High Register): FAH

7	6	5	4	3	2	1	0
-	-	-	-	P4FSRH3	P4FSRH2	P4FSRH1	P4FSRH0
-	-	-	-	R/W	R/W	R/W	R/W

Initial value: 00H

P4FSRH[3:2] P45 Function Select

P4FSRH3	P4FSRH2	Description
0	0	I/O Port
0	1	LCD_S19 Function
1	0	T3O/PWM3O Function
1	1	RXD2/MISO2 Function

P4FSRH[1:0] P44 Function Select

P4FSRH1	P4FSRH0	Description
0	0	I/O Port
0	1	AVREF Function
1	0	AN0 Function
1	1	TXD2/MOSI2 Function

P4FSRL (Port 4 Function Selection Low Register): F9H

7	6	5	4	3	2	1	0
P4FSRL7	P4FSRL6	P4FSRL5	P4FSRL4	P4FSRL3	P4FSRL2	P4FSRL1	P4FSRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

P4FSRL[7:6] P43 Function Select

P4FSRL7	P4FSRL6	Description
0	0	I/O Port (EINT43 function possible when input)
0	1	VLC3 Function
1	0	AN1 Function
1	1	XCK2 Function

P4FSRL[5:4] P42 Function Select

P4FSRL5	P4FSRL4	Description
0	0	I/O Port (EINT42 function possible when input)
0	1	VLC2 Function
1	0	AN2 Function
1	1	SS2 Function

P4FSRL[3:2] P41 Function Select

P4FSRL3	P4FSRL2	Description
0	0	I/O Port (EINT41 function possible when input)
0	1	VLC1 Function
1	0	AN3 Function
1	1	reserved

P4FSRL[1:0] P40 Function Select

P4FSRL1	P4FSRL0	Description
0	0	I/O Port (EINT40 function possible when input)
0	1	VLC0 Function
1	0	AN4 Function
1	1	reserved

6.7 P5 port

6.7.1 P5 port description

P5 is a 4-bit I/O port. P5 control registers consist of P5 data register (P5), P5 direction register (P5IO) and P5 pull-up resistor selection register (P5PU). Refer to the port function selection registers for the P5 function selection.

6.7.2 Register description for P5

P5 (P5 Data Register): B0H

7	6	5	4	3	2	1	0
–	–	–	–	P53	P52	P51	P50
–	–	–	–	R/W	R/W	R/W	R/W

Initial value: 00H

P5[3:0]	I/O Data

P5IO (P5 Direction Register): D1H

7	6	5	4	3	2	1	0
–	–	–	–	P53IO	P52IO	P51IO	P50IO
–	–	–	–	R/W	R/W	R/W	R/W

Initial value: 00H

P5IO[3:0]	P5 Data I/O Direction	
	0	Input
	1	Output

P5PU (P5 Pull-up Resistor Selection Register): 95H

7	6	5	4	3	2	1	0
–	–	–	–	P53PU	P52PU	P51PU	P50PU
–	–	–	–	R/W	R/W	R/W	R/W

Initial value: 00H

P5PU[3:0]	Configure Pull-up Resistor of P5 Port	
	0	Disable
	1	Enable

P5FSR (Port 5 Function Selection Register): FFH

7	6	5	4	3	2	1	0
P5FSR7	P5FSR6	P5FSR5	P5FSR4	P5FSR3	P5FSR2	P5FSR1	P5FSR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

P5FSR[7:6] P53 Function Select

P5FSR7	P5FSR6	Description
0	0	I/O Port
0	1	reserved
1	0	reserved
1	1	BUZ0 Function

P5FSR[5:4] P52 Function Select

P5FSR5	P5FSR4	Description
0	0	I/O Port
0	1	reserved
1	0	reserved
1	1	reserved

P5FSR[3:2] P51 Function Select

P5FSR3	P5FSR2	Description
0	0	I/O Port
0	1	XOUT2/SXOUT Function
1	0	reserved
1	1	reserved

P5FSR[1:0] P50 Function Select

P5FSR1	P5FSR0	Description
0	0	I/O Port
0	1	XIN2/SXIN Function
1	0	reserved
1	1	reserved

NOTE: Refer to the configure option for the P52/RESETB.

7 Interrupt controller

A96G150 supports up to 23 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 A96G150 has following features:

- Request receive from the 23 interrupt sources
- 6 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 A96G150 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 10 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.

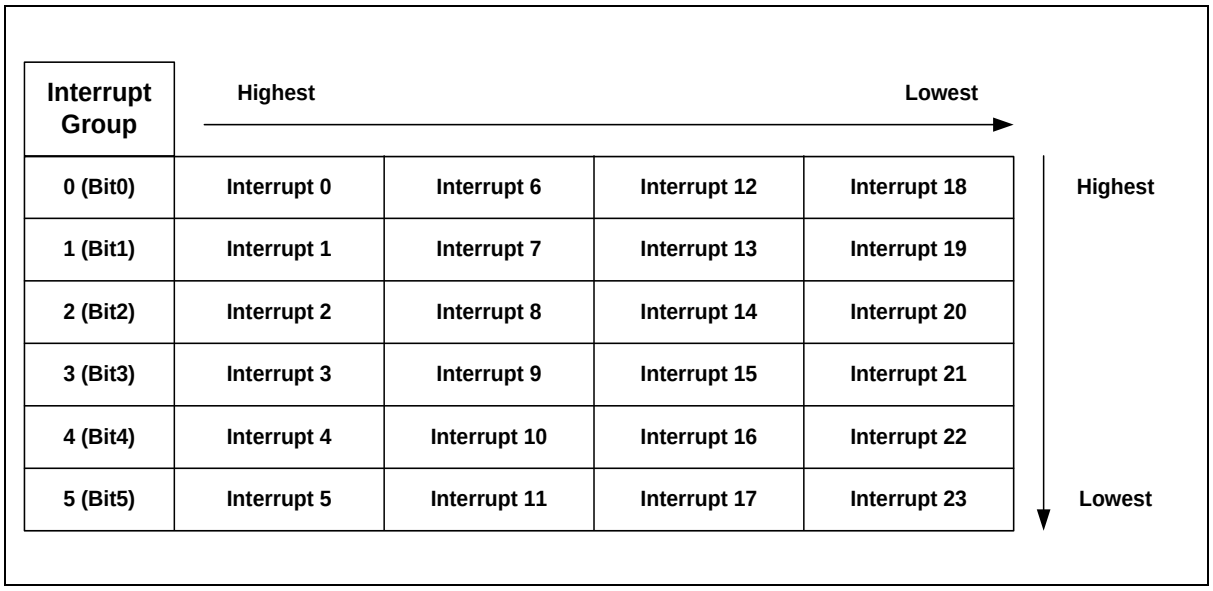


Figure 10. Interrupt Group Priority Level

7.1 External interrupt

External interrupts on INT0, INT1, INT5, INT6 and INT11 pins receive various interrupt requests depending on the external interrupt polarity 0 high/low register (EIPOL0H/L) and external interrupt polarity 1 register (EIPOL1) as shown in Figure 11.

Each external interrupt source has enable/disable bits. External interrupt flag 0 register (EIFLAG0) and external interrupt flag 1 register 1 (EIFLAG1) indicate status of external interrupts.

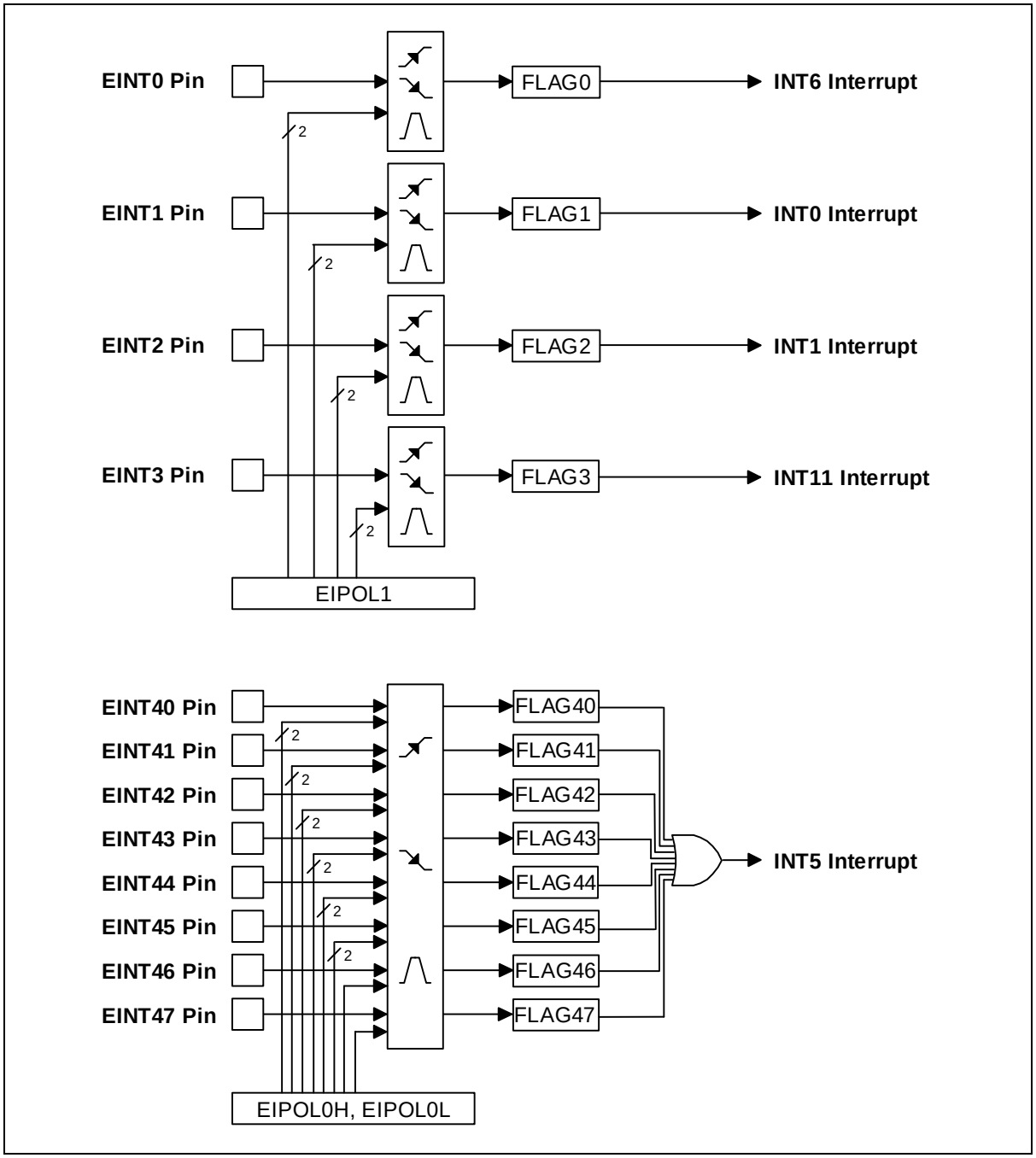
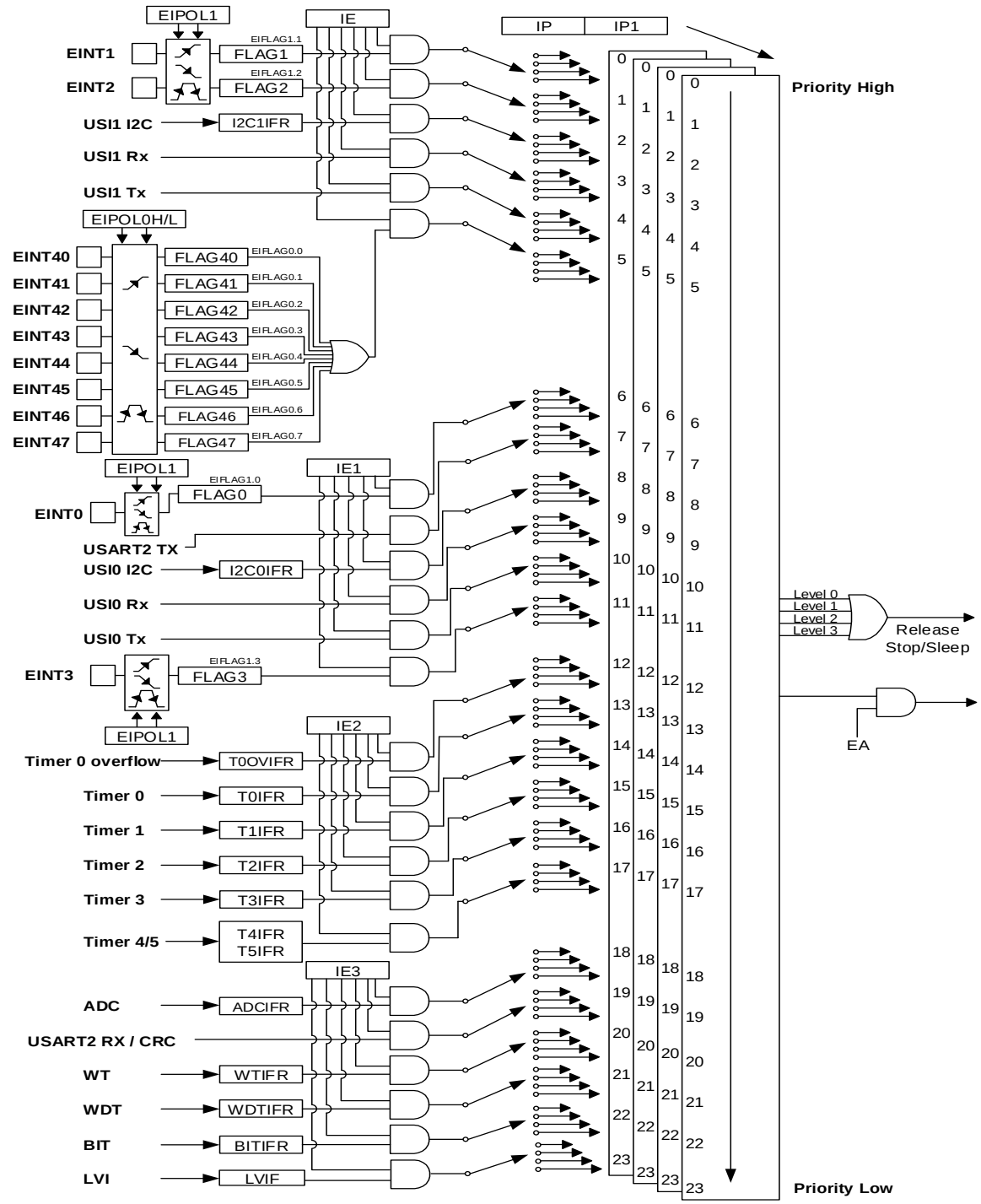


Figure 11. External Interrupt Description

7.2 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 12. Interrupt Controller Block Diagram

7.3 Interrupt vector table

Interrupt controller of A96G150 supports 24 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 1	INT0	IE.0	1	Maskable	0003H
External Interrupt 2	INT1	IE.1	2	Maskable	000BH
USI1 I2C Interrupt	INT2	IE.2	3	Maskable	0013H
USI1 Rx Interrupt	INT3	IE.3	4	Maskable	001BH
USI1 Tx Interrupt	INT4	IE.4	5	Maskable	0023H
External Interrupt 40 - 47	INT5	IE.5	6	Maskable	002BH
External Interrupt 0	INT6	IE1.0	7	Maskable	0033H
USART2 TX Interrupt	INT7	IE1.1	8	Maskable	003BH
USI0 I2C Interrupt	INT8	IE1.2	9	Maskable	0043H
USI0 Rx Interrupt	INT9	IE1.3	10	Maskable	004BH
USI0 Tx Interrupt	INT10	IE1.4	11	Maskable	0053H
External Interrupt 3	INT11	IE1.5	12	Maskable	005BH
T0 Overflow Interrupt	INT12	IE2.0	13	Maskable	0063H
T0 Match Interrupt	INT13	IE2.1	14	Maskable	006BH
T1 Match Interrupt	INT14	IE2.2	15	Maskable	0073H
T2 Match Interrupt	INT15	IE2.3	16	Maskable	007BH
T3 Match Interrupt	INT16	IE2.4	17	Maskable	0083H
T4/T5 Match Interrupt	INT17	IE2.5	18	Maskable	008BH
ADC Interrupt	INT18	IE3.0	19	Maskable	0093H
USART2 RX / CRC Interrupt	INT19	IE3.1	20	Maskable	009BH
WT Interrupt	INT20	IE3.2	21	Maskable	00A3H
WDT Interrupt	INT21	IE3.3	22	Maskable	00ABH
BIT Interrupt	INT22	IE3.4	23	Maskable	00B3H
LVI Interrupt	INT23	IE3.5	24	Maskable	00BBH

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.

7.4 Interrupt sequence

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

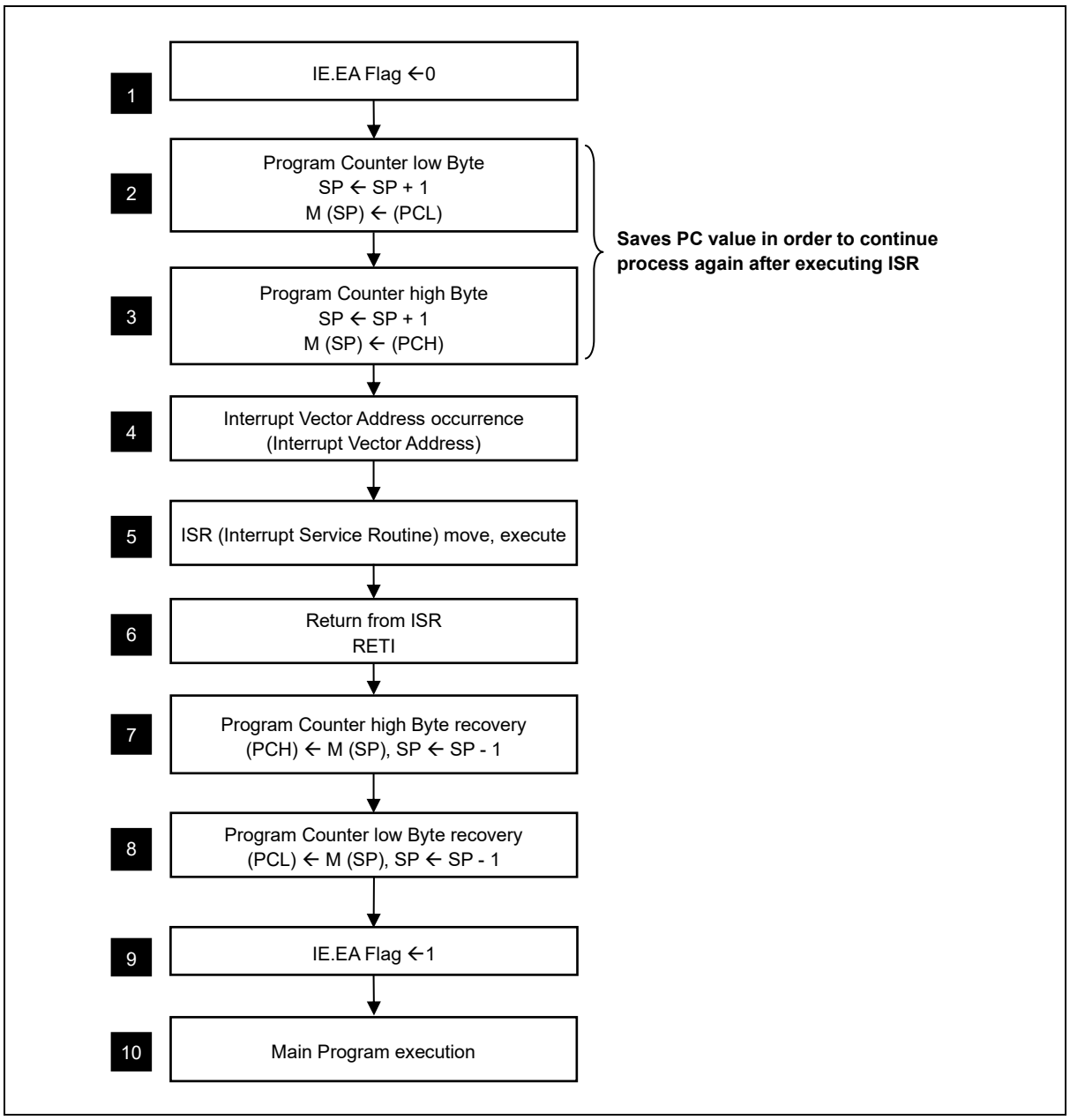


Figure 13. Interrupt Sequence Flow

7.5 Effective timing after controlling interrupt bit

Case A in Figure 14 shows the effective time after controlling Interrupt Enable Registers (IE, IE1, IE2, and IE3).

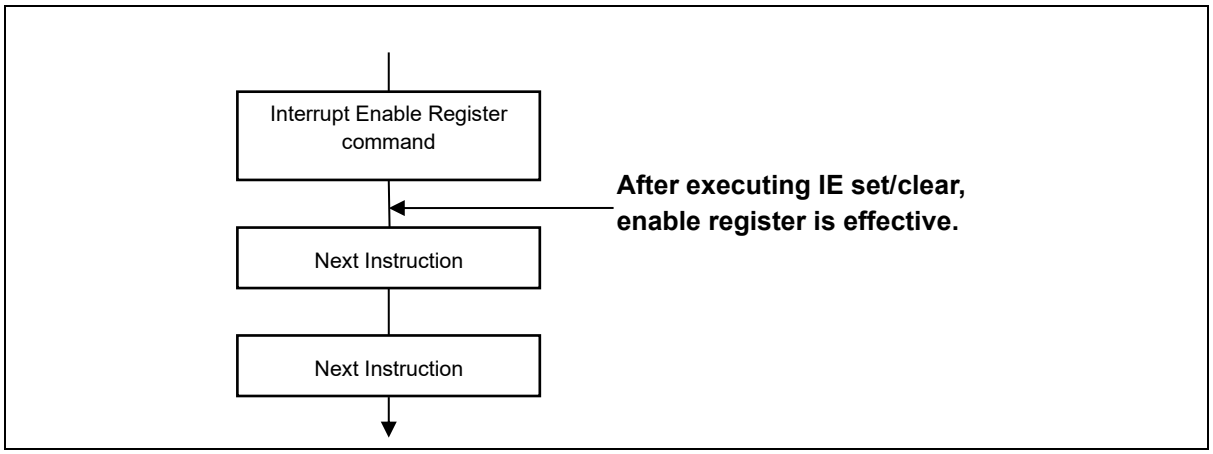


Figure 14. Effective Timing of Interrupt Enable Register

Case B in Figure 15 shows the effective time after controlling Interrupt Flag Registers.

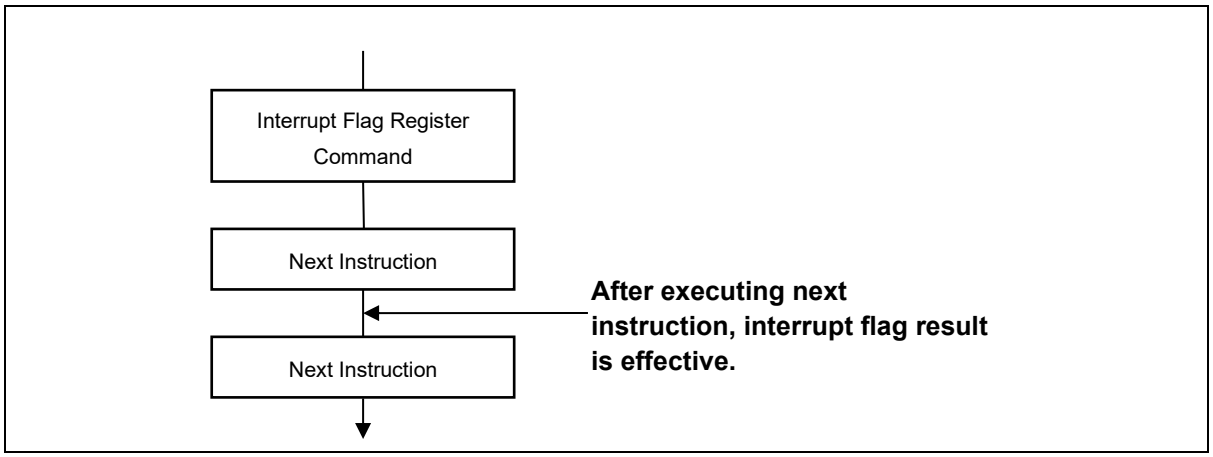


Figure 15. Effective Timing of Interrupt Flag Register

7.6 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.

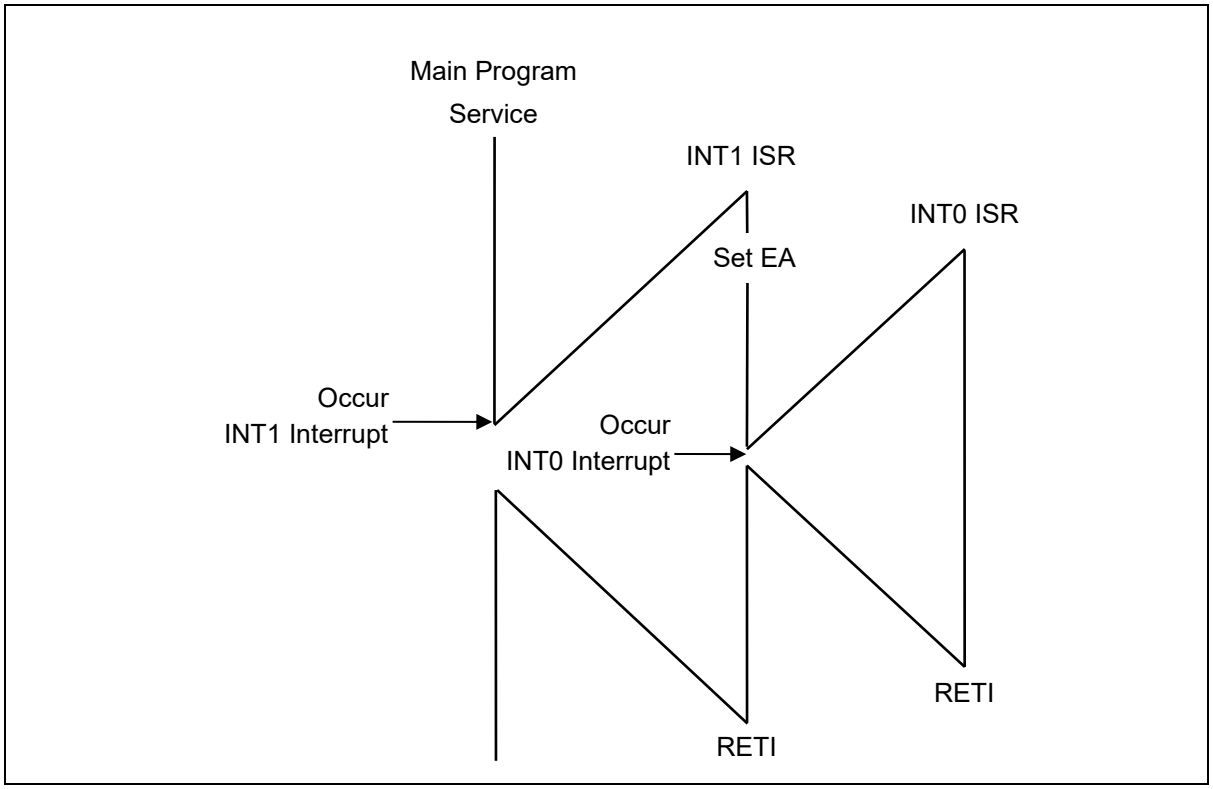


Figure 16. Effective Timing of Multi-interrupt

Figure 16 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.

7.7 Interrupt enable accept timing

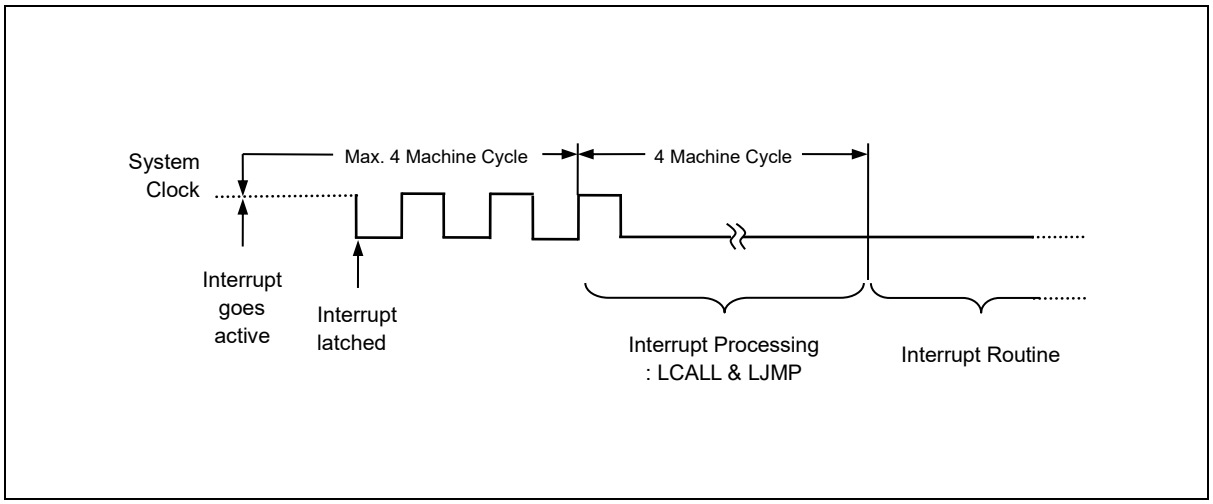


Figure 17. Interrupt Response Timing Diagram

7.8 Interrupt service routine address

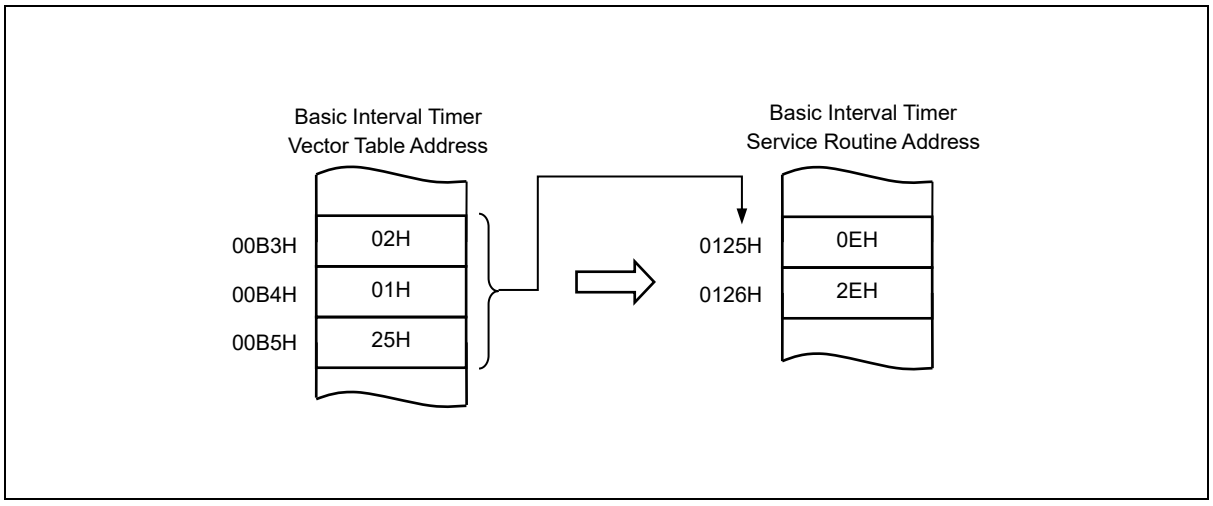


Figure 18. Correspondence between Vector Table Address and the Entry Address of ISR

7.9 Saving/restore general purpose registers

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

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

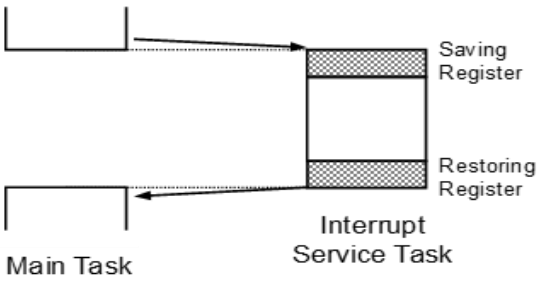
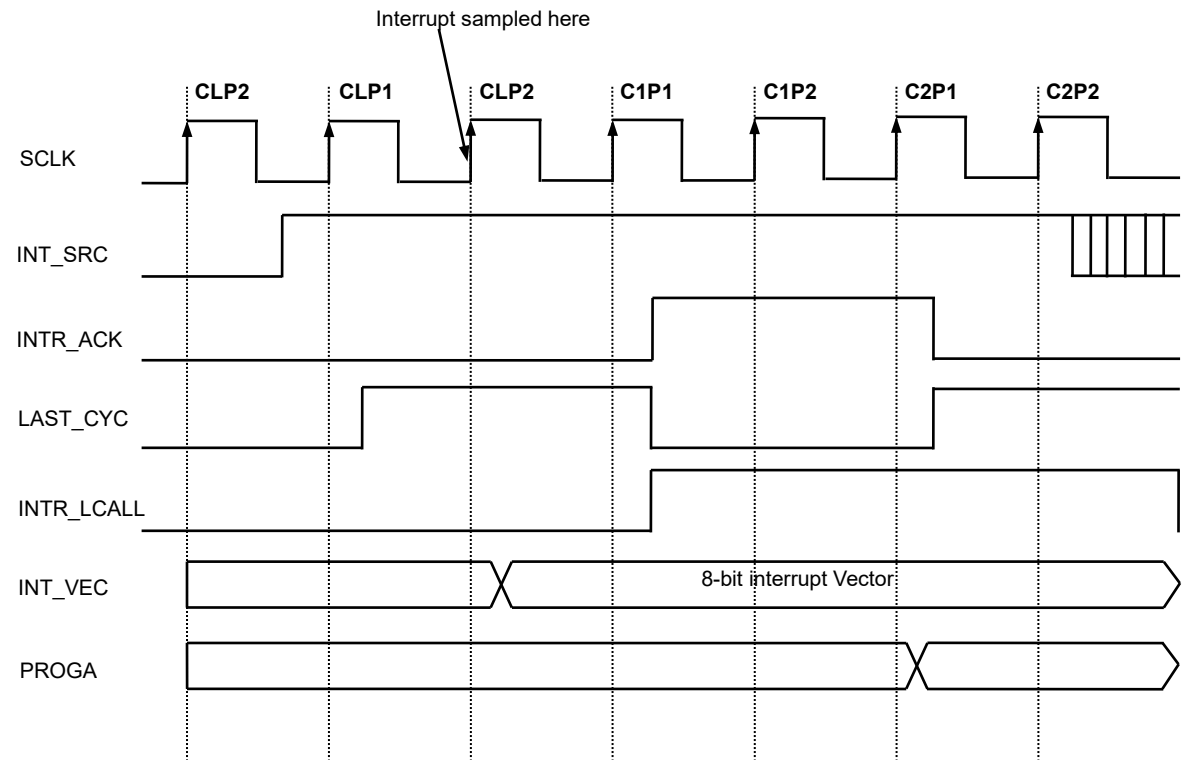


Figure 19. Saving/Restore Process Diagram and Sample Source

7.10 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 20. 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.

7.11 Interrupt register overview

7.11.1 Interrupt Enable register (IE, IE1, IE2, and IE3)

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

7.11.2 Interrupt priority register (IP and IP1)

24 interrupts are divided into 6 groups which have 4 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.

7.11.3 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.

7.11.4 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.

7.11.5 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
IE3	ABH	R/W	00H	Interrupt Enable Register 3
IP	B8H	R/W	00H	Interrupt PriorityRegister
IP1	F8H	R/W	00H	Interrupt PriorityRegister 1
EIFLAG0	C0H	R/W	00H	External Interrupt Flag 0 Register
EIPOL0L	A4H	R/W	00H	External Interrupt Polarity 0 Low Register
EIPOL0H	A5H	R/W	00H	External Interrupt Polarity 0 High Register
EIFLAG1	A6H	R/W	00H	External Interrupt Flag 1 Register
EIPOL1	A7H	R/W	00H	External Interrupt Polarity 1 Register

7.11.6 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 External Interrupt 40 ~ 47 (EINT40 ~ EINT47)	
	0	Disable
	1	Enable
INT4E	Enable or Disable USI1 Tx Interrupt	
	0	Disable
	1	Enable
INT3E	Enable or Disable USI1 Rx Interrupt	
	0	Disable
	1	Enable
INT2E	Enable or Disable USI1 I2C Interrupt	
	0	Disable
	1	Enable
INT1E	Enable or Disable External Interrupt 2(EINT2)	
	0	Disable
	1	Enable
INT0E	Enable or Disable External Interrupt 1 (EINT1)	
	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 External Interrupt 3 (EINT3)	
	0	Disable
	1	Enable
INT10E	Enable or Disable USI0Tx Interrupt	
	0	Disable
	1	Enable
INT9E	Enable or Disable USI0 Rx Interrupt	
	0	Disable
	1	Enable
INT8E	Enable or Disable USI0 I2C Interrupt	
	0	Disable
	1	Enable
INT7E	Enable or Disable USART2 TX Interrupt	
	0	Disable
	1	Enable
INT6E	Enable or Disable External Interrupt 0 (EINT0)	
	0	Disable
	1	Enable

IE2 (Interrupt Enable Register 2): AAH

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

Initial value: 00H

INT17E	Enable or Disable Timer 4/5 Match Interrupt	
	0	Disable
	1	Enable
INT16E	Enable or Disable Timer 3 Match Interrupt	
	0	Disable
	1	Enable
INT15E	Enable or Disable Timer 2 Match Interrupt	
	0	Disable
	1	Enable
INT14E	Enable or Disable Timer 1 Match Interrupt	
	0	Disable
	1	Enable
INT13E	Enable or Disable Timer 0 Match Interrupt	
	0	Disable
	1	Enable
INT12E	Enable or Disable Timer 0 Overflow Interrupt	
	0	Disable
	1	Enable

IE3 (Interrupt Enable Register 3): ABH

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

Initial value: 00H

INT23E	Enable or Disable LVI Interrupt	
	0	Disable
	1	Enable
INT22E	Enable or Disable BIT Interrupt	
	0	Disable
	1	Enable
INT21E	Enable or Disable WDT Interrupt	
	0	Disable
	1	Enable
INT20E	Enable or Disable WT Interrupt	
	0	Disable
	1	Enable
INT19E	Enable or Disable USART2 RX / CRC Interrupt	
	0	Disable
	1	Enable
INT18E	Enable or Disable ADC 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
FLAG47	FLAG46	FLAG45	FLAG44	FLAG43	FLAG42	FLAG41	FLAG40
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

EIFLAG0[7:0] When an External Interrupt 40-47 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 Interrupt 40 ~ 47 not occurred

1 External Interrupt 40 ~ 47 occurred

EIPOL0H (External Interrupt Polarity 0High Register): A5H

7	6	5	4	3	2	1	0
POL47		POL46		POL45		POL44	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

EIPOL0H[7:0] External interrupt (EINT47, EINT46, EINT45, EINT44) 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 = 44, 45, 46 and 47

EIPOL0L (External Interrupt Polarity 0Low Register): A4H

7	6	5	4	3	2	1	0
POL43		POL42		POL41		POL40	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

EIPOL0L[7:0] External interrupt (EINT40, EINT41, EINT42, EINT43) 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 = 40, 41, 42 and 43

EIFLAG1 (External Interrupt Flag 1 Register): A6H

7	6	5	4	3	2	1	0
T0OVIFR	T0IFR	–	–	FLAG3	FLAG2	FLAG1	FLAG0
R/W	R/W	–	–	R/W	R/W	R/W	R/W

Initial value: 00H

T0OVIFR — When T0 overflow 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 overflow Interrupt no generation

1 T0 overflow Interrupt generation

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

EIFLAG1[3:0] — When an External Interrupt (EINT0~EINT3) is occurred, the flag becomes ‘1’. The flag is cleared by writing ‘0’ to the bit or automatically cleared by INT_ACK signal. Writing “1” has no effect.

0 External Interrupt not occurred

1 External Interrupt occurred

EIPOL1 (External Interrupt Polarity 1 Register): A7H

7	6	5	4	3	2	1	0
POL3		POL2		POL1		POL0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

EIPOL1[7:0] — External interrupt (EINT0, EINT1, EINT2, EINT3) 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, 2 and 3

8 Clock generator

As shown in Figure 21, a clock generator produces basic clock pulses which provide a system clock for CPU and peripheral hardware.

It contains main/sub-frequency clock oscillator. The main/sub clock can operate easily by attaching a crystal between the XIN1/XIN2/SXIN and XOUT1/XOUT2/SXOUT pin, respectively. The main/sub clock can be also obtained from the external oscillator. For this, it is necessary to place external clock signal into the XIN1/XIN2/SXIN pin and open XOUT1/XOUT2/SXOUT pin.

Default system clock is 1MHz 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)
- Main crystal oscillator (4~12MHz)
- Sub-crystal Oscillator (32.768kHz)
- Internal LSIRC oscillator (128kHz)

8.1 Clock generator block diagram

In this section, a clock generator of A96G150 is described in a block diagram.

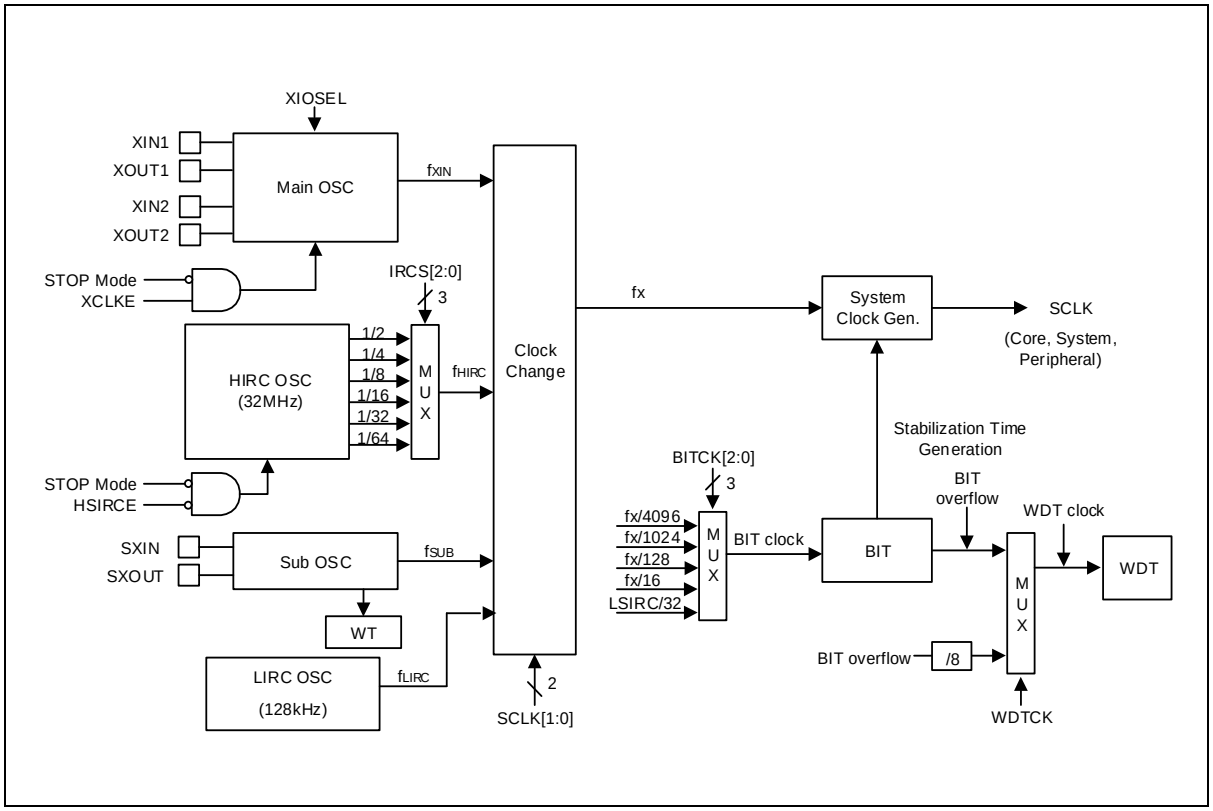


Figure 21. Clock Generator Block Diagram

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
XTFLSR	1038H	R/W	00H	Main Crystal OSC Filter Selection 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	External Main OSC (f_{XIN}) for system clock
1	0	External Sub OSC (f_{SUB}) for system clock
1	1	Internal 128kHz RC OSC (f_{LSIRC}) for system clock

OSCCR (Oscillator Control Register): C8H

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

Initial value: 28H

XIOSEL External Main OSC channel Select Register

0 Selected XIN1/XOUT1 (P07/P06)

1 Selected XIN2/XOUT2 (P50/P51)

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)
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

XCLKE Control the Operation of the External Main Oscillator

0 Disable operation of X-TAL

1 Enable operation of X-TAL

SCLKE Control the Operation of the External Sub Oscillator

0 Disable operation of SX-TAL

1 Enable operation of SX-TAL

XTFLSR (Main Crystal OSC Filter Selection Register): 1038H

7	6	5	4	3	2	1	0
NFSEL1	NFSEL0	MX_FIL_DIS	MX_ISEL1	MX_ISEL0	SUB_FIL_DIS	SUB_ISEL1	SUB_ISEL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

NFSEL[1:0] Noise Filter Selective Option

NFSEL1	NFSEL0	Description
0	0	18ns (Default, 12MHz)
0	1	22ns (12MHz)
1	0	26ns (8MHz)
1	1	30ns (4MHz)

MX_FIL_DIS Main X-TAL noise canceller selection.

0	Using noise filter
1	Bypass noise filter

MX_ISEL[1:0] Current selective option for MX-TAL

MX_ISEL1	MX_ISEL0	Description
0	0	HIGH (~12M)
0	1	MID-HIGH (8~12M)
1	0	MID-LOW (4~8M)
1	1	LOW (~4M)

SUB_FIL_DIS SUB X-TAL noise canceller selection.

0	Using noise filter
1	Bypass noise filter

SUB_ISEL[1:0] Current selective option for SUB-TAL

SUB_ISEL1	SUB_ISEL0	Description
0	0	Low
0	1	Mid-Low
1	0	Mid-High
1	1	High (When using fast Start-up)

NOTES:

1. The External Main Oscillator Range (XRNS) should be changed while the system clock is selected as IRC
2. When using the SUB-TAL, it is recommended to set it to Current High and enable the SCLKE.

9 Basic Interval Timer (BIT)

A96G150 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 A96G150 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 A96G150 is described in a block diagram.

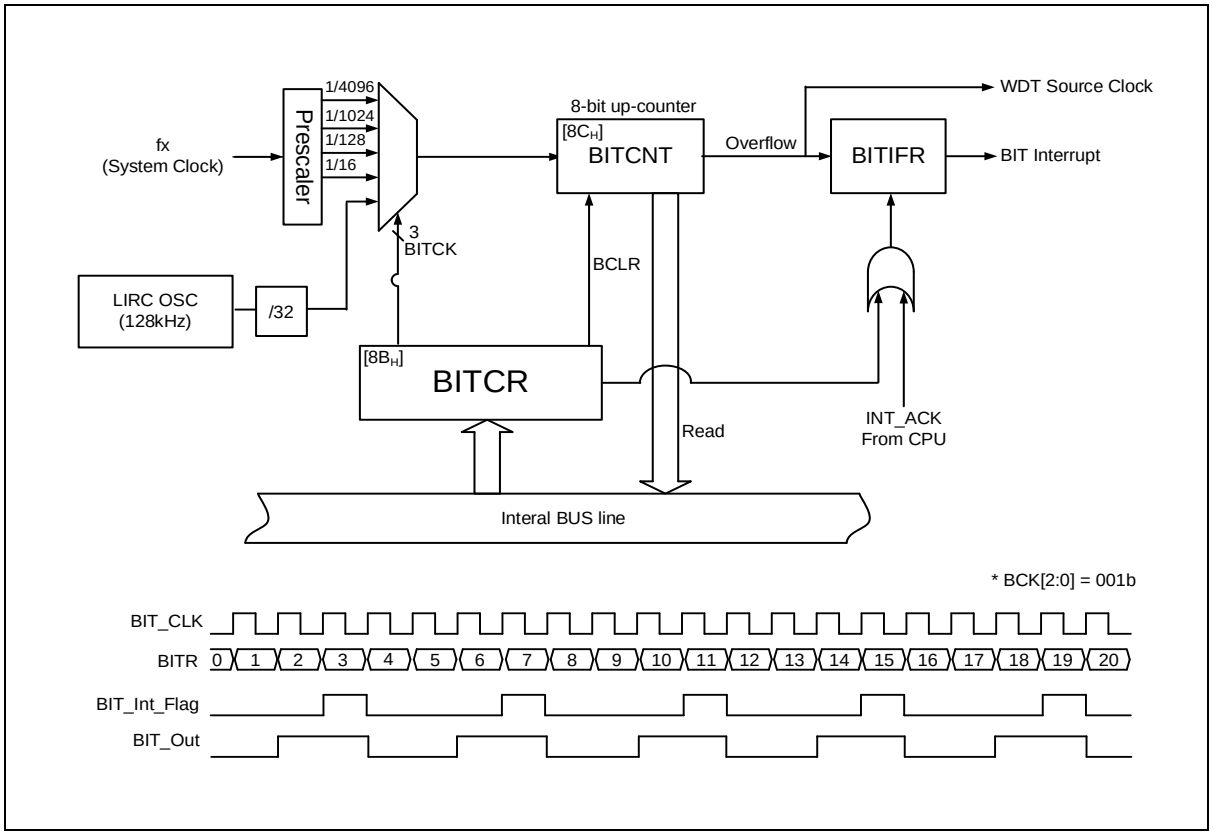


Figure 22. Basic Interval Timer Block Diagram

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 (WDT)

Watchdog timer (WDT) rapidly detects the CPU malfunction such as endless looping caused by noise or something like that, and resumes the CPU to the normal state. Watchdog timer signal for malfunction detection can be used as either a CPU reset or an interrupt request. When the watchdog timer is not being used for malfunction detection, it can be used as a timer to generate an interrupt at fixed intervals.

It is possible to use free running 8-bit timer mode (WDTRSON='0') or watch dog timer mode (WDTRSON='1') by setting WDTCR[6] bit. If WDTCR[5] is set to '1', the WDT counter value is cleared and counts up. After 1 machine cycle, this bit is cleared to '0' automatically.

The WDT consists of an 8-bit binary counter and a watchdog timer data register. When value of the 8-bit binary counter is equal to the 8 bits of WDTCNT, an interrupt request flag is generated. This can be used as a watchdog timer interrupt or a reset signal of CPU in accordance with a bit WDTRSON.

Input clock source of the WDT is BIT overflow. An interval between watchdog timer interrupts is decided by BIT overflow period and WDTDR set value. The equation can be described as the followings:

- WDT Interrupt Interval = (BIT Interrupt Interval) X (WDTDR Value+1)

10.1 WDT interrupt timing waveform

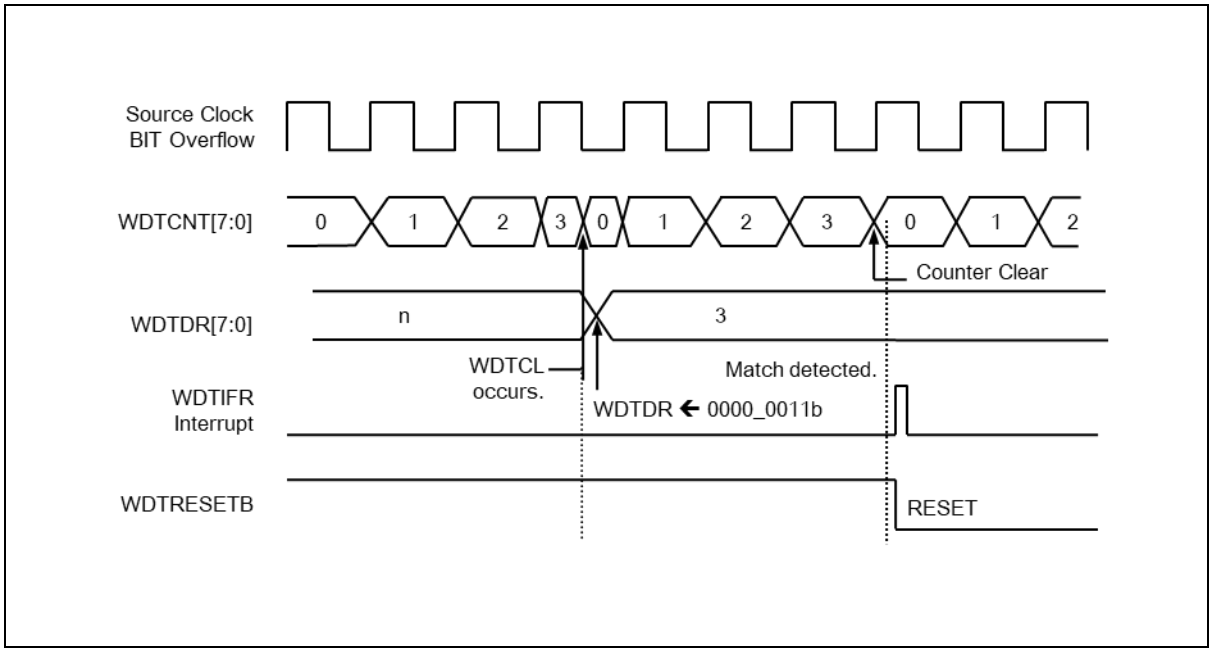


Figure 23. Watchdog Timer Interrupt Timing Waveform

10.2 WDT block diagram

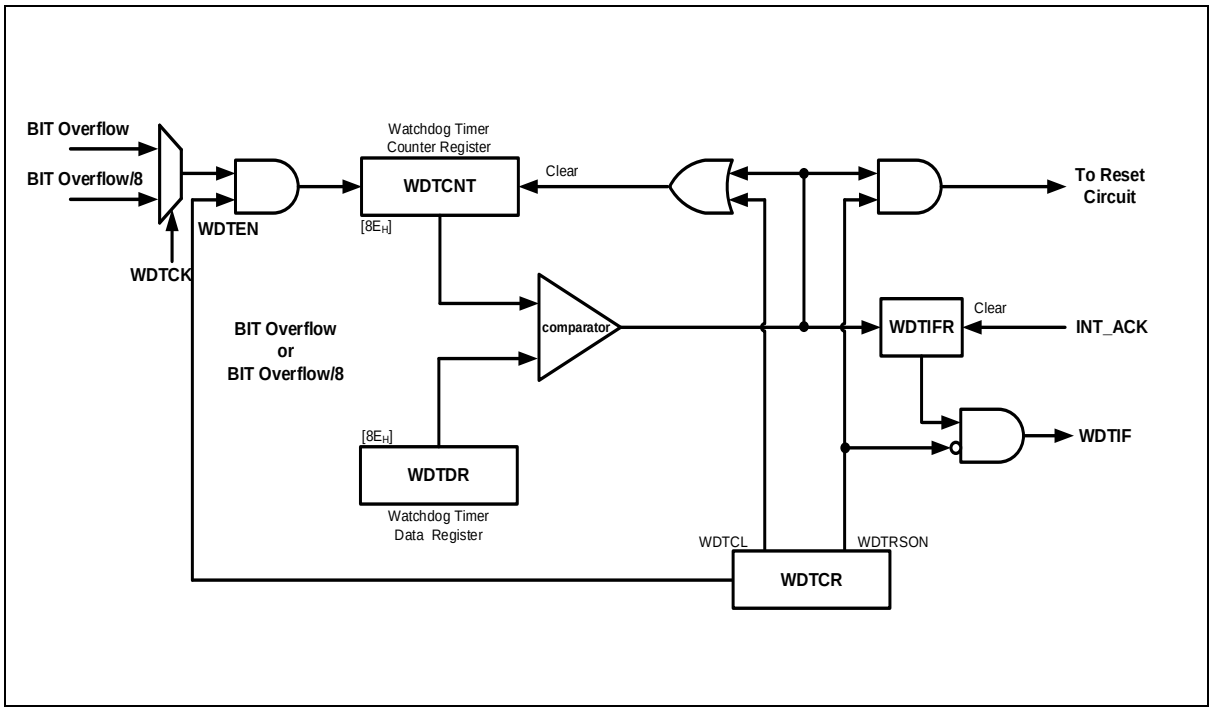


Figure 24. Watchdog Timer Block Diagram

10.3 Register map

Table 12. Watchdog Timer Register Map

Name	Address	Direction	Default	Description
WDTCNT	8EH	R	00H	Watch Dog Timer Counter Register
WDTDR	8EH	W	FFH	Watch Dog Timer Data Register
WDTCR	8DH	R/W	00H	Watch Dog Timer Control Register

10.4 Register description

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

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

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

7	6	5	4	3	2	1	0
WDTDR7	WDTDR6	WDTDR5	WDTDR4	WDTDR3	WDTDR2	WDTDR1	WDTDR0
W	W	W	W	W	W	W	W

Initial value: FFH

WDTDR[7:0] Set a period
WDT Interrupt Interval = (BIT Interrupt Interval) x (WDTDR Value+1)
NOTE: Do not write “0” in the WDTDR register.

WDTCR (Watch Dog Timer Control Register): 8DH

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

Initial value: 00H

WDTEN Control WDT Operation
0 Disable
1 Enable

WDTRSON Control WDT RESET Operation
0 Free Running 8-bit timer
1 Watch Dog Timer RESET ON

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

WDTCK Control WDT Clock Selection Bit
0 BIT overflow for WDT clock
1 BIT overflow/8 for WDT clock

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

11 Watch Timer (WT)

Watch timer (WT) has functions for RTC (Real Time Clock) operation. It is generally used for RTC design. WT consists of a clock source select circuit, a timer counter circuit, an output select circuit and watch timer control registers.

Prior to operate WT, a user needs to determine an input clock source and output interval, and to set WTEN to '1' in watch timer control register (WTCR). It is able to execute simultaneously or individually. To stop or reset WT, clear the WTEN bit in WTCR register.

Although CPU is in STOP mode, a sub clock can be alive so that WT continues its operation. Watch timer counter circuits may be composed of 21-bit counter which contains low 14-bit with binary counter and high 7-bit counter in order to increase resolution. In WTDR, it can control WT clear and set interval value at write time, and it can read 7-bit WT counter value at read time.

11.1 WT block diagram

In this section, watch timer of A96G150 is described in a block diagram.

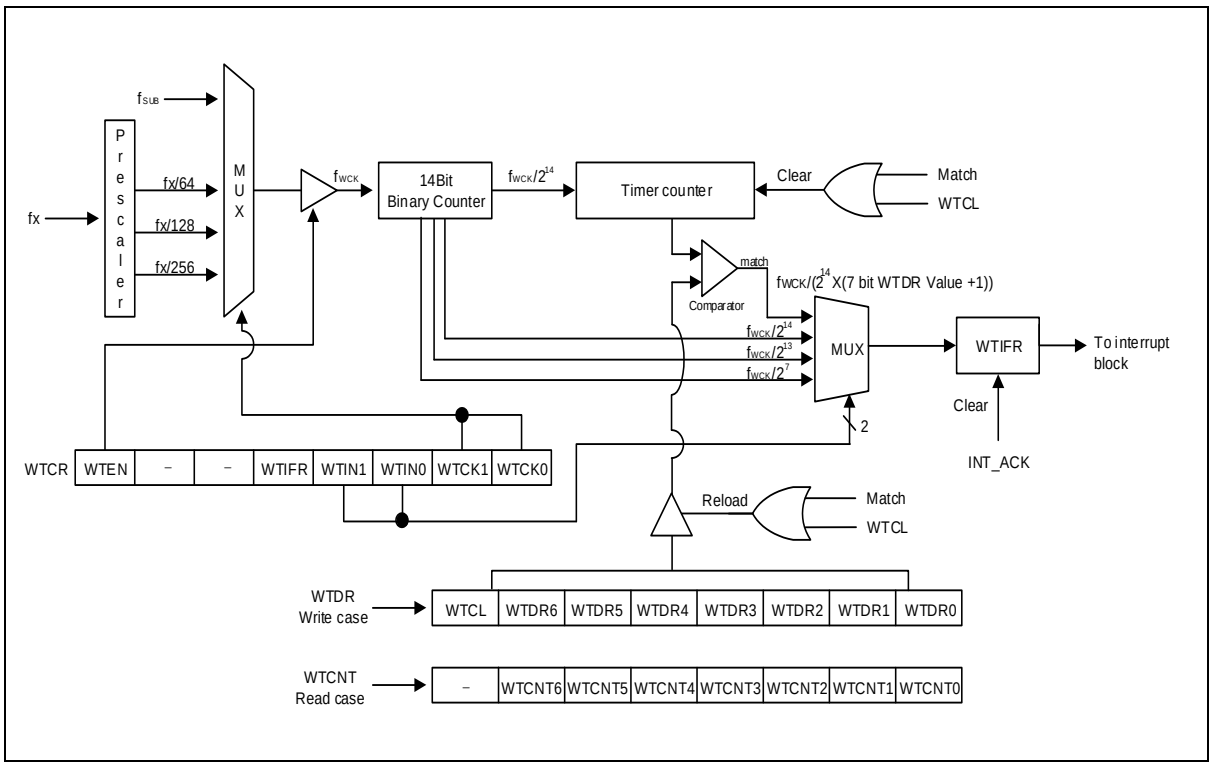


Figure 25. Watch Timer Block Diagram

11.2 Register map

Table 13. Watch Timer Register Map

Name	Address	Direction	Default	Description
WTCNT	89H	R	00H	Watch Timer Counter Register
WTDR	89H	W	7FH	Watch Timer Data Register
WTCR	96H	R/W	00H	Watch Timer Control Register

11.3 Watch Timer register description

WTCNT (Watch Timer Counter Register: Read Case): 89H

7	6	5	4	3	2	1	0
–	WTCNT6	WTCNT5	WTCNT4	WTCNT3	WTCNT2	WTCNT1	WTCNT0
–	R	R	R	R	R	R	R

Initial value: 00H

WTCNT[6:0] WT Counter

WTDR (Watch Timer Data Register: Write Case): 89H

7	6	5	4	3	2	1	0
WTCL	WTDR6	WTDR5	WTDR4	WTDR3	WTDR2	WTDR1	WTDR0
R/W	W	W	W	W	W	W	W

Initial value: 7FH

WTCL Clear WT Counter

0 Free Run

1 Clear WT Counter (auto clear after 1 Cycle)

NOTE: If the WTCL is set to ‘1’, the WTDR register is cleared. Therefore, when the WT Counter is cleared, the WTDR bits must be reset.

WTDR[6:0] Set WT period

WT Interrupt Interval=fwck/(2^14 x(7bit WTDR Value+1))

NOTE: Do not write “0” in the WTDR register.

WTCR (Watch Timer Control Register): 96H

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

Initial value: 00H

WTEN Control Watch Timer

0	Disable
1	Enable

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

0	WT Interrupt no generation
1	WT Interrupt generation

WTIN[1:0] Determine interrupt interval

WTIN1	WTIN0	Description
0	0	f_{WCK}/2^7
0	1	f_{WCK}/2^13
1	0	f_{WCK}/2^14
1	1	f_{WCK}/(2^14 x (7bit WTDR Value+1))

WTCK[1:0] Determine Source Clock

WTCK1	WTCK0	Description
0	0	f_{SUB}
0	1	f_X/256
1	0	f_X/128
1	1	f_X/64

NOTES:

1. f_X– System clock frequency (Where fx= 4.19MHz)
2. f_{SUB}– Sub clock oscillator frequency (32.768kHz)
3. f_{WCK}– Selected Watch timer clock

12 Timer 0/1/2/3/4/5

12.1 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 EINT1. 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

12.1.1 8-bit timer/counter mode

As shown in Figure 26, 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 P52IO bit.

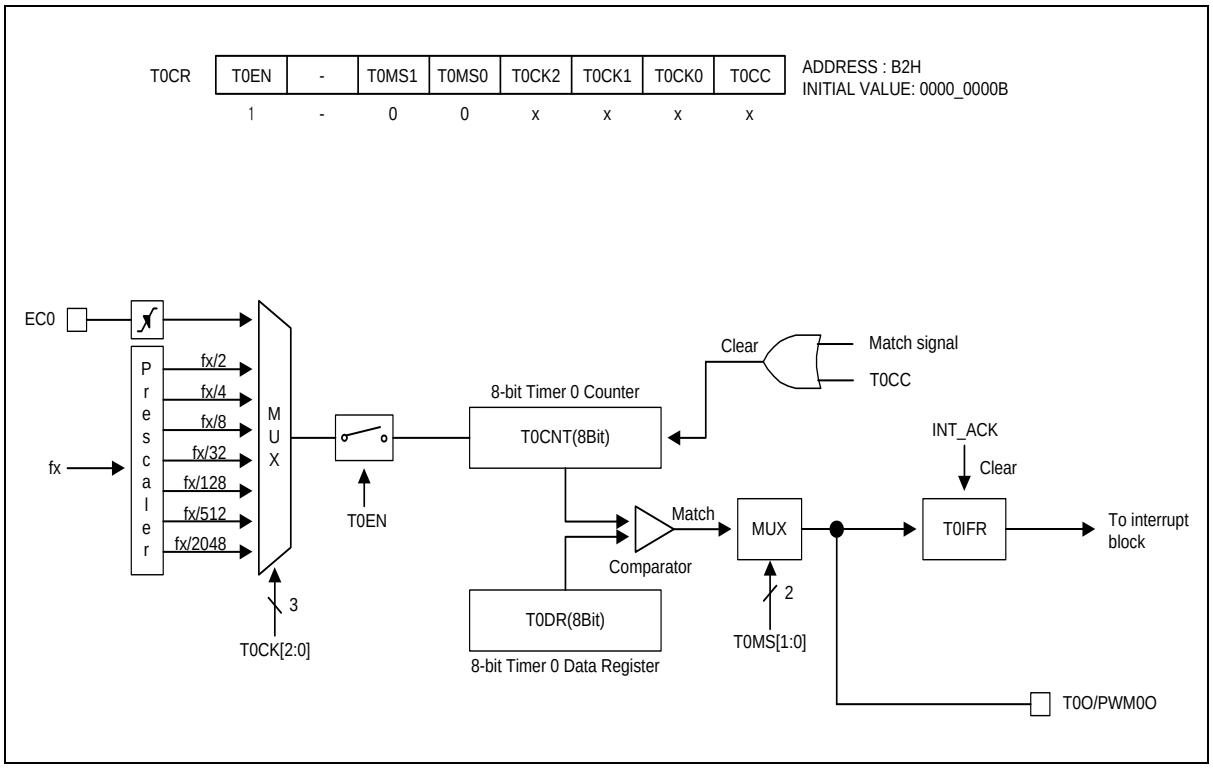


Figure 26. 8-bit Timer/Counter Mode for Timer 0

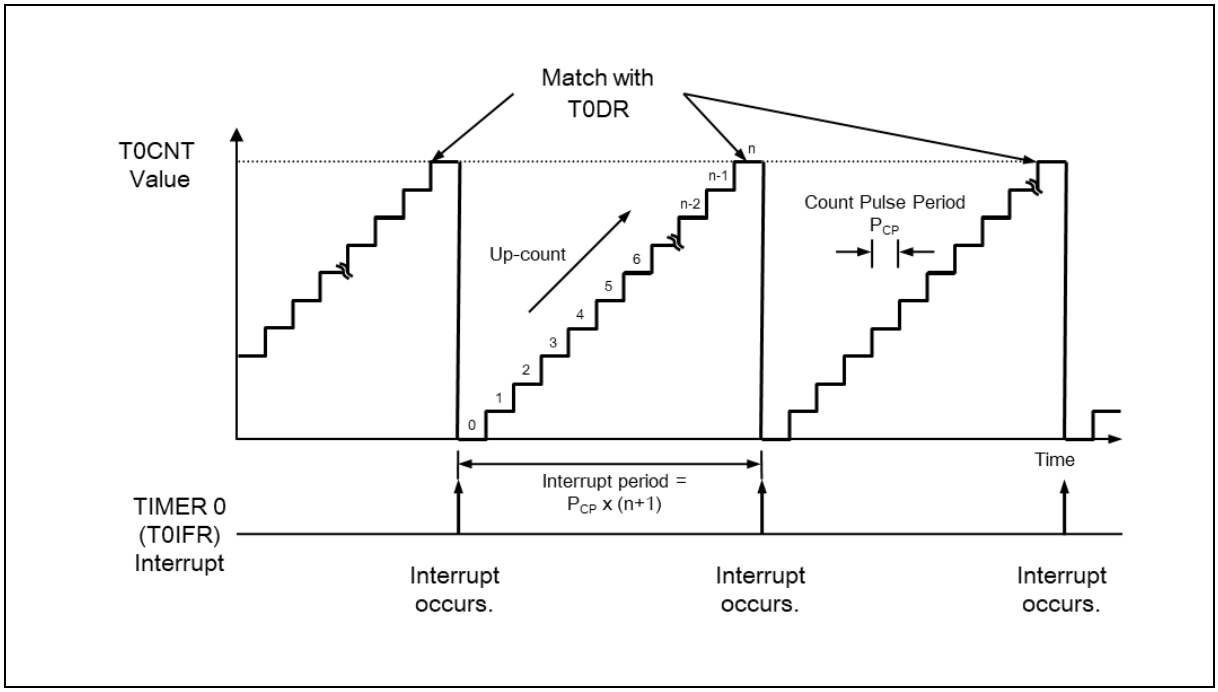


Figure 27. 8-bit Timer/Counter 0 Example

12.1.2 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 P0FSRL[7:6] 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.

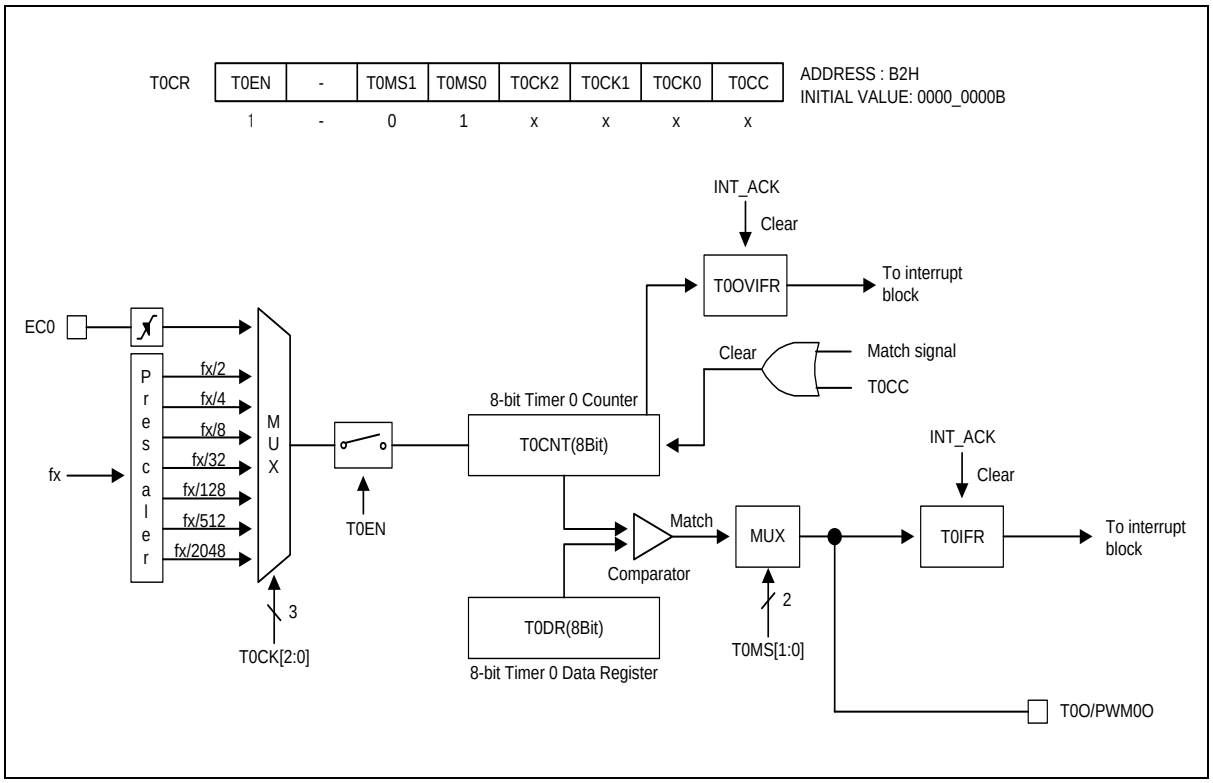


Figure 28. 8-bit PWM Mode for Timer 0

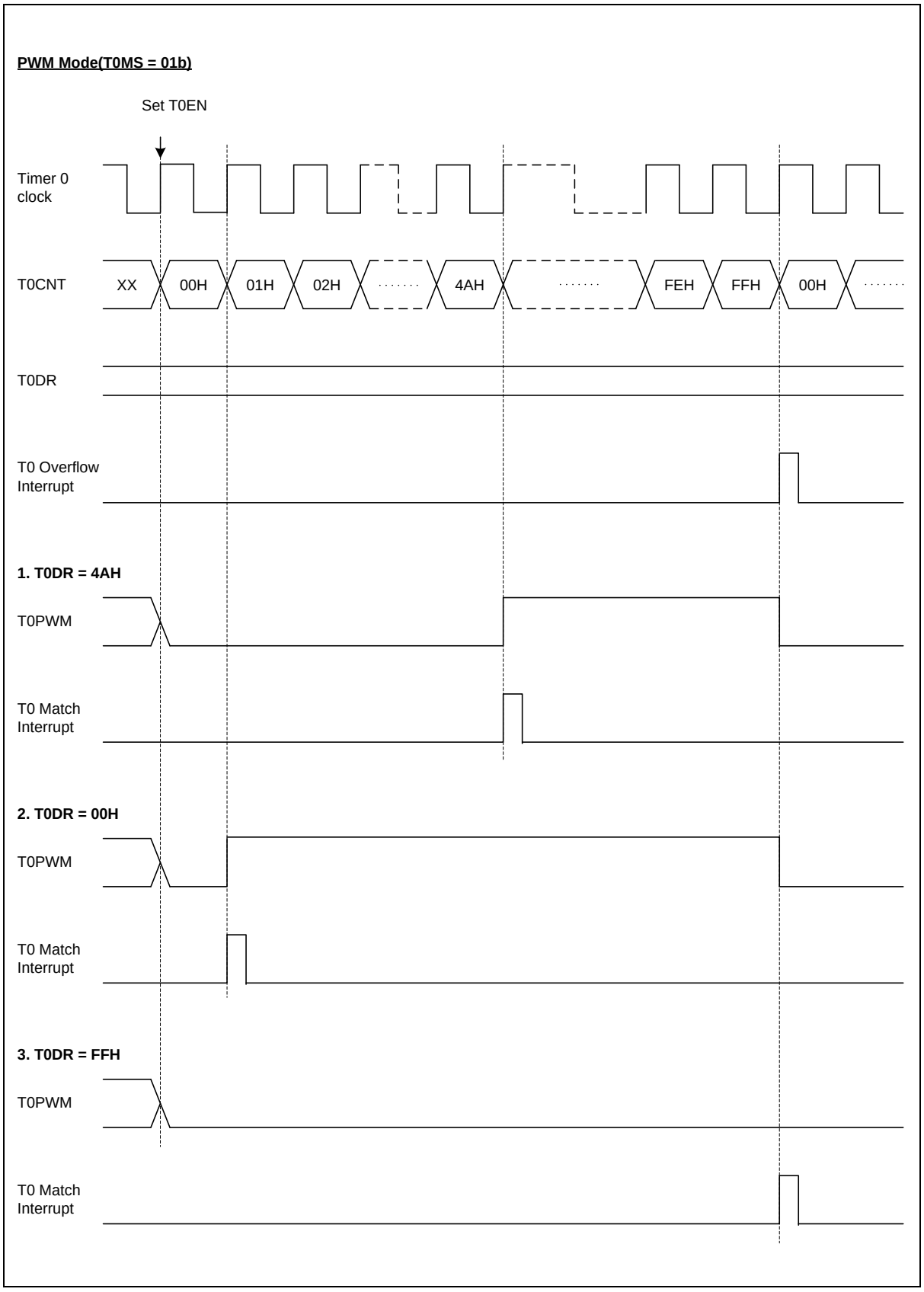


Figure 29. PWM Output Waveforms in PWM Mode for Timer 0

12.1.3 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 EIPOL1 registers setting, the external interrupt EINT1 function is chosen. Of course, the EINT1 pin must be set to an input port.

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

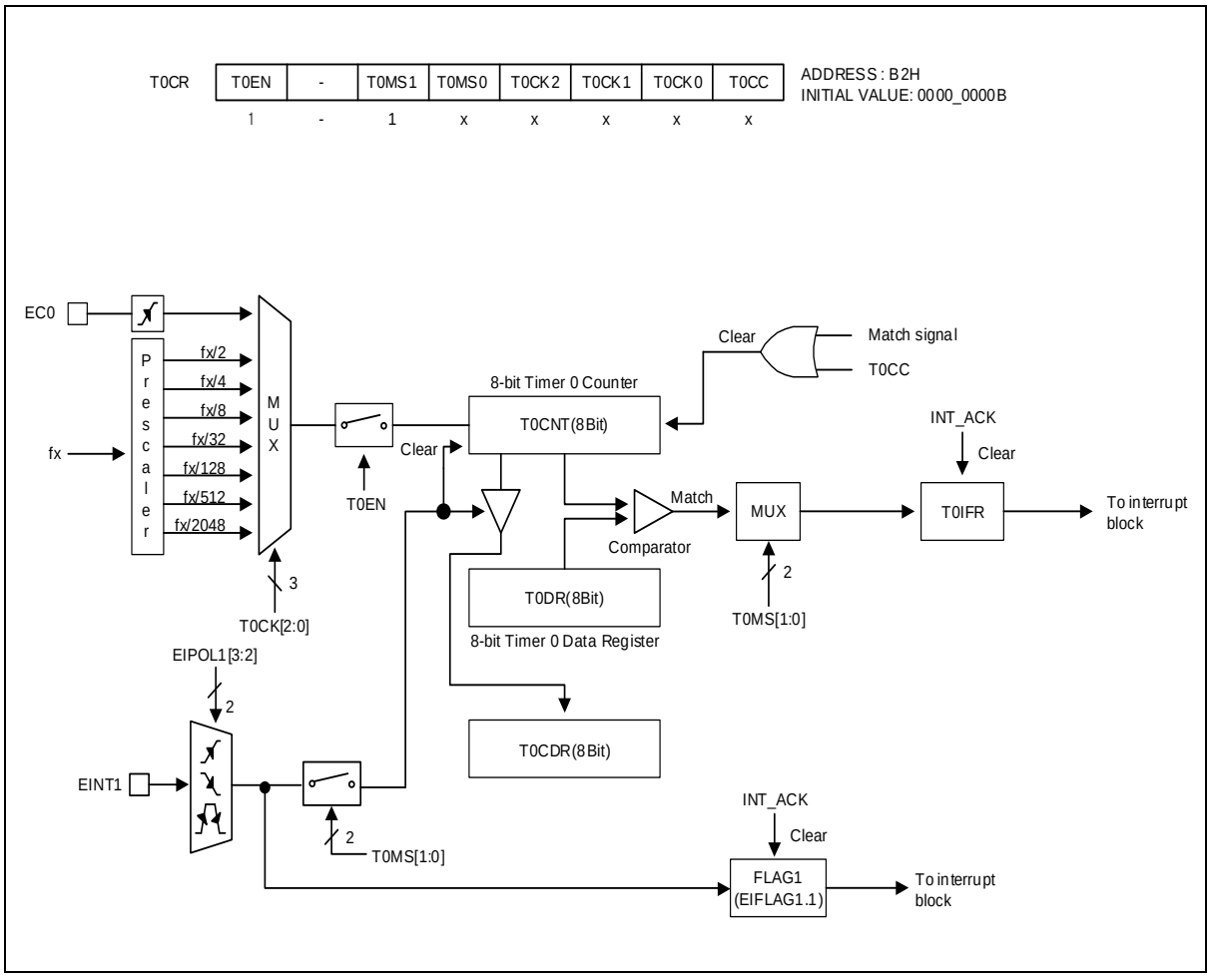


Figure 30. 8-bit Capture Mode for Timer 0

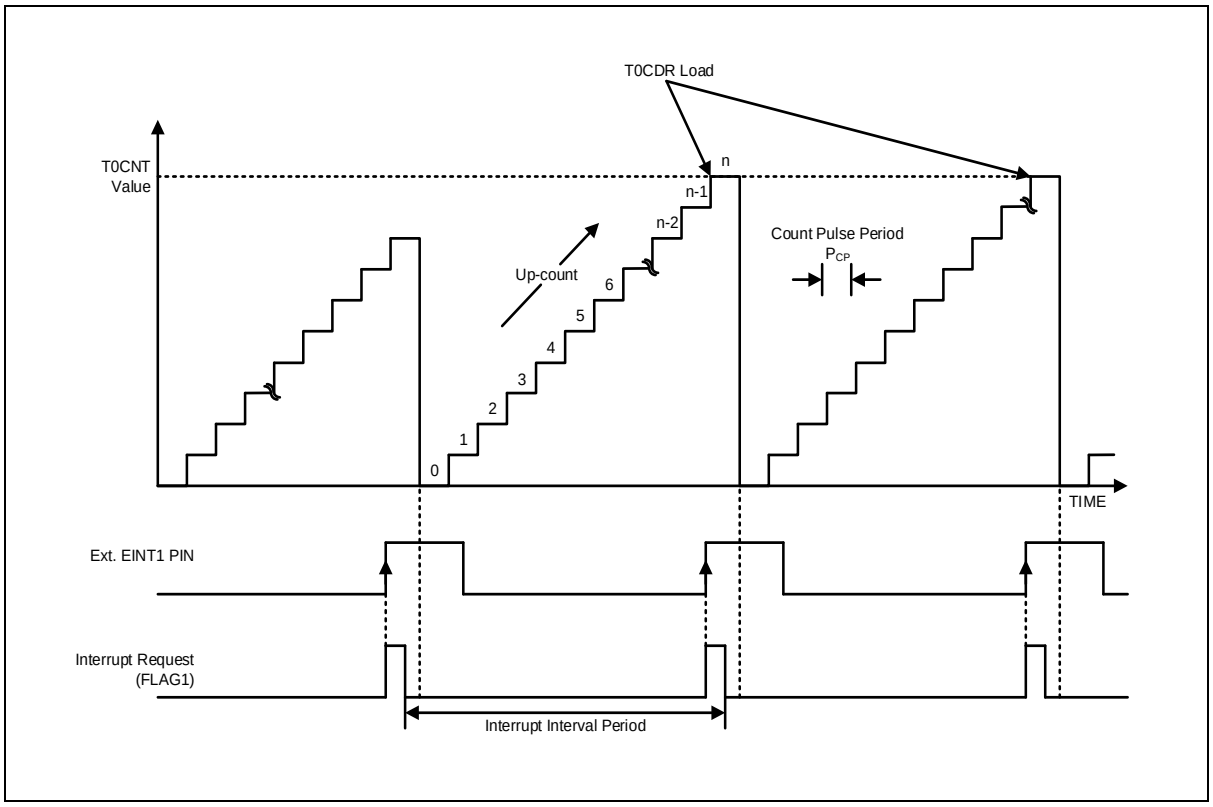


Figure 31. Input Capture Mode Operation for Timer 0

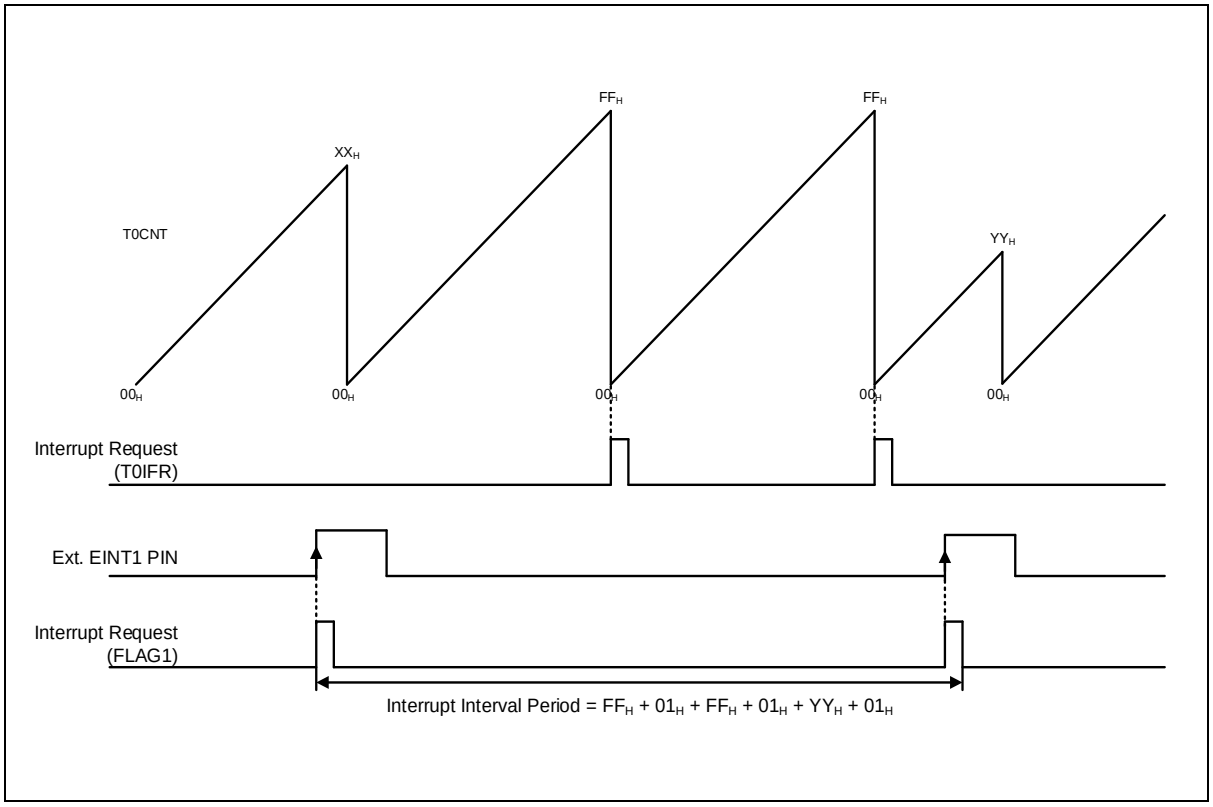


Figure 32. Express Timer Overflow in Capture Mode

12.1.4 Timer 0 block diagram

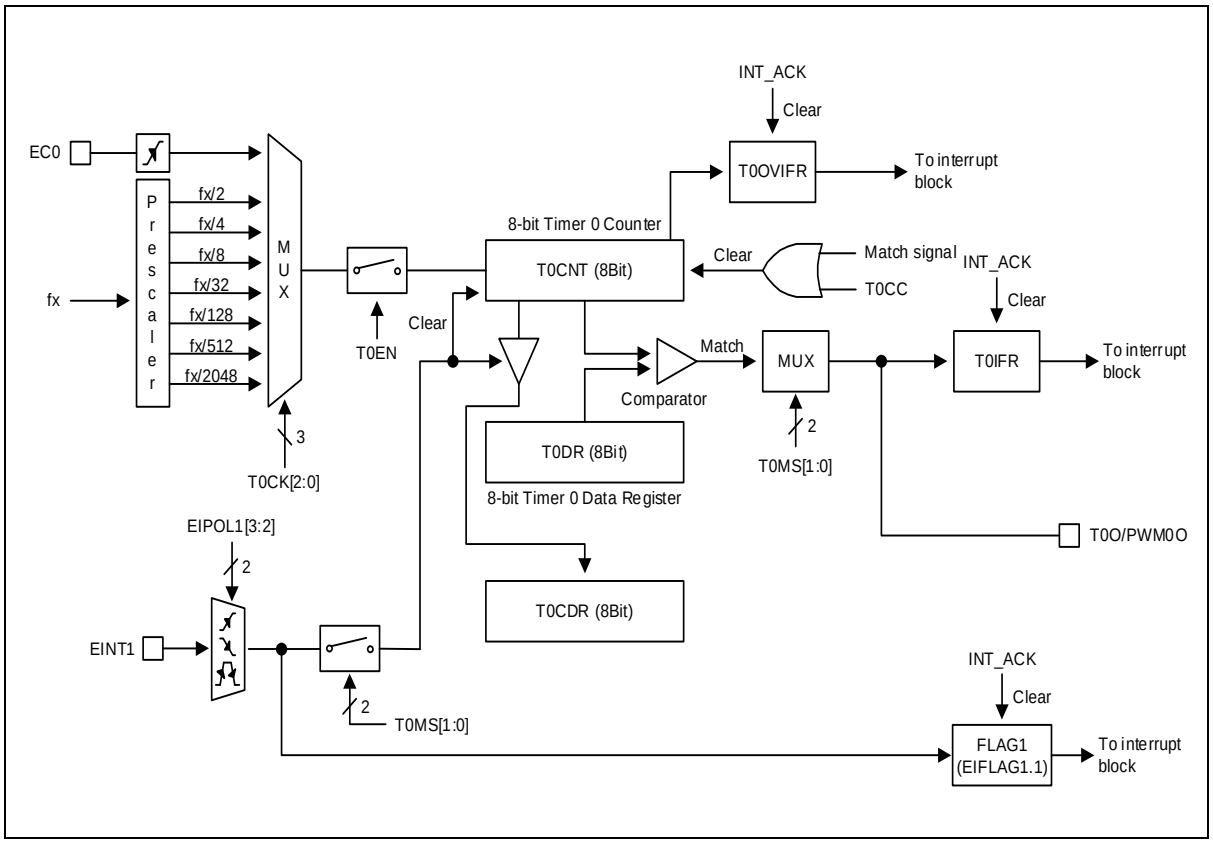


Figure 33. 8-bit Timer 0 Block Diagram

12.1.5 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

12.1.6 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.

12.2 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).

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, 512, 2048 and EC1

In capture mode, the data is captured into input capture data register (T1BDRH/T1BDRL) by EINT2. 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	P0FSRL[5:4]	T1MS[1:0]	T1CK[2:0]	Timer 1
1	10	00	XXX	16 Bit Timer/Counter Mode
1	00	01	XXX	16 Bit Capture Mode
1	10	10	XXX	16 Bit PPG Mode(one-shot mode)
1	10	11	XXX	16 Bit PPG Mode(repeat mode)

12.2.1 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 34. 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, 512 and 2048 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 P14IO bit.

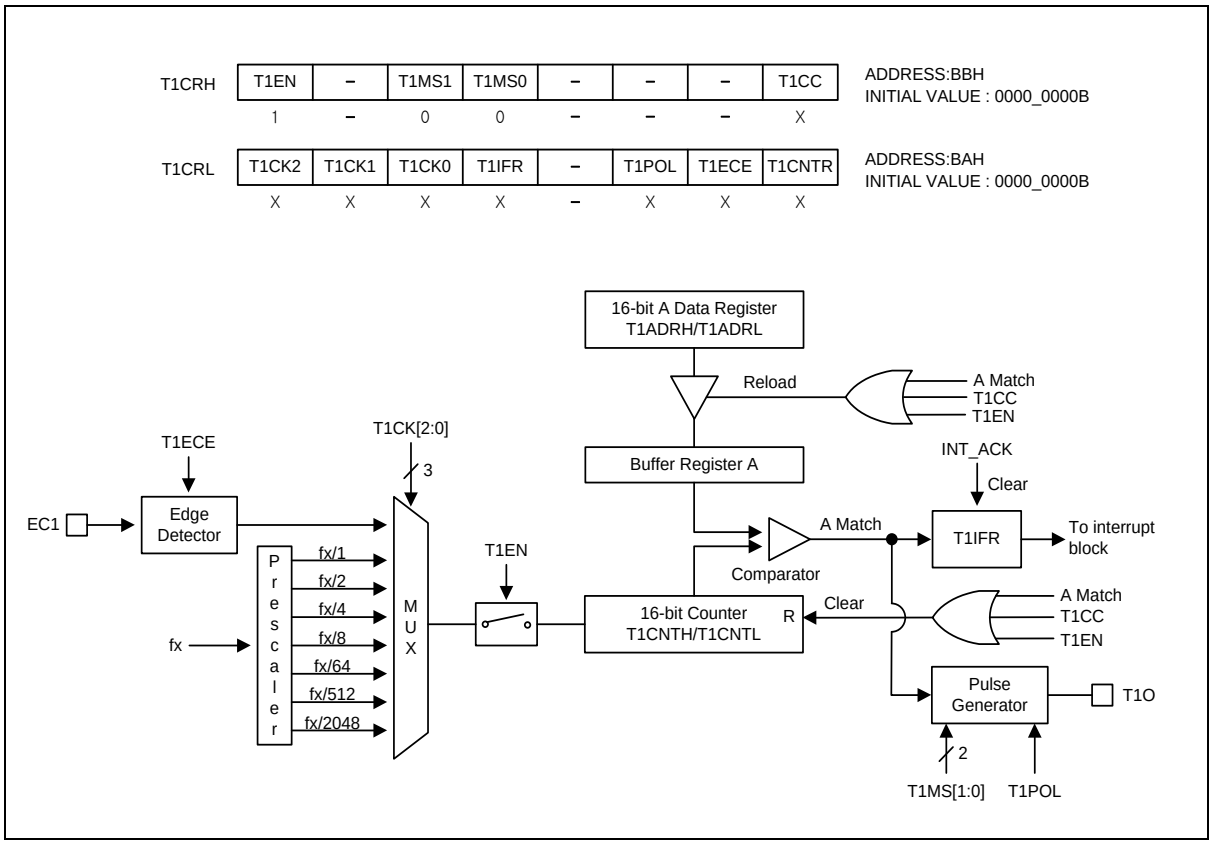


Figure 34. 16-bit Timer/Counter Mode of Timer 1

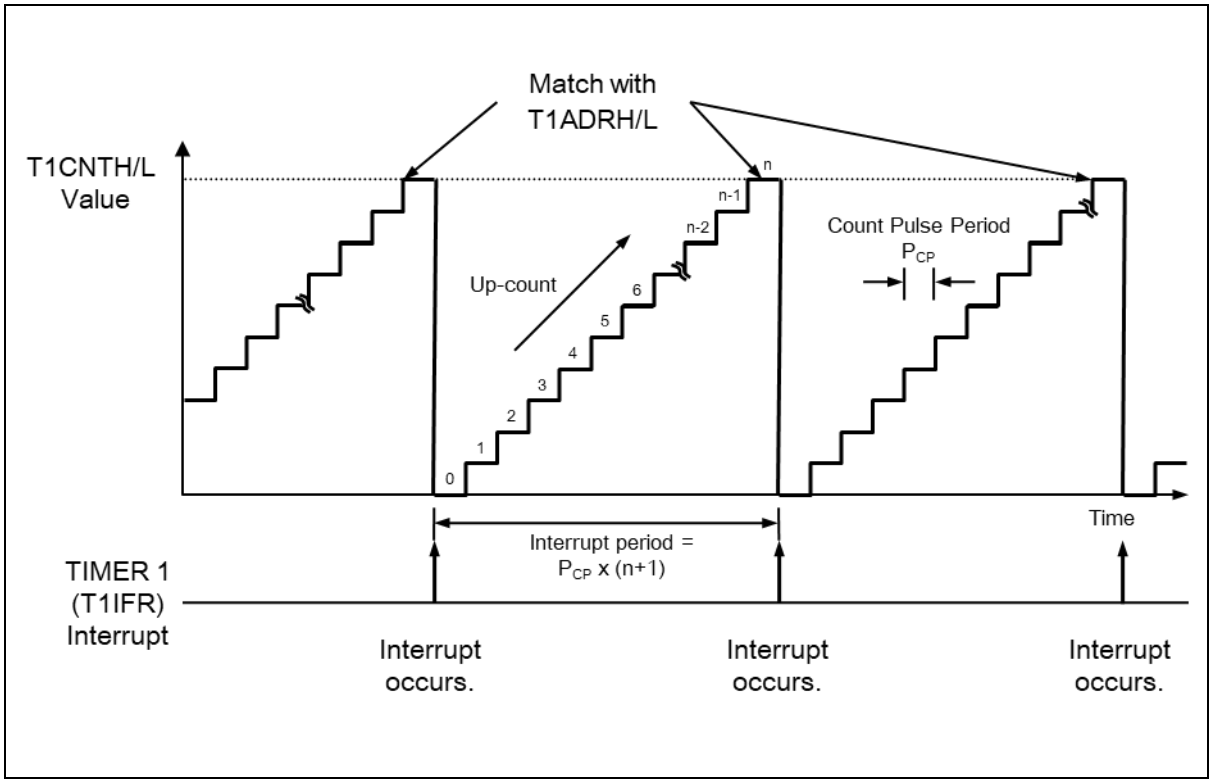


Figure 35. 16-bit Timer/Counter Mode Operation Example

12.2.2 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 EINT2 function is selected. EINT2 pin must be set as an input port.

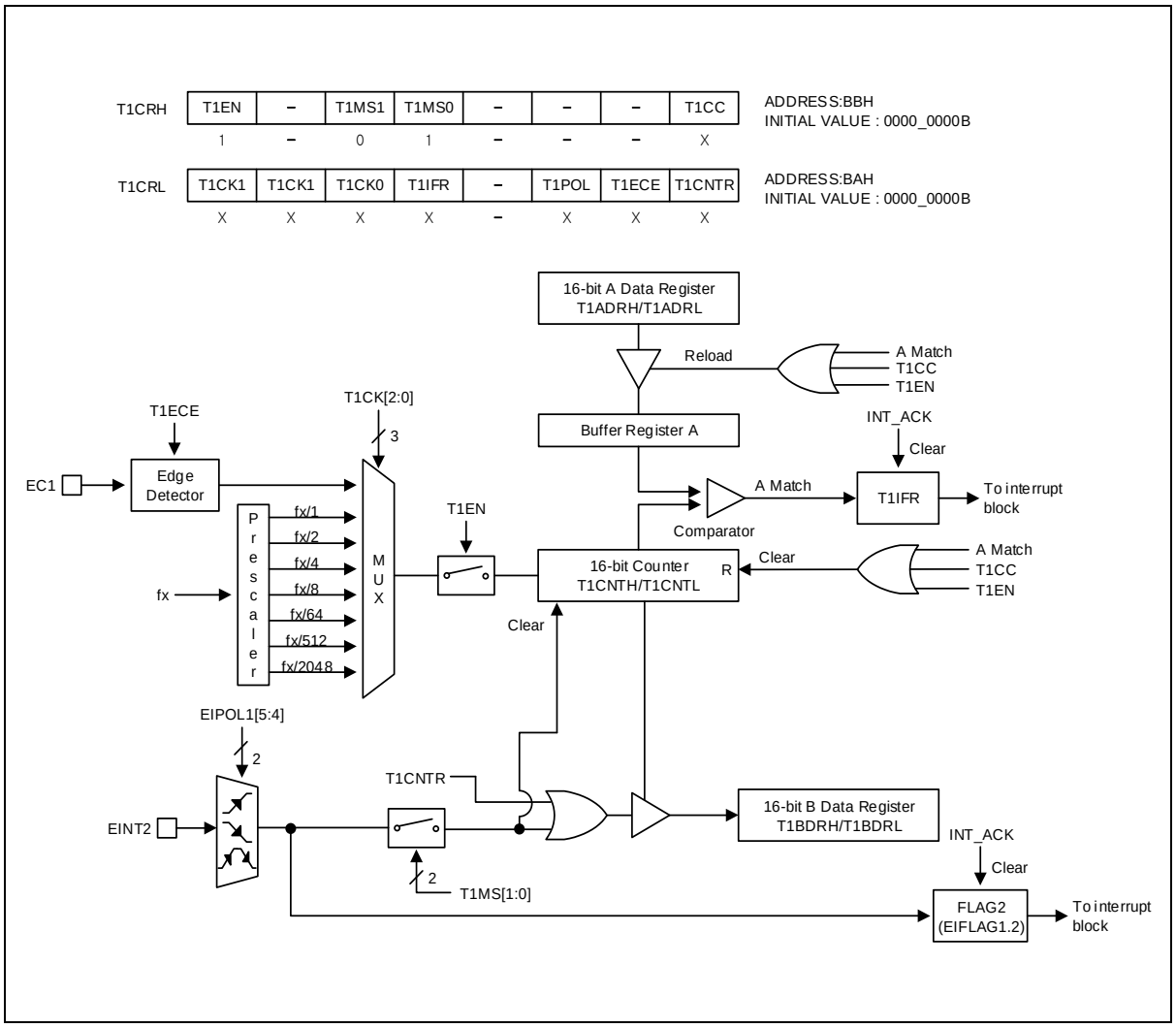


Figure 36. 16-bit Capture Mode of Timer 1

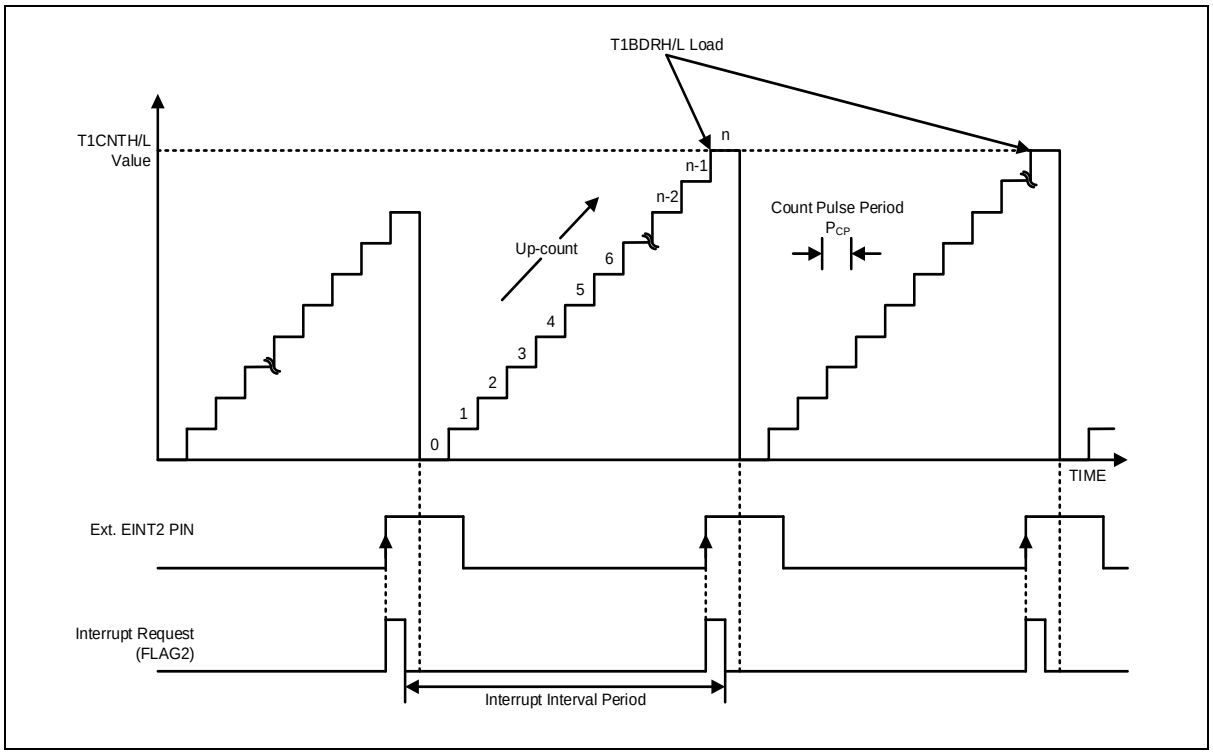


Figure 37. 16-bit Capture Mode Operation Example

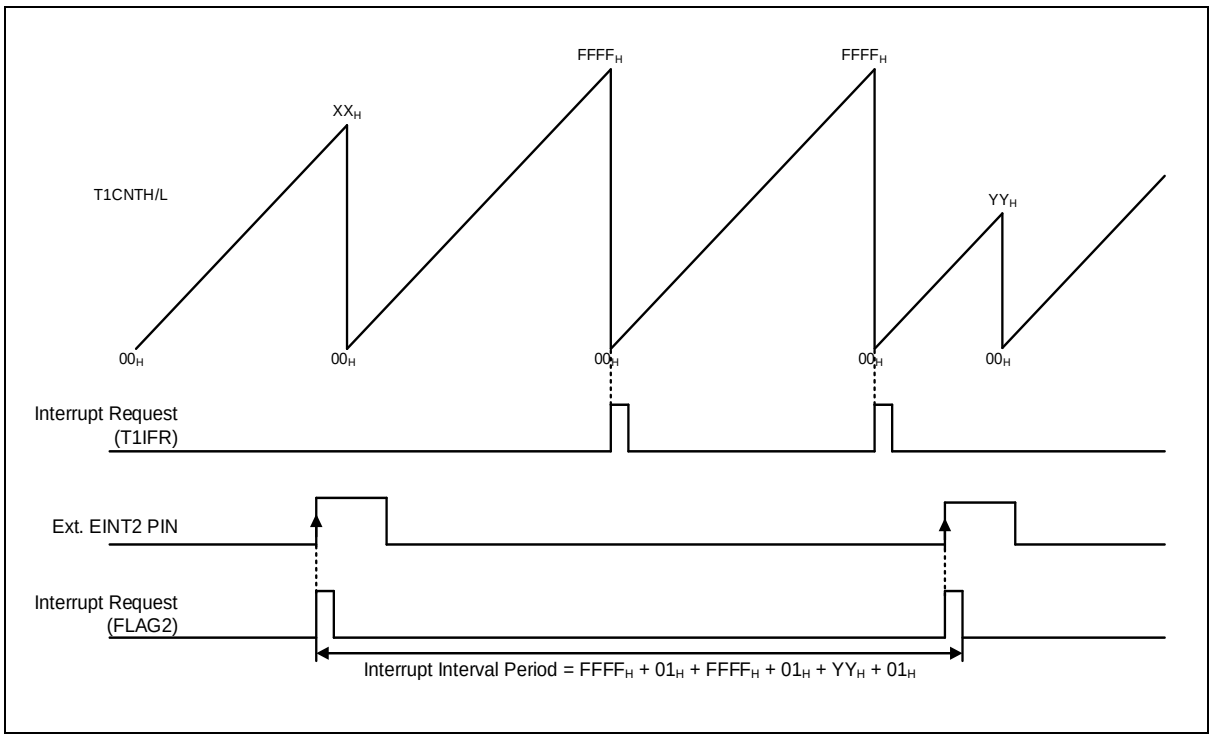
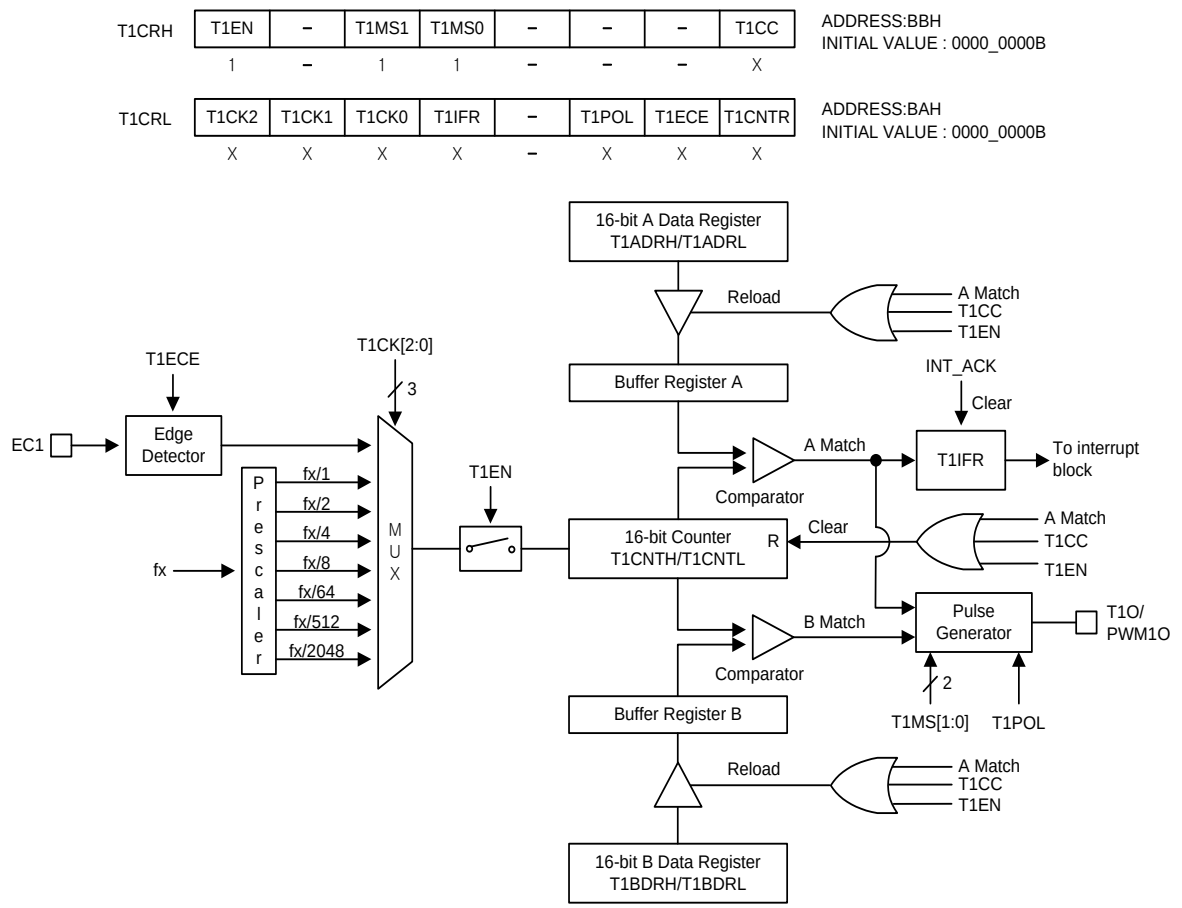


Figure 38. Express Timer Overflow in Capture Mode

12.2.3 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 P0FSRL[5:4] to '10'. 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 39. 16-bit PPG Mode of Timer 1

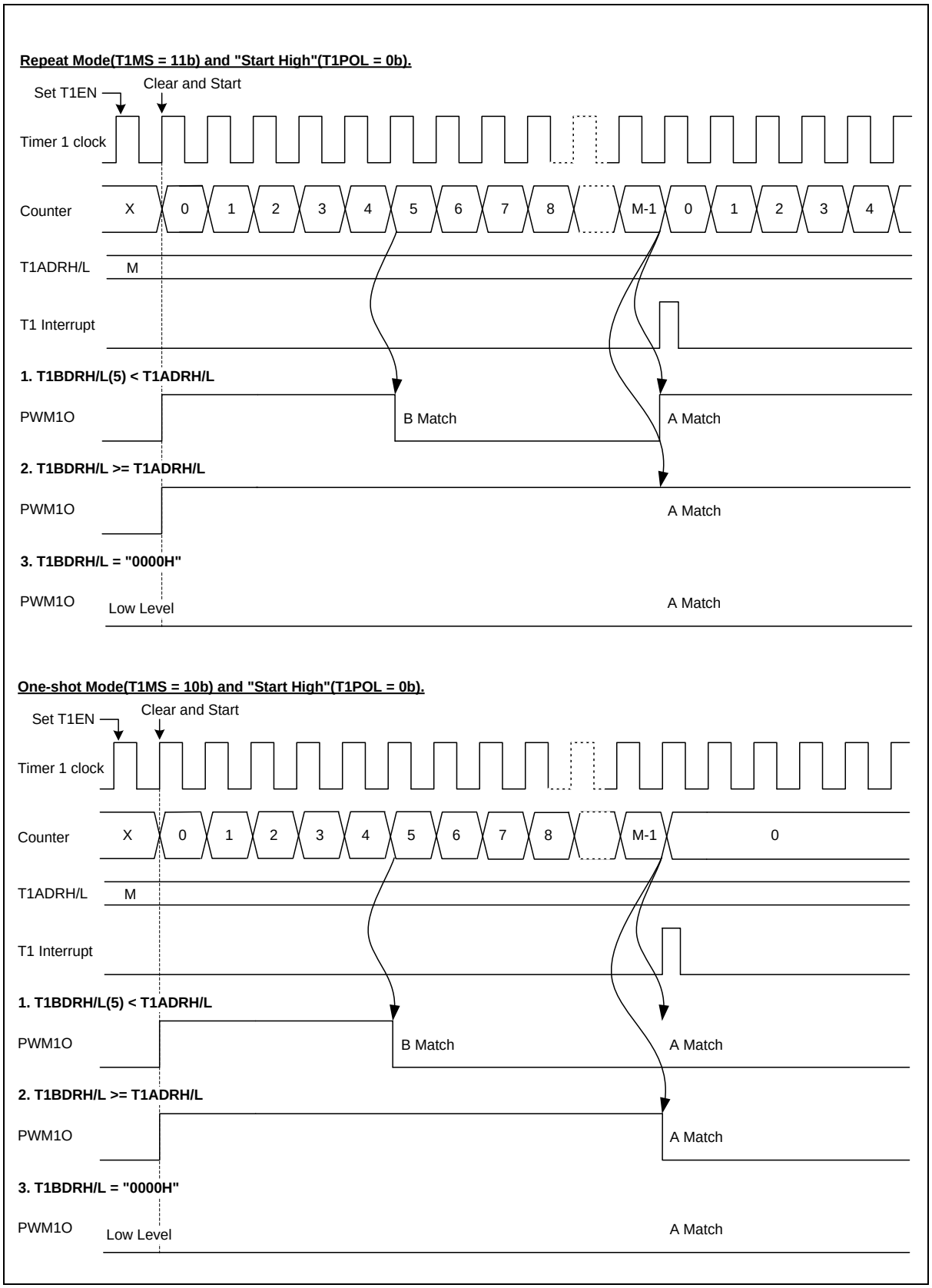


Figure 40. 16-bit PPG Mode Operation Example

12.2.4 16-bit timer 1 block diagram

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

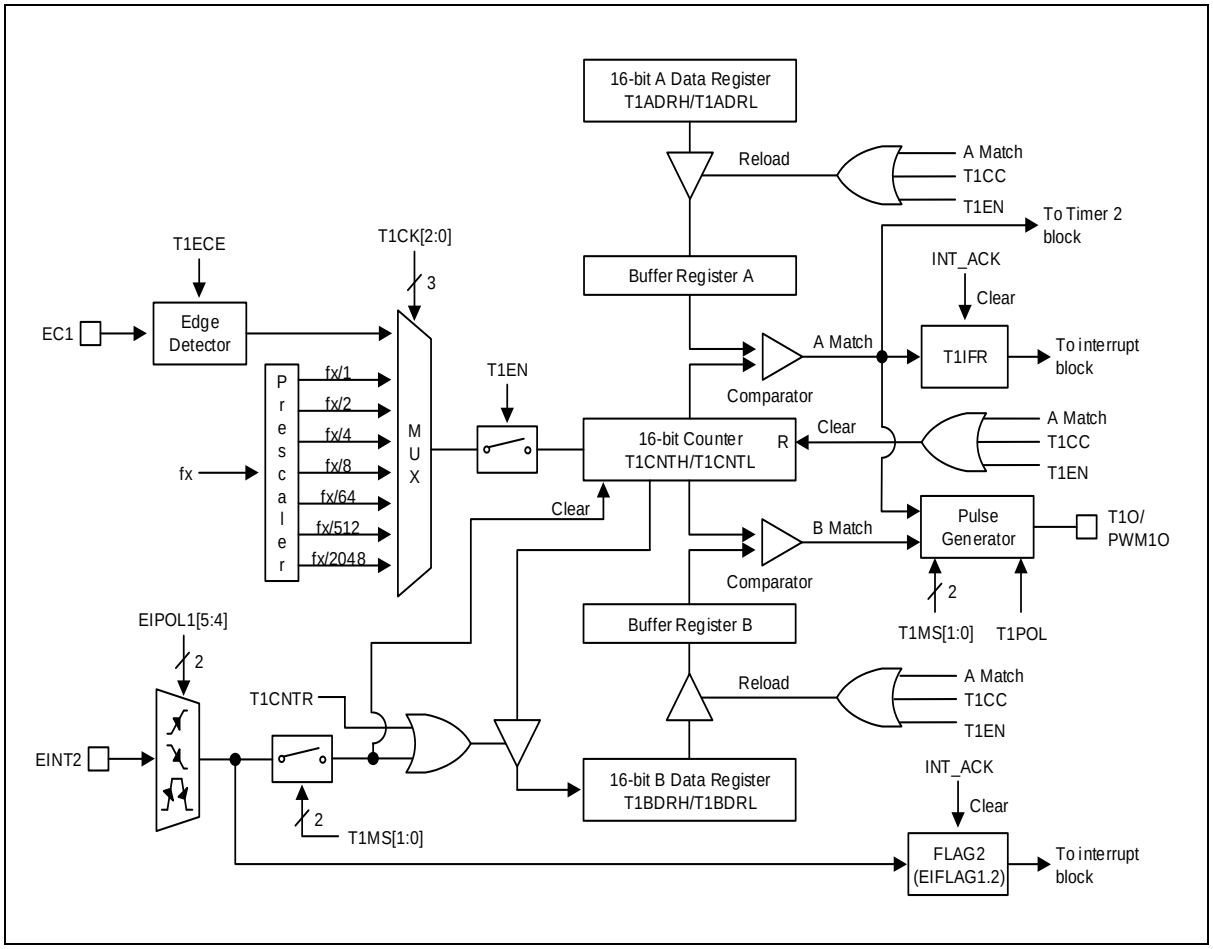


Figure 41. 16-bit Timer 1 Block Diagram

12.2.5 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
T1CRH	BBH	R/W	00H	Timer 1 Control High Register
T1CRL	BAH	R/W	00H	Timer 1 Control Low Register

12.2.6 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

T1CRH (Timer 1 Control High Register): BBH

7	6	5	4	3	2	1	0
T1EN	–	T1MS1	T1MS0	–	–	–	T1CC
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)	
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)
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	–	T1POL	T1ECE	T1CNTR
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/512
0	1	0	fx/64
0	1	1	fx/8
1	0	0	fx/4
1	0	1	fx/2
1	1	0	fx/1
1	1	1	External clock (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

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)

12.3 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 EINT3. 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.

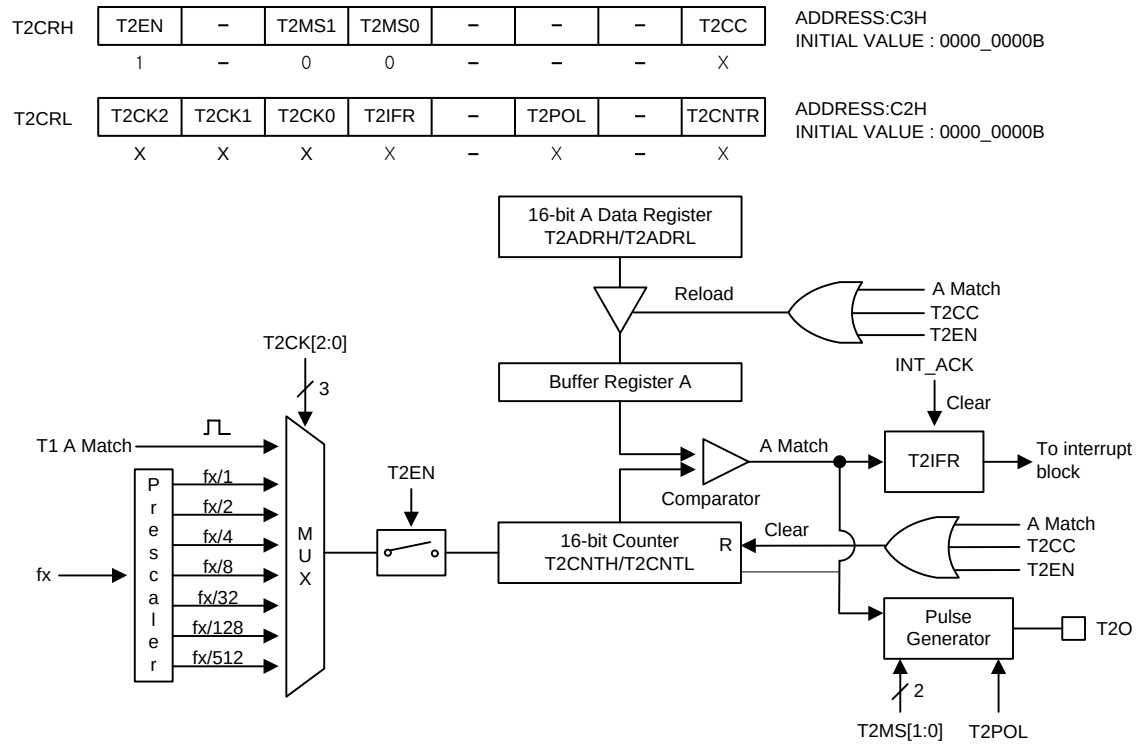
Table 18. TIMER 2 Operating Modes

T2EN	P0FSRL[3:2]	T2MS[1:0]	T2CK[2:0]	Timer 2
1	10	00	XXX	16 Bit Timer/Counter Mode
1	00	01	XXX	16 Bit Capture Mode
1	10	10	XXX	16 Bit PPG Mode(one-shot mode)
1	10	11	XXX	16 Bit PPG Mode(repeat mode)

12.3.1 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 42. 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 42. 16-bit Timer/Counter Mode of Timer 2

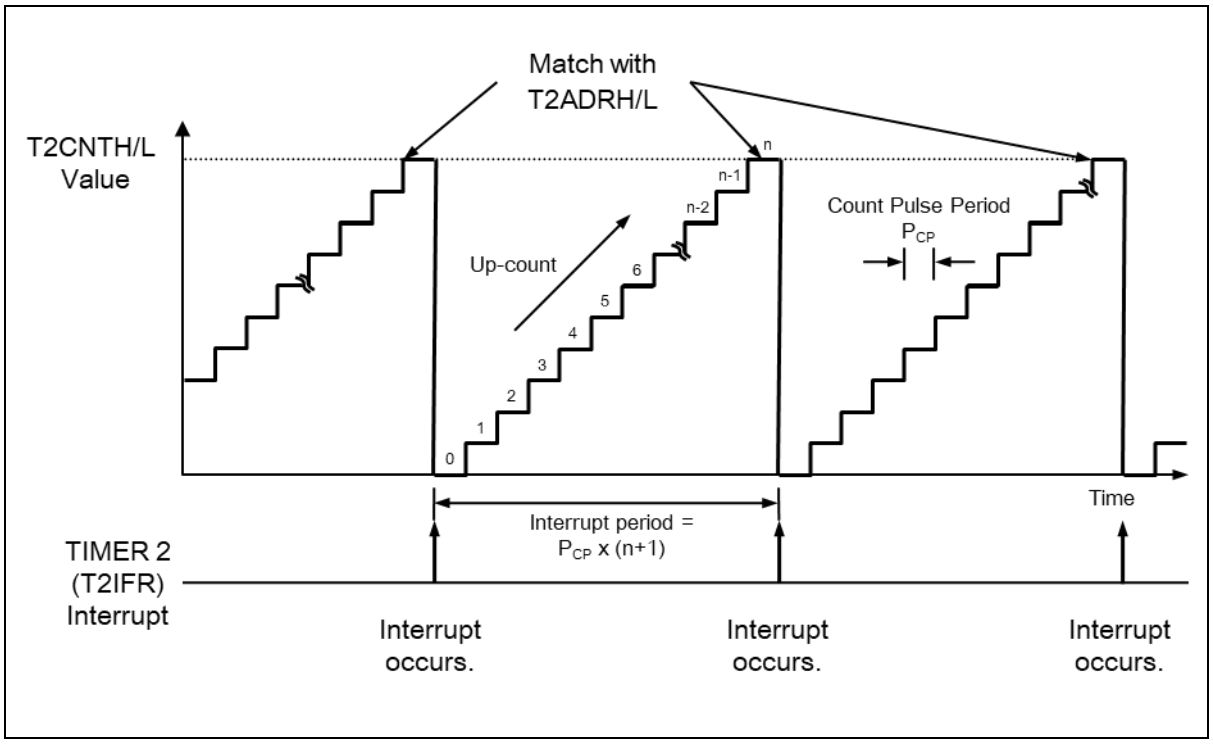


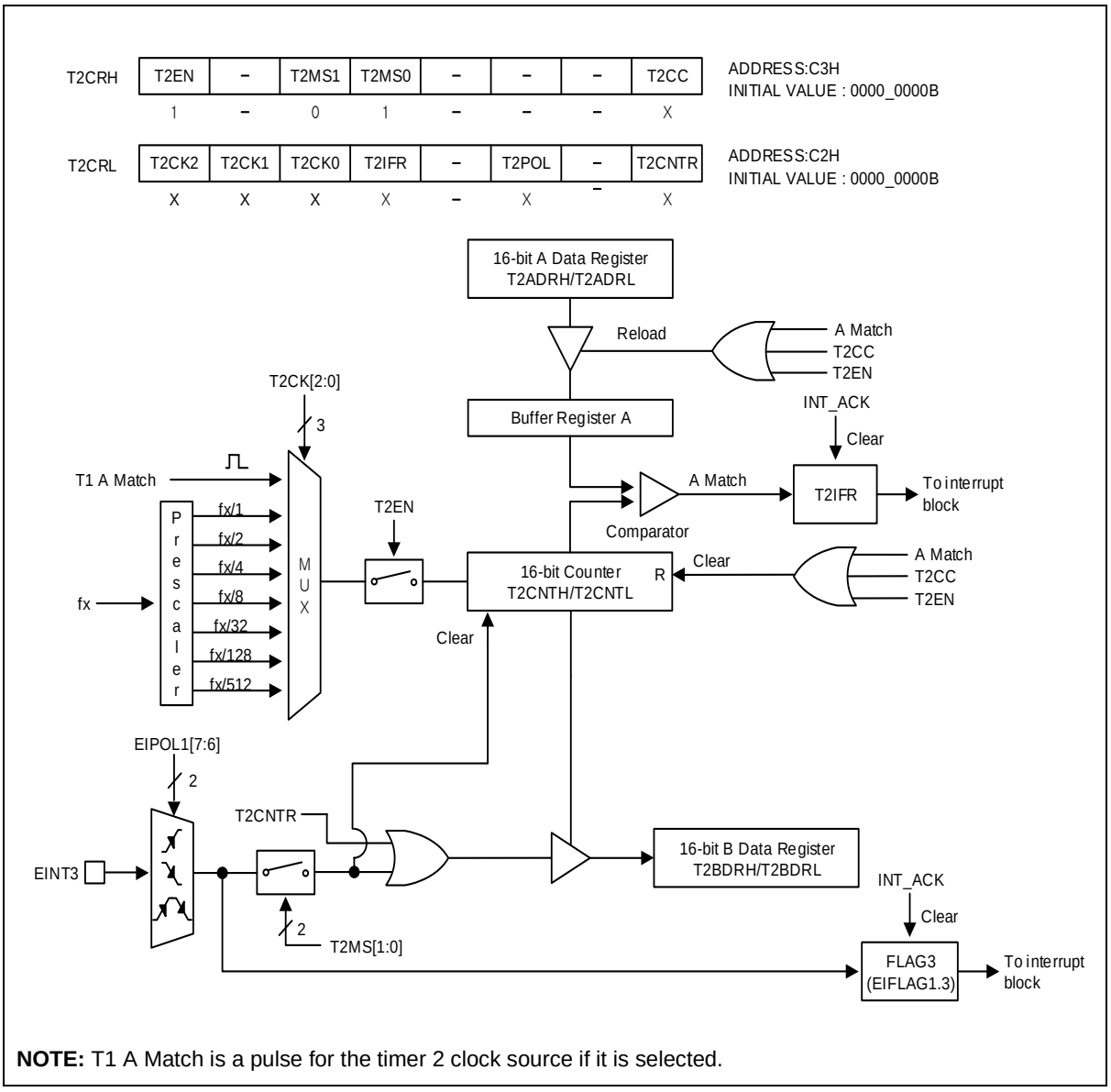
Figure 43. 16-bit Timer/Counter Mode Operation Example

12.3.2 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 EINT3 function is selected. EINT3 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 44. 16-bit Capture Mode of Timer 2

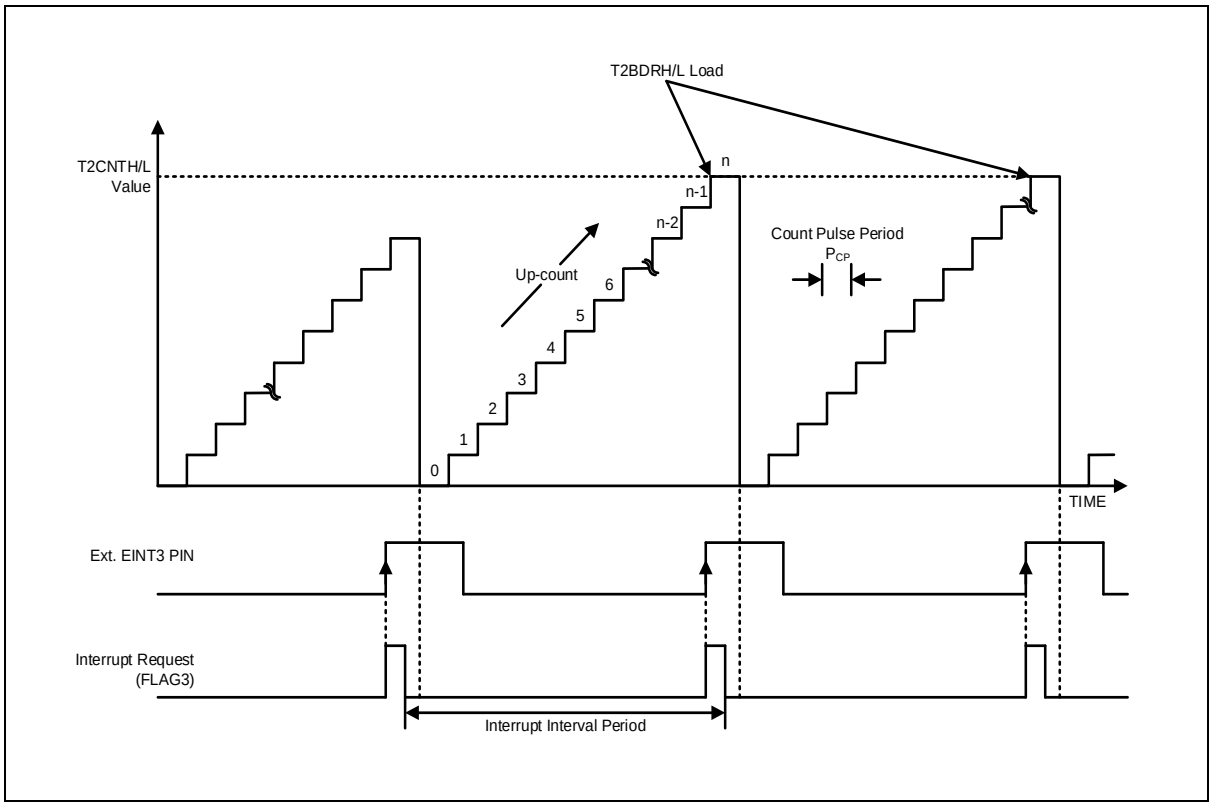


Figure 45. 16-bit Capture Mode Operation Example

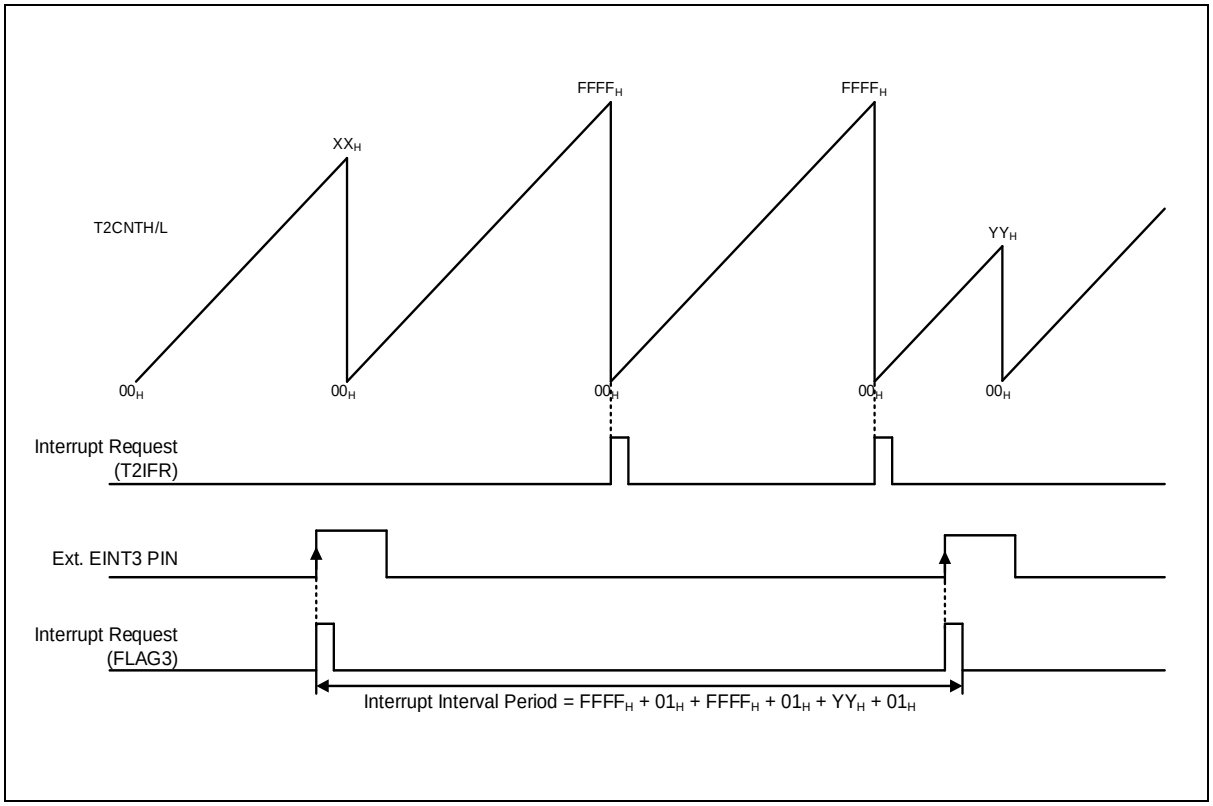
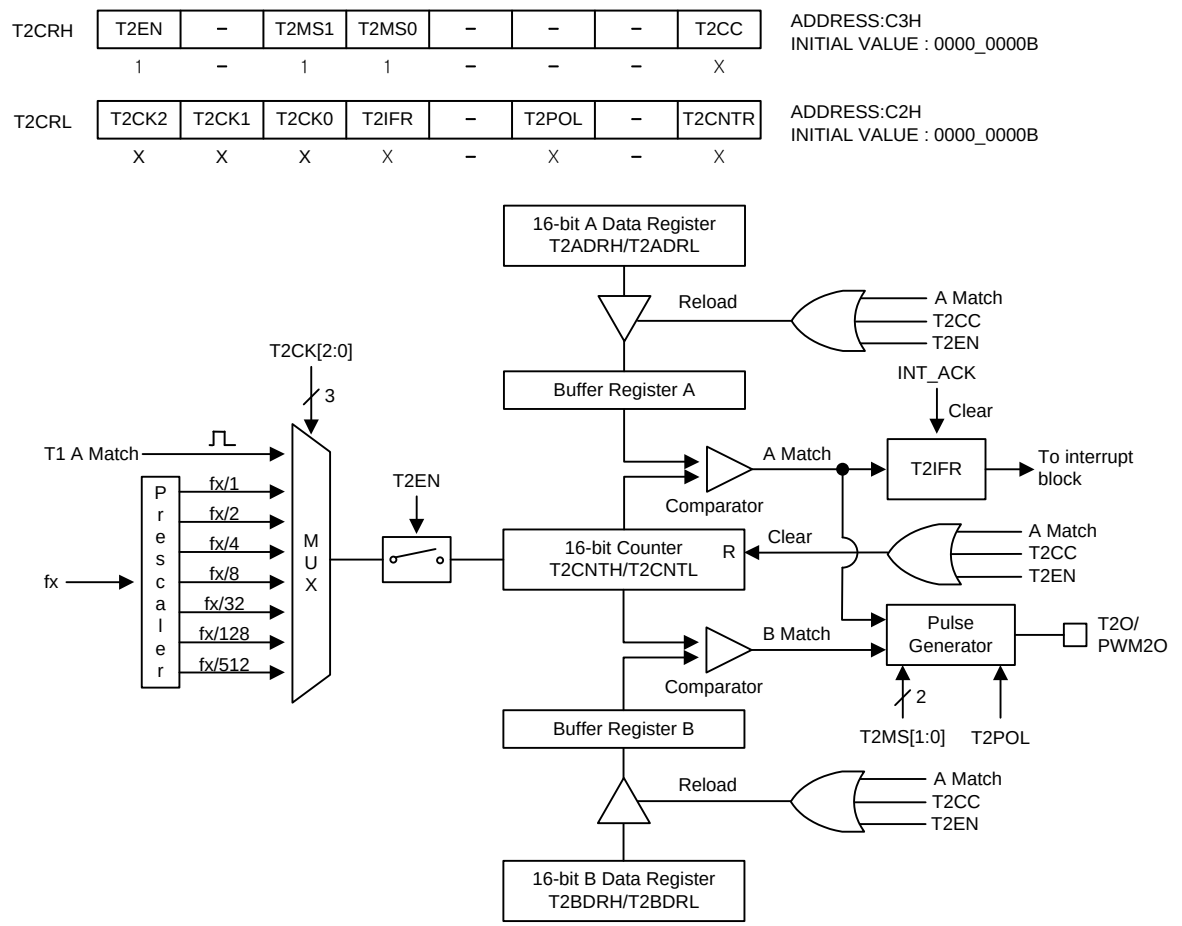


Figure 46. Express Timer Overflow in Capture Mode

12.3.3 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 P0FSRL[3:2] to '10'. 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 47. 16-bit PPG Mode of Timer 2

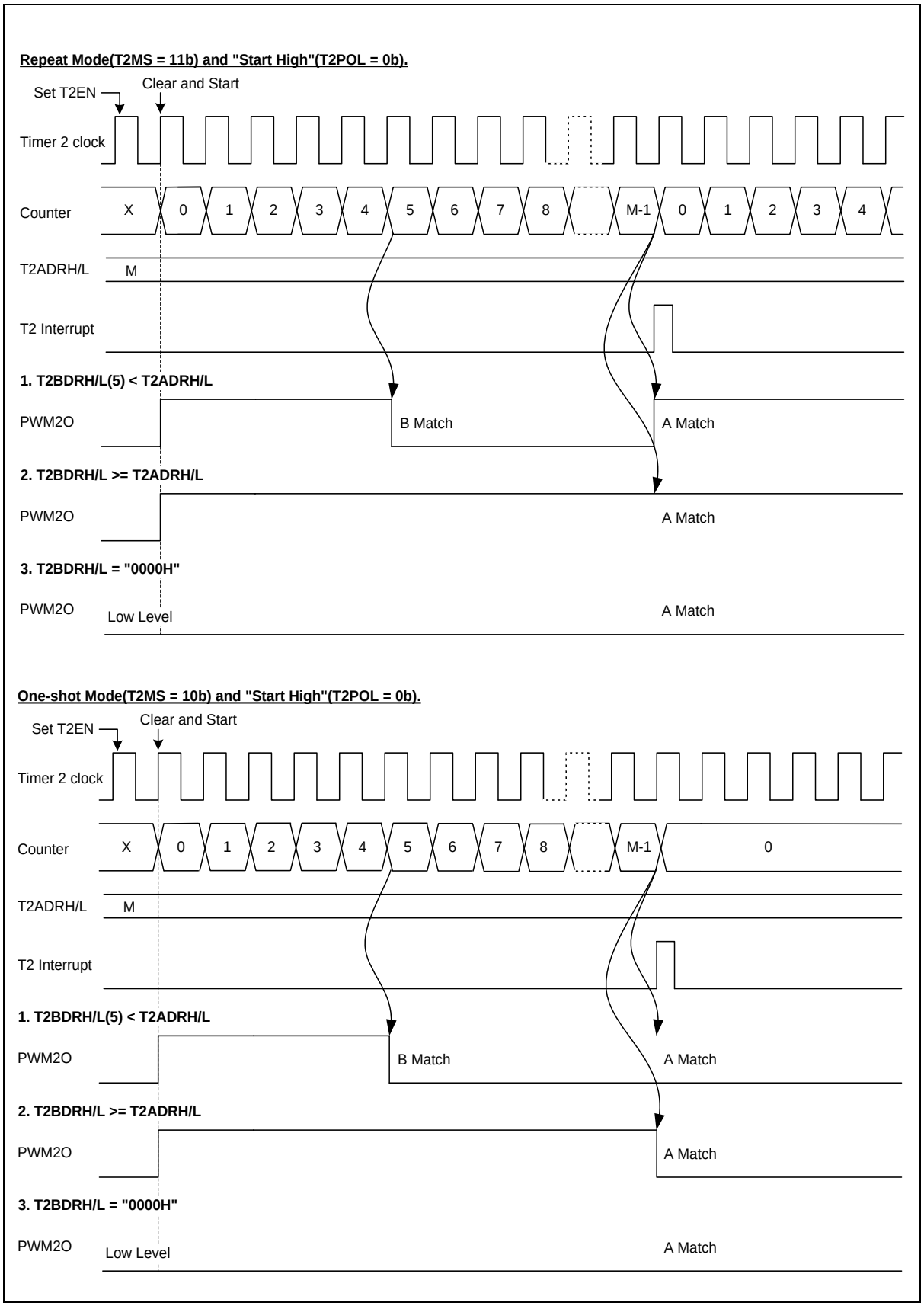
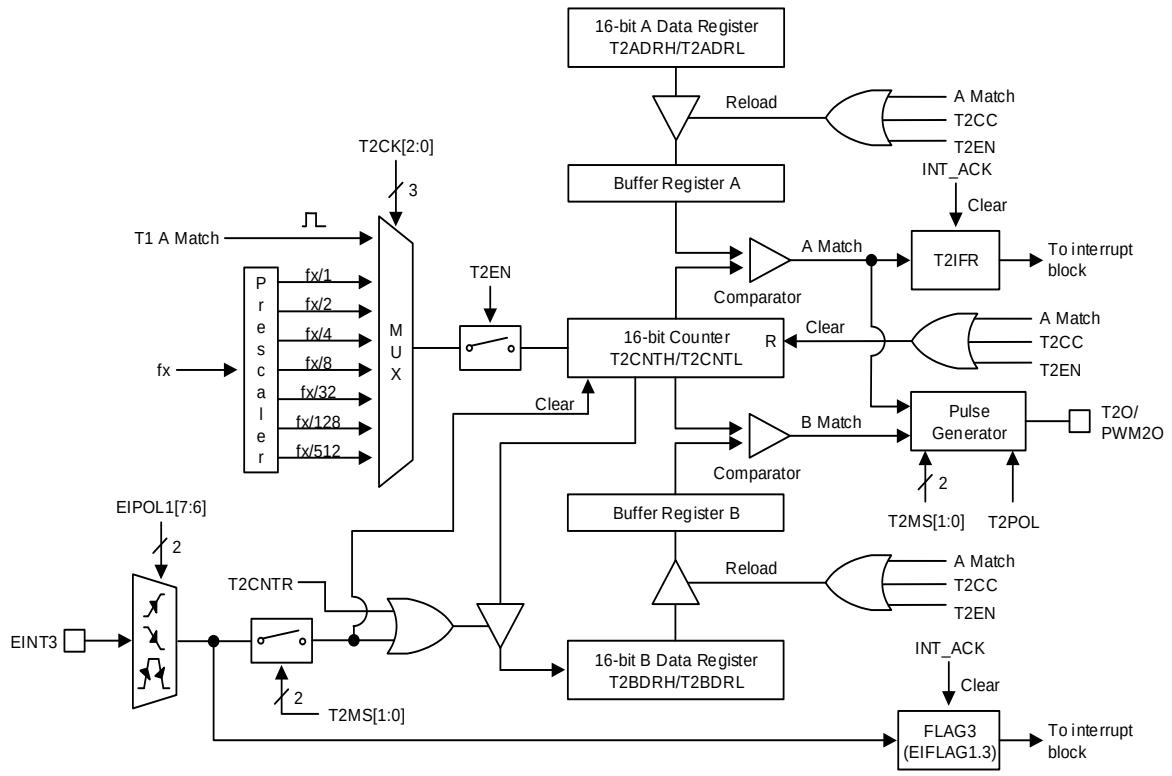


Figure 48. 16-bit PPG Mode Operation Example

12.3.4 16-bit timer 2 block diagram

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



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

Figure 49. 16-bit Timer 2 Block Diagram

12.3.5 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

12.3.6 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)		

12.4 Timer 3

A 16-bit timer 3 consists of a multiplexer, timer 3 A data high/low register, timer 3 B data high/low register and timer 3 control high/low register (T3ADRH, T3ADRL, T3BDRH, T3BDRL, T3CRH, and T3CRL).

Timer 3 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 3 can be clocked by an internal or an external clock source (EC3). The clock source is selected by a clock selection logic controlled by clock selection bits (T3CK[2:0]).

- TIMER 3 clock source: fx/1, fx/2, fx/4, fx/8, fx/64, fx/512, fx/2048 and EC3

In capture mode, data is captured into input capture data registers (T3BDRH/T3BDRL) by EINT43. Timer 3 results in the comparison between counter and data register through T3O port in timer/counter mode. In addition, timer 3 outputs PWM waveform through PWM3O port in the PPG mode.

Table 20. TIMER 3 Operating Modes

T3EN	P0FSRH[1:0]	T3MS[1:0]	T3CK[2:0]	Timer 3
1	10	00	XXX	16 Bit Timer/Counter Mode
1	00	01	XXX	16 Bit Capture Mode
1	10	10	XXX	16 Bit PPG Mode(one-shot mode)
1	10	11	XXX	16 Bit PPG Mode(repeat mode)

12.4.1 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 50. The counter register is increased by internal or external clock input. Timer 3 can use the input clock with one of 1, 2, 4, 8, 64, 512 and 2048 prescaler division rates (T3CK[2:0]). When the values of T3CNTH/T3CNTL and T3ADRH/T3ADRL are identical to each other in timer 3, a match signal is generated and the interrupt of Timer 3 occurs. The T3CNTH/T3CNTL values are automatically cleared by the match signal. It can be cleared by software (T3CC) too.

External clock (EC3) counts up the timer at the rising edge. If the EC3 is selected as a clock source by T3CK[2:0], EC3 port should be set as an input port by configuring P16 IO bit.

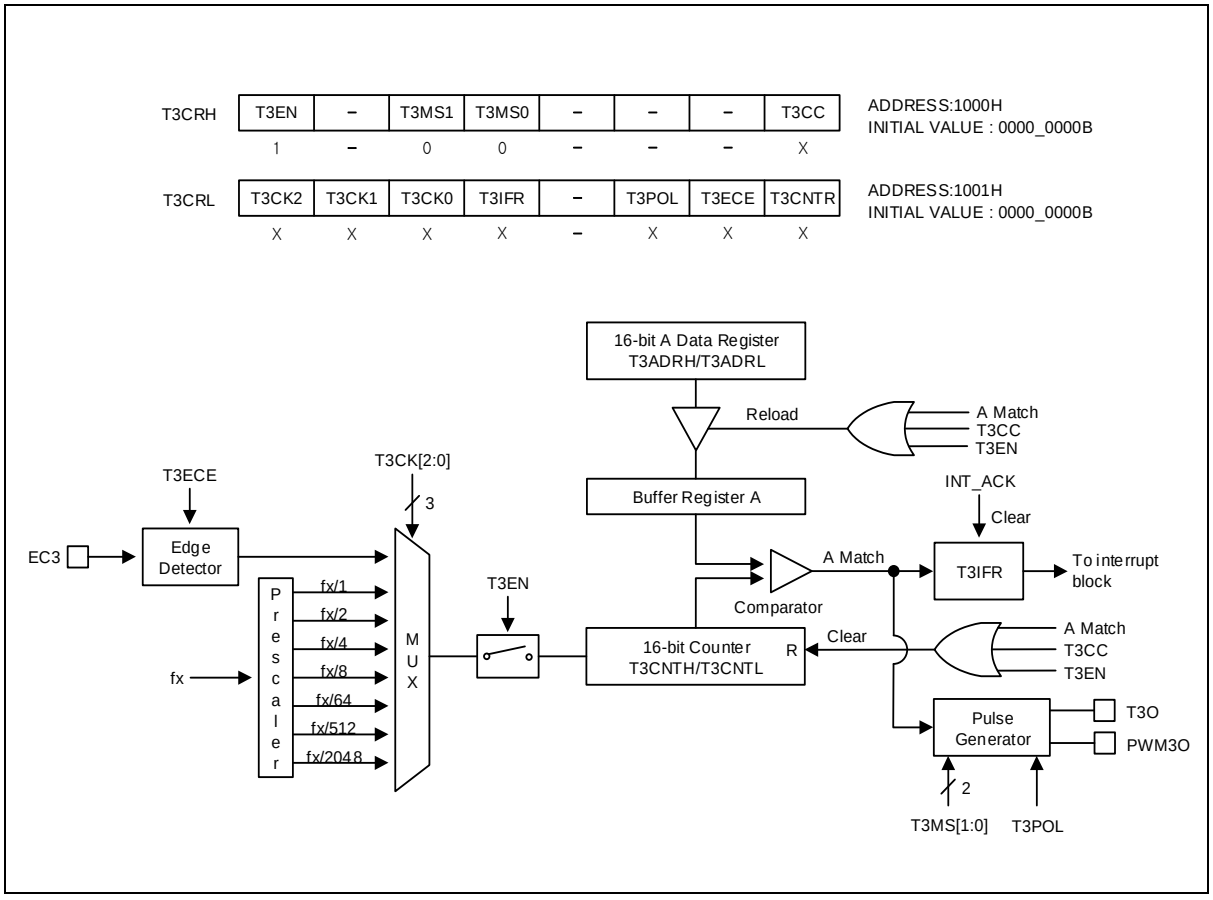


Figure 50. 16-bit Timer/Counter Mode of Timer 3

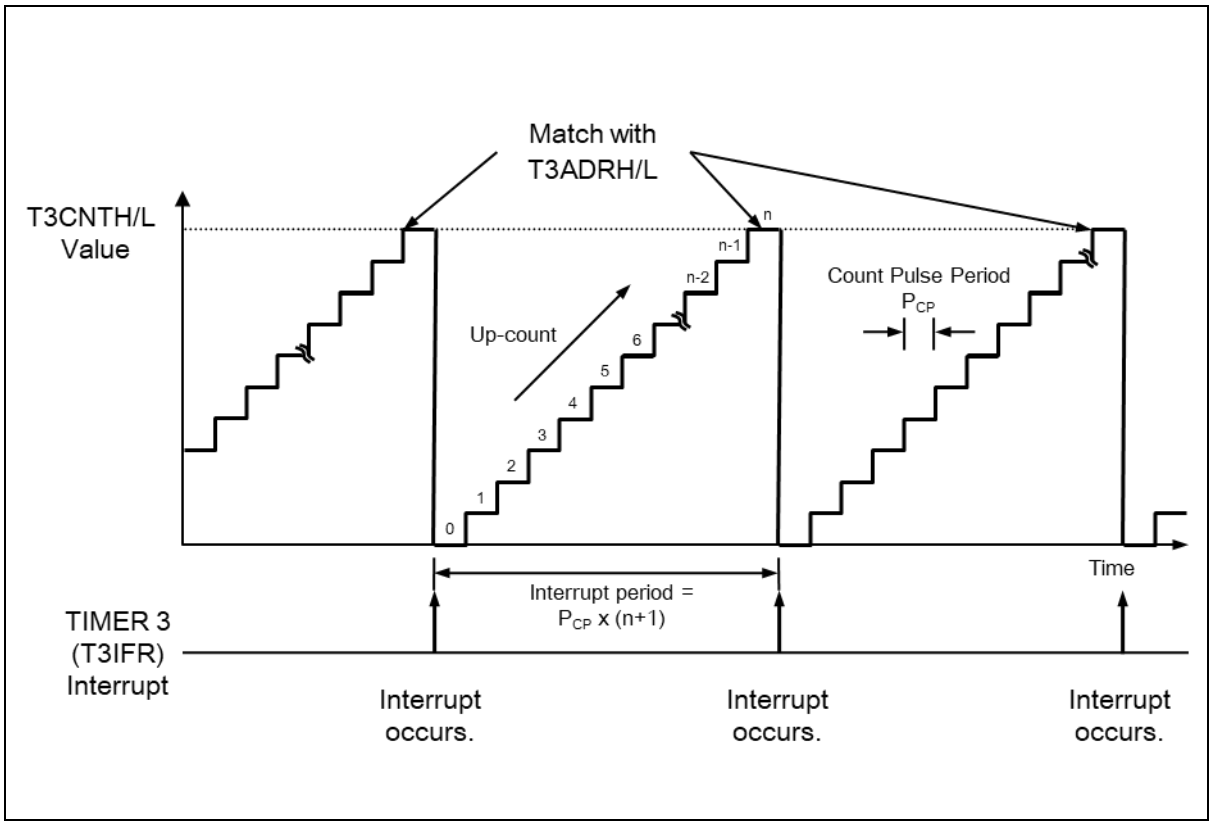


Figure 51. 16-bit Timer/Counter Mode Operation Example

12.4.2 16-bit capture mode

Timer 3 capture mode is set by configuring T3MS[1:0] as '01'. It uses an internal/external clock as a clock source. Basically, the 16-bit timer 3 capture mode has the same function as the 16-bit timer/counter mode, and the interrupt occurs when T3CNTH/T3CNTL is equal to T3ADRH/T3ADRL. The T3CNTH, T3CNTL values are automatically cleared by a match signal. It can be cleared by software (T3CC) 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 T3BDRH/T3BDRL.

According to EIPOL0L registers setting, the external interrupt EINT43 function is selected. EINT43 pin must be set as an input port.

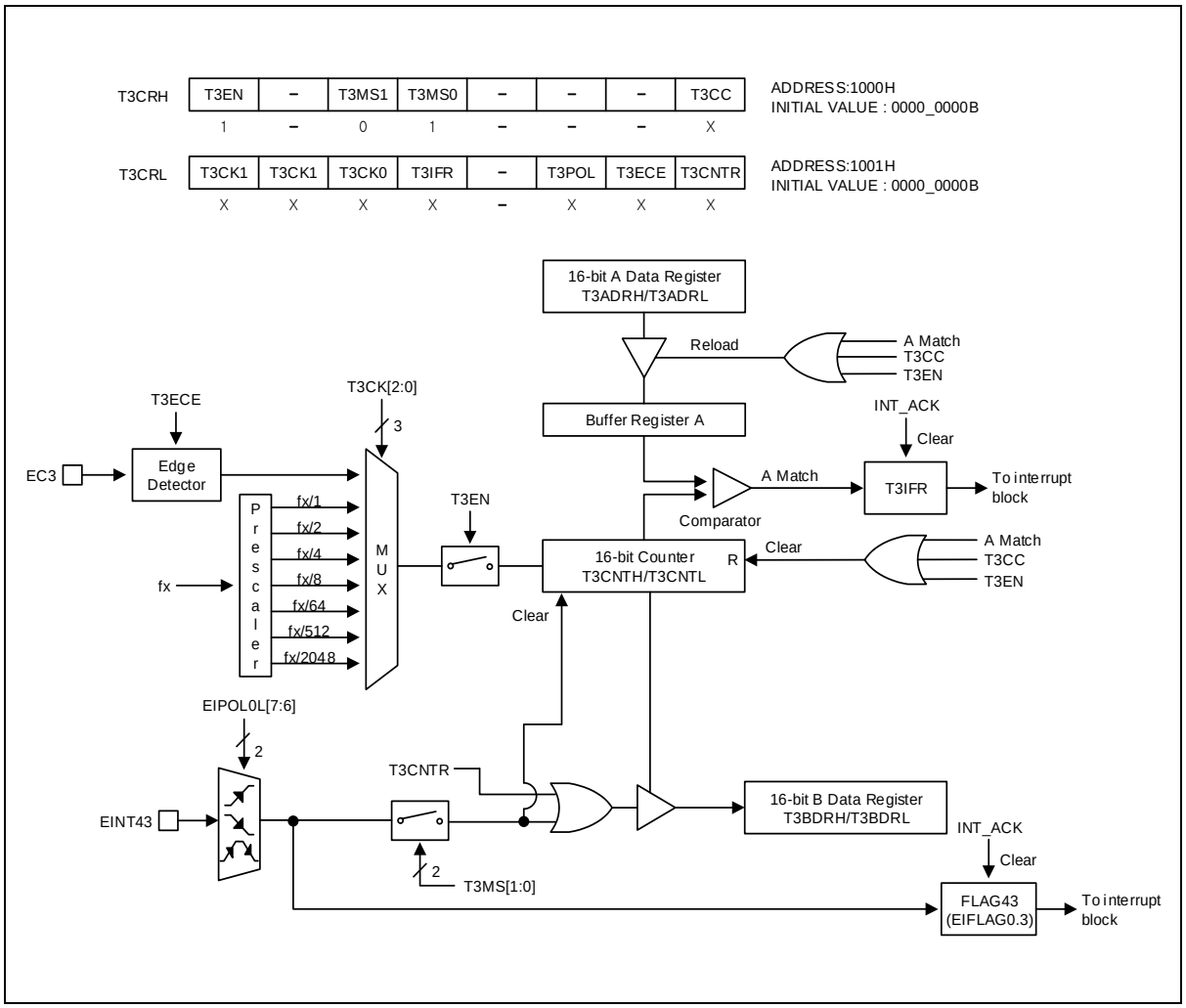


Figure 52. 16-bit Capture Mode of Timer 3

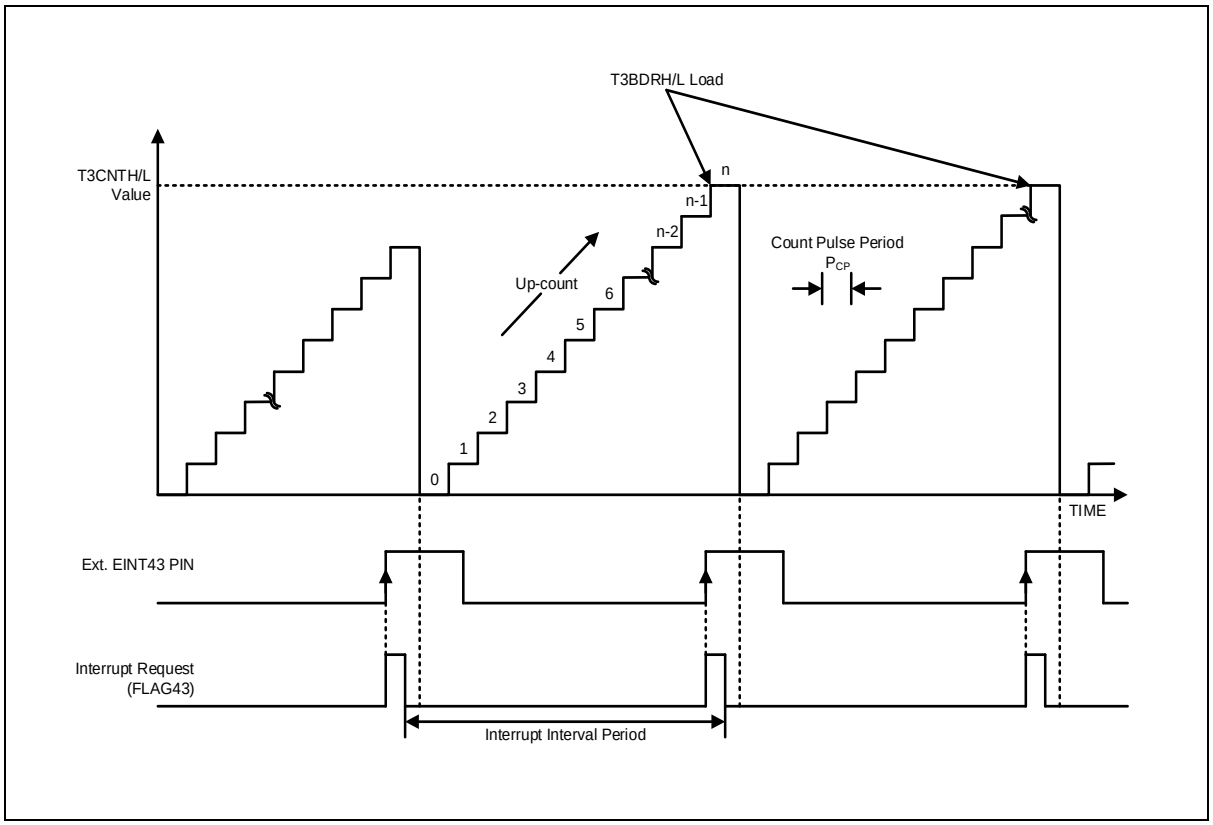


Figure 53. 16-bit Capture Mode Operation Example

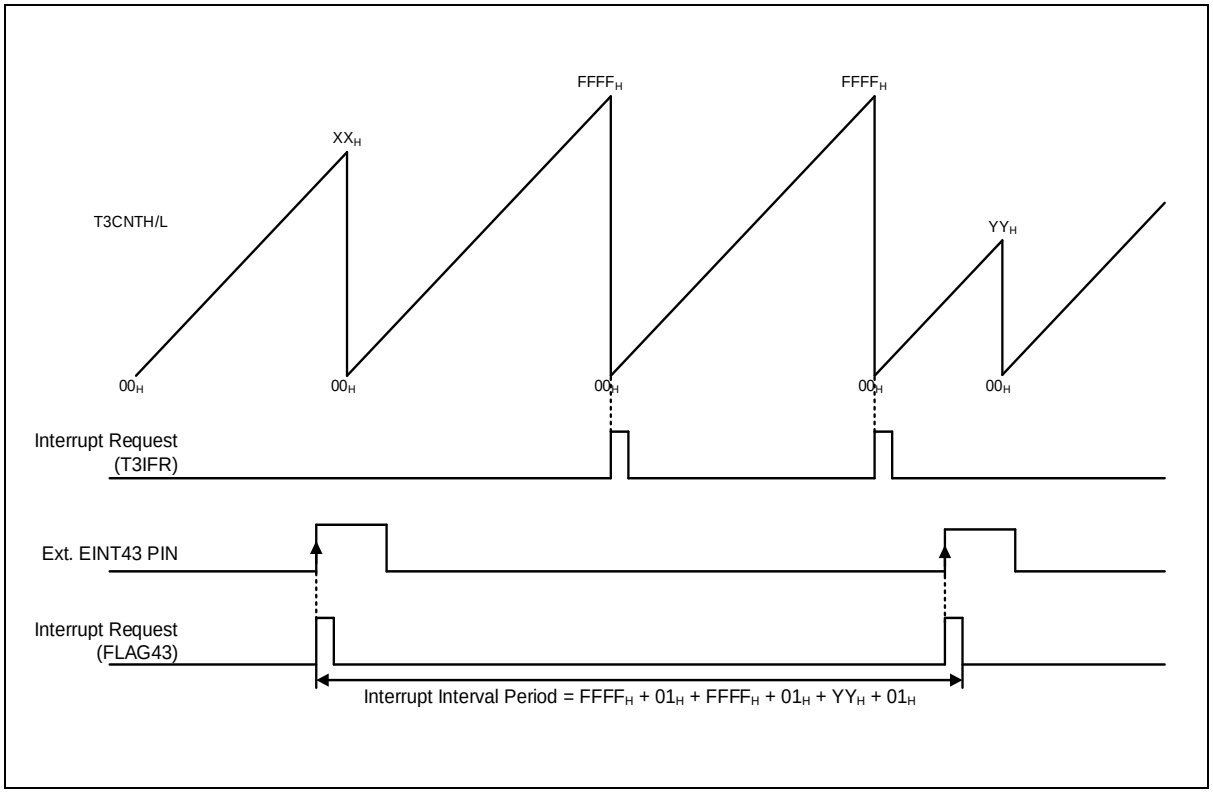
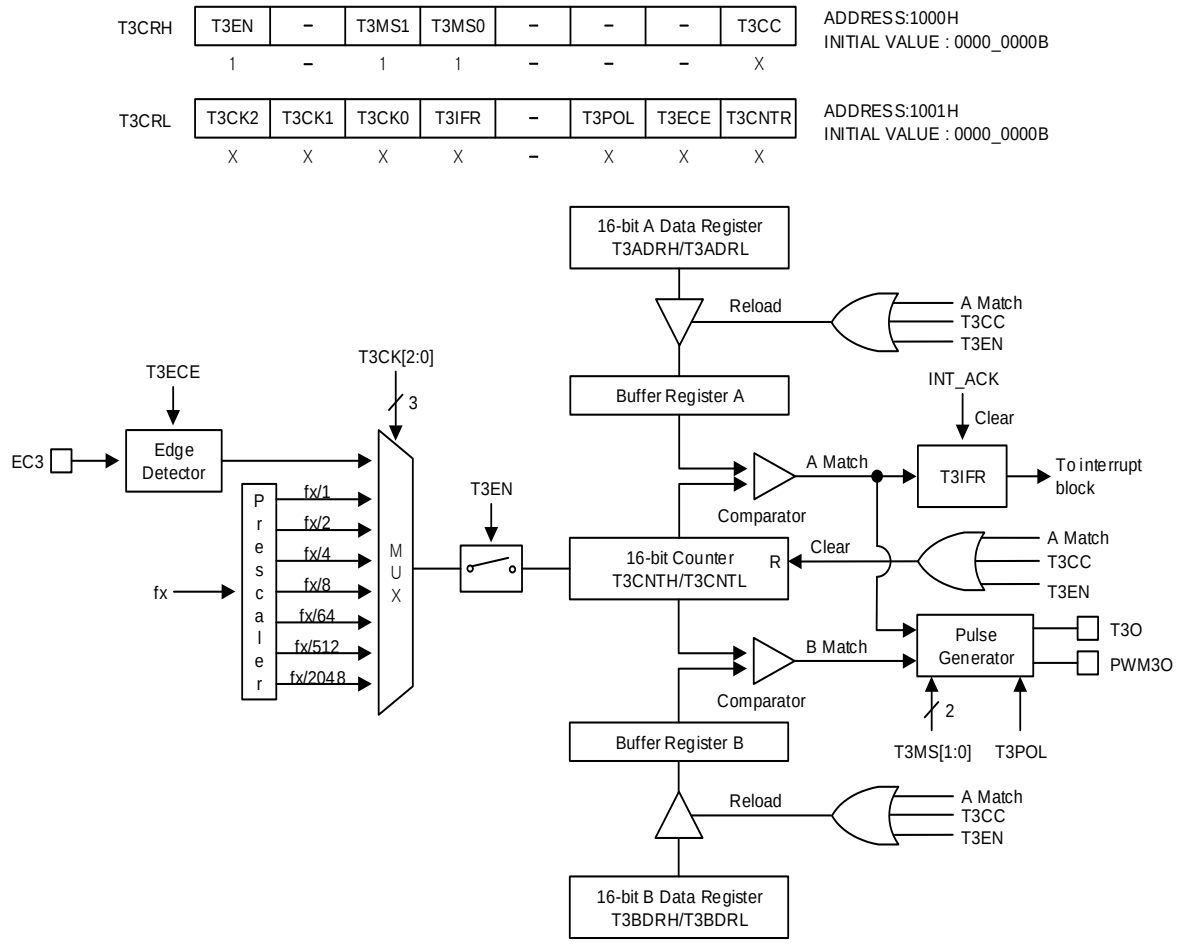


Figure 54. Express Timer Overflow in Capture Mode

12.4.3 16-bit PPG mode

TIMER 3 has a PPG (Programmable Pulse Generation) function. In PPG mode, T3O/PWM3O pin outputs up to 16-bit resolution PWM output. For this function, T3O/PWM3O pin must be configured as a PWM output by setting P0FSRL[1:0] to ‘10’ or P4FSRH[3:2] to ‘10’. Period of the PWM output is determined by T3ADRH/T3ADRL, and duty of the PWM output is determined by T3BDRH/T3BDRL.



NOTE: T3EN is automatically cleared to logic “0” after one pulse is generated at a PPG one-shot mode.

Figure 55. 16-bit PPG Mode of Timer 3

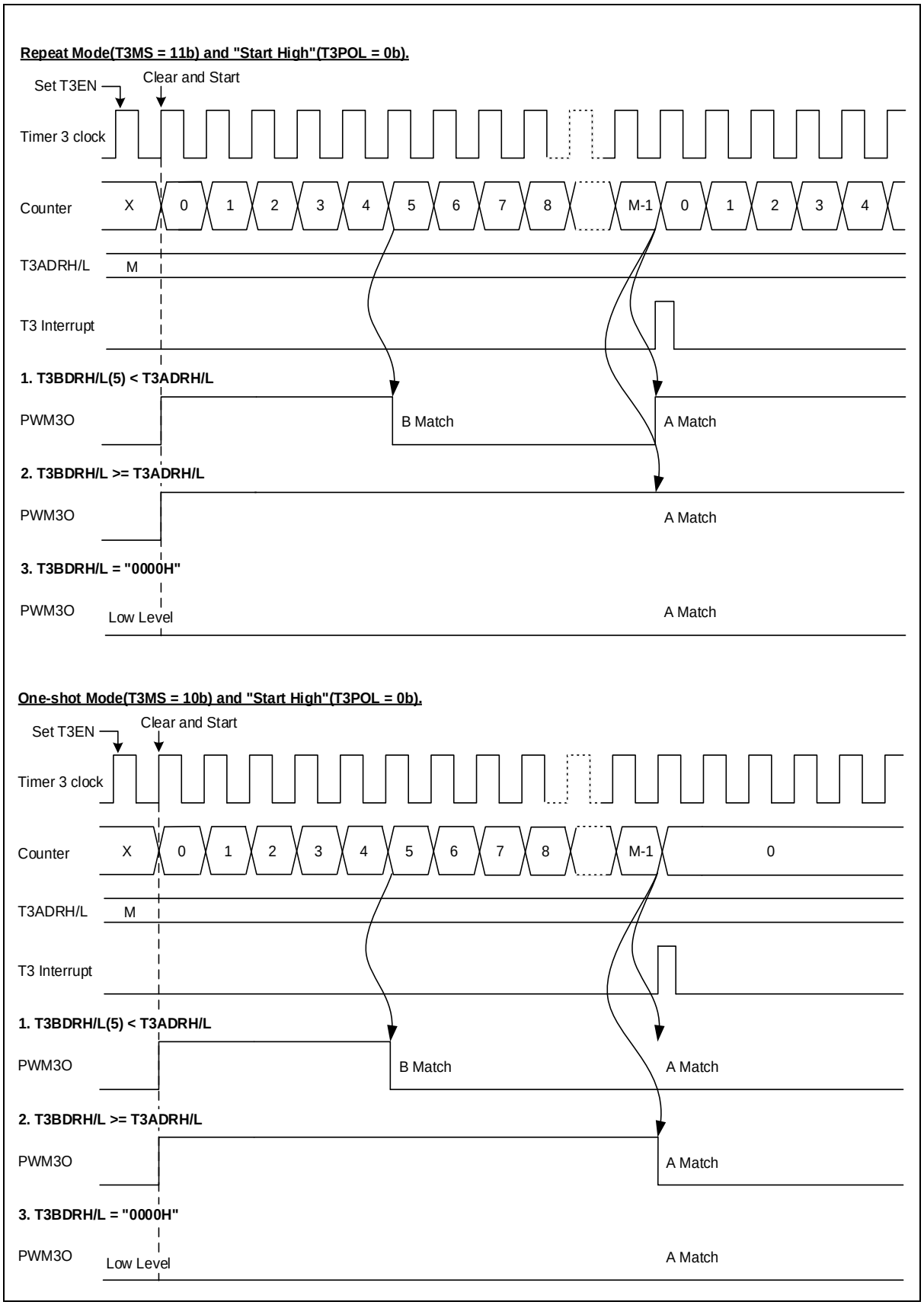


Figure 56. 16-bit PPG Mode Operation Example

12.4.4 16-bit timer 3 block diagram

In this section, a 16-bit timer 3 is described in a block diagram.

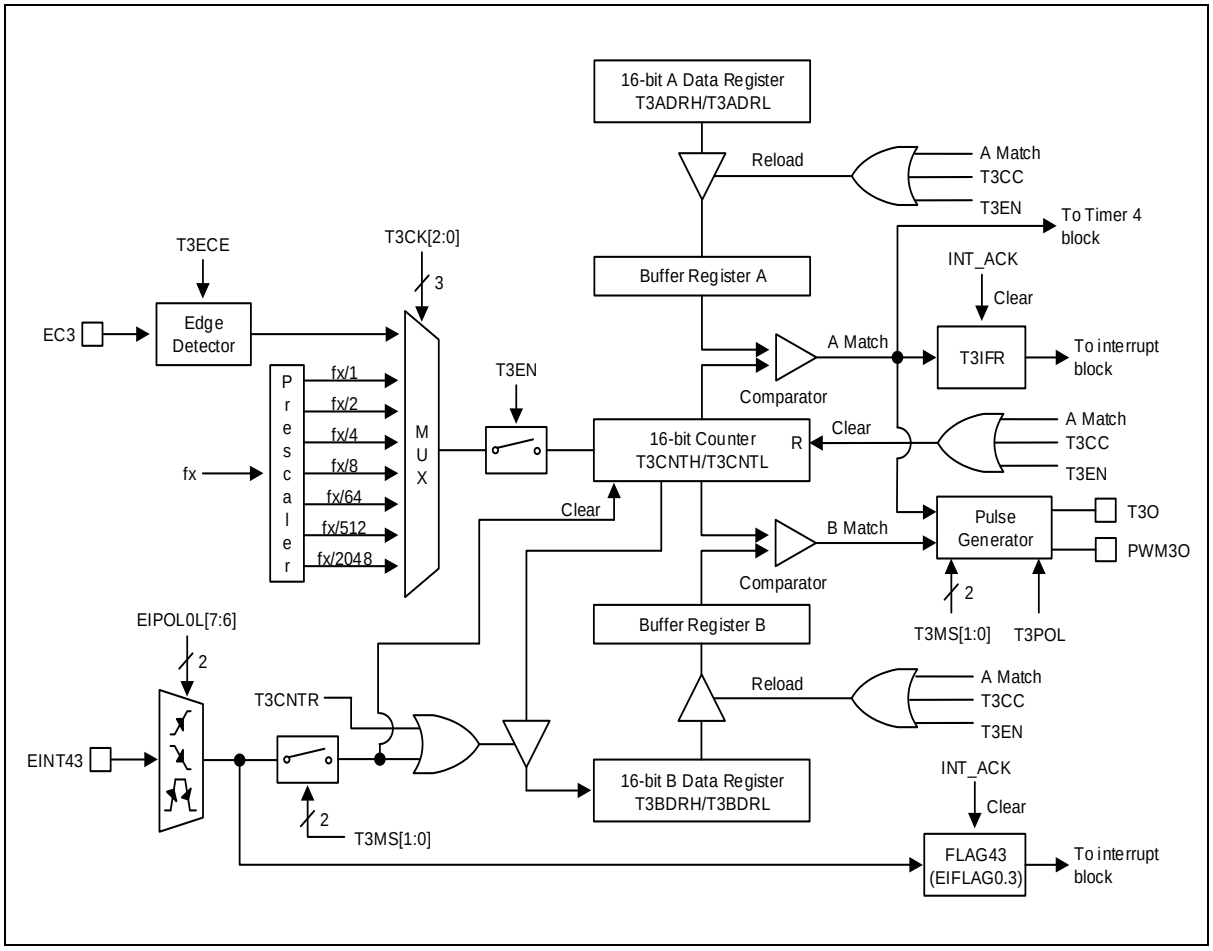


Figure 57. 16-bit Timer 3 Block Diagram

12.4.5 Register map

Table 21. TIMER 3 Register Map

Name	Address	Direction	Default	Description
T3ADRH	1002H	R/W	FFH	Timer 3 A Data High Register
T3ADRL	1003H	R/W	FFH	Timer 3 A Data Low Register
T3BDRH	1004H	R/W	FFH	Timer 3 B Data High Register
T3BDRL	1005H	R/W	FFH	Timer 3 B Data Low Register
T3CRH	1000H	R/W	00H	Timer 3 Control High Register
T3CRL	1001H	R/W	00H	Timer 3 Control Low Register

12.4.6 Register description

T3ADRH (Timer 3 A data High Register): 1002H

7	6	5	4	3	2	1	0
T3ADRH7	T3ADRH6	T3ADRH5	T3ADRH4	T3ADRH3	T3ADRH2	T3ADRH1	T3ADRH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: FFH

T3ADRH[7:0] T3 A Data High Byte

T3ADRL (Timer 3 A Data Low Register): 1003H

7	6	5	4	3	2	1	0
T3ADRL7	T3ADRL6	T3ADRL5	T3ADRL4	T3ADRL3	T3ADRL2	T3ADRL1	T3ADRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: FFH

T3ADRL[7:0] T3 A Data Low Byte

NOTE: Do not write “0000H” in the T3ADRH/T3ADRL register when PPG mode

T3BDRH (Timer 3 B Data High Register): 1004H

7	6	5	4	3	2	1	0
T3BDRH7	T3BDRH6	T3BDRH5	T3BDRH4	T3BDRH3	T3BDRH2	T3BDRH1	T3BDRH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: FFH

T3BDRH[7:0] T3 B Data High Byte

T3BDRL (Timer 3 B Data Low Register): 1005H

7	6	5	4	3	2	1	0
T3BDRL7	T3BDRL6	T3BDRL5	T3BDRL4	T3BDRL3	T3BDRL2	T3BDRL1	T3BDRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: FFH

T3BDRL[7:0] T3 B Data Low Byte

T3CRH (Timer 3 Control High Register): 1000H

7	6	5	4	3	2	1	0
T3EN	–	T3MS1	T3MS0	–	–	–	T3CC
R/W	–	R/W	R/W	–	–	–	R/W

Initial value: 00H

T3EN	Control Timer 3		
	0	Timer 3 disable	
	1	Timer 3 enable (Counter clear and start)	
T3MS[1:0]	Control Timer 3 Operation Mode		
	T3MS1	T3MS0	Description
	0	0	Timer/counter mode (T3O: toggle at A match)
	0	1	Capture mode (The A match interrupt can occur)
	1	0	PPG one-shot mode (PWM3O)
	1	1	PPG repeat mode (PWM3O)
T3CC	Clear Timer 3 Counter		
	0	No effect	
	1	Clear the Timer 3 counter (When write, automatically cleared “0” after being cleared counter)	

T3CRL (Timer 3 Control Low Register): 1001H

7	6	5	4	3	2	1	0
T3CK2	T3CK1	T3CK0	T3IFR	–	T3POL	T3ECE	T3CNTR
R/W	R/W	R/W	R/W	–	R/W	R/W	R/W

Initial value: 00H

T3CK[2:0] Select Timer 3 clock source. fx is main system clock frequency

T3CK2	T3CK1	T3CK0	Description
0	0	0	fx/2048
0	0	1	fx/512
0	1	0	fx/64
0	1	1	fx/8
1	0	0	fx/4
1	0	1	fx/2
1	1	0	fx/1
1	1	1	External clock (EC3)

T3IFR When T3 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 T3 Interrupt no generation

1 T3 Interrupt generation

T3POL T3O/PWM3O Polarity Selection

0 Start High (T3O/PWM3O is low level at disable)

1 Start Low (T3O/PWM3O is high level at disable)

T3ECE Timer 3 External Clock Edge Selection

0 External clock falling edge

1 External clock rising edge

T3CNTR Timer 3 Counter Read Control

0 No effect

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

12.5 Timer 4

A 16-bit timer 4 consists of a multiplexer, timer 4 A data high/low register, timer 4 B data high/low register and timer 4 control high/low register (T4ADRH, T4ADRL, T4BDRH, T4BDRL, T4CRH, and T4CRL).

Timer 4 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 4 can be a divided clock of a system clock selected from prescaler output and T3 A Match (timer 3 A match signal). The clock source is selected by a clock selection logic controlled by clock selection bits (T4CK[2:0]).

- TIMER 4 clock source: fx/1, fx/2, fx4, fx/8, fx/32, fx/128, fx/512 and T3 A Match

In capture mode, data is captured into input capture data registers (T4BDRH/T4BDRL) by EINT44. In timer/counter mode, whenever counter value is equal to T4ADRH/L, T4O port toggles. In addition, the TIMER 4 outputs PWM waveform to PWM4O port in the PPG mode.

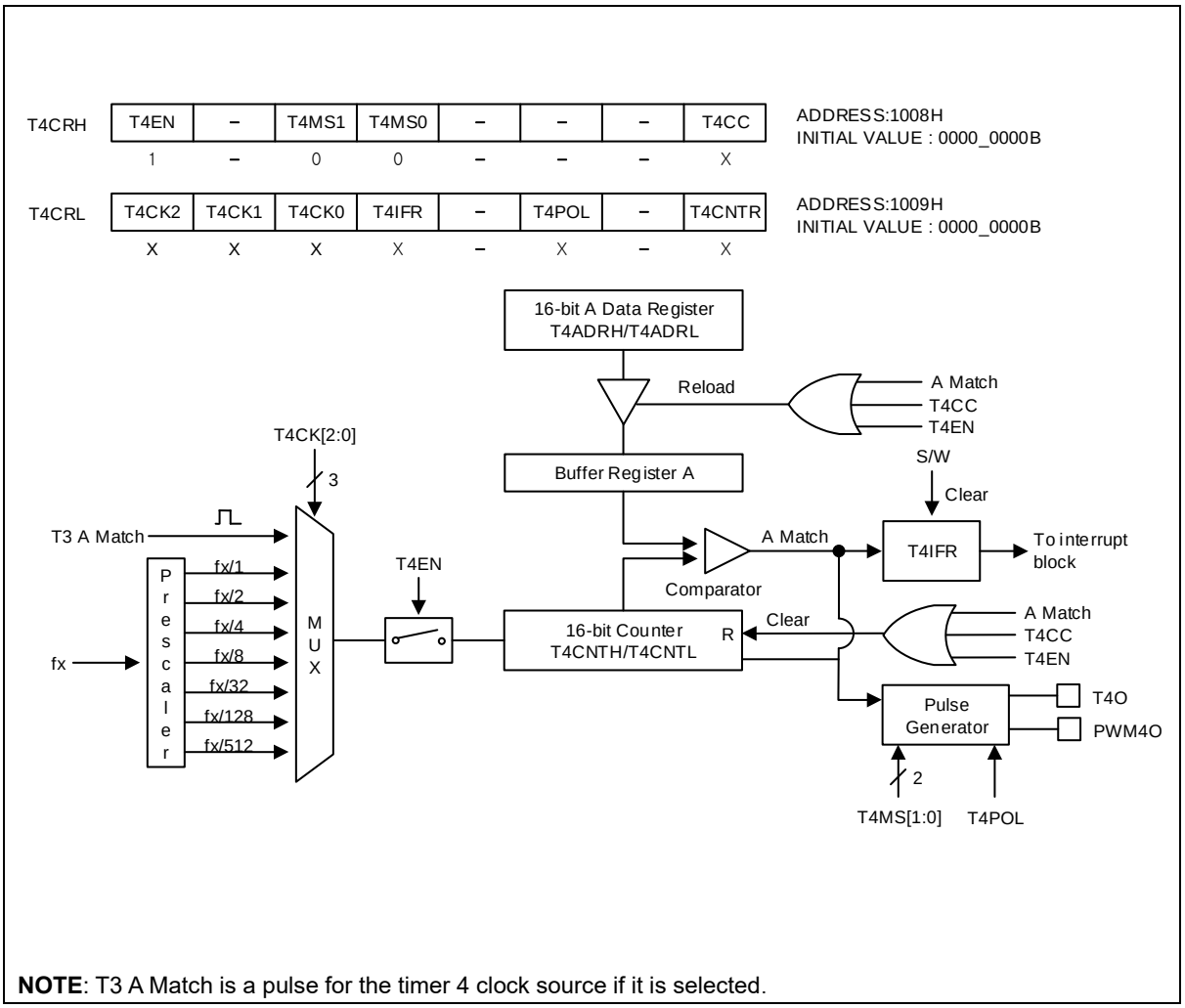
Table 22. TIMER 4 Operating Modes

T4EN	P2FSRH[7:6]	T4MS[1:0]	T4CK[2:0]	Timer 4
1	10	00	XXX	16 Bit Timer/Counter Mode
1	00	01	XXX	16 Bit Capture Mode
1	10	10	XXX	16 Bit PPG Mode(one-shot mode)
1	10	11	XXX	16 Bit PPG Mode(repeat mode)

12.5.1 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 58. The counter register is increased by internal or timer 4 A match clock input.

Timer 4 can use the input clock with one of 1, 2, 4, 8, 32, 128, 512 and T3 A Match prescaler division rates (T4CK[2:0]). When the values of T4CNTH/T4CNTL and T4ADRH/T4ADRL are identical to each other in timer 4, a match signal is generated and the interrupt of Timer 4 occurs. The T4CNTH/T4CNTL values are automatically cleared by the match signal. It can be cleared by software (T4CC) too.



NOTE: T3 A Match is a pulse for the timer 4 clock source if it is selected.

Figure 58. 16-bit Timer/Counter Mode of Timer 4

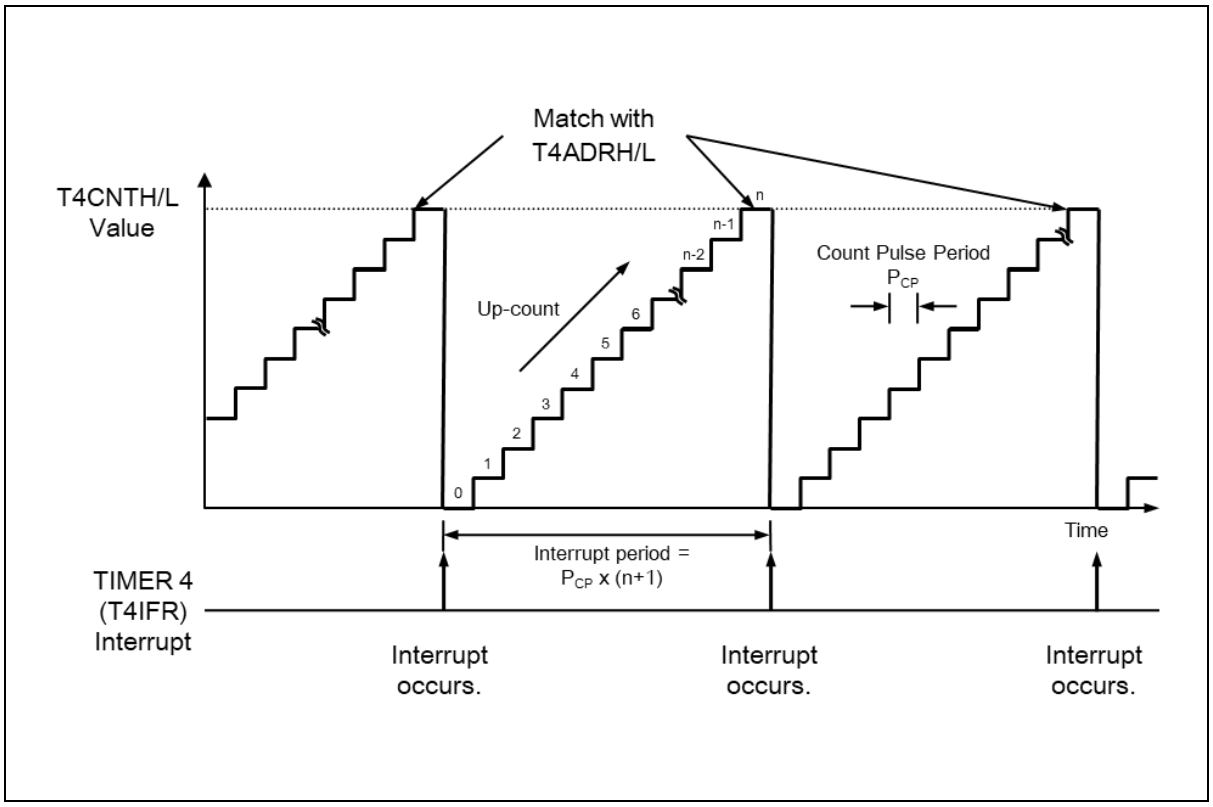


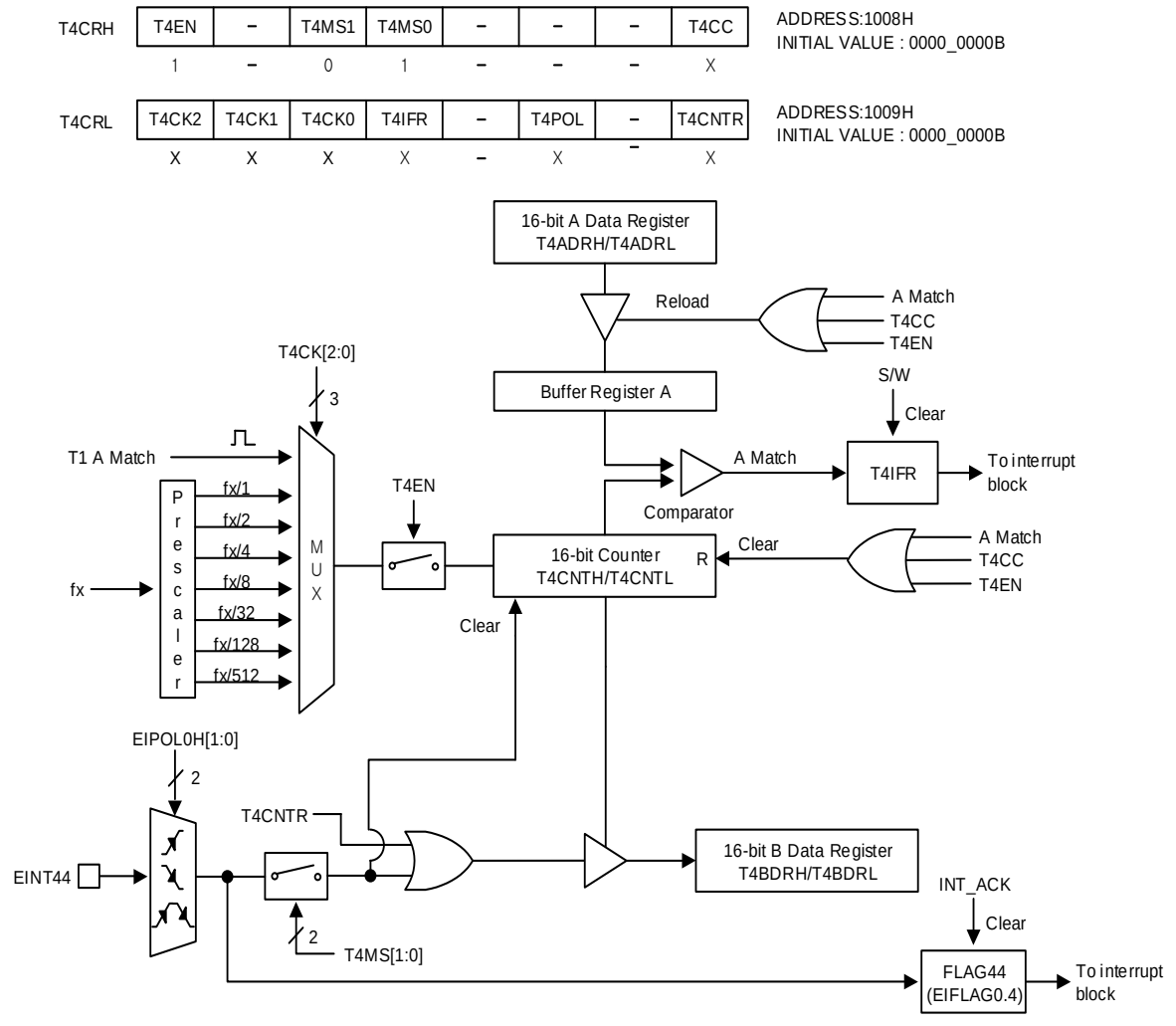
Figure 59. 16-bit Timer/Counter Mode Operation Example

12.5.2 16-bit capture mode

Timer 4 capture mode is set by configuring T4MS[1:0] as '01'. It uses an internal clock as a clock source. Basically, the 16-bit timer 4 capture mode has the same function as the 16-bit timer/counter mode, and the interrupt occurs when T4CNTH/T4CNTL is equal to T4ADRH/T4ADRL. The T4CNTH, T4CNTL values are automatically cleared by a match signal. It can be cleared by software (T4CC) 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 T4BDRH/T4BDRL. In the timer 4 capture mode, timer 4 output (T4O) waveform is not available.

According to EIPOL0L registers setting, the external interrupt EINT44 function is selected. EINT44 pin must be set as an input port.



NOTE: T3 A Match is a pulse for the timer 4 clock source if it is selected.

Figure 60. 16-bit Capture Mode of Timer 4

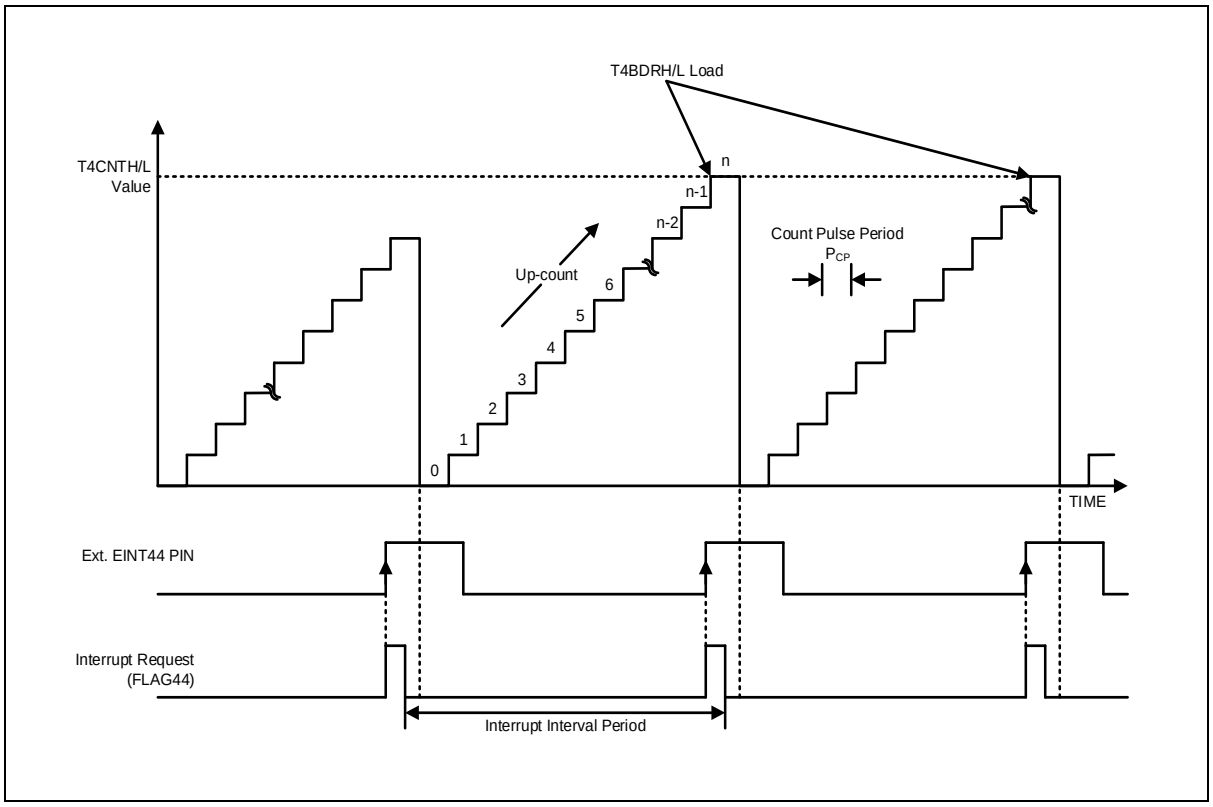


Figure 61. 16-bit Capture Mode Operation Example

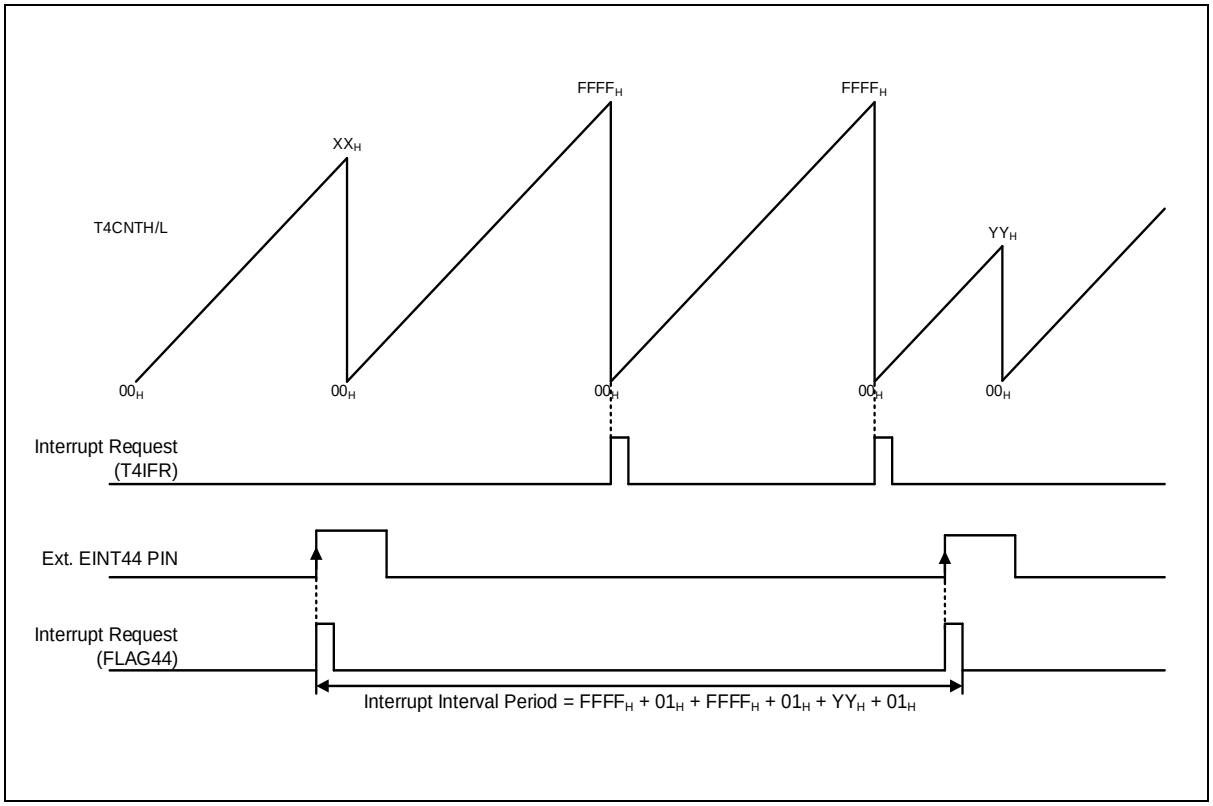
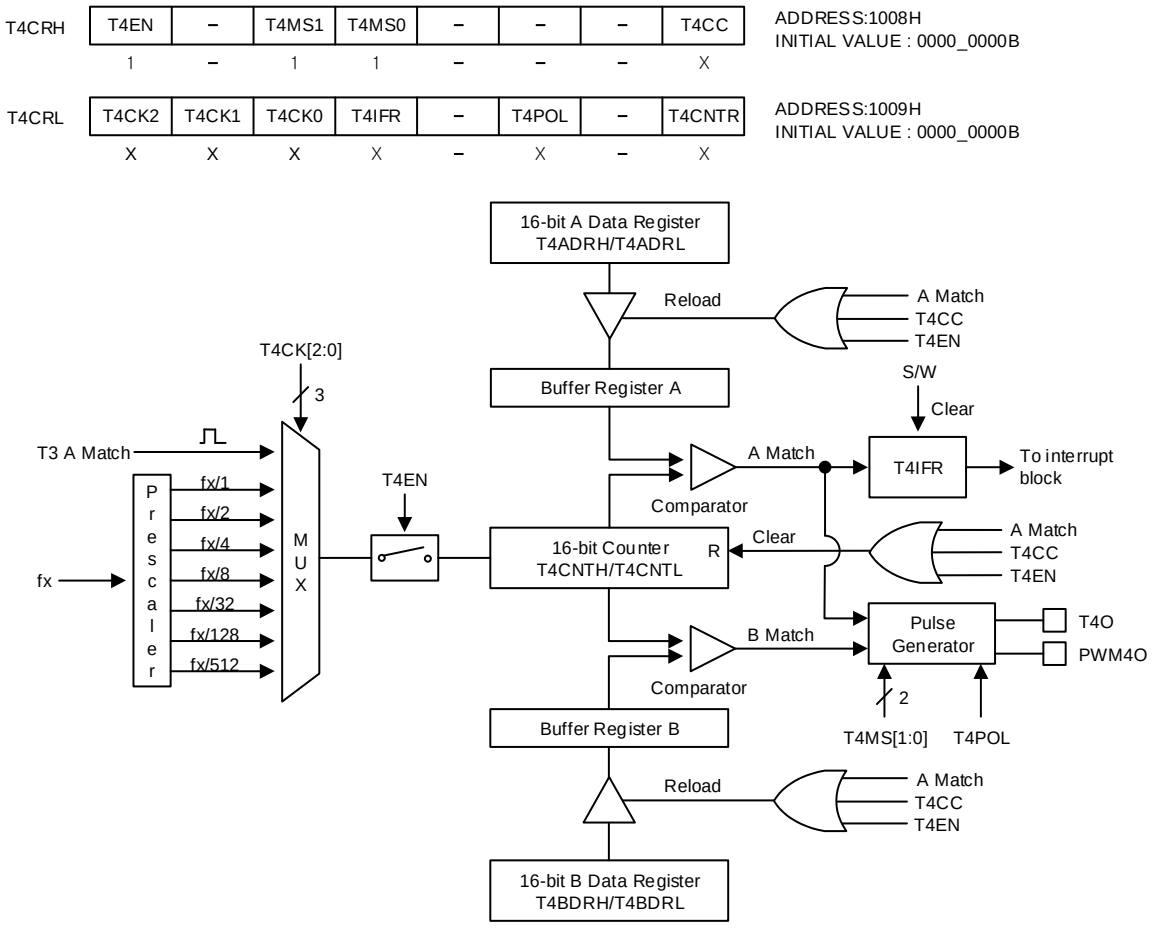


Figure 62. Express Timer Overflow in Capture Mode

12.5.3 16-bit PPG mode

TIMER 4 has a PPG (Programmable Pulse Generation) function. In PPG mode, T4O/PWM4O pin outputs up to 16-bit resolution PWM output. For this function, T4O/PWM4O pin must be configured as a PWM output by setting P2FSRH[7:6] to '10'. Period of the PWM output is determined by T4ADRH/T4ADRL, and duty of the PWM output is determined by T4BDRH/T4BDRL.



NOTES:

1. T4EN is automatically cleared to logic "0" after one pulse is generated at a PPG one-shot mode.
2. T3 A Match is a pulse for the timer 4 clock source if it is selected.

Figure 63. 16-bit PPG Mode of Timer 4

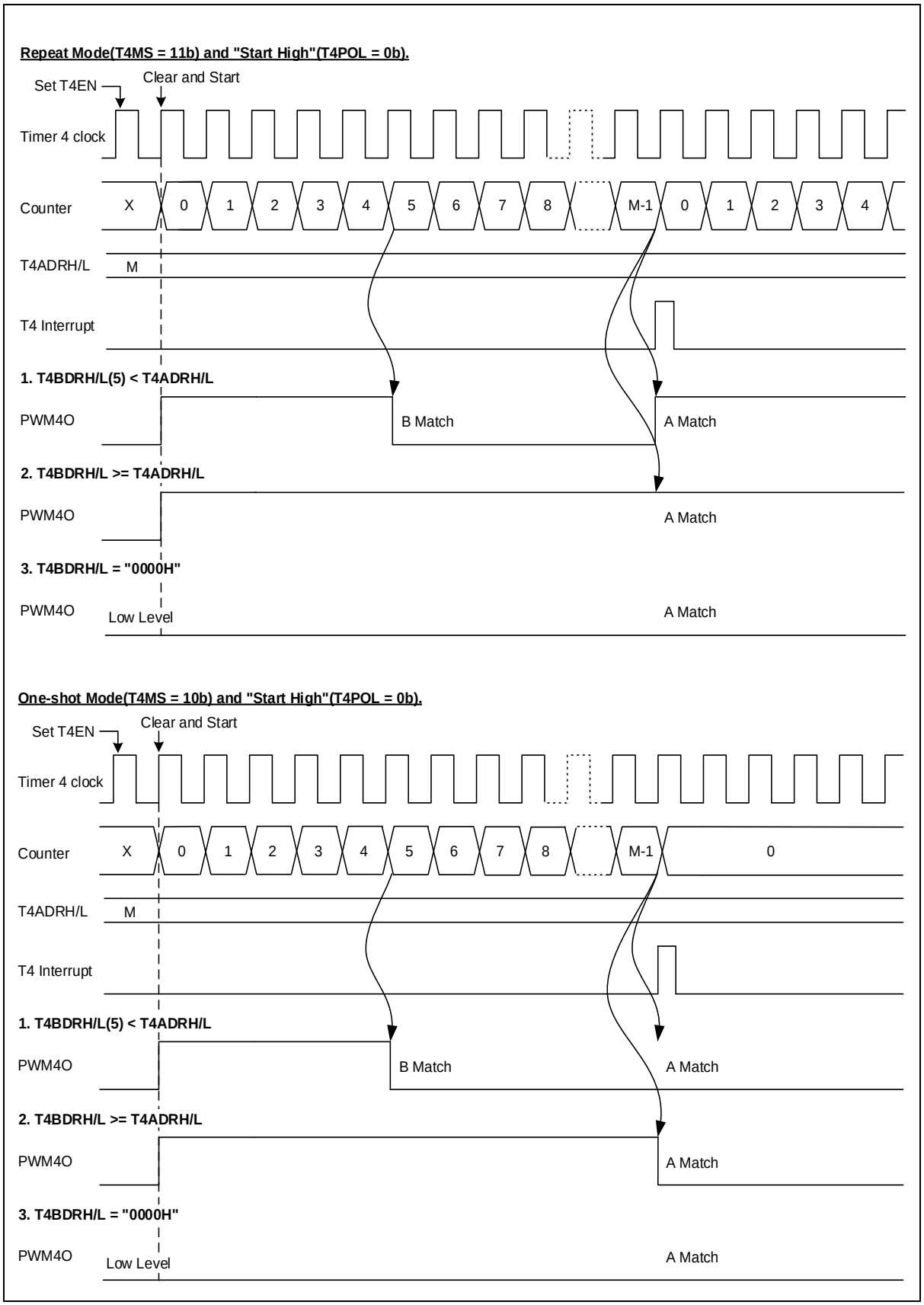
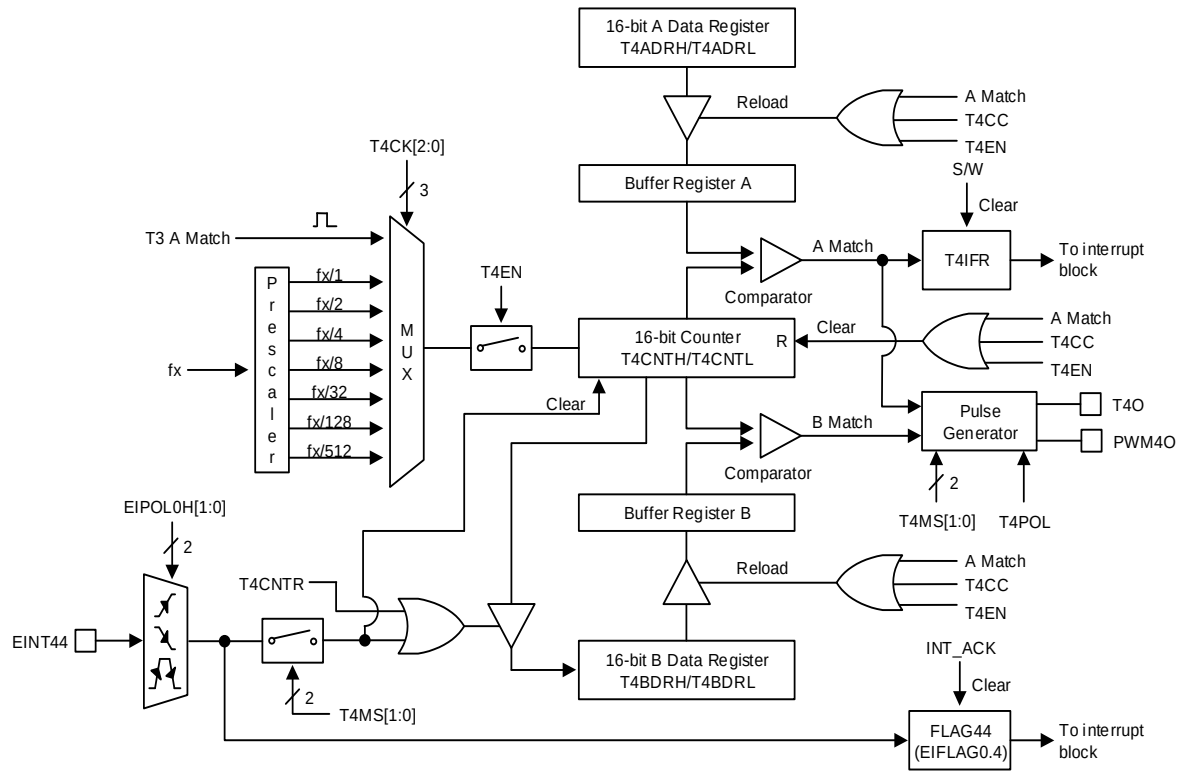


Figure 64. 16-bit PPG Mode Operation Example

12.5.4 16-bit timer 4 block diagram

In this section, a 16-bit timer 4 is described in a block diagram.



NOTE: T3 A Match is a pulse for the timer 4 clock source if it is selected.

Figure 65. 16-bit Timer 4 Block Diagram

12.5.5 Register map

Table 23. TIMER 4 Register Map

Name	Address	Direction	Default	Description
T4ADRH	100AH	R/W	FFH	Timer 4 A Data High Register
T4ADRL	100BH	R/W	FFH	Timer 4 A Data Low Register
T4BDRH	100CH	R/W	FFH	Timer 4 B Data High Register
T4BDRL	100DH	R/W	FFH	Timer 4 B Data Low Register
T4CRH	1008H	R/W	00H	Timer 4 Control High Register
T4CRL	1009H	R/W	00H	Timer 4 Control Low Register

12.5.6 Register description

T4ADRH (Timer 4 A data High Register): 100AH

7	6	5	4	3	2	1	0
T4ADRH7	T4ADRH6	T4ADRH5	T4ADRH4	T4ADRH3	T4ADRH2	T4ADRH1	T4ADRH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: FFH

T4ADRH[7:0] T4 A Data High Byte

T4ADRL (Timer 4 A Data Low Register): 100BH

7	6	5	4	3	2	1	0
T4ADRL7	T4ADRL6	T4ADRL5	T4ADRL4	T4ADRL3	T4ADRL2	T4ADRL1	T4ADRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: FFH

T4ADRL[7:0] T4 A Data Low Byte

NOTE: Do not write “0000H” in the T4ADRH/T4ADRL register when PPG mode.

T4BDRH (Timer 4 B Data High Register): 100CH

7	6	5	4	3	2	1	0
T4BDRH7	T4BDRH6	T4BDRH5	T4BDRH4	T4BDRH3	T4BDRH2	T4BDRH1	T4BDRH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: FFH

T4BDRH[7:0] T4 B Data High Byte

T4BDRL (Timer 4 B Data Low Register): 100DH

7	6	5	4	3	2	1	0
T4BDRL7	T4BDRL6	T4BDRL5	T4BDRL4	T4BDRL3	T4BDRL2	T4BDRL1	T4BDRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: FFH

T4BDRL[7:0] T4 B Data Low

T4CRH (Timer 4 Control High Register): 1008H

7	6	5	4	3	2	1	0
T4EN	–	T4MS1	T4MS0	–	–	–	T4CC
R/W	–	R/W	R/W	–	–	–	R/W

Initial value: 00H

T4EN Control Timer 4

0 Timer 4 disable

1 Timer 4 enable (Counter clear and start)

T4MS[1:0] Control Timer 4 Operation Mode

T4MS1	T4MS0	Description
0	0	Timer/counter mode (T4O: toggle at A match)
0	1	Capture mode (The A match interrupt can occur)
1	0	PPG one-shot mode (PWM4O)
1	1	PPG repeat mode (PWM4O)

T4CC Clear Timer 4 Counter

0 No effect

1 Clear the Timer 4 counter (When write, automatically cleared “0” after being cleared counter)

T4CRL (Timer 4 Control Low Register): 1009H

7	6	5	4	3	2	1	0
T4CK2	T4CK1	T4CK0	T4IFR	–	T4POL	–	T4CNTR
R/W	R/W	R/W	R/W	–	R/W	–	R/W

Initial value: 00H

T4CK[2:0] Select Timer 4 clock source. fx is main system clock frequency

T4CK2	T4CK1	T4CK0	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	T3 A Match

T4IFR When T4 Match Interrupt occurs, this bit becomes ‘1’. For clearing bit, write ‘0’ to this bit. Writing “1” has no effect.

0 T4 interrupt no generation

1 T4 interrupt generation

T4POL T4O/PWM4O Polarity Selection

0 Start High (T4O/PWM4O is low level at disable)

1 Start Low (T4O/PWM4O is high level at disable)

T4CNTR Timer 4 Counter Read Control

0 No effect

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

12.6 Timer 5

A 16-bit timer 5 consists of a multiplexer, timer 5 A data high/low register, timer 5 B data high/low register and timer 5 control high/low register (T5ADRH, T5ADRL, T5BDRH, T5BDRL, T5CRH, and T5CRL).

Timer 5 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 5 can be a divided clock of a system clock selected from prescaler output. The clock source is selected by a clock selection logic controlled by clock selection bits (T5CK[2:0]).

- TIMER 5 clock source: fx/1, fx/2, fx/4, fx/8, fx/32, fx/128, fx/512 and HSIRC

In capture mode, data is captured into input capture data registers (T5BDRH/T5BDRL) by EINT45. In timer/counter mode, whenever counter value is equal to T5ADRH/L, T5O port toggles. In addition, the TIMER 5 outputs PWM waveform to PWM5O port in the PPG mode.

Table 24. TIMER 5 Operating Modes

T5EN	P2FSRH[5:4]	T5MS[1:0]	T5CK[2:0]	Timer 5
1	10	00	XXX	16 Bit Timer/Counter Mode
1	00	01	XXX	16 Bit Capture Mode
1	10	10	XXX	16 Bit PPG Mode(one-shot mode)
1	10	11	XXX	16 Bit PPG Mode(repeat mode)

12.6.1 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 66. The counter register is increased by internal clock input.

Timer 5 can use the input clock with one of 1, 2, 4, 8, 32, 128, 512 and High Frequency Internal Oscillator (HSIRC) prescaler division rates (T5CK[2:0]). When the values of T5CNTH/T5CNTL and T5ADRH/T5ADRL are identical to each other in timer 5, a match signal is generated and the interrupt of Timer 5 occurs. The T5CNTH/T5CNTL values are automatically cleared by the match signal. It can be cleared by software (T5CC) too.

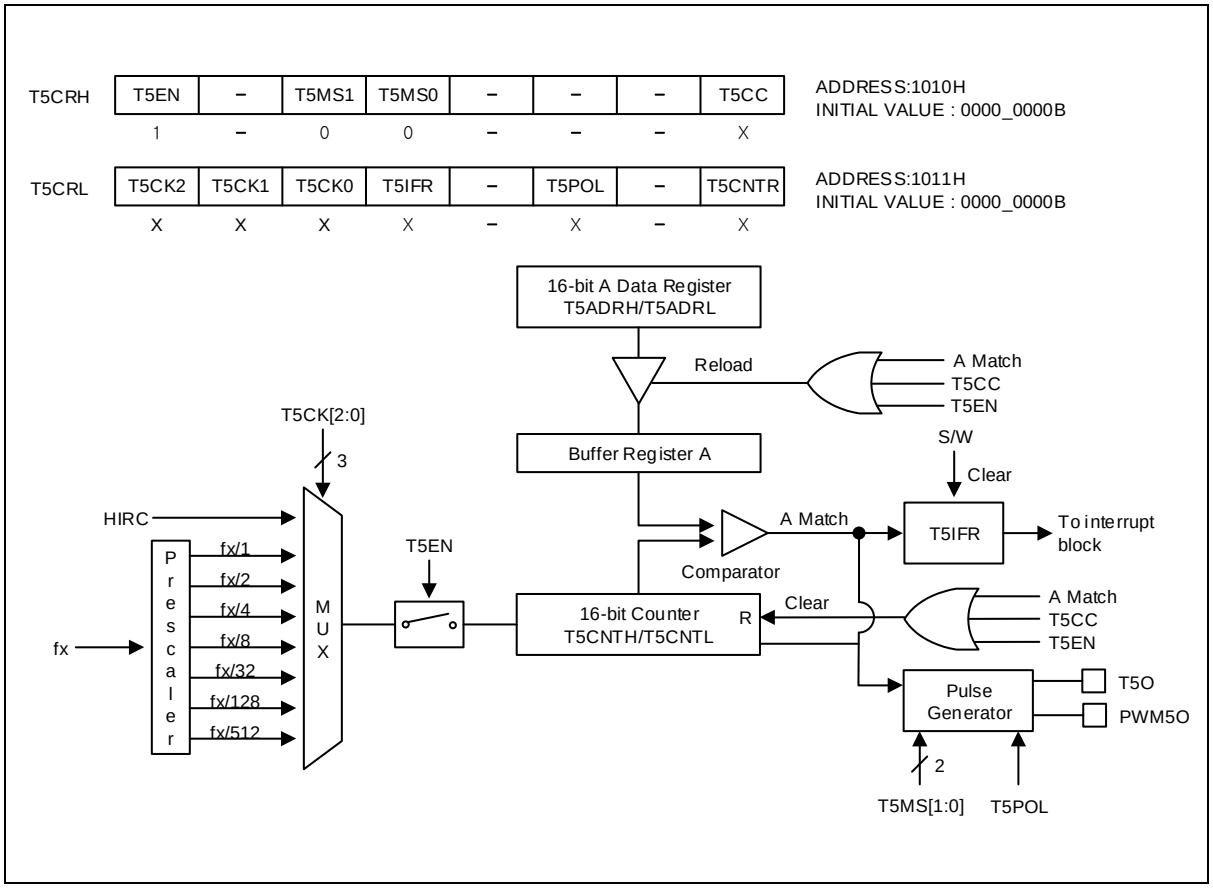


Figure 66. 16-bit Timer/Counter Mode of Timer 5

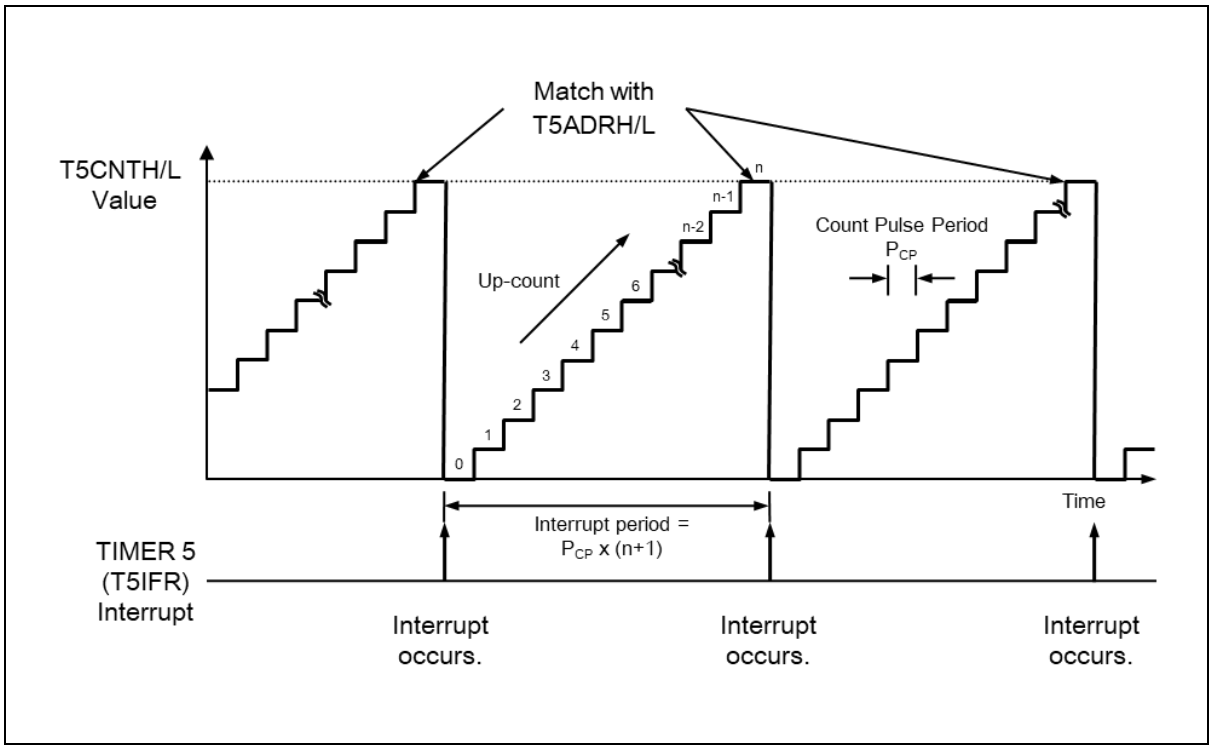


Figure 67. 16-bit Timer/Counter Mode Operation Example

12.6.2 16-bit capture mode

Timer 5 capture mode is set by configuring T5MS[1:0] as '01'. It uses an internal clock as a clock source. Basically, the 16-bit timer 5 capture mode has the same function as the 16-bit timer/counter mode, and the interrupt occurs when T5CNTH/T5CNTL is equal to T5ADRH/T5ADRL. The T5CNTH, T5CNTL values are automatically cleared by a match signal. It can be cleared by software (T5CC) 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 T5BDRH/T5BDRL. In the timer 5 capture mode, timer 5 output (T5O) waveform is not available.

According to EIPOL0H registers setting, the external interrupt EINT45 function is selected. EINT45 pin must be set as an input port.

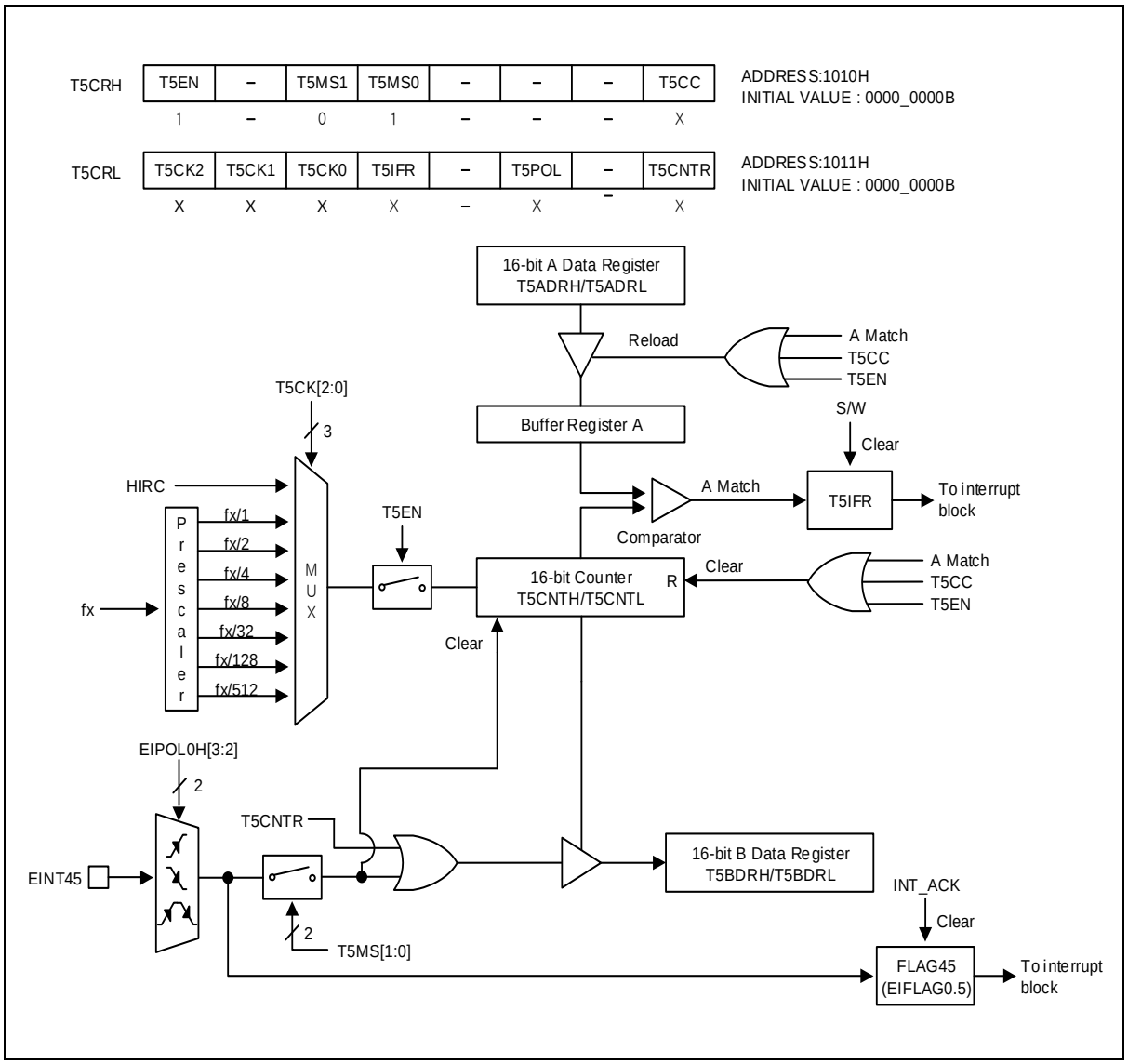


Figure 68. 16-bit Capture Mode of Timer 5

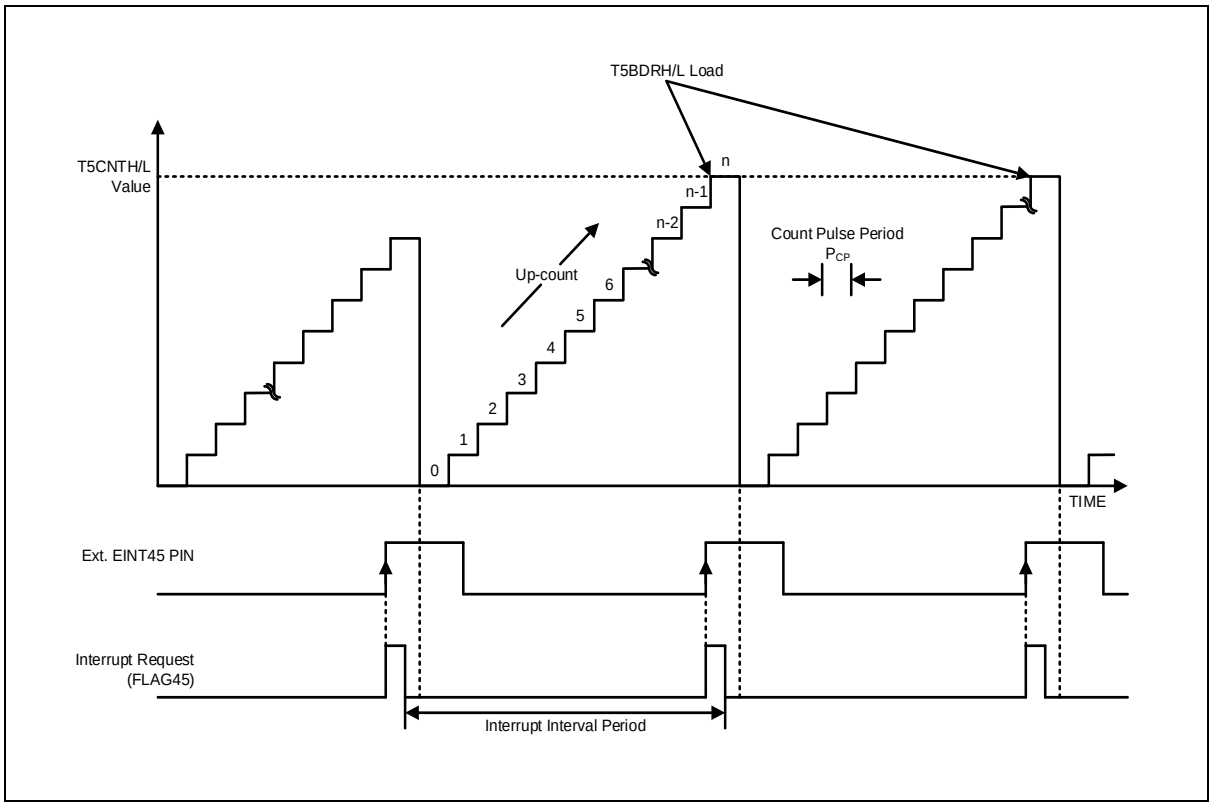


Figure 69. 16-bit Capture Mode Operation Example

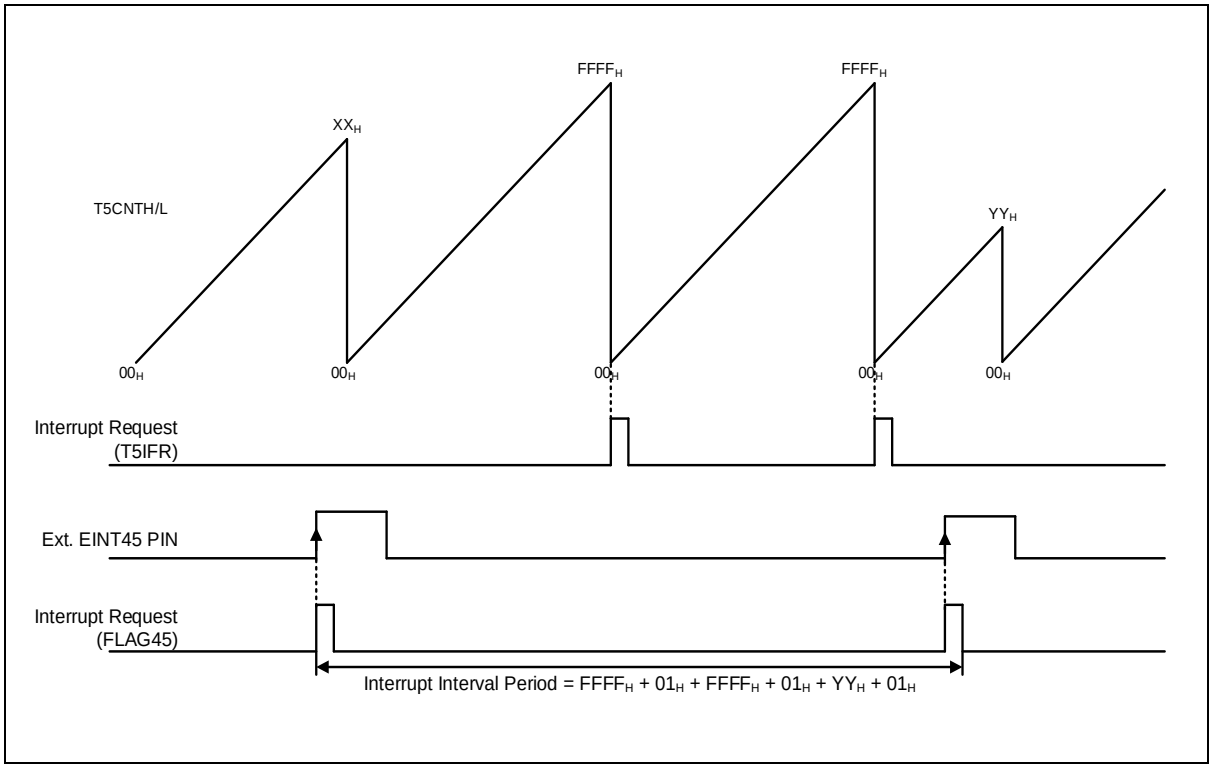
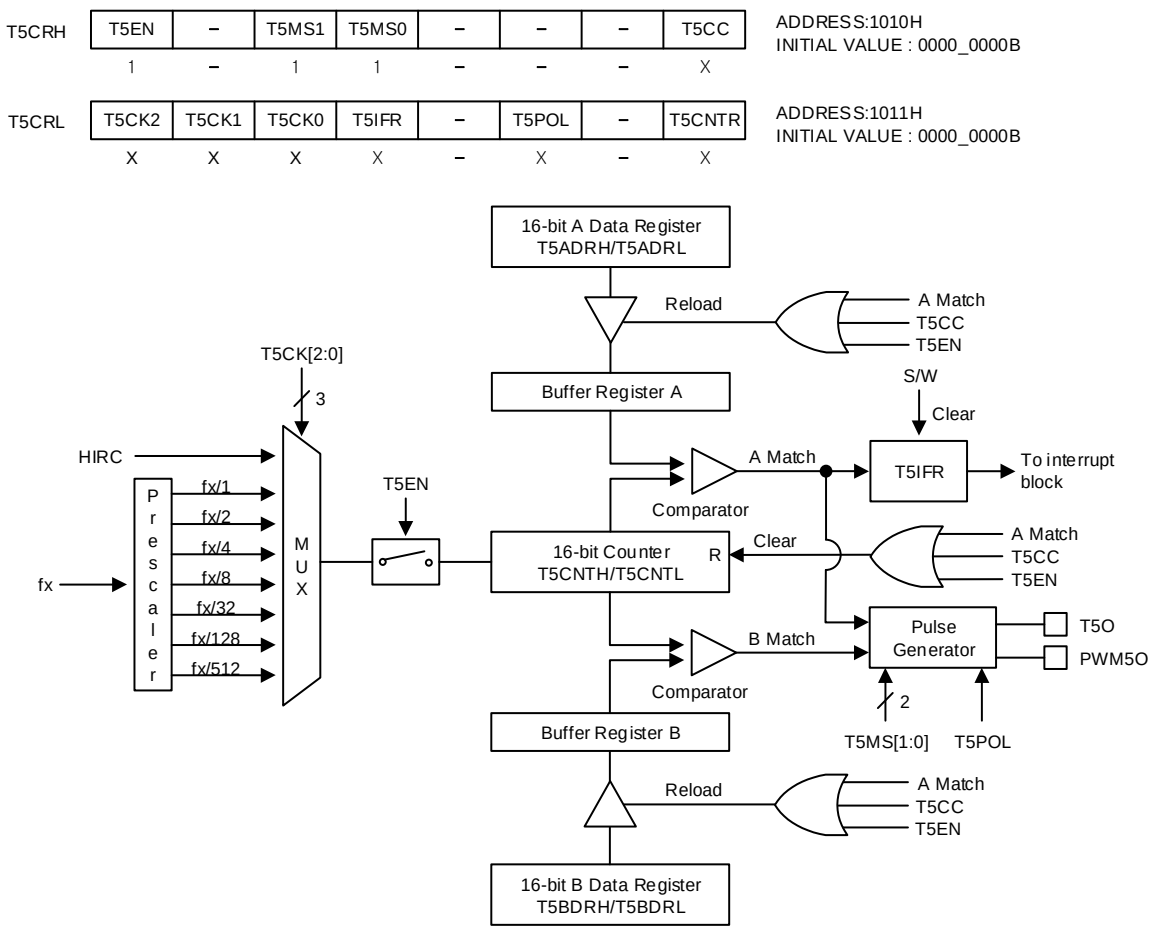


Figure 70. Express Timer Overflow in Capture Mode

12.6.3 16-bit PPG mode

TIMER 5 has a PPG (Programmable Pulse Generation) function. In PPG mode, T5O/PWM5O pin outputs up to 16-bit resolution PWM output. For this function, T5O/PWM5O pin must be configured as a PWM output by setting P2FSRH[5:4] to '10'. Period of the PWM output is determined by T5ADRH/T5ADRL, and duty of the PWM output is determined by T5BDRH/T5BDRL.



NOTE: T5EN is automatically cleared to logic "0" after one pulse is generated at a PPG one-shot mode.

Figure 71. 16-bit PPG Mode of Timer 5

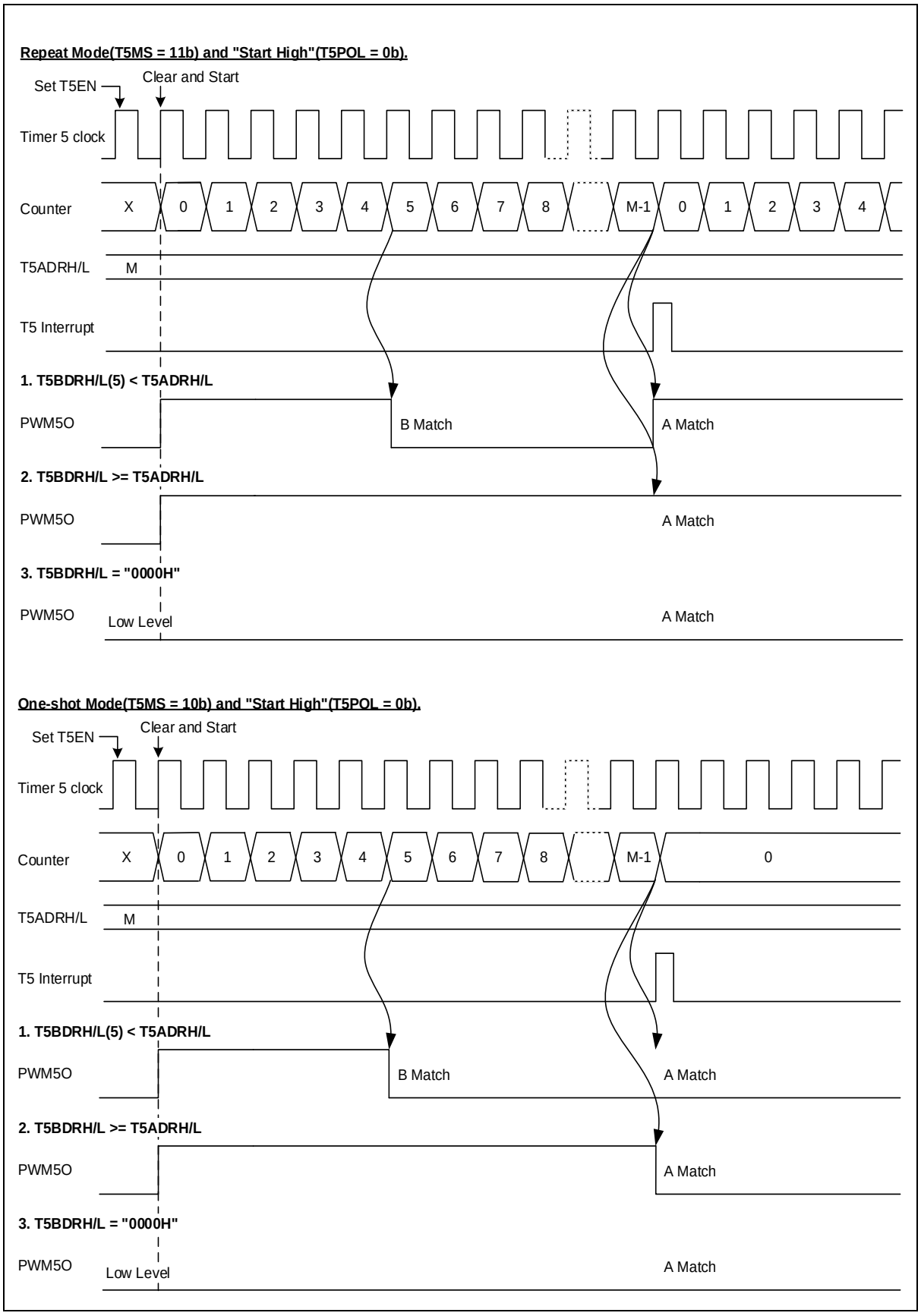


Figure 72. 16-bit PPG Mode Operation Example

12.6.4 16-bit timer 5 block diagram

In this section, a 16-bit timer 5 is described in a block diagram.

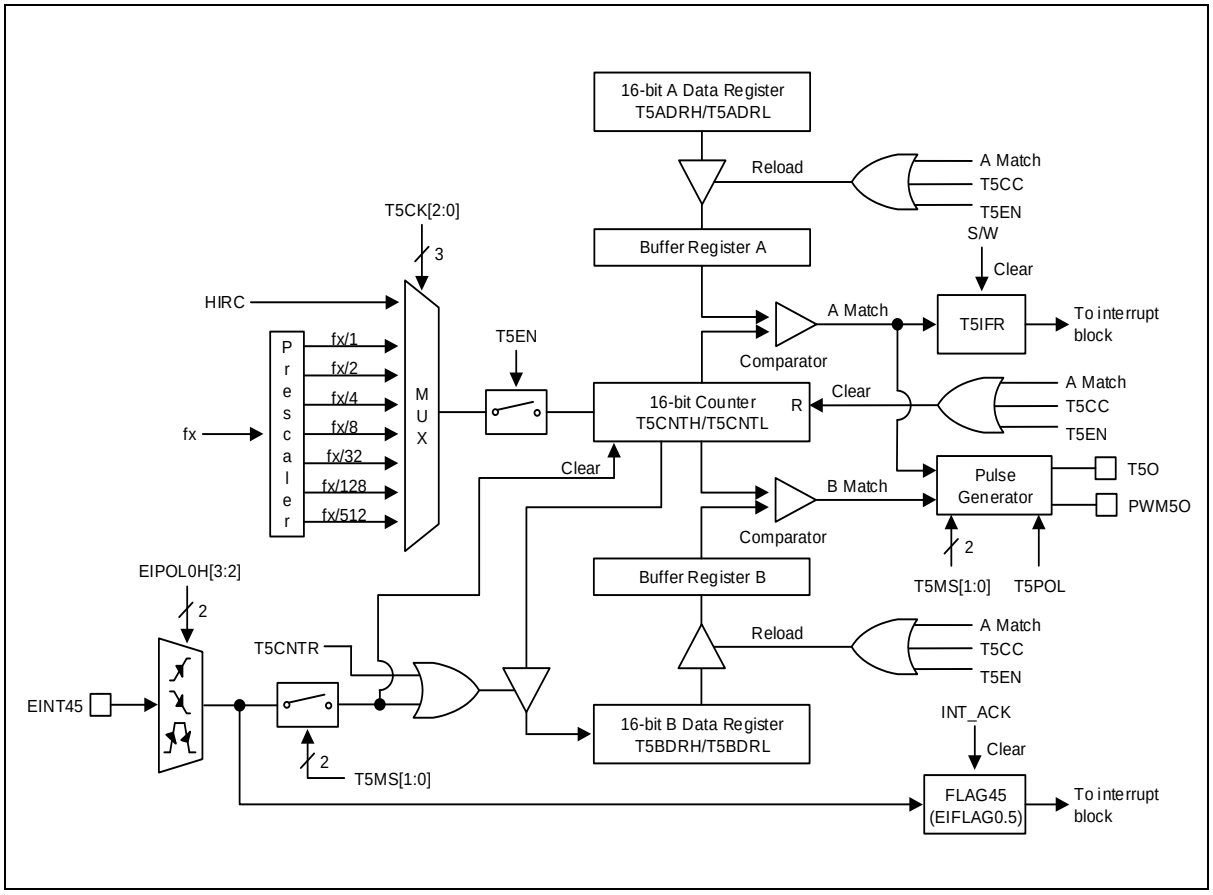


Figure 73. 16-bit Timer 5 Block Diagram

12.6.5 Register map

Table 25. TIMER 5 Register Map

Name	Address	Direction	Default	Description
T5ADRH	1012H	R/W	FFH	Timer 5 A Data High Register
T5ADRL	1013H	R/W	FFH	Timer 5 A Data Low Register
T5BDRH	1014H	R/W	FFH	Timer 5 B Data High Register
T5BDRL	1015H	R/W	FFH	Timer 5 B Data Low Register
T5CRH	1010H	R/W	00H	Timer 5 Control High Register
T5CRL	1011H	R/W	00H	Timer 5 Control Low Register

12.6.6 Register description

T5ADRH (Timer 5 A data High Register): 1012H

7	6	5	4	3	2	1	0
T5ADRH7	T5ADRH6	T5ADRH5	T5ADRH4	T5ADRH3	T5ADRH2	T5ADRH1	T5ADRH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: FFH

T5ADRH[7:0] T5 A Data High Byte

T5ADRL (Timer 5 A Data Low Register): 1013H

7	6	5	4	3	2	1	0
T5ADRL7	T5ADRL6	T5ADRL5	T5ADRL4	T5ADRL3	T5ADRL2	T5ADRL1	T5ADRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: FFH

T5ADRL[7:0] T5 A Data Low Byte

NOTE: Do not write “0000H” in the T5ADRH/T5ADRL register when PPG mode.

T5BDRH (Timer 5 B Data High Register): 1014H

7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
T5BDRH7	T5BDRH6	T5BDRH5	T5BDRH4	T5BDRH3	T5BDRH2	T5BDRH1	T5BDRH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: FFH

T5BDRH[7:0] T5 B Data High Byte

T5BDRL (Timer 5 B Data Low Register): 1015H

7	6	5	4	3	2	1	0
T5BDRL7	T5BDRL6	T5BDRL5	T5BDRL4	T5BDRL3	T5BDRL2	T5BDRL1	T5BDRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: FFH

T5BDRL[7:0] T5 B Data Low

T5CRH (Timer 5 Control High Register): 1010H

7	6	5	4	3	2	1	0
T5EN	–	T5MS1	T5MS0	–	–	–	T5CC
R/W	–	R/W	R/W	–	–	–	R/W

Initial value: 00H

T5EN Control Timer 5

0 Timer 5 disable

1 Timer 5 enable (Counter clear and start)

T5MS[1:0] Control Timer 5 Operation Mode

T5MS1	T5MS0	Description
0	0	Timer/counter mode (T5O: toggle at A match)
0	1	Capture mode (The A match interrupt can occur)
1	0	PPG one-shot mode (PWM5O)
1	1	PPG repeat mode (PWM5O)

T5CC Clear Timer 5 Counter

0 No effect

1 Clear the Timer 5 counter (When write, automatically cleared “0” after being cleared counter)

T5CRL (Timer 5 Control Low Register): 1011H

7	6	5	4	3	2	1	0
T5CK2	T5CK1	T5CK0	T5IFR	–	T5POL	–	T5CNTR
R/W	R/W	R/W	R/W	–	R/W	–	R/W

Initial value: 00H

T5CK[2:0] Select Timer 5 clock source. fx is main system clock frequency

T4CK2	T4CK1	T4CK0	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	HSIRC Direct (32MHz)

T5IFR When T5 Match Interrupt occurs, this bit becomes ‘1’. For clearing bit, write ‘0’ to this bit. Writing “1” has no effect.

0 T5 interrupt no generation

1 T5 interrupt generation

T5POL T5O/PWM5O Polarity Selection

0 Start High (T5O/PWM5O is low level at disable)

1 Start Low (T5O/PWM5O is high level at disable)

T5CNTR Timer 5 Counter Read Control

0 No effect

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

13 Buzzer driver

A buzzer of A96G150 consists of 8-bit counter, a buzzer data register (BUZDR), and a buzzer control register (BUZCR). It outputs square wave (61.035Hz to 125.0KHz @ 8MHz) through P53/BUZO pin, and its buzzer data register (BUZDR) controls the buzzer frequency (refer to the following expression). In a buzzer control register (BUZCR), BUCK[1:0] bits select a source clock divided by prescaler.

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

Table 26. Buzzer Frequency at 8MHz

BUZDR[7:0]	Buzzer frequency (KHz)			
	BUZCR[2:1]=00	BUZCR[2:1]=01	BUZCR[2:1]=10	BUZCR[2:1]=11
0000_0000	125KHz	62.5KHz	31.25KHz	15.625KHz
0000_0001	62.5KHz	31.25KHz	15.625KHz	7.812KHz
…	…	…	…	…
1111_1101	492.126Hz	246.063Hz	123.031Hz	61.515Hz
1111_1110	490.196Hz	245.098Hz	122.549Hz	61.274Hz
1111_1111	488.281Hz	244.141Hz	122.07Hz	61.035Hz

13.1 Buzzer driver block diagram

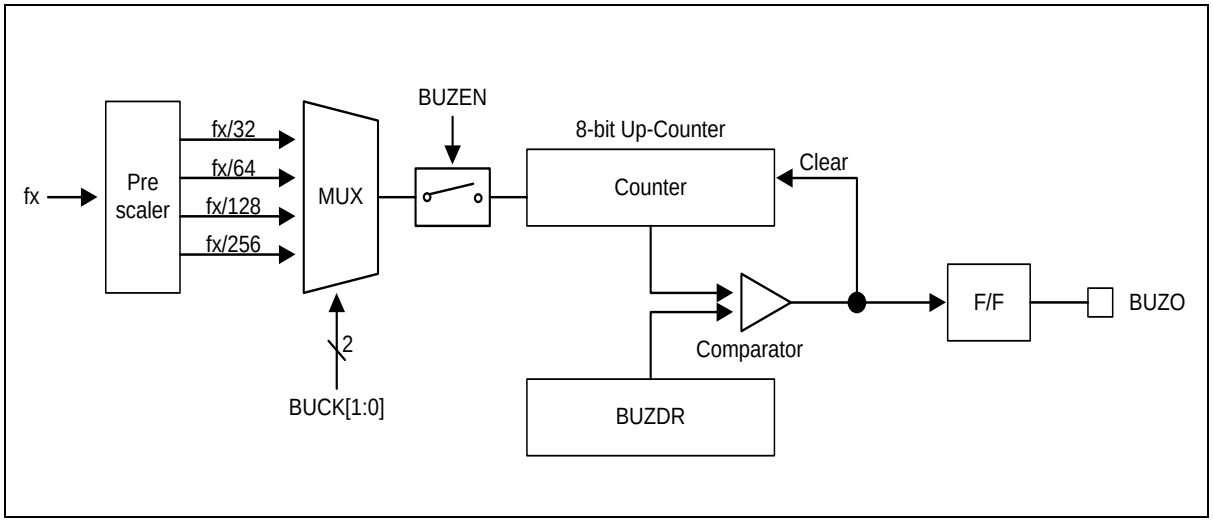


Figure 74. Buzzer Driver Block Diagram

13.2 Register map

Table 27. Buzzer Driver Register Map

Name	Address	Direction	Default	Description
BUZDR	8FH	R/W	FFH	Buzzer Data Register
BUZCR	97H	R/W	00H	Buzzer Control Register

13.3 Register description

BUZDR (Buzzer Data Register): 8FH

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

Initial value: FFH

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

BUZCR (Buzzer Control Register): 97H

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

Initial value: 00H

BUCK[1:0] Buzzer Driver Source Clock Selection

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

BUZEN Buzzer Driver Operation Control

0	Buzzer Driver disable
1	Buzzer Driver enable

NOTE: fx: System clock oscillation frequency.

14 12-bit ADC

Analog-to-digital converter (ADC) of A96G150 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’.

14.1 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 fx/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.

14.2 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.

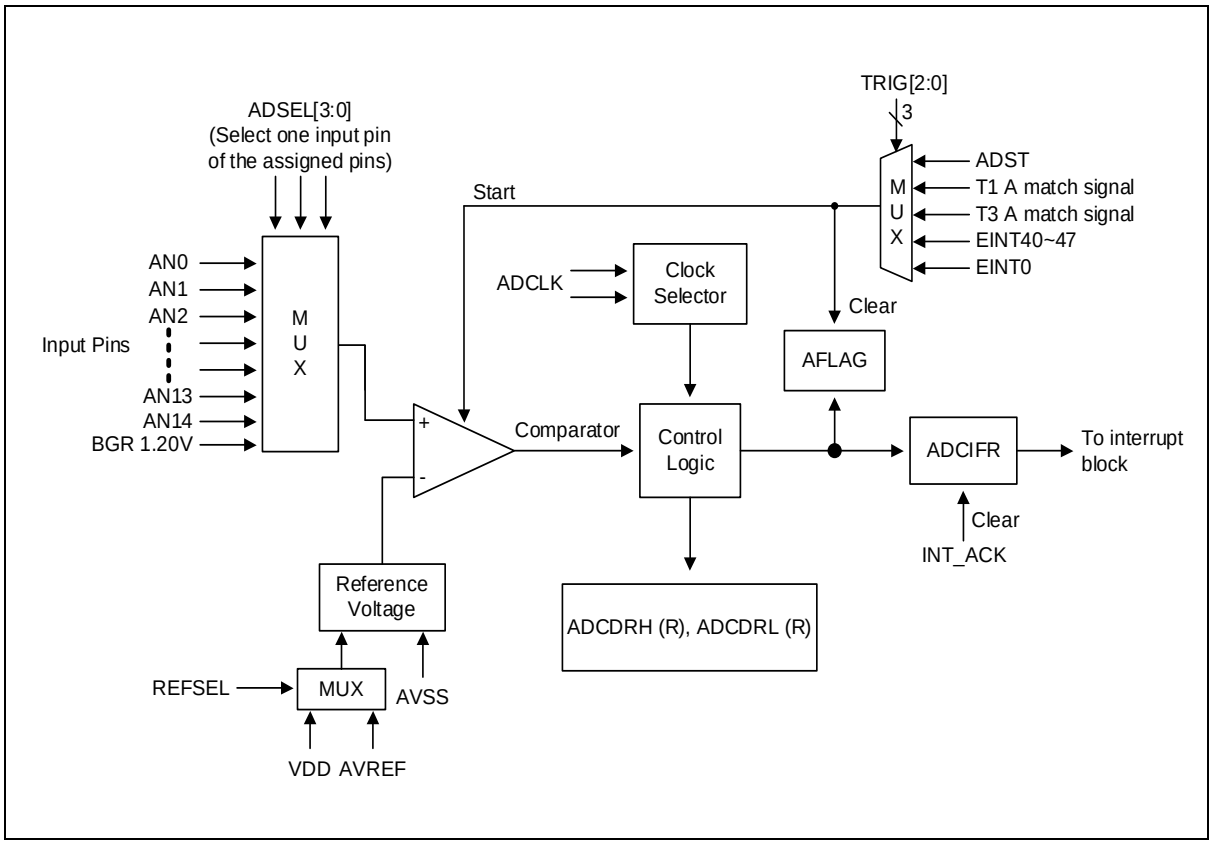


Figure 75. 12-bit ADC Block Diagram

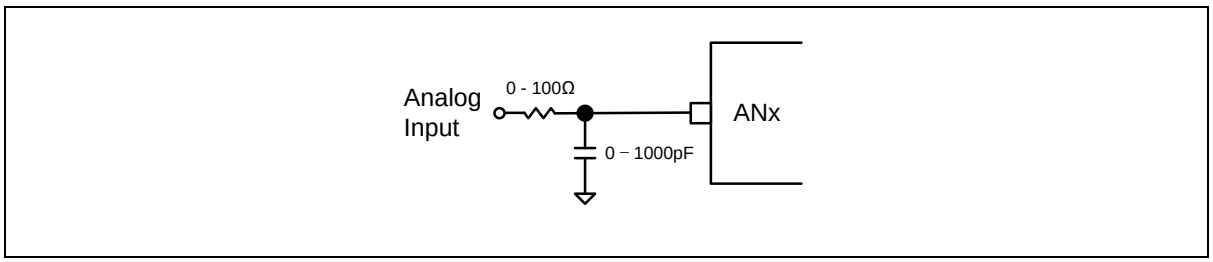


Figure 76. A/D Analog Input Pin with a Capacitor

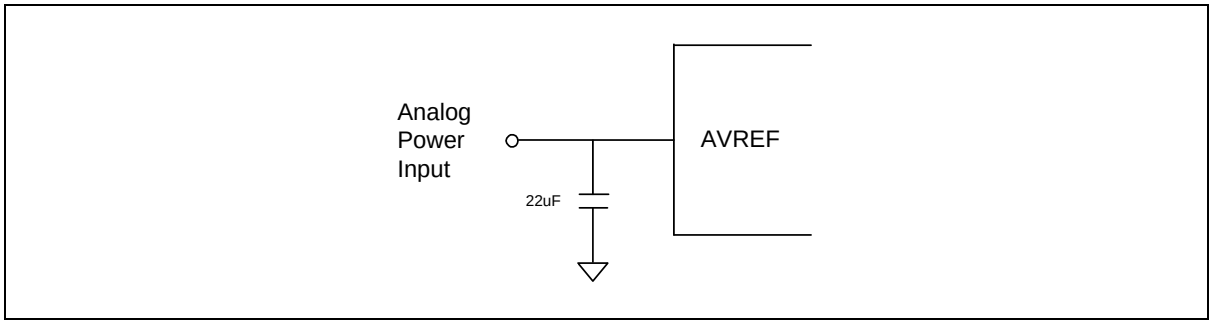


Figure 77. A/D Power (AVREF) Pin with a Capacitor

14.3 ADC operation

In this section, control registers and align bits are introduced in Figure 78, and ADC operation flow sequence is introduced in Figure 79.

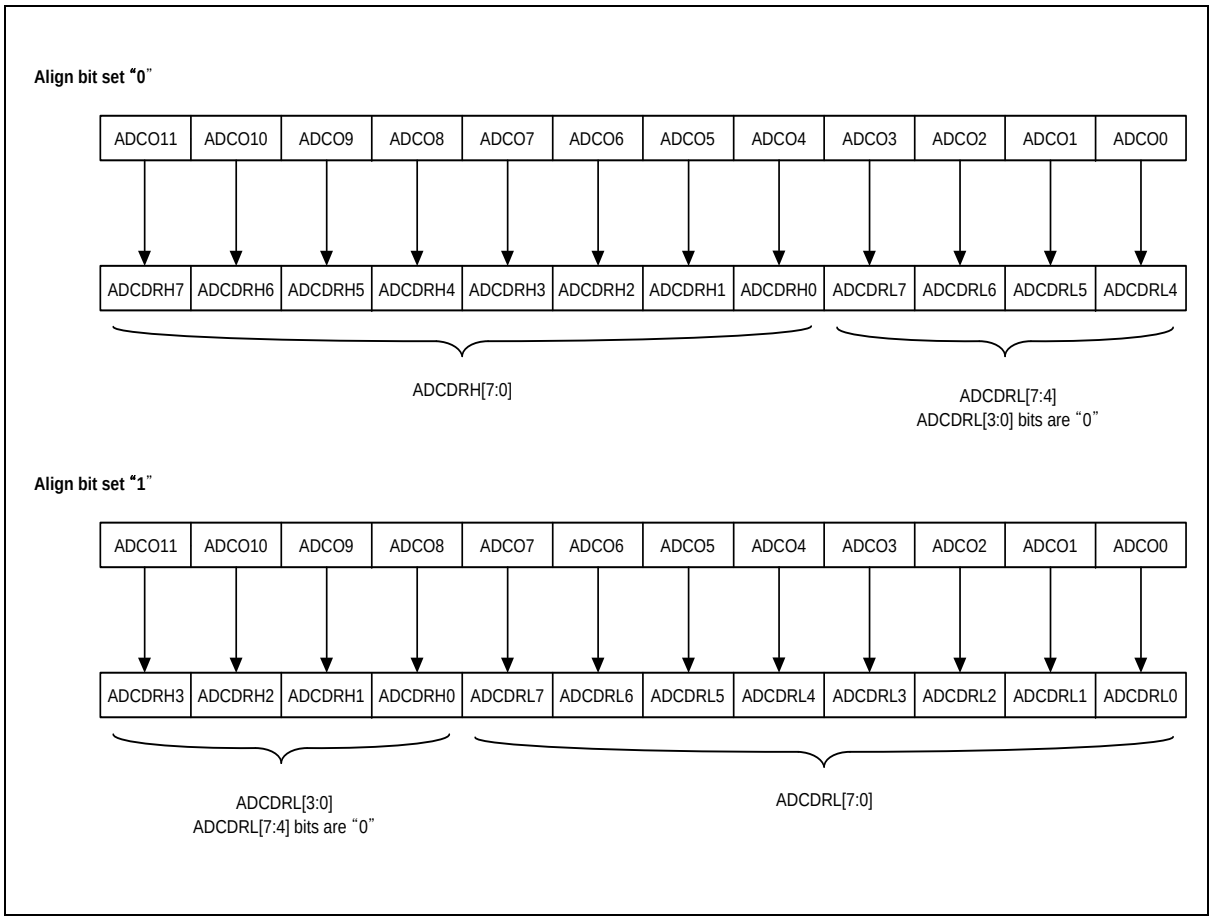


Figure 78. Control Registers and Align Bits

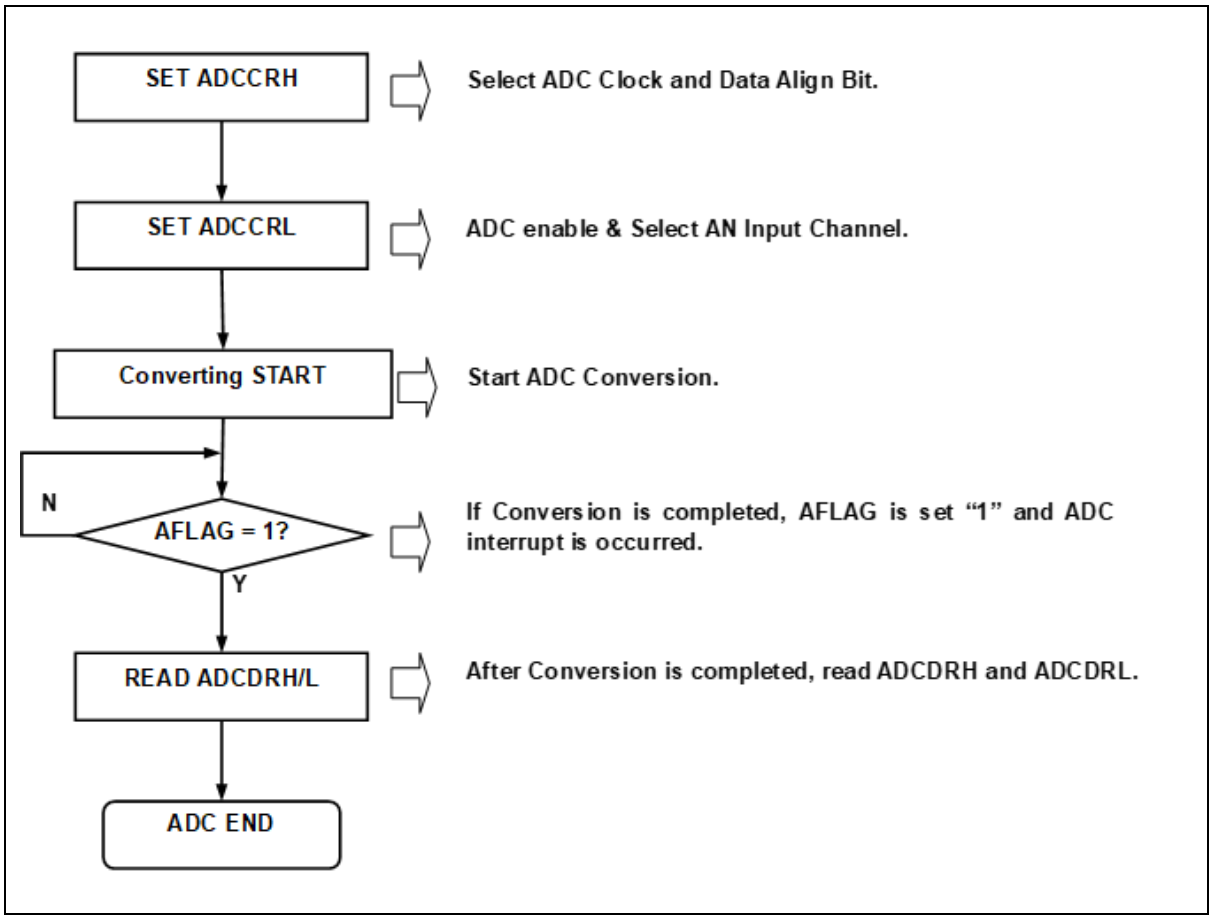


Figure 79. ADC Operation Flow Sequence

14.4 Register map

Table 28. 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

14.5 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 (**Keep always '0'**)

0	External input signal source select (AN0, AN1 … AN14)
1	Reserved

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

TRIG2	TRIG1	TRIG0	Description
0	0	0	ADST
0	0	1	Timer 1 A match signal
0	1	0	Timer 3 A match signal
0	1	1	EINT40~47
1	0	0	EINT0
1	0	1	Not used
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 External Reference (AVREF)

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	BGR 1.20V

15 Combination of USART, SPI, and I2C (USI)

USI stands for the combination of USART, SPI and I2C. A96G150 has two USI function blocks, USI0 and USI1, which are identical to each other functionally. Each USI block consists of USI control registers 1/2/3/4, USI status registers 1/2, USI baud-rate generation register, USI data register, USI SDA hold time register, USI SCL high period register, USI SCL low period register, and USI slave address register (USInCR1, USInCR2, USInCR3, USInCR4, USInST1, USInST2, USInBD, USInDR, USInSDHR, USInSCHR, USInSCLR, USInSAR). The ‘n’ means ‘0’ or ‘1’.

USI operates in one of the following modes selected by USIn selection bits (USInMS[1:0]):

- Asynchronous mode (UART)
- Synchronous mode (USART)
- SPI mode
- I2C mode

15.1 USIn UART mode

Universal synchronous and asynchronous serial receiver and transmitter (USART) are highly flexible serial communication devices. Main features are listed below:

- Full Duplex Operation (Independent Serial Receive and Transmit Registers)
- Asynchronous or Synchronous Operation
- 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 are Supported by Hardware
- Data OverRun Detection
- Framing Error Detection
- Three Separate Interrupts on TX Completion, TX Data Register Empty and RX Completion
- Double Speed Asynchronous communication mode

The USIn comprises clock generator, transmitter and receiver. Clock generation logic consists of synchronization logic for external clock input used by synchronizing or SPI slave operation, and the 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. A write buffer allows continuous transfer of data without any delay between frames.

Receiver is the most complex part of the UART module due to its clock and data recovery units. The recovery unit is used for asynchronous data reception. In addition to the recovery unit, the receiver includes a parity checker, a shift register, a two-level receive FIFO (USInDR) and control logic. The receiver supports the same frame formats as the transmitter and can detect frame error, data overrun and parity errors.

15.2 USIn UART block diagram

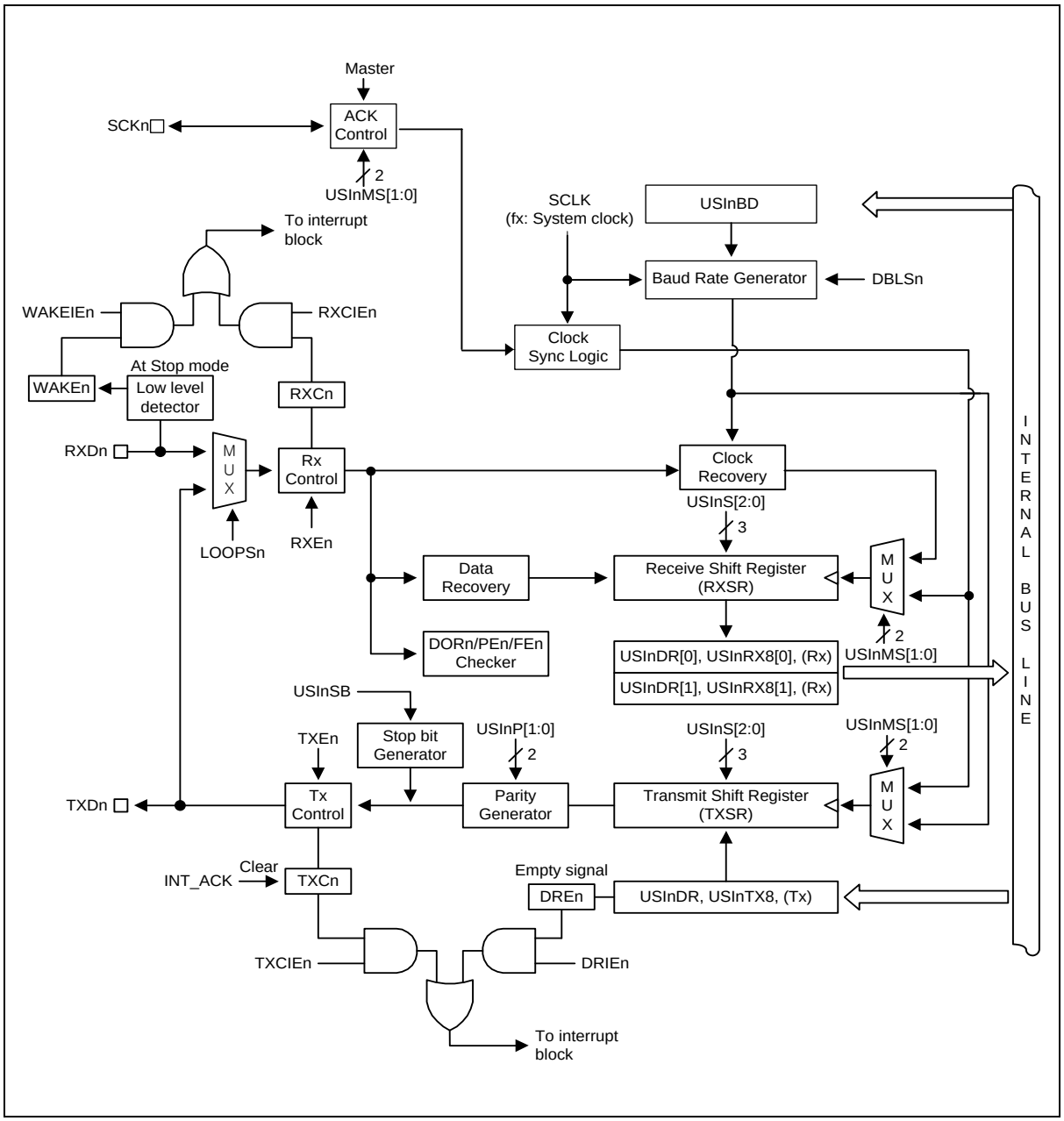


Figure 80. USIn USART Block Diagram (n = 0 and 1)

15.3 USIn clock generation

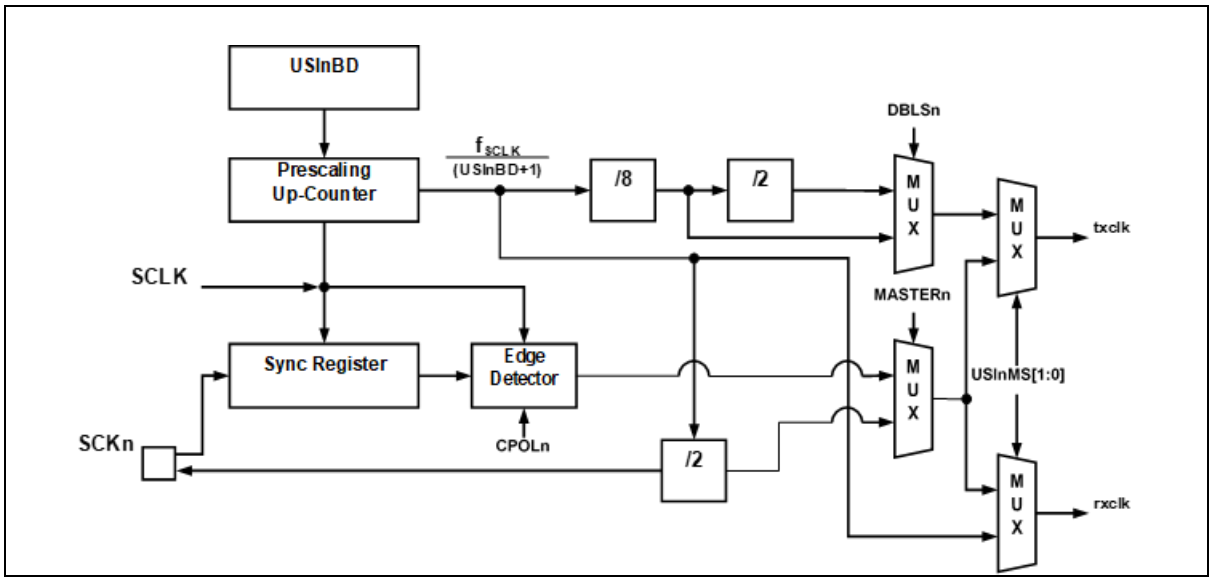


Figure 81. Clock Generation Block Diagram (USIn)

Clock generation logic generates base clock signal for the transmitter and the receiver. The USIn supports four modes of clock operation such as normal asynchronous mode, double speed asynchronous mode, master synchronous mode and slave synchronous mode.

The clock generation scheme for master SPI mode and slave SPI mode is the same as master synchronous and slave synchronous operation mode. The USInMS[1:0] bits in USInCR1 register selects one from asynchronous operation and synchronous operation. Asynchronous double speed mode is controlled by the DBLSn bit in the USInCR2 register. The MASTERn bit in USInCR3 register controls whether the clock source is internal (master mode, output pin) or external (slave mode, input pin). The SCKn pin is active only when the USIn operates in synchronous or SPI mode.

Table 29 shows the equations for calculating the baud rate (in bps).

Table 29. Equations for Calculating USIn Baud Rate Register Setting

Operating mode	Equation for calculating baud rate
Asynchronous Normal Mode (DBLSn=0)	$\text{Baud Rate} = \frac{\text{fx}}{16(\text{USInBD} + 1)}$
Asynchronous Double Speed Mode (DBLSn=1)	$\text{Baud Rate} = \frac{\text{fx}}{8(\text{USInBD} + 1)}$
Synchronous or SPI Master Mode	$\text{Baud Rate} = \frac{\text{fx}}{2(\text{USInBD} + 1)}$

15.4 USIn external clock (SCKn)

External clocking is used in synchronous mode of operation. External clock input from the SCKn pin is sampled by a synchronization logic to remove meta-stability. The output from the synchronization logic must be passed through an edge detector before it is used by the transmitter and receiver. This process introduces two CPU clock period delay. The maximum frequency of the external SCKn pin is limited up-to 1MHz.

15.5 USIn synchronous mode operation

When synchronous mode or SPI mode is used, SCKn pin will be used as either clock input (slave) or clock output (master).Data sampling and transmitter are issued on the different edge of SCKn clock respectively. For example, if data input on RXDn (MISOn in SPI mode) pin is sampled on the rising edge of SCKn clock, data output on TXDn (MOSIn in SPI mode) pin is altered on the falling edge.

CPOLn bit in USInCR1 register selects which SCKn clock edge will be used both for data sampling and data change. As shown in Figure 82, when the CPOLn is zero, data will be changed at rising SCKn edge and sampled at falling SCKn edge.

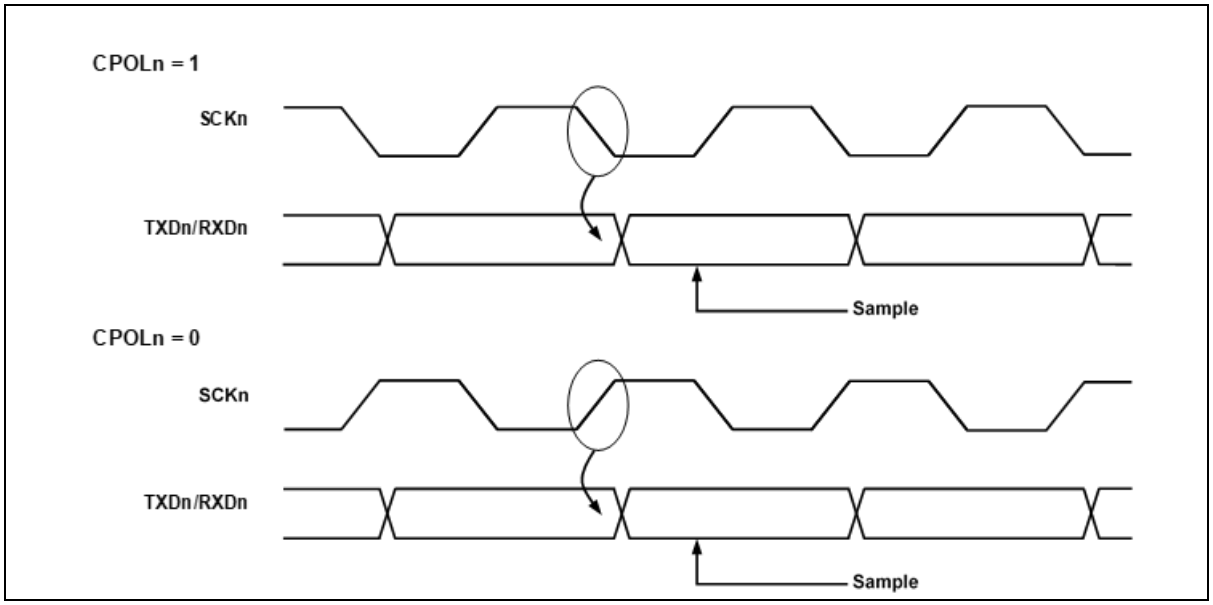


Figure 82. Synchronous Mode SCKn Timing (USIn)

15.6 USIn UART data format

A serial frame is defined to have one character of data bits with synchronization bits (start and stop bits), and optionally a parity bit for error detection. USART supports 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 a start bit followed by the least significant data bit (LSB). The next data bits, up to nine, are succeeding, ending with the most significant bit (MSB). If parity function is enabled, parity bit is inserted between the last data bit and the stop bit.

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. Figure 83 shows a possible combination of the frame formats. Bits inside brackets are optional.

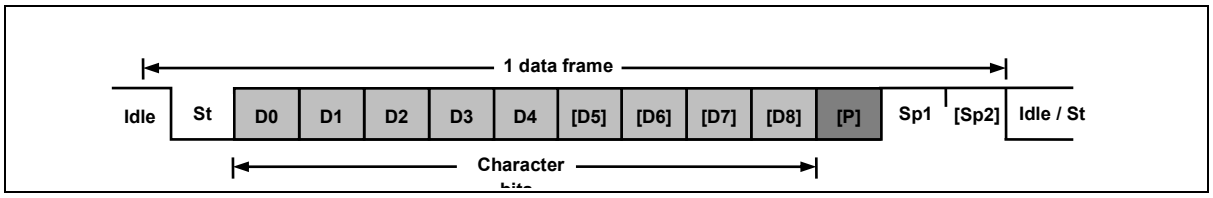


Figure 83. Frame Formats (USIn)

1 data frame consists of the following bits

- Idle: No communication on communication line (TXDn/RXDn)
- 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 used by the UART is determined by the USInS[2:0], USInPM[1:0] bits in USInCR1 register and USInSB bit in USInCR3register. The transmitter and the receiver use the same figures.

15.7 USIn UART parity bit

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

- P_{even} = Dn-1 ^ … ^ D3 ^ D2 ^ D1 ^ D0 ^ 0
- P_{odd} = Dn-1 ^ … ^ D3 ^ D2 ^ D1 ^ D0 ^ 1
- P_{even}: Parity bit using even parity
- P_{odd}: Parity bit using odd parity
- D_n: Data bit n of the character

15.8 USIn UART transmitter

The UART transmitter is enabled by setting the TXEn bit in USInCR2 register. When the Transmitter is enabled, the TXDn pin should be set to TXDn function for the serial output pin of UART by the P1FSRL[7:6], P2FSRL[3:2] and P2FSRL[7:6].The baud-rate, operation mode and frame format must be set up once before doing any transmission. In synchronous operation mode, the SCKn pin is used as transmission clock, so it should be selected to do SCKn function by P1FSRH[7:6], P1FSRL[5:4] and P2FSRH[1:0].

15.8.1 USIn UART sending TX data

A data transmission is initiated by loading the transmit buffer (USInDRregister I/O location) with the data to be transmitted. The data be written in transmit buffer is moved to the shift register when the shift register is ready to send a new frame. The shift register is loaded with the new data if it is in idle state or immediately after the last stop bit of the previous frame is transmitted. When the shift register is loaded with new data, it will transfer one complete frame according to the settings of control registers. If the 9-bit characters are used in asynchronous or synchronous operation mode, the ninth bit must be written to the USInTX8 bit in USInCR3 register before it is loaded to the transmit buffer (USInDR register).

15.8.2 USIn UART transmitter flag and interrupt

The USART transmitter has two flags which indicate its state. One is USART data register empty flag (DREn) and the other is transmit completion flag (TXCn). Both flags can be interrupt sources.

DREn flag indicates whether the transmit buffer is ready to receive new data. This bit is set when the transmit buffer is empty and cleared when the transmit buffer contains data to be transmitted but has not yet been moved into the shift register. And also this flag can be cleared by writing ‘0’ to this bit position. Writing ‘1’ to this bit position is prohibited.

When the data register empty interrupt enable (DRIEn) bit in USInCR2 register is set and the global interrupt is enabled, USInST1 status register empty interrupt is generated while DREn flag is set.

The transmit complete (TXCn) flag bit is set when the entire frame in the transmit shift register has been shifted out and there is no more data in the transmit buffer.

The TXCn flag is automatically cleared when the transmit complete interrupt serve routine is executed, or it can be cleared by writing ‘0’ to TXCn bit in USInST1 register.

When the transmit complete interrupt enable (TXCIEn) bit in USInCR2 register is set and the global interrupt is enabled, UART transmit complete interrupt is generated while TXCn flag is set.

15.8.3 USIn UART parity generator

The parity generator calculates the parity bit for the serial frame data to be sent. When parity bit is enabled (USInPM1=1), the transmitter control logic inserts the parity bit between the MSB and the first stop bit of the frame to be sent.

15.8.4 USIn UART disabling transmitter

Disabling the transmitter by clearing the TXEn bit will not become effective until ongoing transmission is completed. When the Transmitter is disabled, the TXDn pin can be used as a normal general purpose I/O (GPIO).

15.9 USIn UART receiver

The USART receiver is enabled by setting the RXEn bit in the USInCR2 register. When the receiver is enabled, the RXDn pin should be set to RXDn function for the serial input pin of UART by P1FSRH[1:0], P2FSRL[1:0] and P2FSRL[5:4]. The baud-rate, mode of operation and frame format must be set before serial reception. In synchronous or SPI operation mode the SCKn pin is used as transfer clock input, so it should be selected to do SCKn function by P1FSRH[7:6], P1FSRL[5:4] and P2FSRH[1:0]. In SPI operation mode the SSn input pin in slave mode or can be configured as SSn output pin in master mode. This can be done by setting USInSSEN bit in USInCR3 register.

15.9.1 USIn UART receiver RX data

When UART is in synchronous or asynchronous operation mode, the receiver starts data reception when it detects a valid start bit (LOW) on RXDn pin. Each bit after start bit is sampled at pre-defined baud-rate (asynchronous) or sampling edge of SCKn (synchronous), and shifted into the receive shift register until the first stop bit of a frame is received. Even if there's the second stop bit in the frame, the second stop bit is ignored by the receiver. That is, receiving the first stop bit means that a complete serial frame is presented 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 USInDR register.

If 9-bit characters are used (USInS[2:0] = "111"), the ninth bit is stored in the USInRX8 bit position in the USInCR3 register. The ninth bit must be read from the USInRX8 bit before reading the low 8 bits from the USInDR register. Likewise, the error flags FEn, DORn, PEn must be read before reading the data from USInDR register. It's because the error flags are stored in the same FIFO position of the receive buffer.

15.9.2 USIn UART receiver flag and interrupt

The UART receiver has one flag that indicates the receiver state.

The receive complete (RXCn) flag indicates whether there are unread data in the receive buffer. This flag is set when there is unread data in the receive buffer and cleared when the receive buffer is empty. If the receiver is disabled (RXEn=0), the receiver buffer is flushed and the RXCn flag is cleared.

When the receive complete interrupt enable (RXCIEn) bit in the USInCR2 register is set and global interrupt is enabled, the UART receiver complete interrupt is generated while RXCn flag is set.

The UART receiver has three error flags which are frame error (FEn), data overrun (DORn) and parity error (PEn). These error flags can be read from the USInST1 register. As the received data is 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 USInDR register, read the USInST1 register first which contains error flags.

The frame error (FEn) flag indicates the state of the first stop bit. The FEn flag is ‘0’ when the stop bit was correctly detected as “1”, and the FEn flag is “1” 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.

The data overrun (DORn) flag indicates data loss due to a receive buffer full condition. DORn occurs when the receive buffer is full, and another new data is presented in the receive shift register which are to be stored into the receive buffer. After the DORn flag is set, all the incoming data are lost. To avoid data loss or clear this flag, read the receive buffer.

The parity error (PEn) flag indicates that the frame in the receive buffer had a parity error when received. If parity check function is not enabled (USInPM1=0), the PE bit is always read “0”.

15.9.3 USIn UART parity checker

If parity bit is enabled (USInPM1=1), the Parity Checker calculates the parity of the data bits in incoming frame and compares the result with the parity bit from the received serial frame.

15.9.4 USIn UART disabling receiver

In contrast to transmitter, disabling the Receiver by clearing RXEn bit makes the Receiver inactive immediately. When the receiver is disabled, the receiver flushes the receive buffer, the remaining data in the buffer is all reset, and the RXDn pin can be used as a normal general purpose I/O (GPIO).

15.9.5 USIn Asynchronous data reception

To receive asynchronous data frame, the UART 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 RXDn pin.

The data recovery logic does sampling and low pass filtering the incoming bits, and removing the noise of RXDn pin.

The next figure illustrates the sampling process of the start bit of an incoming frame. The sampling rate is 16 times of the baud-rate in normal mode and 8 times the baud-rate for double speed mode (DBLSn=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.

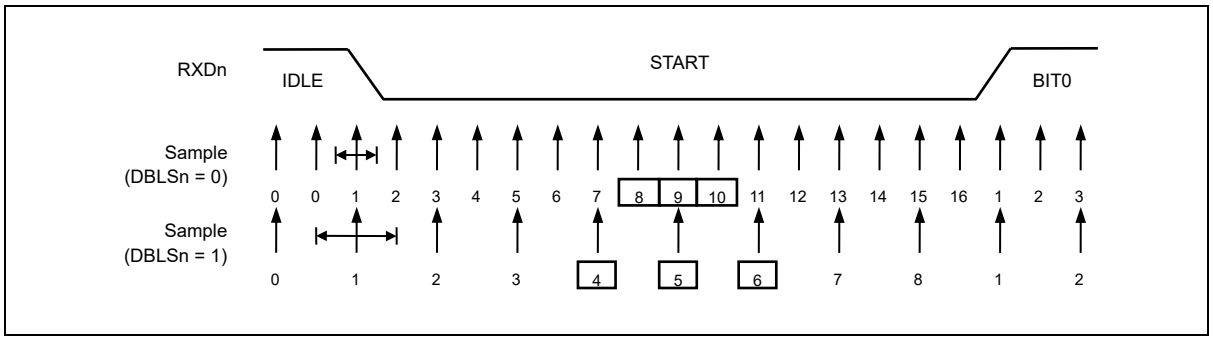


Figure 84. Asynchronous Start Bit Sampling (USIn)

When the receiver is enabled (RXEn=1), the clock recovery logic tries to find a high-to-low transition on the RXDn line, the start bit condition. After detecting high to low transition on RXDn line, the clock recovery logic uses samples 8, 9 and 10 for normal 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 same to the clock recovery process. The data recovery logic samples 16 times for each incoming bits for normal mode and 8 times for double speed mode, and uses sample 8, 9 and 10 to decide data value. If more than 2 samples have low levels, the received bit is considered to a logic ‘0’ and 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.

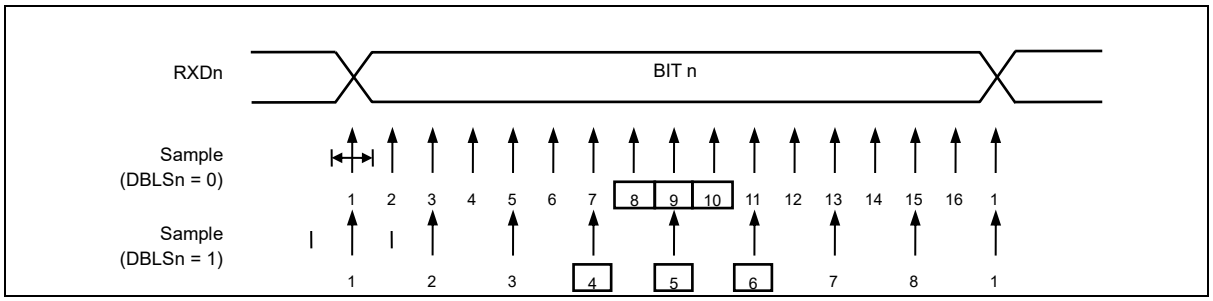


Figure 85. Asynchronous Sampling of Data and Parity Bit (USIn)

The process for detecting stop bit is same as clock and data recovery process. That is, if 2 or more samples of 3 center values have high level, correct stop bit is detected, else a frame error (FEn) flag is set. After deciding whether the first stop bit is valid or not, the Receiver goes to idle state and monitors the RXDn line to check a valid high to low transition is detected (start bit detection).

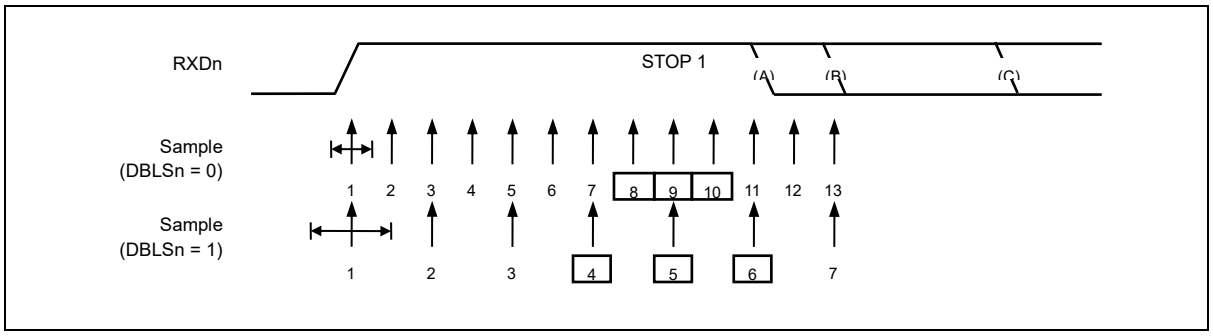


Figure 86. Stop Bit Sampling and Next Start Bit Sampling (USIn)

15.10 USIn SPI mode

The USIn 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 and slave operation
- Supports all four SPIn modes of operation (mode 0, 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 (USInMS[1:0]=”11”), the slave select (SSn) pin becomes active LOW input in slave mode operation, or can be output in master mode operation if USInSSEN bit is set to ‘0’.

Note that during SPI mode of operation, the pin RXDn is renamed as MISOn and TXDn is renamed as MOSIn for compatibility to other SPI devices.

15.11 USIn SPI clock formats and timing

To accommodate a wide variety of synchronous serial peripherals from different manufacturers, the USIn has a clock polarity bit (CPOLn) and a clock phase control bit (CPHAn) to select one of four clock formats for data transfers. CPOLn selectively insert an inverter in series with the clock. CPHAn chooses between two different clock phase relationships between the clock and data. Note that CPHAn and CPOLn bits in USInCR1 register have different meanings according to the USInMS[1:0] bits which decides the operating mode of USIn.

Table 30 shows four combinations of CPOLn and CPHAn for SPI mode 0, 1, 2, and 3.

Table 30. CPOLn Functionality

SPI Mode	CPOLn	CPHAn	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)

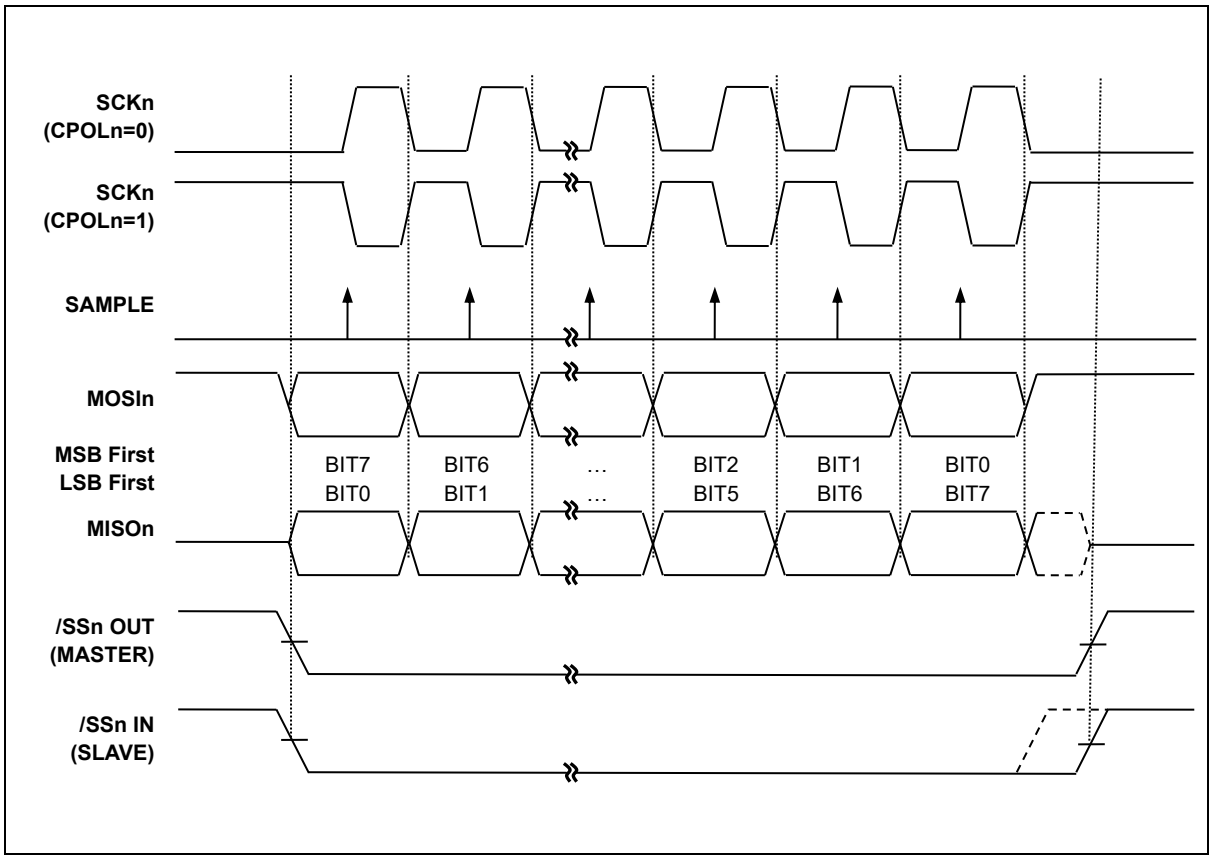


Figure 87. USIn SPI Clock Formats when CPHAn = 0

When CPHAn = 0, the slave begins to drive its MISOn output with the first data bit value when SSn goes to active low. The first SCKn edge causes both the master and the slave to sample the data bit value on their MISOn and MOSIn inputs, respectively.

At the second SCKn edge, the USIn shifts the second data bit value out to the MOSIn and MISOn outputs of the master and slave, respectively. Unlike the case of CPHAn=1, when CPHAn=0, the slave's SSn 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 SSn input.

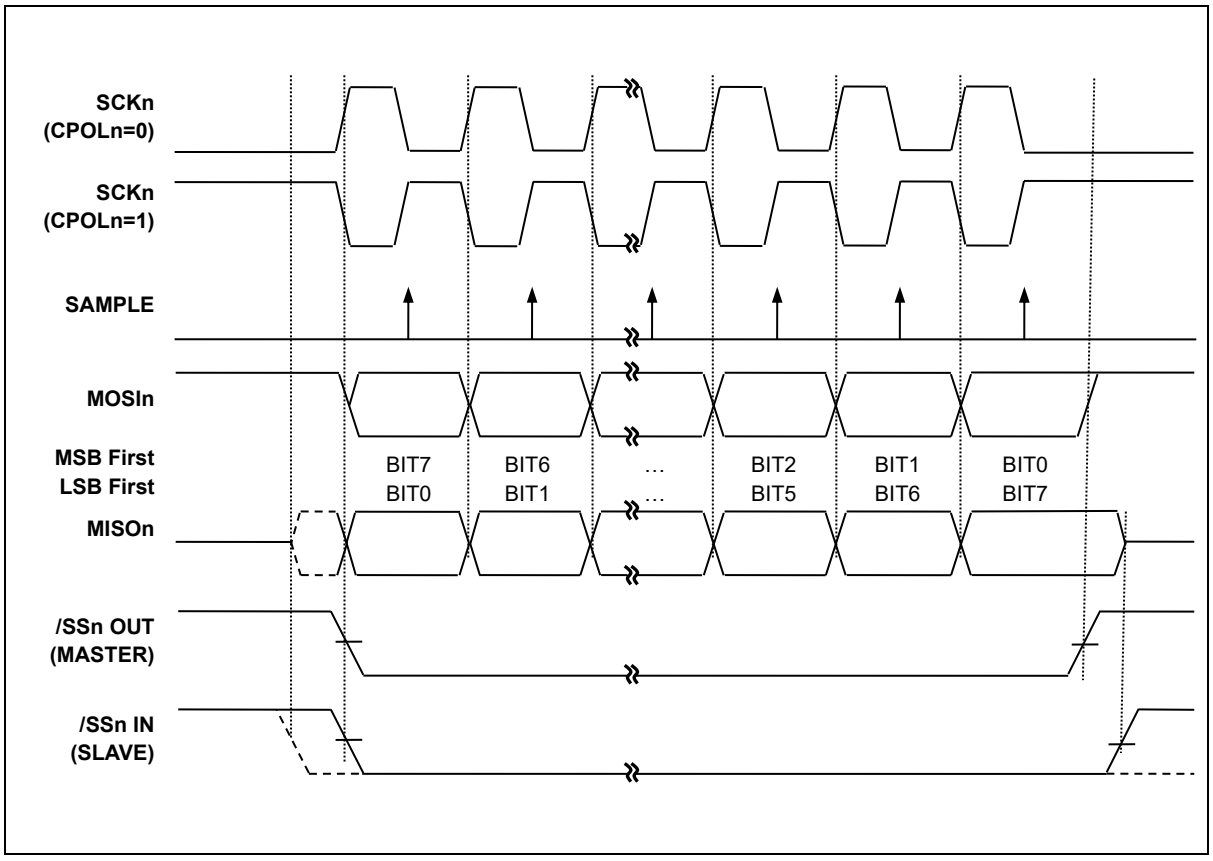


Figure 88. USIn SPI Clock Formats when CPHAn = 1

When CPHAn = 1, the slave begins to drive its MISOn output when SSn goes active low, but the data is not defined until the first SCKn edge. The first SCKn edge shifts the first bit of data from the shifter onto the MOSIn output of the master and the MISOn output of the slave. The next SCKn edge causes both the master and slave to sample the data bit value on their MISOn and MOSIn inputs, respectively.

At the third SCKn edge, the USIn shifts the second data bit value out to the MOSIn and MISOn output of the master and slave respectively. When CPHAn=1, the slave's SSn input is not required to go to its inactive high level between transfers.

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

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

15.12 USIn SPI block diagram

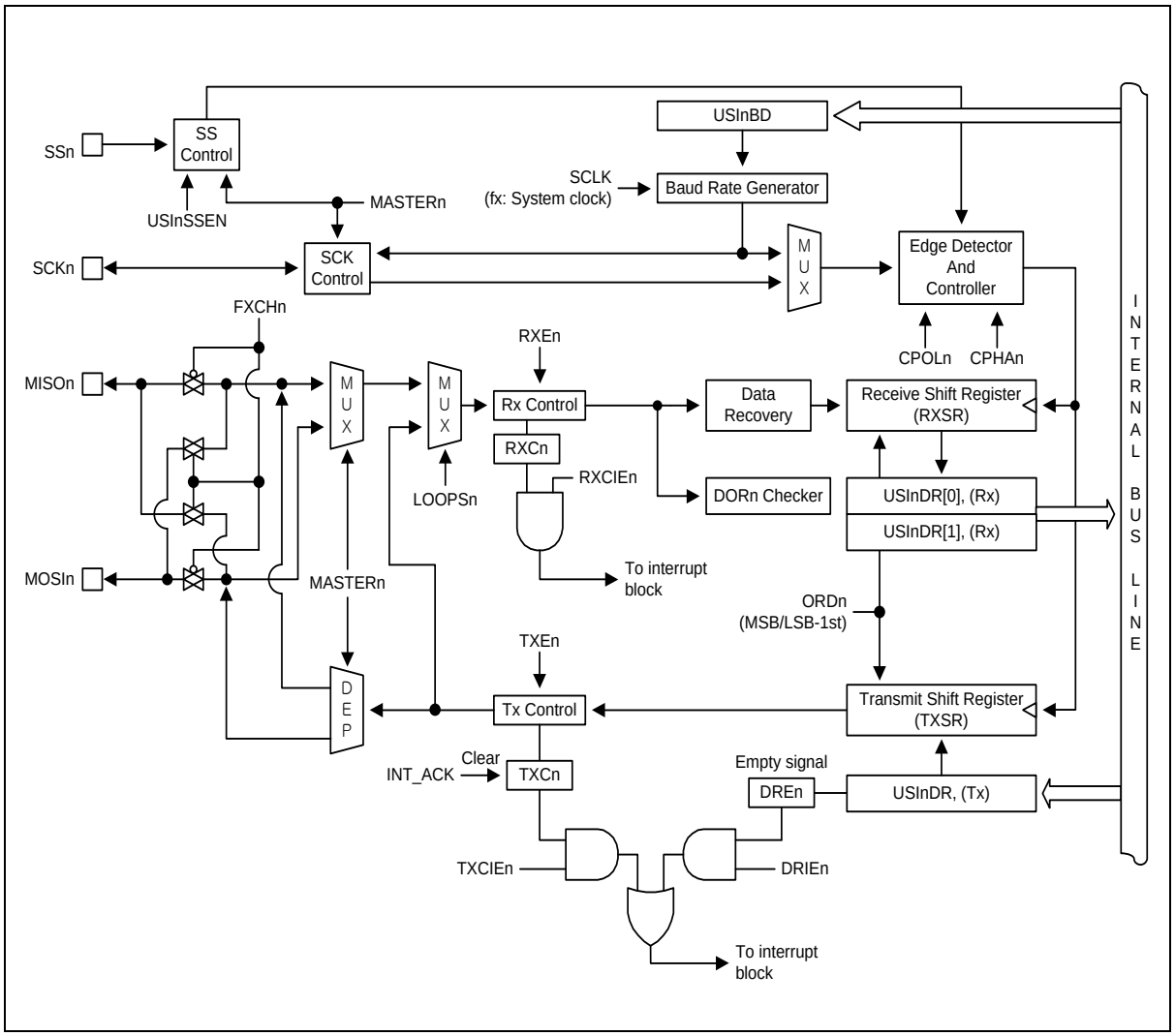


Figure 89. USIn SPI Block Diagram (n = 0 and 1)

15.13 USIn I2C mode

The USIn can be set to operate in industrial standard serial communication protocols mode. The I2C mode uses 2 bus lines serial data line (SDAn) and serial clock line (SCLn) to exchange data. Because both SDAn and SCLn lines are open-drain output, each line needs pull-up resistor. The features are as shown below.

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

15.14 USIn I2C bit transfer

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

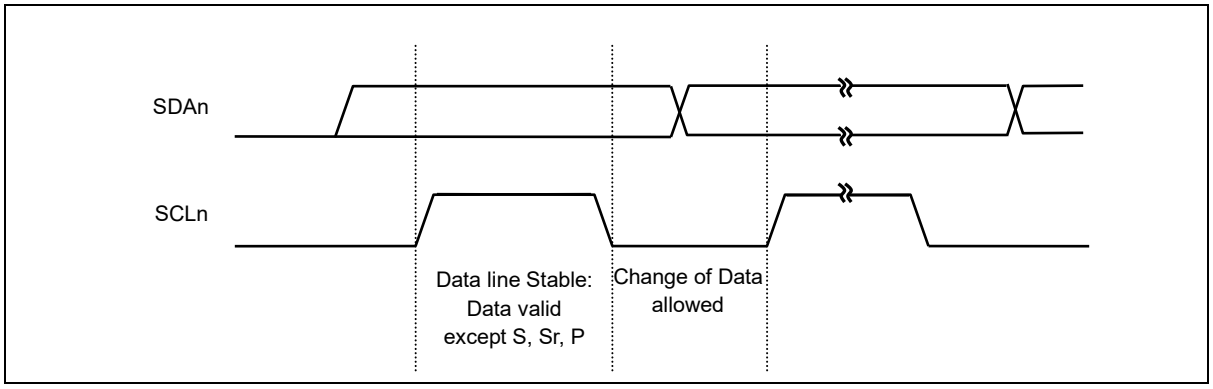


Figure 90. Bit Transfer on the I2C-Bus (USIn)

15.15 USIn I2C START/repeated START/STOP

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

A low to high transition on the SDAn line while SCLn 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.

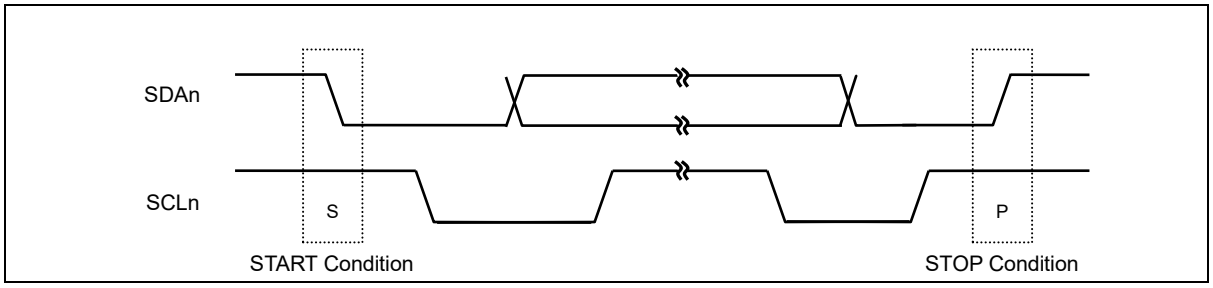


Figure 91. START and STOP Condition (USIn)

15.16 USIn I2C data transfer

Every byte put on the SDAn 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 SCLn 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 SCLn.

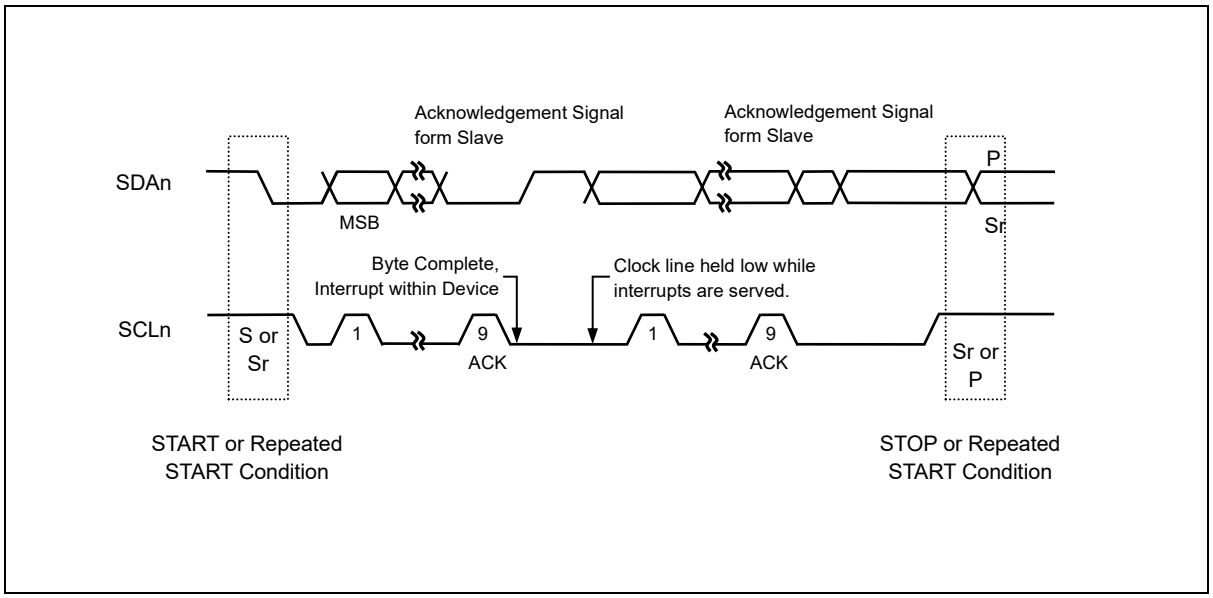


Figure 92. Data Transfer on the I2C-Bus (USIn)

15.17 USIn I2C acknowledge

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

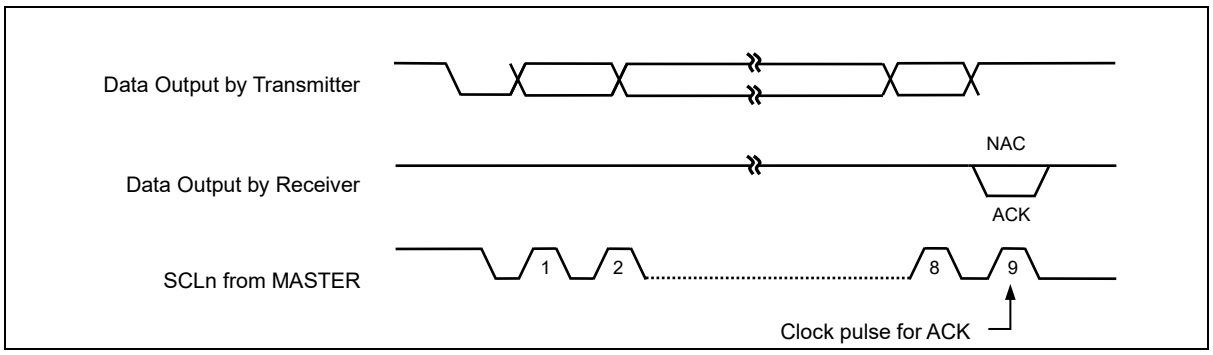


Figure 93. Acknowledge on the I2C-Bus (USIn)

15.18 USIn I2C synchronization/arbitration

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

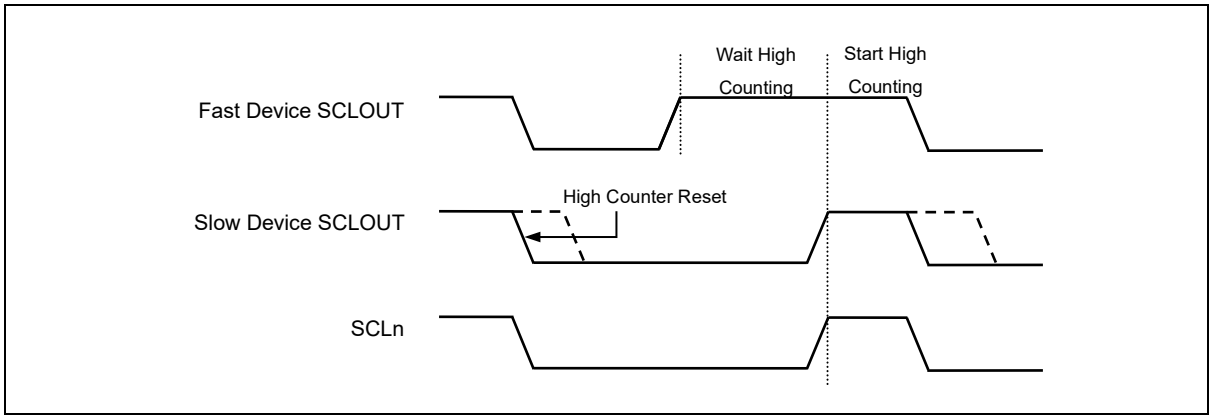


Figure 94. Clock Synchronization during Arbitration Procedure (USIn)

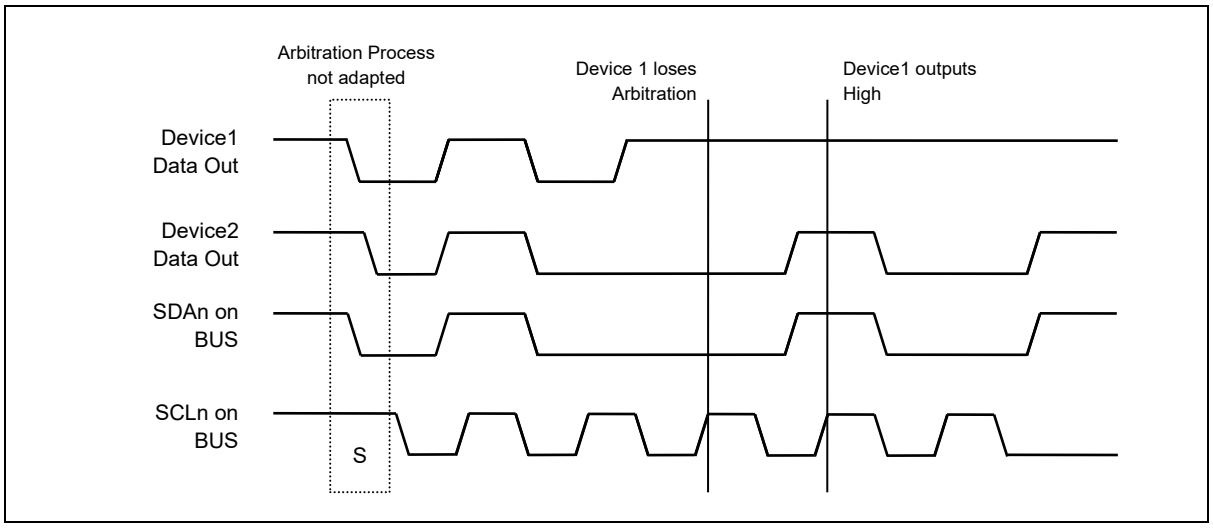


Figure 95. Arbitration Procedure of Two Masters (USIn)

15.19 USIn I2C operation

The I2C is byte-oriented and interrupt based. Interrupts are issued after all bus events except for a transmission of a START condition. Because the I2C is interrupt based, the application software is free to carry on other operations during an I2C byte transfer.

Note that when an I2C interrupt is generated, IICnIFR flag in USInCR4 register is set, it is cleared by writing any value to USInST2. When I2C interrupt occurs, the SCLn line is hold LOW until writing any value to USInST2. When the IICnIFR flag is set, the USInST2 contains a value indicating the current state of the I2C bus. According to the value in USInST2, software can decide what to do next.

I2C can operate in 4 modes by configuring master/slave, transmitter/receiver. The operating mode is configured by a winning master. A more detailed explanation follows below.

15.19.1 USIn I2C master transmitter

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

1. Enable I2C by setting USInMS[1:0] bits in USInCR1 and USInEN bit in USInCR2. This provides main clock to the peripheral.
2. Load SLAn+W into the USInDR where SLAn is address of slave device and W is transfer direction from the viewpoint of the master. For master transmitter, W is ‘0’. Note that USInDR is used for both address and data.
3. Configure baud rate by writing desired value to both USInSCLR and USInSCHR for the Low and High period of SCLn line.
4. Configure the USInSDHR to decide when SDAn changes value from falling edge of SCLn. If SDAn should change in the middle of SCLn LOW period, load half the value of USInSCLR to the USInSDHR.
5. Set the STARTCn bit in USInCR4. This transmits a START condition. And also configure how to handle interrupt and ACK signal. When the STARTCn bit is set, 8-bit data in USInDR is transmitted out according to the baud-rate.
6. This is ACK signal processing stage for address packet transmitted by master. When 7-bit address and 1-bit transfer direction is transmitted to target slave device, the master can know whether the slave acknowledged or not in the 9th high period of SCLn. If the master gains bus mastership, I2C generates GCALL interrupt regardless of the reception of ACK from the slave device. When I2C loses bus mastership during arbitration process, the MLOSTn bit in USInST2 is set, and I2C waits in idle state or can be operate as an addressed slave.

To operate as a slave when the MLOSTn bit in USInST2 is set, the ACKnEN bit in USInCR4 must be set and the received 7-bit address must equal to the USInSLA[6:0] bits in USInSAR. In this case I2C operates as a slave transmitter or a slave receiver (go to appropriate section). In this stage, I2C holds the SCLn LOW. This is because to decide whether I2C continues serial transfer or stops communication. The following steps continue assuming that I2C does not lose mastership during first data transfer.

I2C (Master) can choose one of the following cases regardless of the reception of ACK signal from slave:

Case 1: Master receives ACK signal from slave, so continues data transfer because slave can receive more data from master. In this case, load data to transmit to USInDR.
Case 2: Master stops data transfer even if it receives ACK signal from slave. In this case, set the STOPCn bit in USInCR4.
Case 3: Master transmits repeated START condition with not checking ACK signal. In this case, load SLAn+R/W into the USInDR and set STARTCn bit in USInCR4.

After doing one of the actions above, write any arbitrary to USInST2 to release SCLn line. For the case 1, move to step 7. For the case 2, move to step 9 to handle STOP interrupt. For the case 3, move to step 6 after transmitting the data in USInDR and if transfer direction bit is '1' go to master receiver section.

7. 1-Byte of data is being transmitted. During data transfer, bus arbitration continues.

8. This is ACK signal processing stage for data packet transmitted by master. I2C holds the SCLn LOW. When I2C loses bus mastership while transmitting data arbitrating other masters, the MLOSTn bit in USInST2 is set. If then, I2C waits in idle state. When the data in USInDR is transmitted completely, I2C generates TENDn interrupt.

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

   Case 1: Master receives ACK signal from slave, so continues data transfer because slave can receive more data from master. In this case, load data to transmit to USInDR.
   Case 2: Master stops data transfer even if it receives ACK signal from slave. In this case, set the STOPCn bit in USInCR4.
   Case 3: Master transmits repeated START condition with not checking ACK signal. In this case, load SLAn+R/W into the USInDR and set the STARTCn bit in USInCR4.

After doing one of the actions above, write any arbitrary to USInST2 to release SCLn line. For the case 1, move to step 7. For the case 2, move to step 9 to handle STOP interrupt. For the case 3, move to step 6 after transmitting the data in USInDR, and if transfer direction bit is '1' go to master receiver section.

9. This is the final step for master transmitter function of I2C, handling STOP interrupt. The STOP bit indicates that data transfer between master and slave is over. To clear USInST2, write any value to USInST2. After this, I2C enters idle state.

15.19.2 USIn I2C master receiver

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

1. Enable I2C by setting USInMS[1:0] bits in USInCR1and USInEN bit in USInCR2. This provides main clock to the peripheral.
2. Load SLAn+R into the USInDR where SLA is address of slave device and R is transfer direction from the viewpoint of the master. For master receiver, R is '1'. Note that USInDR is used for both address and data.
3. Configure baud rate by writing desired value to both USInSCLR and USInSCHR for the Low and High period of SCLn line.
4. Configure the USInSDHR to decide when SDAn changes value from falling edge of SCLn. If SDAn should change in the middle of SCLn LOW period, load half the value of USInSCLR to the USInSDHR.
5. Set the STARTCn bit in USInCR4. This transmits a START condition. And also configure how to handle interrupt and ACK signal. When the STARTCn bit is set, 8-bit data in USInDR is transmitted out according to the baud-rate.
6. This is ACK signal processing stage for address packet transmitted by master. When 7-bit address and 1-bit transfer direction is transmitted to target slave device, the master can know whether the slave acknowledged or not in the 9th high period of SCLn. If the master gains bus mastership, I2C generates GCALL interrupt regardless of the reception of ACK from the slave device. When I2C loses bus mastership during arbitration process, the MLOSTn bit in USInST2 is set, and I2C waits in idle state or can be operate as an addressed slave.

To operate as a slave when the MLOSTn bit in USInST2 is set, the ACKnEN bit in USInCR4 must be set and the received 7-bit address must equal to the USInSLA[6:0] bits in USInSAR. In this case I2C operates as a slave transmitter or a slave receiver (go to appropriate section). In this stage, I2C holds the SCLn LOW. This is because to decide whether I2C continues serial transfer or stops communication. The following steps continue assuming that I2C does not lose mastership during first data transfer.

I2C (Master) can choose one of the following cases according to the reception of ACK signal from slave:

Case 1: Master receives ACK signal from slave, so continues data transfer because slave can prepare and transmit more data to master. Configure ACKnEN bit in USInCR4 to decide whether I2C ACKnowledges the next data to be received or not.
Case 2: Master stops data transfer because it receives no ACK signal from slave. In this case, set the STOPCn bit in USInCR4.
Case 3: Master transmits repeated START condition due to no ACK signal from slave. In this case, load SLAn+R/W into the USInDR and set STARTCn bit in USInCR4.

After doing one of the actions above, write arbitrary value to USInST2 to release SCLn line. For the case 1, move to step 7. For the case 2, move to step 9 to handle STOP interrupt. For the case 3, move to step 6 after transmitting the data in USInDR and if transfer direction bit is '0' go to master transmitter section.

7. 1-Byte of data is being received.

8. This is ACK signal processing stage for data packet transmitted by slave. I2C holds the SCLn LOW. When 1-Byte of data is received completely, I2C generates TENDn interrupt.

   I2C can choose one of the following cases according to the RXACKn flag in USInST2:

   Case 1: Master continues receiving data from slave. To do this, set ACKnEN bit in USInCR4 to ACKnowledge the next data to be received.
   Case 2: Master wants to terminate data transfer when it receives next data by not generating ACK signal. This can be done by clearing ACKnEN bit in USInCR4.
   Case 3: Because no ACK signal is detected, master terminates data transfer. In this case, set the STOPCn bit in USInCR4.
   Case 4: No ACK signal is detected, and master transmits repeated START condition. In this case, load SLAn+R/W into the USInDR and set the STARTCn bit in USInCR4.

After doing one of the actions above, write arbitrary value to USInST2 to release SCLn line. For the case 1 and case 2, move to step 7. For the case 3, move to step 9 to handle STOP interrupt. For the case 4, move to step 6 after transmitting the data in USInDR, and if transfer direction bit is '0' go to master transmitter section.

9. This is the final step for master receiver function of I2C, handling STOP interrupt. The STOP bit indicates that data transfer between master and slave is over. To clear USInST2, write any value to USInST2. After this, I2C enters idle state.

15.19.3 USIn I2C slave transmitter

To operate I2C in slave transmitter, follow the recommended steps below:

1. If the main operating clock (SCLK) of the system is slower than that of SCLn, load value 0x00 into USInSDHR to make SDAn change within one system clock period from the falling edge of SCLn. Note that the hold time of SDAn is calculated by SDAH x period of SCLK where SDAH is multiple of number of SCLK coming from USInSDHR. When the hold time of SDAn is longer than the period of SCLK, I2C (slave) cannot transmit serial data properly.
2. Enable I2C by setting USInMS[1:0] bits in USInCR1, IICnIEbit in USInCR4 and USInEN bit in USInCR2. This provides main clock to the peripheral.
3. When a START condition is detected, I2C receives one byte of data and compares it with USInSLA[6:0] bits in USInSAR. If the GCALLn bit in USInSAR is enabled, I2C compares the received data with value 0x00, the general call address.
4. If the received address does not equal to USInSLA[6:0] bits in USInSAR, I2C enters idle state i.e., waits for another START condition. Else if the address equals to USInSLA[6:0] bits and the ACKnEN bit is enabled, I2C generates SSELn interrupt and the SCLn line is held LOW. Note that even if the address equals to USInSLA[6:0] bits, when the ACKnEN bit is disabled, I2C enters idle state. When SSELn interrupt occurs, load transmit data to USInDR and write arbitrary value to USInST2 to release SCLn line.
5. 1-Byte of data is being transmitted.
6. In this step, I2C generates TENDn interrupt and holds the SCLn line LOW regardless of the reception of ACK signal from master. Slave can select one of the following cases.

Case 1: No ACK signal is detected and I2C waits STOP or repeated START condition.
Case 2: ACK signal from master is detected. Load data to transmit into USInDR.

After doing one of the actions above, write arbitrary value to USInST2 to release SCLn line. For the case 1, move to step 7 to terminate communication. For the case 2, move to step 5. In either case, a repeated START condition can be detected. For that case, move step 4.

7. This is the final step for slave transmitter function of I2C, handling STOP interrupt. The STOPCn bit indicates that data transfer between master and slave is over. To clear USInST2, write any value to USInST2. After this, I2C enters idle state.

15.19.4 USIn I2C slave receiver

To operate I2C in slave receiver, follow the recommended steps below:

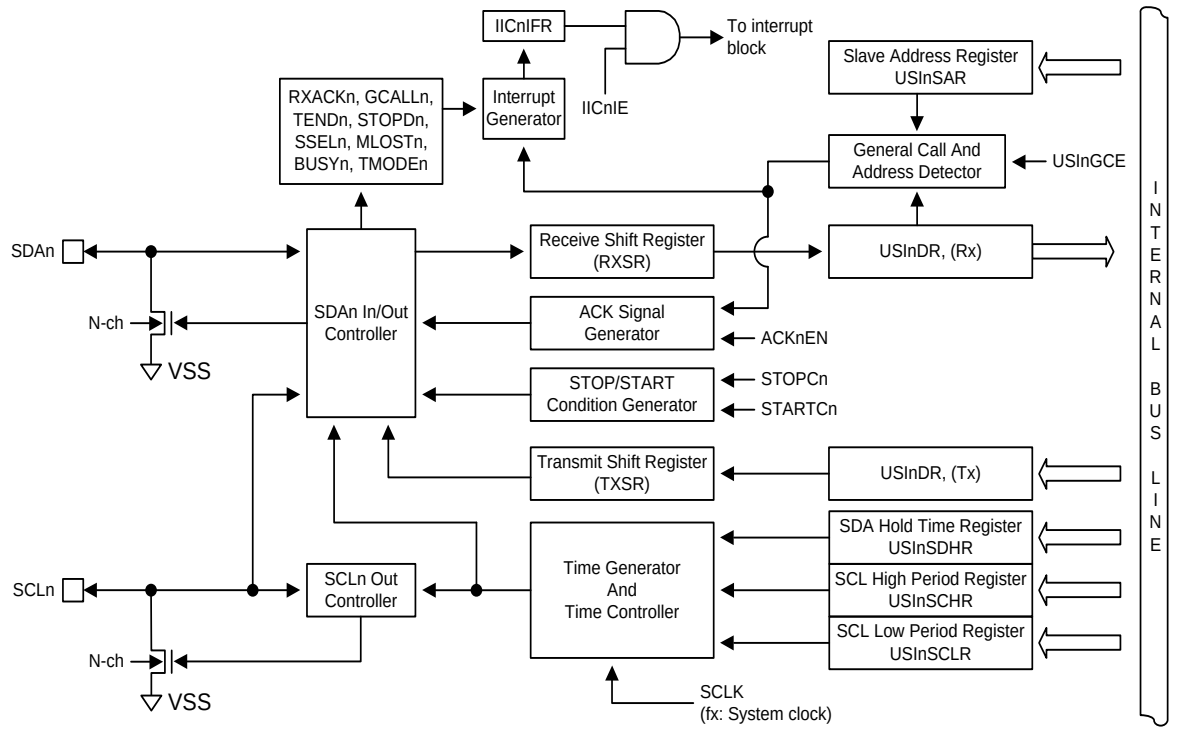
1. If the main operating clock (SCLK) of the system is slower than that of SCLn, load value 0x00 into USInSDHR to make SDAn change within one system clock period from the falling edge of SCLn. Note that the hold time of SDAn is calculated by SDAH x period of SCLK where SDAH is multiple of number of SCLK coming from USInSDHR. When the hold time of SDAn is longer than the period of SCLK, I2C (slave) cannot transmit serial data properly.
2. Enable I2C by setting USInMS[1:0] bits in USInCR1, IICnIEbit in USInCR4 and USInEN bit in USInCR2. This provides main clock to the peripheral.
3. When a START condition is detected, I2C receives one byte of data and compares it with USInSLA[6:0] bits in USInSAR. If the GCALLn bit in USInSAR is enabled, I2Cn compares the received data with value 0x00, the general call address.
4. If the received address does not equal to SLAn bits in USInSAR, I2C enters idle state i.e., waits for another START condition. Else if the address equals to SLAn bits and the ACKnEN bit is enabled, I2C generates SSELn interrupt and the SCLn line is held LOW. Note that even if the address equals to SLAn bits, when the ACKnEN bit is disabled, I2C enters idle state. When SSELn interrupt occurs and I2C is ready to receive data, write arbitrary value to USInST2 to release SCLn line.
5. 1-Byte of data is being received.
6. In this step, I2C generates TENDn interrupt and holds the SCLn line LOW regardless of the reception of ACK signal from master. Slave can select one of the following cases.

Case 1: No ACK signal is detected (ACKnEN=0) and I2C waits STOP or repeated START condition.
Case 2: ACK signal is detected (ACKnEN=1) and I2C can continue to receive data from master.

After doing one of the actions above, write arbitrary value to USInST2 to release SCLn line. For the case 1, move to step 7 to terminate communication. For the case 2, move to step 5. In either case, a repeated START condition can be detected. For that case, move step 4.

7. This is the final step for slave receiver function of I2C, handling STOP interrupt. The STOPCn bit indicates that data transfer between master and slave is over. To clear USInST2, write any value to USInST2. After this, I2C enters idle state

15.20 USIn I2C block diagram



NOTE: When the USIn block is an I2C mode and the corresponding port is a sub-function for SCLn/SDAn pin, The SCLn/SDAn pins are automatically set to the N-channel open-drain outputs and the input latch is read in the case of reading the pins. The corresponding pull-up resistor is determined by the control register.

Figure 96. USIn I2C Block Diagram

15.21 Register map

Table 31. USI Register Map

Name	Address	Direction	Default	Description
USI0BD	E3H	R/W	FFH	USI0 Baud Rate Generation Register
USI0DR	E5H	R/W	00H	USI0 Data Register
USI0SDHR	E4H	R/W	01H	USI0 SDA Hold Time Register
USI0SCHR	E7H	R/W	3FH	USI0 SCL High Period Register
USI0SCLR	E6H	R/W	3FH	USI0 SCL Low Period Register
USI0SAR	DDH	R/W	00H	USI0 Slave Address Register
USI0CR1	D9H	R/W	00H	USI0 Control Register 1
USI0CR2	DAH	R/W	00H	USI0 Control Register 2
USI0CR3	DBH	R/W	00H	USI0 Control Register 3
USI0CR4	DCH	R/W	00H	USI0 Control Register 4
USI0ST1	E1H	R/W	80H	USI0 Status Register 1
USI0ST2	E2H	R	00H	USI0 Status Register 2
USI1BD	F3H	R/W	FFH	USI1 Baud Rate Generation Register
USI1DR	F5H	R/W	00H	USI1 Data Register
USI1SDHR	F4H	R/W	01H	USI1 SDA Hold Time Register
USI1SCHR	F7H	R/W	3FH	USI1 SCL High Period Register
USI1SCLR	F6H	R/W	3FH	USI1 SCL Low Period Register
USI1SAR	EDH	R/W	00H	USI1 Slave Address Register
USI1CR1	E9H	R/W	00H	USI1 Control Register 1
USI1CR2	EAH	R/W	00H	USI1 Control Register 2
USI1CR3	EBH	R/W	00H	USI1 Control Register 3
USI1CR4	ECH	R/W	00H	USI1 Control Register 4
USI1ST1	F1H	R/W	80H	USI1 Status Register 1
USI1ST2	F2H	R	00H	USI1 Status Register 2

15.22 USIn register description

USInBD (USIn Baud- Rate Generation Register: For UART and SPI mode): E3H/F3H, n = 0, 1

7	6	5	4	3	2	1	0
USInBD7	USInBD6	USInBD5	USInBD4	USInBD3	USInBD2	USInBD1	USInBD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: FFH

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

NOTE: In common with USInSAR register, USInBD register is used for slave address register when the USIn I2C mode.

USInDR (USIn Data Register: For UART, SPI, and I2C mode): E5H/F5H, n = 0, 1

7	6	5	4	3	2	1	0
USInDR7	USInDR6	USInDR5	USInDR4	USInDR3	USInDR2	USInDR1	USInDR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

USInDR[7:0] The USIn 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 USInDR register. Reading the USInDR register returns the contents of the receive buffer.

Write to this register only when the DREn flag is set. In SPI master mode, the SCK clock is generated when data are written to this register.

USInSDHR (USInSDA Hold Time Register: For I2C mode): E4H/F4H, n = 0, 1

7	6	5	4	3	2	1	0
USInSDHR7	USInSDHR6	USInSDHR5	USInSDHR4	USInSDHR3	USInSDHR2	USInSDHR1	USInSDHR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 01H

USInSDHR[7:0] The register is used to control SDAn output timing from the falling edge of SCl in I2C mode.

NOTES:

1. That SDAn is changed after t_{SCLK} X (USInSDHR+2), in master SDAn change in the middle of SCLn.
2. In slave mode, configure this register regarding the frequency of SCLn from master.
3. The SDAn is changed after t_{SCLK} X (USInSDHR+2) in master mode. So, to insure operation in slave mode, the value t_{SCLK} X (USInSDHR+2) must be smaller than the period of SCL.

USInSCHR (USInSCL High Period Register: For I2C mode): E7H/F7H, n = 0, 1

7	6	5	4	3	2	1	0
USInSCHR7	USInSCHR6	USInSCHR5	USInSCHR 4	USInSCHR 3	USInSCHR 2	USInSCHR 1	USInSCHR 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 3FH

USInSCHR[7:0] This register defines the high period of SCLn when it operates in I2C master mode.

The base clock is SCLK, the system clock, and the period is calculated by the formula: t_{SCLK} X (4 X USInSCHR +2) where

t_{SCLK} is the period of SCLK.

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

$$f_{I2C} = \frac{1}{t_{SCLK} X\ (4\ X\ (USInSCLR + USInSCHR) + 4)}$$

USInSCLR (USInSCL Low Period Register: For I2C mode): E6H/F6H, n = 0, 1

7	6	5	4	3	2	1	0
USInSCLR7	USInSCLR6	USInSCLR5	USInSCLR 4	USInSCLR 3	USInSCLR 2	USInSCLR 1	USInSCLR 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 3FH

USInSCLR[7:0] This register defines the high period of SCLn when it operates in I2C master mode.

The base clock is SCLK, the system clock, and the period is calculated by the formula: t_{SCLK} X (4 X USInSCLR +2) where

t_{SCLK} is the period of SCLK.

USInSAR (USIn Slave Address Register: For I2C mode): DDH/EDH, n = 0, 1

7	6	5	4	3	2	1	0
USInSLA6	USInSLA5	USInSLA4	USInSLA3	USInSLA2	USInSLA1	USInSLA0	USInGCE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

USInSLA[6:0] These bits configure the slave address of I2C when it operates in I2C slave mode.

USInGCE This bit decides whether I2C allows general call address or not in I2C slave mode.

0 Ignore general call address

1 Allow general call address

USInCR1 (USIn Control Register 1: For UART, SPI, and I2C mode): D9H/E9H, n = 0, 1

7	6	5	4	3	2	1	0
USInMS1	USInMS0	USInPM1	USInPM0	USInS2	USInS1 ORDn	USInS0 CPHAn	CPOLn
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

USInMS[1:0] Selects operation mode of USIn

USInMS1	USInMS0	Operation mode
0	0	Asynchronous Mode (UART)
0	1	Synchronous Mode
1	0	I2C mode
1	1	SPI mode

USInPM[1:0] Selects parity generation and check methods (only UART mode)

USInPM1	USInPM0	Parity
0	0	No Parity
0	1	Reserved
1	0	Even Parity
1	1	Odd Parity

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

USInS2	USInS1	USInS0	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

ORDn This bit in the same bit position with USInS1. The MSB of the data byte is transmitted first when set to '1' and the LSB when set to '0' (only SPI mode)

0	LSB-first
1	MSB-first

CPHAn This bit is in the same bit position with USInS0. This bit determines if data are sampled on the leading or trailing edge of SCKn (only SPI mode).

CPOLn	CPHAn	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)

CPOLn This bit determines the clock polarity of ACK in synchronous or SPI mode.

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

USInCR2 (USIn Control Register 2: For UART, SPI, and I2C mode): DAH/EAH, n = 0, 1

7	6	5	4	3	2	1	0
DRIEn	TXCIEn	RXCIEn	WAKEIEn	TXEn	RXEn	USInEN	DBLSn
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

DRIEn — Interrupt enable bit for data register empty (only UART and SPI mode).
- 0 Interrupt from DREn is inhibited (use polling)
- 1 When DREn is set, request an interrupt

TXCIEn — Interrupt enable bit for transmit complete (only UART and SPI mode).
- 0 Interrupt from TXCn is inhibited (use polling)
- 1 When TXCn is set, request an interrupt

RXCIEn — Interrupt enable bit for receive complete (only UART and SPI mode).
- 0 Interrupt from RXCn is inhibited (use polling)
- 1 When RXCn is set, request an interrupt

WAKEIEn — Interrupt enable bit for asynchronous wake in STOP mode. When device is in stop mode, if RXDn goes to low level an interrupt can be requested to wake-up system. (only UART mode). At that time the DRIEn bit and USInST1 register value should be set to ‘0b’ and “00H”, respectively.
- 0 Interrupt from Wake is inhibited
- 1 When WAKEn is set, request an interrupt

NOTES:
1. WAKEIEn must set after USInEN setting ‘1’.
2. When using the RXD to wake up from STOP mode, the WAKEIE bit must be cleared to ‘0’ after waking up

TXEn — Enables the transmitter unit (only UART and SPI mode).
- 0 Transmitter is disabled
- 1 Transmitter is enabled

RXEn — Enables the receiver unit (only UART and SPI mode).
- 0 Receiver is disabled
- 1 Receiver is enabled

USInEN — Activate USIn function block by supplying.
- 0 USIn is disabled
- 1 USIn is enabled

DBLSn — This bit selects receiver sampling rate (only UART).
- 0 Normal asynchronous operation
- 1 Double Speed asynchronous operation

USInCR3 (USIn Control Register 3: For UART, SPI, and I2C mode): DBH/EBH, n = 0, 1

7	6	5	4	3	2	1	0
MASTERn	LOOPSn	DISSCKn	USInSSEN	FXCHn	USInSB	USInTX8	USInRX8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Initial value: 00H

MASTERn	Selects master or slave in SPI and synchronous mode operation and controls the direction of SCKn pin	
	0	Slave mode operation (External clock for SCKn).
	1	Master mode operation (Internal clock for SCKn).
LOOPSn	Controls the loop back mode of USIn for test mode (only UART and SPI mode)	
	0	Normal operation
	1	Loop Back mode
DISSCKn	In synchronous mode of operation, selects the waveform of SCKn output	
	0	ACK is free-running while UART is enabled in synchronous master mode
	1	ACK is active while any frame is on transferring
USInSSEN	This bit controls the SSn pin operation (only SPI mode)	
	0	Disable
	1	Enable
FXCHn	SPI port function exchange control bit (only SPI mode)	
	0	No effect
	1	Exchange MOSIn and MISOn function
USInSB	Selects the length of stop bit in asynchronous or synchronous mode of operation.	
	0	1 Stop Bit
	1	2 Stop Bit
USInTX8	The ninth bit of data frame in asynchronous or synchronous mode of operation. Write this bit first before loading the USInDR register	
	0	MSB (9^{th} bit) to be transmitted is ‘0’
	1	MSB (9^{th} bit) to be transmitted is ‘1’
USInRX8	The ninth bit of data frame in asynchronous or synchronous mode of operation. Read this bit first before reading the receive buffer (only UART mode).	
	0	MSB (9^{th} bit) received is ‘0’
	1	MSB (9^{th} bit) received is ‘1’

USI0CR4 (USIn Control Register 4: For I2C mode): DCH/ECH, n = 0, 1

7	6	5	4	3	2	1	0
IICnIFR	–	RESETn	IICnIE	ACKnEN	IMASTERn	STOPCn	STARTCn
R	–	R/W	R/W	R/W	R	R/W	R/W

Initial value: 00H

IICnIFR — This is an interrupt flag bit for I2C mode. When an interrupt occurs, this bit becomes '1'. This bit is cleared when write any values in the USInST2. Writing "1" has no effect.

0 I2C interrupt no generation

1 I2C interrupt generation

RESETn — Initialize Internal registers of I2C

0 No operation

1 Initialize I2C, auto cleared

IICnIE — Interrupt Enable bit for I2C mode

0 Interrupt from I2C is inhibited (use polling)

1 Enable interrupt for I2C

ACKnEN — Controls ACK signal Generation at ninth SCLn period.

0 No ACK signal is generated (SDAn =1)

1 ACK signal is generated (SDAn =0)

NOTE **ACK signal is output (SDA =0) for the following 3 cases.**
When received address packet equals to USInSLA bits in USInSAR. When received address packet equals to value 0x00 with GCALLn enabled.
When I2C operates as a receiver (master or slave)

IMASTERn — Represent operating mode of I2C

0 I2C is in slave mode

1 I2C is in master mode

STOPCn — When I2C is master, STOP condition generation

0 No effect

1 STOP condition is to be generated

STARTCn — When I2C is master, START condition generation

0 No effect

1 START or repeated START condition is to be generated

USInST1 (USIn Status Register 1: For UART and SPI mode): E1H/F1H, n = 0, 1

7	6	5	4	3	2	1	0
DREn	TXCn	RXCn	WAKEn	USInRST	DORn	FEn	PEn
R/W	R/W	R	R/W	R/W	R	R/W	R/W

Initial value: 80H

DREn — The DREn flag indicates if the transmit buffer (USInDR) is ready to receive new data. If DREn is '1', the buffer is empty and ready to be written. This flag can generate a DREn interrupt.

0 Transmit buffer is not empty.

1 Transmit buffer is empty.

TXCn — 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 TXCn interrupt is executed. This flag can generate a TXCn interrupt. This bit is automatically cleared.

0 Transmission is ongoing.

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

RXCn — 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 RXCn flag can be used to generate an RXCn interrupt.

0 There is no data unread in the receive buffer

1 There are more than 1 data in the receive buffer

WAKEn — This flag is set when the RXDn pin is detected low while the CPU is in STOP mode. This flag can be used to generate a WAKEn interrupt. This bit is set only when in asynchronous mode of operation. This bit should be cleared by program software. (only UART mode)

0 No WAKE interrupt is generated.

1 WAKE interrupt is generated

USInRST — This is an internal reset and only has effect on USIn. Writing '1' to this bit initializes the internal logic of USIn and this bit is automatically cleared to '0'.

0 No operation

1 Reset USIn

DORn — 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

FEn — 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. (only UART mode)

0 No Frame Error

1 Frame Error detected

PEn — This bit is set if the next character in the receive buffer has a Parity Error to be received while Parity Checking is enabled. This bit is valid until the receive buffer is read. (only UART mode)

0 No Parity Error

1 Parity Error detected

USInST2 (USIn Status Register 2: For I2C mode): E2H/F2H, n = 0, 1

7	6	5	4	3	2	1	0
GCALLn	TENDn	STOPDn	SSELn	MLOSTn	BUSYn	TMODEn	RXACKn
R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

GCALLn[NOTE] This bit has different meaning depending on whether I2C is master or slave. When I2C is a master, this bit represents whether it received AACK (address ACK) from slave.

0 No AACK is received (Master mode)

1 AACK is received (Master mode)

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

0 General call address is not detected (Slave mode)

1 General call address is detected (Slave mode)

TENDn[NOTE] This bit is set when 1-byte of data is transferred completely

0 1 byte of data is not completely transferred

1 1 byte of data is completely transferred

STOPDn[NOTE] This bit is set when a STOP condition is detected.

0 No STOP condition is detected

1 STOP condition is detected

SSELn[NOTE] This bit is set when I2C is addressed by other master.

0 I2C is not selected as a slave

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

MLOSTn[NOTE] This bit represents the result of bus arbitration in master mode.

0 I2C maintains bus mastership

1 I2C maintains bus mastership during arbitration process

BUSYn This bit reflects bus status.

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

1 I2C bus is busy

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

0 I2C is a receiver

1 I2C is a transmitter

RXACKn This bit shows the state of ACK signal

0 No ACK is received

1 ACK is received at ninth SCL period

NOTE: These bits can be a source of interrupt. When an I2C interrupt occurs except for STOP mode, the SCLn line is hold LOW. To release SCLn, write arbitrary value to USInST2. When USInST2 is written, the TENDn, STOPDn, SSELn, MLOSTn, and RXACKn bits are cleared.

15.23 Baud rate settings (example)

Table 32. Example1 of USI0BD and USI1BDSettings for Commonly Used Oscillator Frequencies

Baud rate (bps)	fx = 1.00MHz		fx = 1.8432MHz		fx = 2.00MHz	
	USI0BD/USI1BD	Error	USI0BD/USI1BD	Error	USI0BD/USI1BD	Error
2400	25	0.2%	47	0.0%	51	0.2%
4800	12	0.2%	23	0.0%	25	0.2%
9600	6	-7.0%	11	0.0%	12	0.2%
14.4k	3	8.5%	7	0.0%	8	-3.5%
19.2k	2	8.5%	5	0.0%	6	-7.0%
28.8k	1	8.5%	3	0.0%	3	8.5%
38.4k	1	-18.6%	2	0.0%	2	8.5%
57.6k	—	—	1	-25.0%	1	8.5%
76.8k	—	—	1	0.0%	1	-18.6%
115.2k	—	—	—	—	—	—
230.4k	—	—	—	—	—	—
2400	25	0.2%	47	0.0%	51	0.2%
4800	12	0.2%	23	0.0%	25	0.2%
9600	6	-7.0%	11	0.0%	12	0.2%
14.4k	3	8.5%	7	0.0%	8	-3.5%
19.2k	2	8.5%	5	0.0%	6	-7.0%
28.8k	1	8.5%	3	0.0%	3	8.5%
38.4k	1	-18.6%	2	0.0%	2	8.5%
57.6k	—	—	1	-25.0%	1	8.5%
76.8k	—	—	1	0.0%	1	-18.6%
115.2k	—	—	—	—	—	—
230.4k	—	—	—	—	—	—

Table 33. Example2 of USI0BD and USI1BDSettings for Commonly Used Oscillator Frequencies

Baud rate (bps)	fx = 8.00MHz		fx = 11.0592MHz	
	USI0BD/USI1BD	Error	USI0BD/USI1BD	Error
2400	207	0.2%	—	—
4800	103	0.2%	143	0.0%
9600	51	0.2%	71	0.0%
14.4k	34	-0.8%	47	0.0%
19.2k	25	0.2%	35	0.0%
28.8k	16	2.1%	23	0.0%
38.4k	12	0.2%	17	0.0%
57.6k	8	-3.5%	11	0.0%
76.8k	6	-7.0%	8	0.0%
115.2k	3	8.5%	5	0.0%
230.4k	1	8.5%	2	0.0%
250k	1	0.0%	2	-7.8%
0.5M	—	—	—	—
1M	—	—	—	—

16 USART2

Universal Synchronous and Asynchronous serial Receiver and Transmitter (USART) is a highly flexible serial communication device. USART2 of A96G150 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

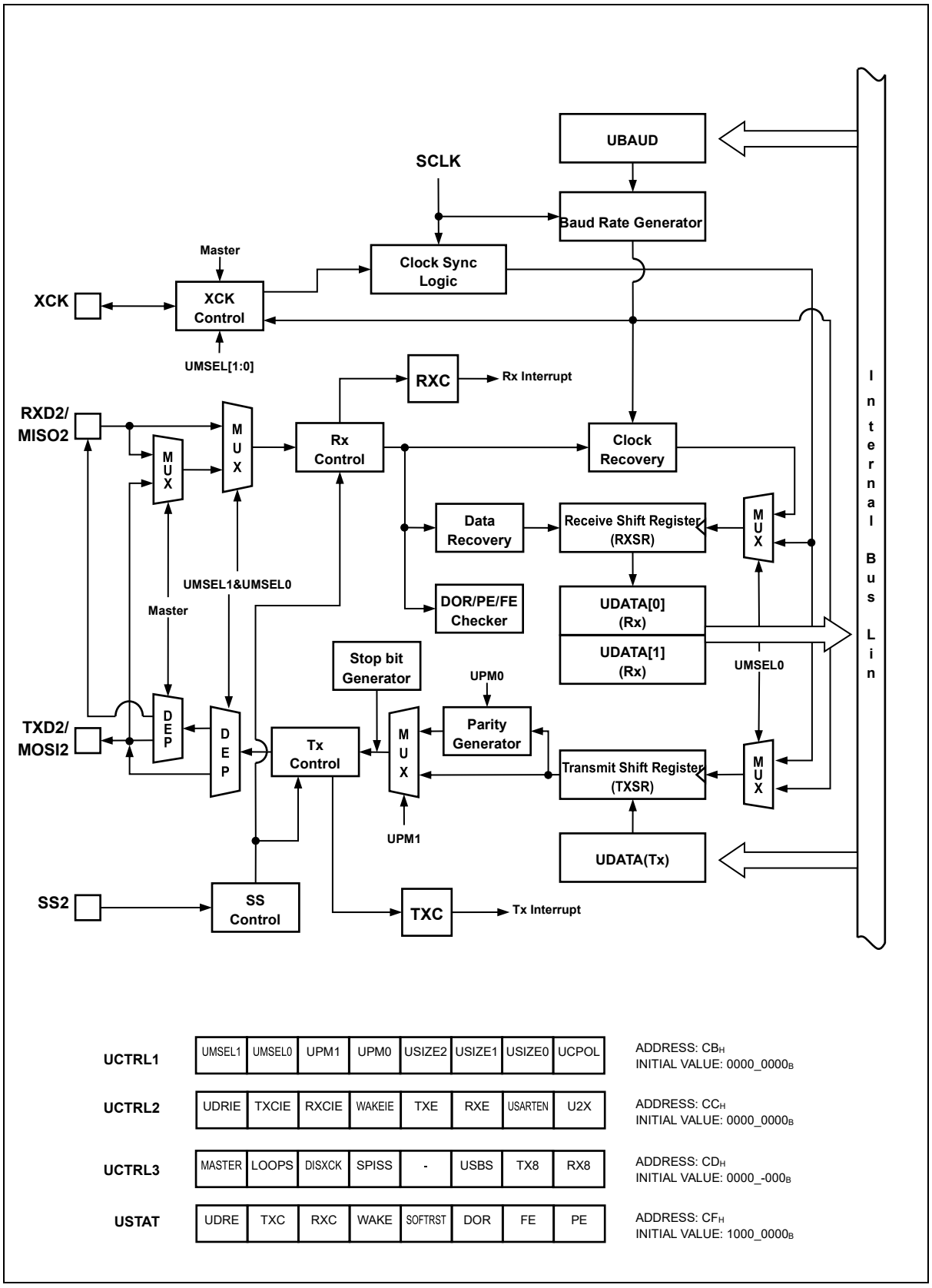
USART2 has three main parts such as a Clock Generator, Transmitter and Receiver.

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 USART2 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.

16.1 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 97. USART2 Block Diagram

16.2 Clock generation

Clock generation logic generates a base clock signal for the Transmitter and the Receiver. USART2 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 USART2 operates in Synchronous or SPI mode.

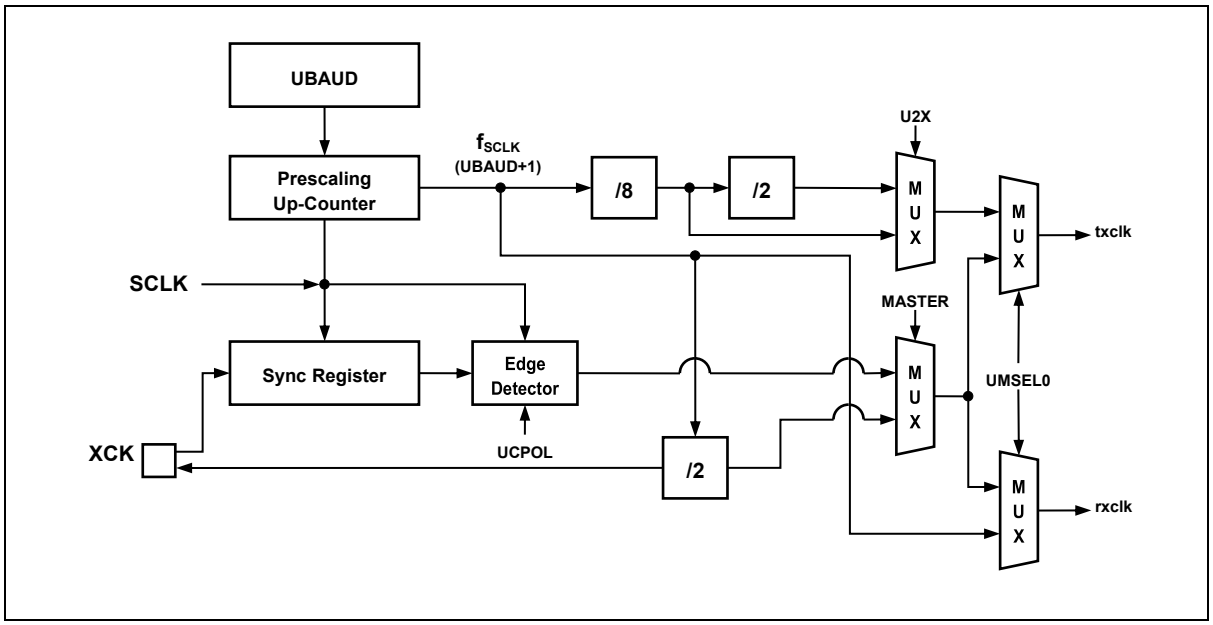


Figure 98. Clock Generation Block Diagram

Table 34 contains equations for calculating the baud rate (in bps).

Table 34. 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	$\text{Baud Rate} = \frac{\text{fSCLK}}{2(\text{UBAUDx} + 1)}$

16.3 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:

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

, where fXCK is frequency of XCK, and fSCLK is frequency of main system clock (SCLK).

16.4 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 at 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 99, when UCPOL is zero, data will be changed at XCK rising edge and sampled at XCK falling edge.

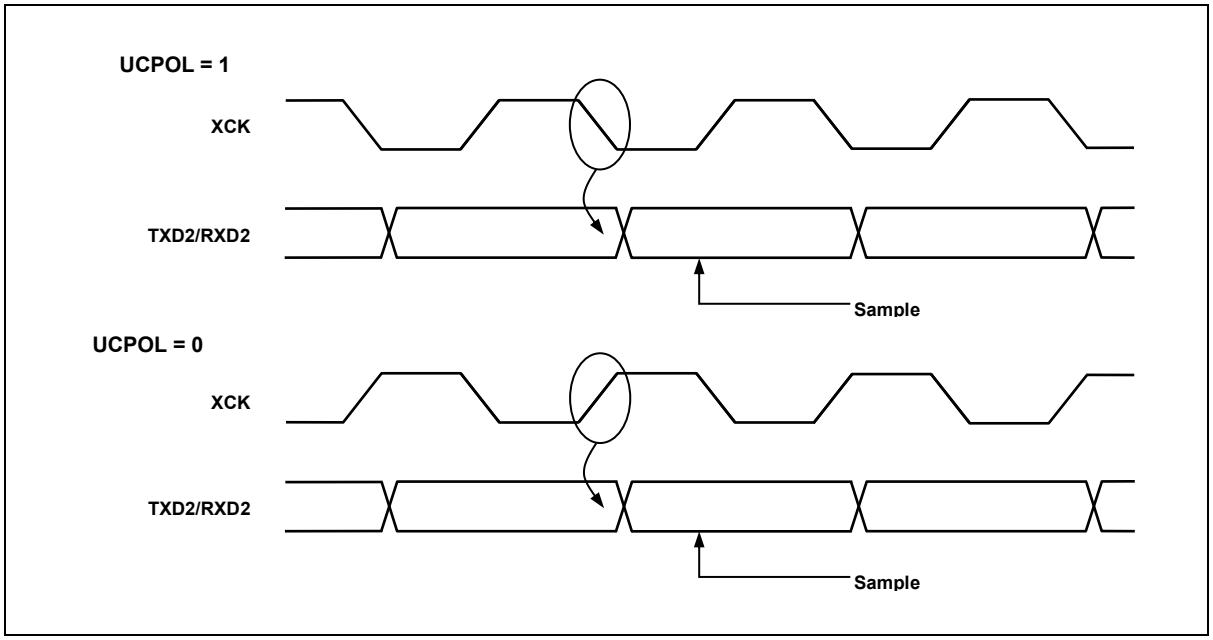


Figure 99. Synchronous Mode XCK Timing

16.5 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.

USART2 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.

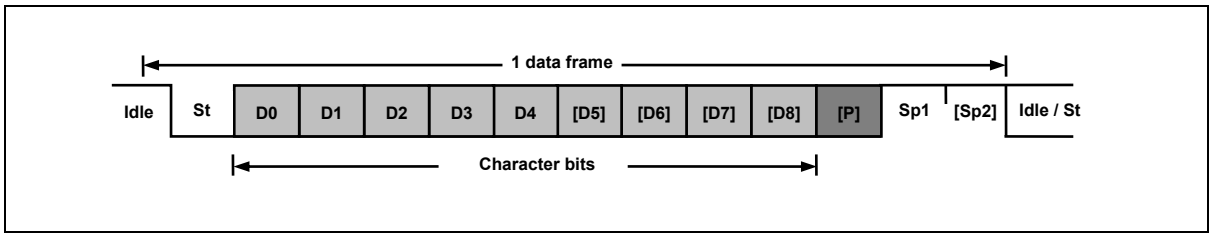


Figure 100. A Frame Format

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 USART2 is set by USIZE[2:0], UPM[1:0] and USBS bits in UCTRL1 register. The Transmitter and the Receiver use the same settings.

16.6 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

16.7 USART2 transmitter

USART2 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 USART2. 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 USART2 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.

16.7.1 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).

16.7.2 Transmitter flag and interrupt

The USART2 Transmitter has 2 flags which indicate its state. One is USART2 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, USART2 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, USART2 Transmit Complete Interrupt is generated while TXC flag is set.

16.7.3 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.

16.7.4 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.

16.8 USART2 receiver

USART2 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 USART2 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 USART2 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.

16.8.1 Receiving Rx data

When USART2 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.

16.8.2 Receiver flag and interrupt

The USART2 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, USART2 Receiver Complete Interrupt is generated while RXC flag is set.

The USART2 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 USART2 is in SPI mode.

16.8.3 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.

16.8.4 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 USART2 is not overridden on the RXD2 pin, so the RXD2 pin becomes normal GPIO or primary function pin.

16.8.5 Asynchronous data reception

To receive asynchronous data frame, the USART2 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 101 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.

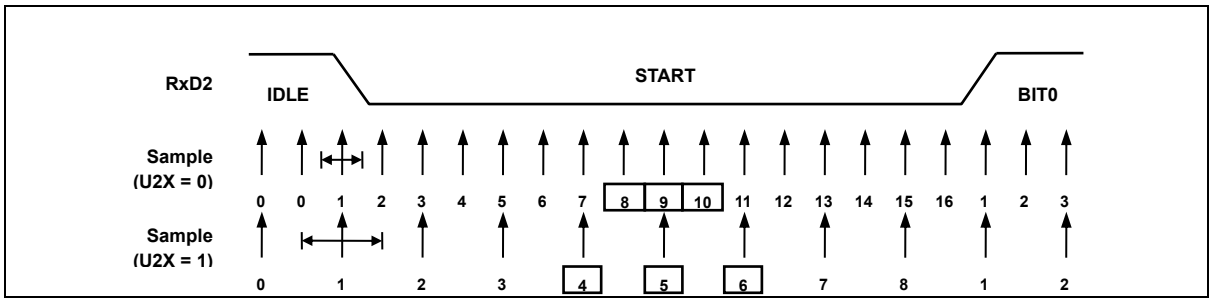


Figure 101. Start Bit Sampling

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.

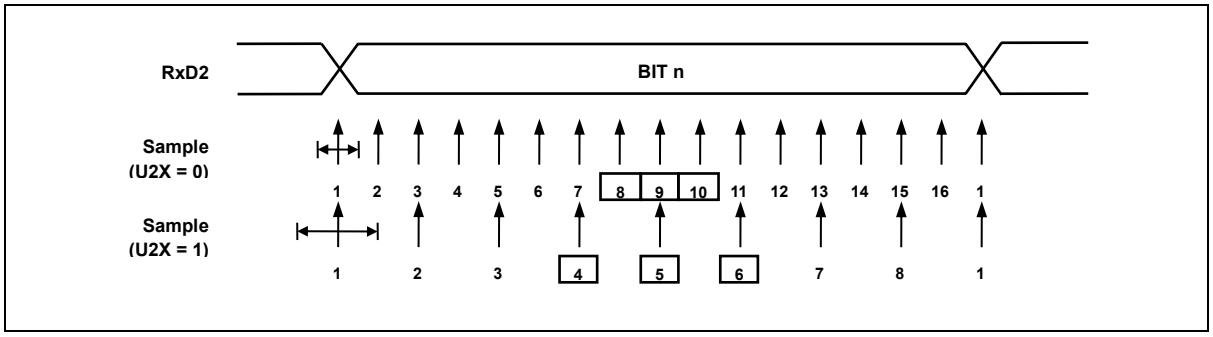


Figure 102. Sampling of Data and Parity Bit

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).

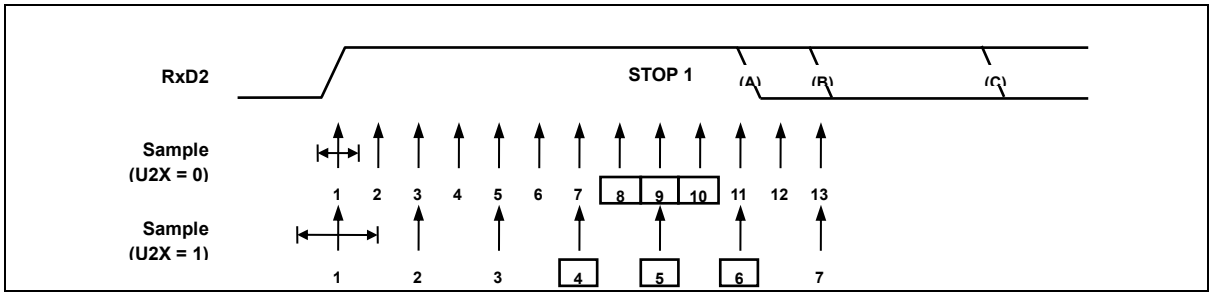


Figure 103. Stop Bit Sampling and Next Start Bit Sampling

16.9 SPI mode

The USART2 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.

16.9.1 SPI clock formats and timing

To accommodate a wide variety of synchronous serial peripherals from different manufacturers, the USART2 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 USART2.

Table 35 shows four combinations of UCPOL and UCPHA for SPI mode 0, 1, 2, and 3.

Table 35. 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)

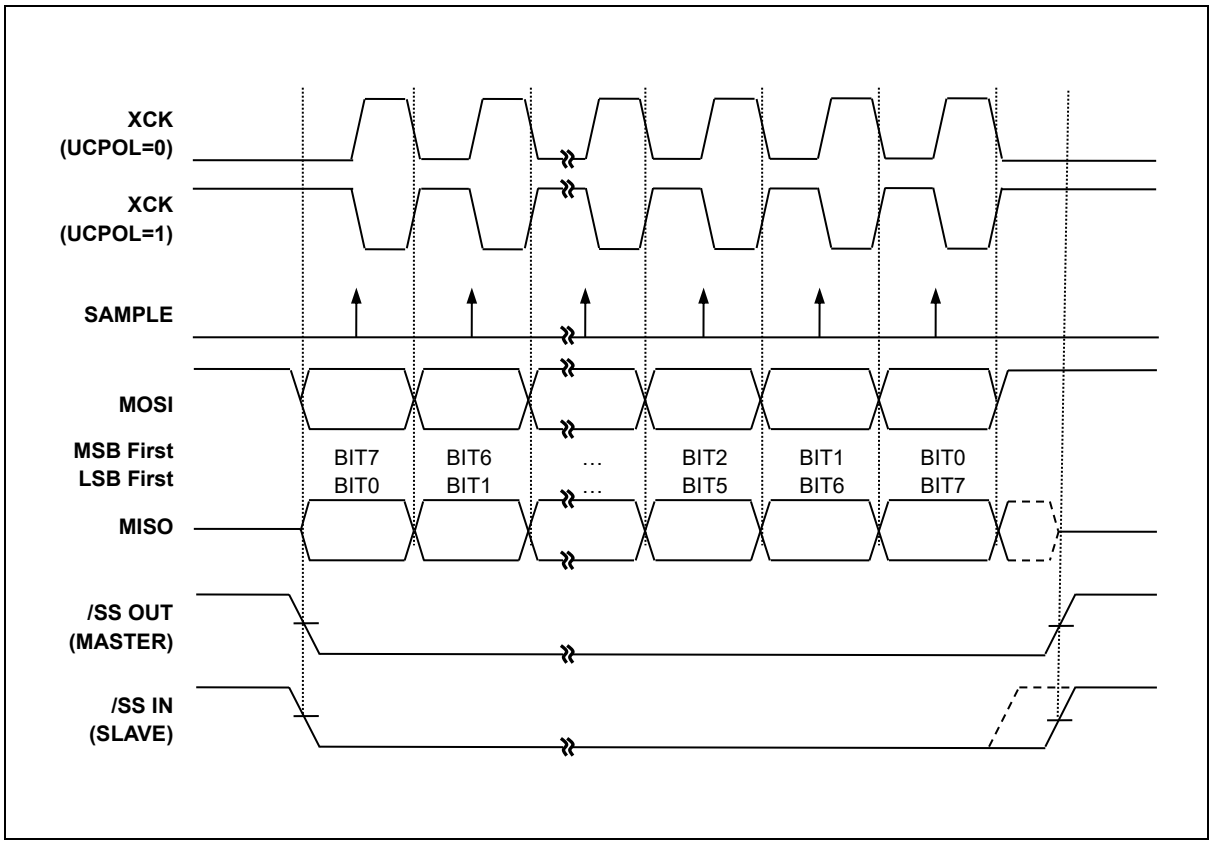


Figure 104. SPI Clock Formats when UCPHA = 0

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 USART2 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.

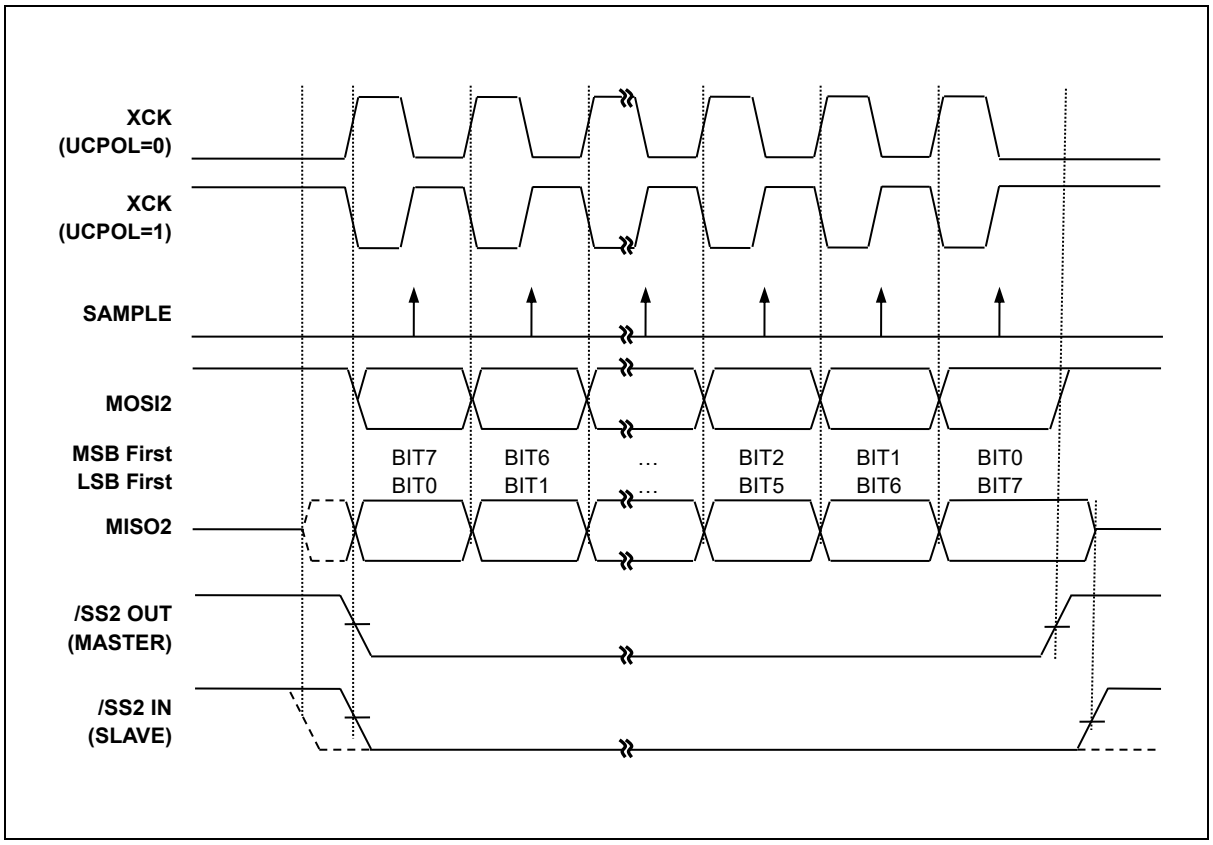


Figure 105. SPI Clock Formats when UCPHA = 1

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 USART2 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 USART2 resources, SPI mode of operation is similar to that of synchronous or asynchronous operation. An SPI transfer is initiated by checking for USART2 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.

16.10 Receiver time out (RTO)

This USART2 system supports the time out function. This function is occur the interrupts when stop bit are not in RX line during URTOC setting value. RTO count stops in RXD signal live state and RTO clear and start is executed by stop bit recognition.

Example condition is listed in Table 36.

Table 36. 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 • Error rate = 0.68 ⇨ **UBAUD = 8**
Real baud rate at sysclk 16Mhz	111,111 bps
1 bit time	9us
Maximum count time	9us * 65536(16bit count) = 589.8ms

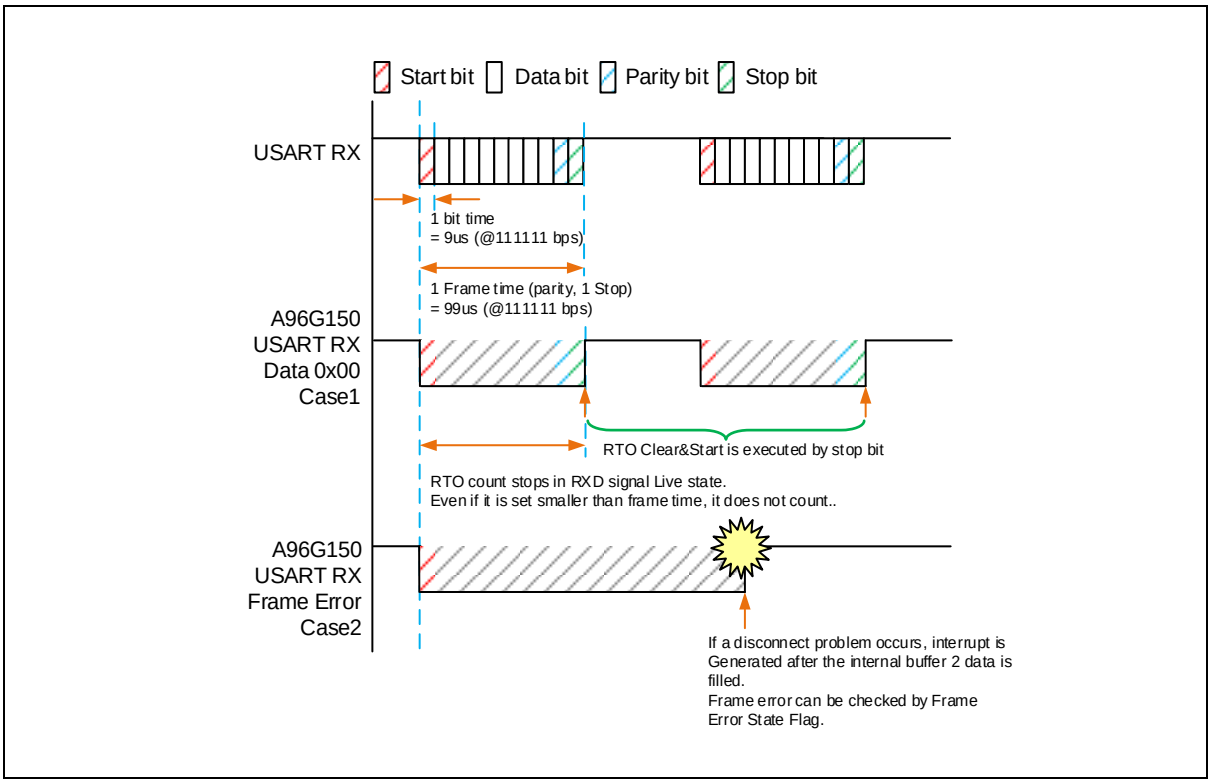


Figure 106. Example for RTO in USART2

16.11 Register map

Table 37. USART2 Register Map

Name	Address	Direction	Default	Description
UCTRL1	CBH	R/W	00H	USART2 Control 1 Register
UCTRL2	CCH	R/W	00H	USART2 Control 2 Register
UCTRL3	CDH	R/W	00H	USART2 Control 3 Register
UCTRL4	1018H	R/W	00H	USART2 Control 4 Register
USTAT	CFH	R	80H	USART2 Status Register
UBAUD	FCH	R/W	FFH	USART2 Baud Rate Generation Register
UDATA	FDH	R/W	00H	USART2 Data Register
FPCR	1019H	R/W	00H	USART2 Floating Point Counter Register
RTOCH	101AH	R	00H	Receiver Time Out Counter High Register
RTOCL	101BH	R	00H	Receiver Time Out Counter Low Register

16.12 Register description

UCTRL1 (USART2 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 USART2

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 (USART2 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 USART2 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

NOTES:
1. WAKEIE must set after USARTEN setting '1'.
2. When using the RXD to wake up from STOP mode, the WAKEIE bit must be cleared to '0' after waking up.

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 USART2 module by supplying clock.
- 0 USART2 is disabled (clock is halted)
- 1 USART2 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 (USART2 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 USART2, 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 (USART2 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 (USART2 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:

Example 1

- 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**

Example 2

- 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

16.13 Baud rate settings (example)

Table 38. 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	ERROR
2400	25	0.2%	51	0.2%	47	0.0%	95	0.0%	51	0.2%	103	0.2%
4800	12	0.2%	25	0.2%	23	0.0%	47	0.0%	25	0.2%	51	0.2%
9600	6	-7.0%	12	0.2%	11	0.0%	23	0.0%	12	0.2%	25	0.2%
14.4K	3	8.5%	8	-3.5%	7	0.0%	15	0.0%	8	-3.5%	16	2.1%
19.2K	2	8.5%	6	-7.0%	5	0.0%	11	0.0%	6	-7.0%	12	0.2%
28.8K	1	8.5%	3	8.5%	3	0.0%	7	0.0%	3	8.5%	8	-3.5%
38.4K	1	-18.6%	2	8.5%	2	0.0%	5	0.0%	2	8.5%	6	-7.0%
57.6K	-	-	1	8.5%	1	-25.0%	3	0.0%	1	8.5%	3	8.5%
76.8K	-	-	1	-18.6%	1	0.0%	2	0.0%	1	-18.6%	2	8.5%
115.2K	-	-	-	-	-	-	1	0.0%	-	-	1	8.5%
230.4K	-	-	-	-	-	-	-	-	-	-	-	-

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

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

16.14 0% Error baud rate

USART2 system of A96G150 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.

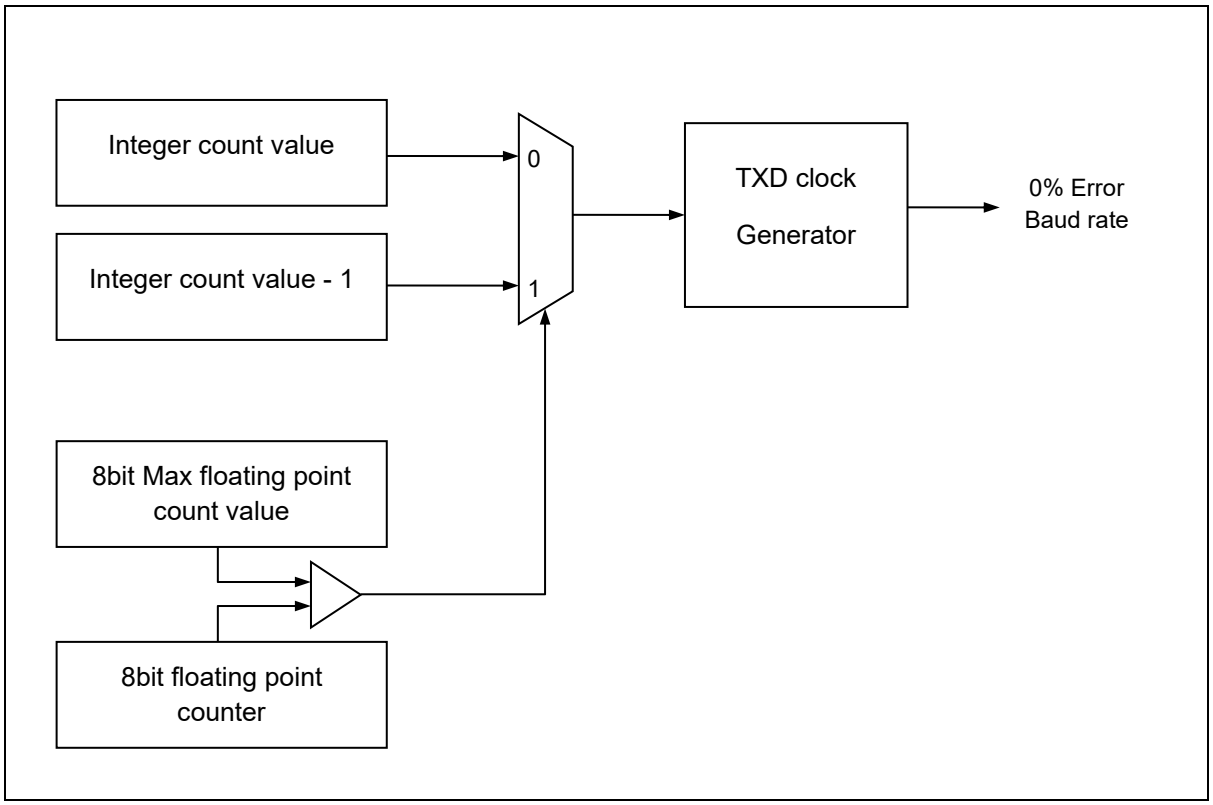


Figure 107. 0% Error Baud Rate Block Diagram

17 LCD Driver

The LCD driver is controlled by the LCD control register (LCD_CR) and LCD driver bias and contrast control register (LCD_BCCR). The LCLK[1:0] determines the frequency of COM signal scanning of each segment output. A RESET clears the LCD control register LCD_CR and LCD_BCCR values to logic ‘0’.

The LCD display can continue operating during IDLE and STOP modes.

The clock and duty for LCD driver is automatically initialized by hardware, whenever LCD_CR register data value is rewritten. So, don’t rewrite LCD_CR frequently.

17.1 LCD Display RAM organization

Display data are stored to the display data area in the external data memory. The display data which stored to the display external data area (address 1050H-1067H) are read automatically and sent to the LCD driver by the hardware.

The LCD driver generates the segment signals and common signals in accordance with the display data and drive method. Therefore, display patterns can be changed by only overwriting the contents of the display external data area with a program.

Figure 108 shows the correspondence between the display external data area and the COM/SEG pins. The LCD is turned on when the display data is “1” and turned off when “0”.

The LCD_DR registers are set to dummy values by default, therefore the registers must be cleared to ‘0x00’ before use.

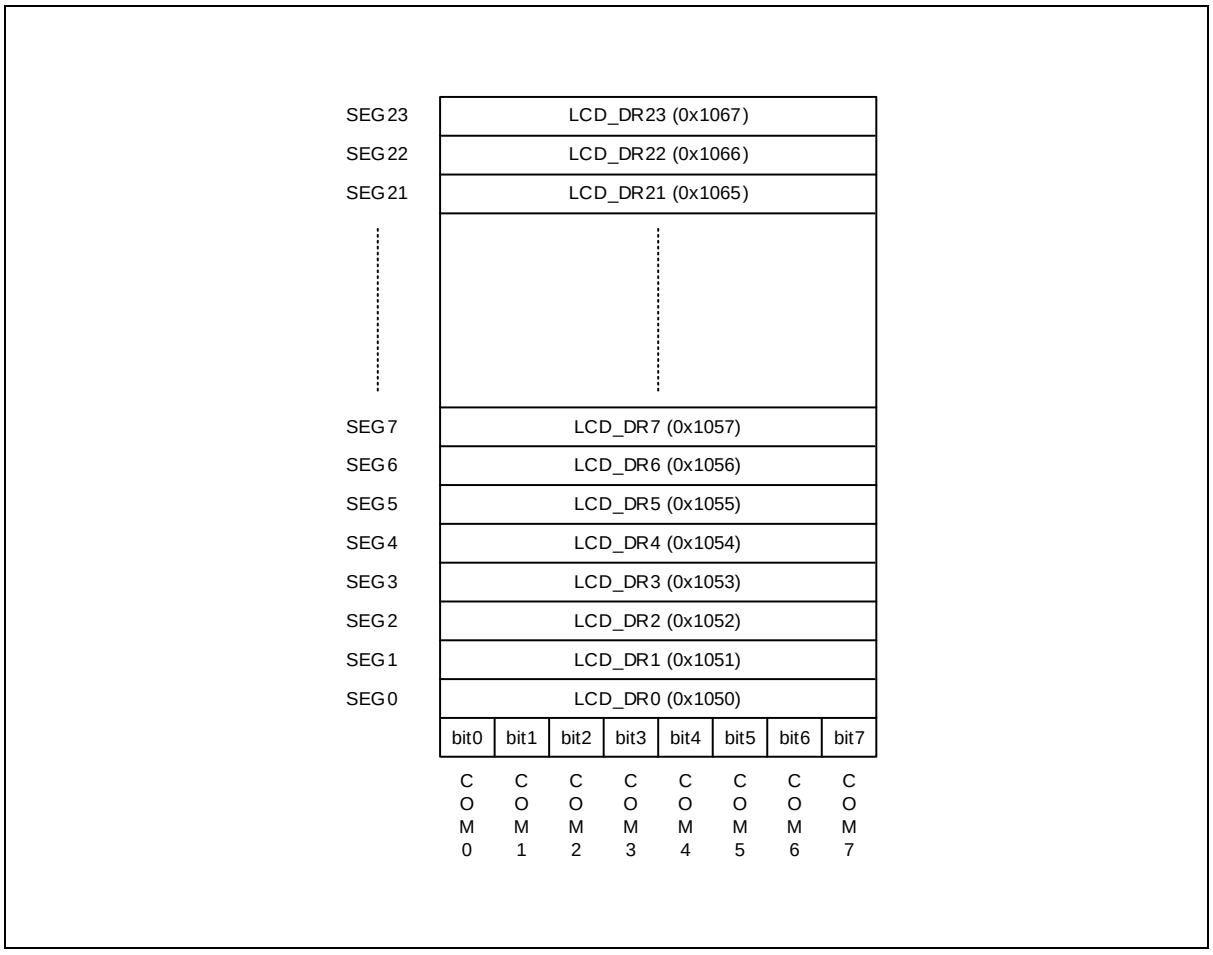


Figure 108. LCD Display RAM

17.2 LCD Signal waveform

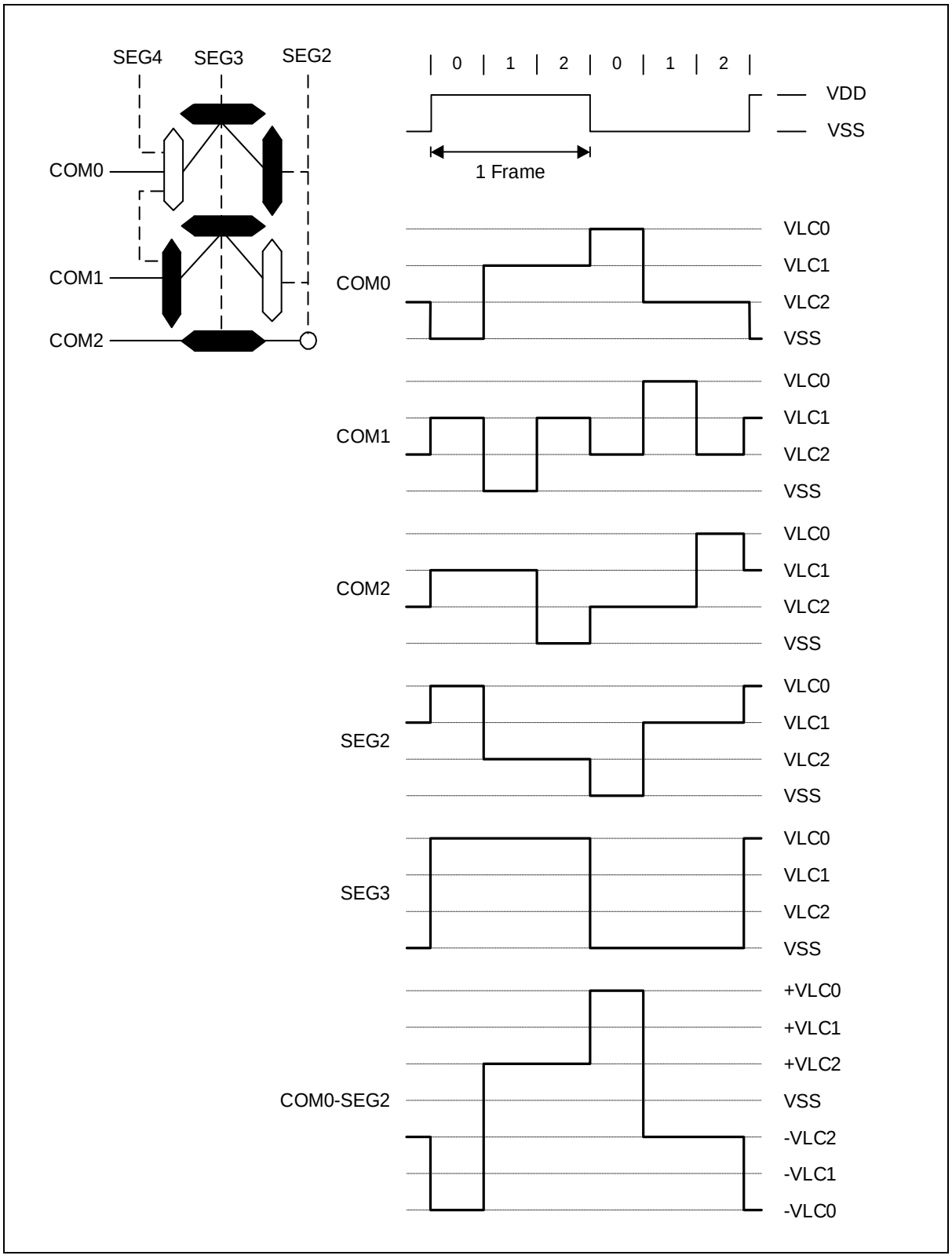


Figure 109. LCD Signal Waveforms (1/3 Duty, 1/3 Bias)

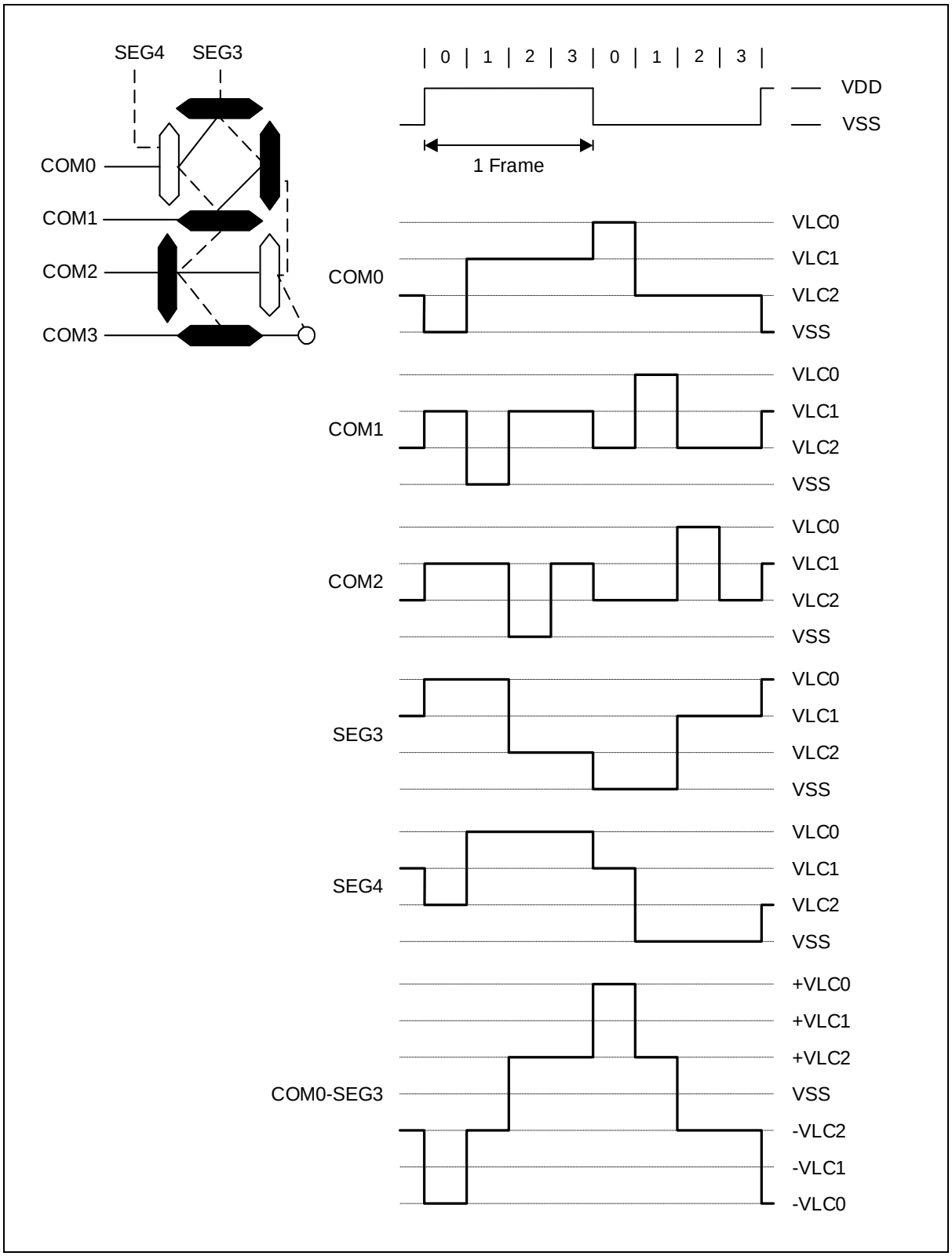


Figure 110. LCD Signal Waveforms (1/4 Duty, 1/3 Bias)

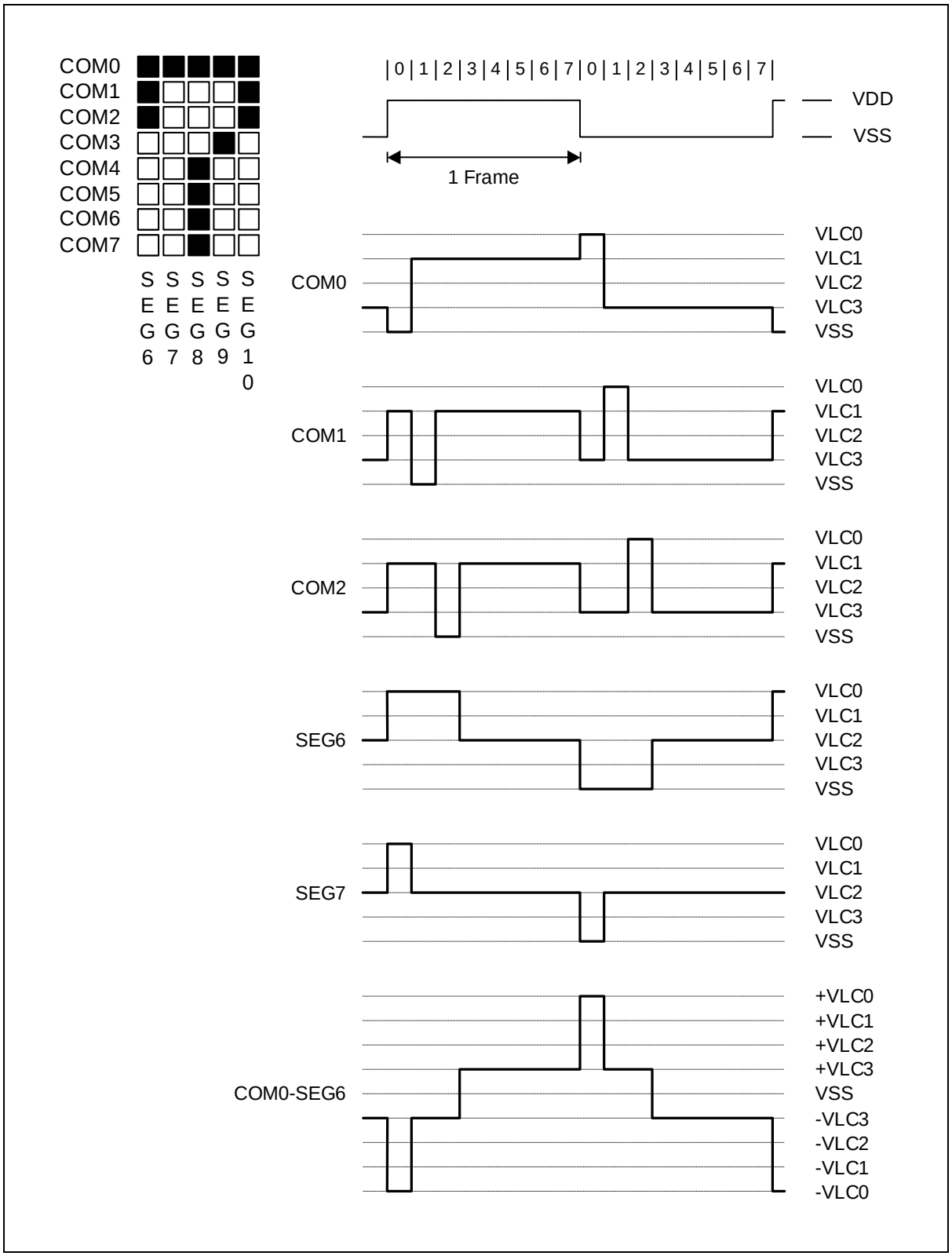
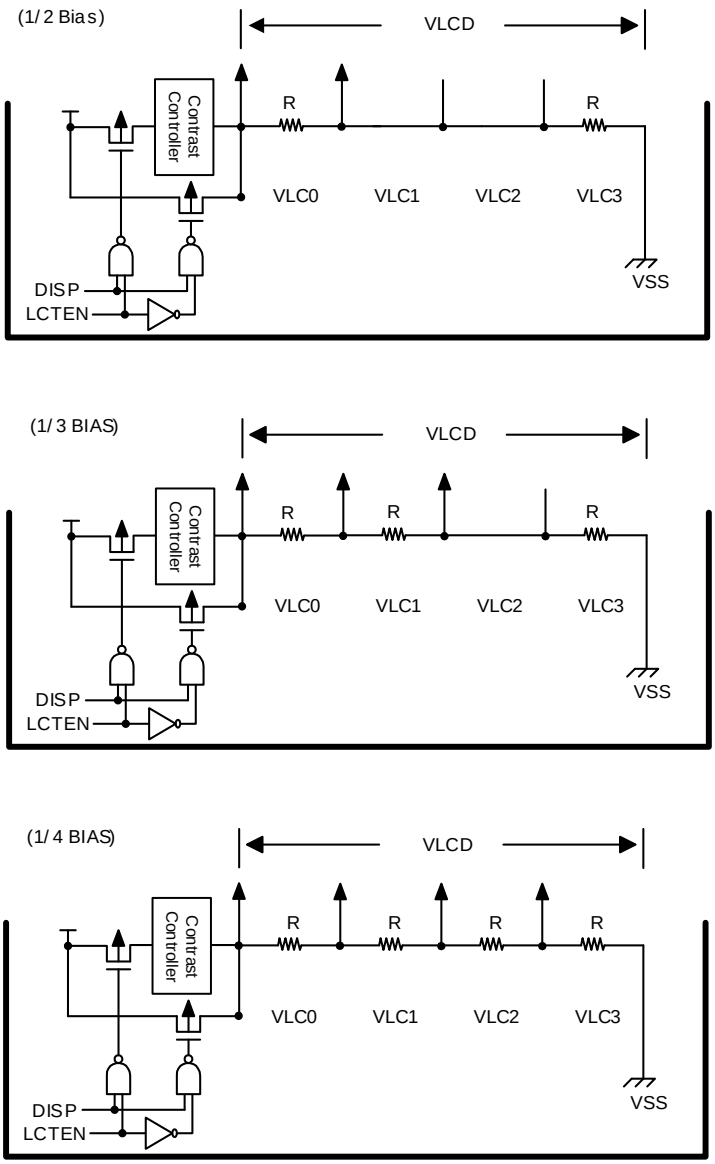


Figure 111. LCD Signal Waveforms (1/8 Duty, 1/4 Bias)

17.3 Internal resistor bias connection

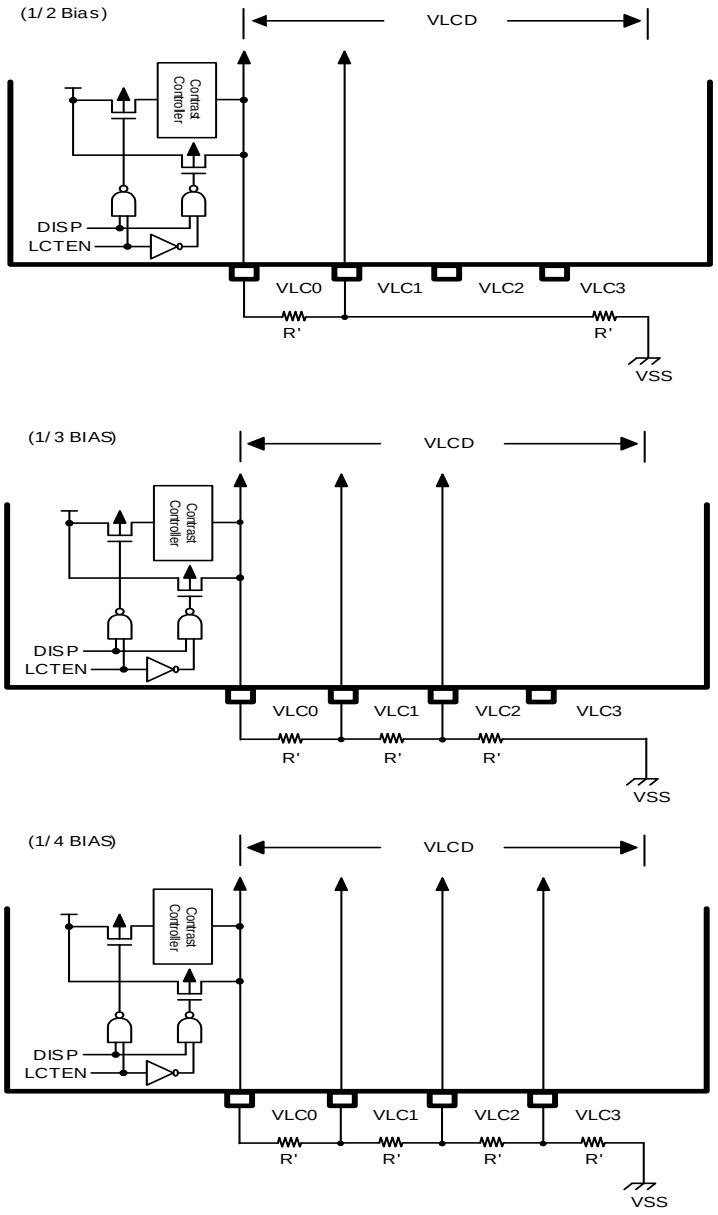


NOTES:

1. When 1/2 bias are selected, the VLC1, VLC2 and VLC3 are internally connected.
2. When 1/3 bias are selected, the VLC2 and VLC3 are internally connected.
3. When the internal resistors are selected, all the VLC0, VLC1, VLC2 and VLC3 pins can be used for normal I/O.

Figure 112. Internal Resistor Bias Connection

17.4 External resistor bias connection

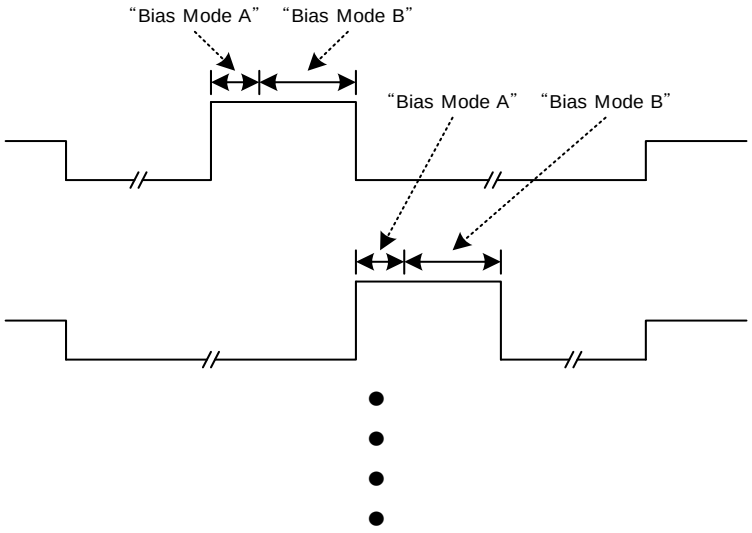


NOTES:

1. When the external resistor bias is selected, the internal resistors for bias are disconnected.
2. When the external resistor bias is selected, the dividing resistors should be connected like the above figure and the needed bias pins should be selected as the LCD bias function pins (VLC0, VLC1, VLC2 and VLC3) by PxFSR register.
3. it must set the LCDDR, LCDEPEN, VLE_EN of the LCDBSSRH register to '1'. And the others must be set to '0'.

Figure 113. External Resistor Bias Connection

17.5 LCD Automatic bias control timing



NOTE: Refer to the below when the LCD automatic bias is on.

- “Bias Mode A”: Always RLCD1 (10kΩ).
- “Bias Mode B”: A LCD bias dividing resistor which is selected by IRSEL[1:0] bits

Figure 114. LCD Automatic Bias Control Timing Diagram

17.6 Block diagram

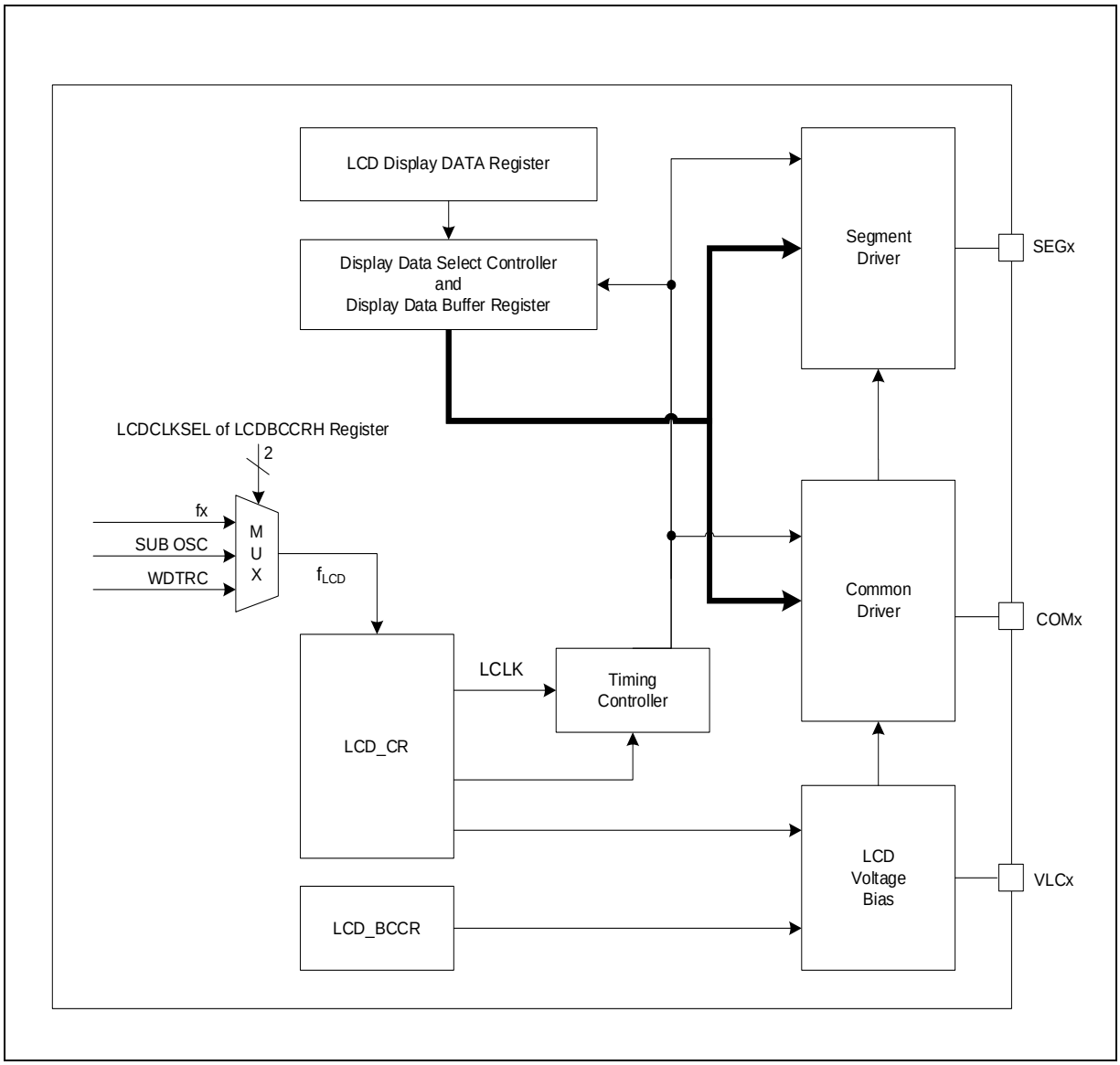


Figure 115. Block Diagram Figure

17.7 Register map

Table 39. LCD Register Map

Name	Address	Direction	Default	Description
LCDCR	1048H	R/W	00H	LCD Driver Control Register
LCDBCCRH	1049H	R/W	00H	LCD Automatic Bias and Contrast Control High Register
LCDBCCRL	104AH	R/W	00H	LCD Automatic Bias and Contrast Control Low Register
LCDBSSRH	104BH	R/W	00H	LCD source Selection High Register
LCDBSSRL	104CH	R/W	00H	LCD source Selection Low Register
LCDDR0~23	1050H~1067H	R/W	XXH	LCD Display Data Register

17.8 Register description

LCDCR (LCD Driver Control Register): 1048H

7	6	5	4	3	2	1	0
IRSEL[1]	IRSEL[0]	DBS[2]	DBS[1]	DBS[0]	LCLK[1]	LCLK[0]	DISP
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

IRSEL[1:0] Internal LCD Bias Dividing Resistor Selection bits.

IRSEL1	IRSEL0	Description
0	0	RLCD3, 105/105/80[kΩ] @(1/2)/(1/3)/(1/4) bias.
0	1	RLCD1, 10/10/10[kΩ] @(1/2)/(1/3)/(1/4) bias.
1	0	RLCD2, 66/66/50[kΩ] @(1/2)/(1/3)/(1/4) bias.
1	1	RLCD4, 320/320/240[kΩ] @(1/2)/(1/3)/(1/4) bias.

DBS[2:0] LCD Duty and Bias Selection bits.

DBS2	DBS1	DBS0	Description
0	0	0	1/8 duty, 1/4 bias.
0	0	1	1/6 duty, 1/4 bias.
0	1	0	1/5 duty, 1/3 bias.
0	1	1	1/4 duty, 1/3 bias.
1	0	0	1/3 duty, 1/3 bias.
1	0	1	1/3 duty, 1/2 bias.
others			Reserved

LCLK[1:0] LCD Clock divider Selection bits

f_{LCD} (LCD clock source) is selected by LCDCLKSEL[1:0].

LCLK1	LCLK0	Description
0	0	f_{LCD}/256 (128Hz @ f_{LCD} = 32.768kHz)
0	1	f_{LCD}/128 (256Hz @ f_{LCD} = 32.768kHz)
1	0	f_{LCD}/64 (512Hz @ f_{LCD} = 32.768kHz)
1	1	f_{LCD}/32 (1024Hz @ f_{LCD} = 32.768kHz)

DISP LCD Display Control bit.

0	Display off
1	Normal display on

LCDBCCRH (LCD Automatic Bias and Contrast Control High Register): 1049H

7	6	5	4	3	2	1	0
LCDCLKSEL[1]	LCDCLKSEL[0]	-	LCDABC	-	BMSEL[2]	BMSEL[1]	BMSEL[0]
R/W	R/W	-	R/W	-	R/W	R/W	R/W

Initial value: 00H

LCDCLKSEL[1:0] LCD Clock Selection bits.

LCDCLKSEL1	LCDCLKSEL0	Description
0	0	SUB OSC
0	1	WDTRC Clock
1	0	fx (system clock frequency)
1	1	fx (system clock frequency)

LCDABC LCD Automatic Bias Control bit.

0	LCD automatic bias is off
1	LCD automatic bias is on

BMSEL[2:0] “Bias Mode A” Time Selection bits.

BMSEL2	BMSEL1	BMSEL0	Description
0	0	0	“Bias Mode A” for 1-clock of $f_{LCD.}$
0	0	1	“Bias Mode A” for 2-clock of f_{LCD}
0	1	0	“Bias Mode A” for 3-clock of f_{LCD}
0	1	1	“Bias Mode A” for 4-clock of f_{LCD}
1	0	0	“Bias Mode A” for 5-clock of f_{LCD}
1	0	1	“Bias Mode A” for 6-clock of f_{LCD}
1	1	0	“Bias Mode A” for 7-clock of f_{LCD}
1	1	1	“Bias Mode A” for 8-clock of f_{LCD}

LCDBCCRL (LCD Automatic Bias and Contrast Control Low Register): 104AH

7	6	5	4	3	2	1	0
-	-	LCTEN	-	VLCD[3]	VLCD[2]	VLCD[1]	VLCD[0]
-	-	R/W	-	R/W	R/W	R/W	R/W

Initial value: 00H

LCTEN LCD Driver Contrast Control bit.

0 Disable LCD driver contrast.

1 Enable LCD driver contrast.

VLCD[3:0] VLC0 Voltage Control when the contrast is enabled.

VLCD3	VLCD2	VLCD1	VLCD0	Description
0	0	0	0	VLC0 = VDD x 16/31 step
0	0	0	1	VLC0 = VDD x 16/30 step
0	0	1	0	VLC0 = VDD x 16/29 step
0	0	1	1	VLC0 = VDD x 16/28 step
0	1	0	0	VLC0 = VDD x 16/27 step
0	1	0	1	VLC0 = VDD x 16/26 step
0	1	1	0	VLC0 = VDD x 16/25 step
0	1	1	1	VLC0 = VDD x 16/24 step
1	0	0	0	VLC0 = VDD x 16/23 step
1	0	0	1	VLC0 = VDD x 16/22 step
1	0	1	0	VLC0 = VDD x 16/21 step
1	0	1	1	VLC0 = VDD x 16/20 step
1	1	0	0	VLC0 = VDD x 16/19 step
1	1	0	1	VLC0 = VDD x 16/18 step
1	1	1	0	VLC0 = VDD x 16/17 step
1	1	1	1	VLC0 = VDD x 16/16 step

NOTE: The above LCD contrast step is based on 1/3 bias with 66kΩ RLCD and on 1/4 bias with 50kΩ RLCD

LCDBSSRH (LCD source Selection High Register): 104BH

7	6	5	4	3	2	1	0
-	-	-	-	VLE_EN	-	LCDDR	LCDEPEN
-	-	-	-	R/W	-	R/W	R/W

Initial value: 00H

VLE_EN	LCD External Bias Control bit	
	0	External Bias Disable
	1	External Bias Enable
LCDDR	LCD Driving Resistor for Bias Select.	
	0	Internal LCD driving resistors for bias
	1	External LCD driving resistors for bias
LCDEPEN	LCD External Bias Path Enable bit	
	0	Disable
	1	Enable

LCDBSSRL (LCD source Selection Low Register): 104CH

7	6	5	4	3	2	1	0
VLC3EN	VLC2EN	VLC1EN	VLC0EN	VLC_REG3	VLC_REG2	VLC_REG1	VLC_REG0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

VLC3EN	External Bias VLC3 Enable bit.	
	0	Disable VLC3
	1	Enable VLC3
VLC2EN	External Bias VLC2 Enable bit.	
	0	Disable VLC2
	1	Enable VLC2
VLC1EN	External Bias VLC1 Enable bit.	
	0	Disable VLC1
	1	Enable VLC1
VLC0EN	External Bias VLC0 Enable bit.	
	0	Disable VLC0
	1	Enable VLC0

18 Cyclic Redundancy Check (CRC)

Using the CRC, it can be monitor the memory of the specified area. This is a one-time operation, and reset is required for continuous operation. In CRC MNT mode, when the CRC read is finished, CRC_FLAG occurs.

In CRC validate mode, if the CRC validate fail after the CRC reading is finished, CRC_FLAG occurs. CRC_FAIL indicates the status of validate results when the CRC read is finished. If the CRC_FLAG is generated and the interrupt is enabled, interrupt service routine is served. CRC-FLAG is not cleared by hardware. CRC-TYPE 0~3 are not supported. Validate is done by comparing the CRC_MNT register and the CRC register value. CRC are not automatically initialized, you need to calculate a new CRC after CRC_H, CRC_L Clear.

Table 40. CRC Mode

CRC TYPE	CRC mode	CRC input	Condition of CRC_FLAG	Condition of CRC reset
CRC_TYPE = 4	MNT	Flash Data	After CRC reading	-
CRC_TYPE = 6	Validate	Flash Data	After CRC reading & Validate fail	Validate fail

18.1 Block diagram

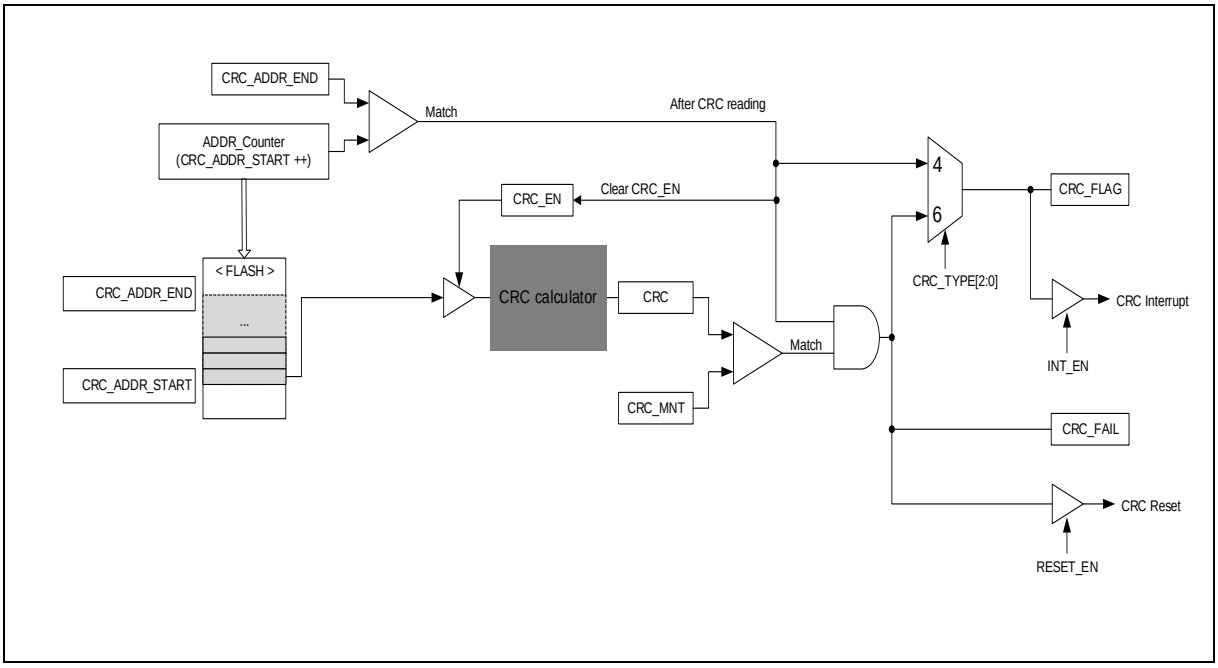


Figure 116. CRC Block Diagram

18.2 Register map

Table 41. CRC Register Map

Name	Address	Direction	Default	Description
CRC_CON	1070H	R/W	00H	CRC Control Register
CRC_H	1072H	R/W	00H	CRC High Register
CRC_L	1073H	R/W	00H	CRC Low Register
CRC_MNT_H	1074H	R/W	00H	CRC Monitor High Register
CRC_MNT_L	1075H	R/W	00H	CRC Monitor Low Register
CRC_ADDR_START_H	1079H	R/W	00H	CRC Start Address High Register
CRC_ADDR_START_M	107AH	R/W	00H	CRC Start Address Middle Register
CRC_ADDR_START_L	107BH	R/W	00H	CRC Start Address Low Register
CRC_ADDR_END_H	107CH	R/W	00H	CRC End Address High Register
CRC_ADDR_END_M	107DH	R/W	00H	CRC End Address Middle Register
CRC_ADDR_END_L	107EH	R/W	00H	CRC End Address Low Register

18.3 Register description

CRC_CON (CRC Control Register): 1070H

7	6	5	4	3	2	1	0
CRC_FLAG	CRC_INTEN	CRC_ RESETEN	CRC_EN	CRC_FAIL	CRC_TYPE2	CRC_TYPE1	CRC_TYPE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

CRC_FLAG	CRC flag. The flag is cleared only by writing a '0' to the bit. So, the flag should be cleared by software. Writing "1" has no effect. * When go to interrupt service routine, CRC_FLAG is not cleared.	
	0	CRC flag not occur
	1	CRC flag occur
INT_EN	Enable CRC interrupt	
	0	CRC interrupt disable
	1	CRC interrupt enable
RESET_EN	Enable CRC reset	
	0	CRC reset disable
	1	CRC reset enable
CRC_EN	Enable CRC operation, it is cleared automatically after the CRC monitoring is finished	
	0	CRC disable
	1	CRC enable
CRC_FAIL	Status of CRC validate.	
	0	Validate pass
	1	Validate fail
CRC_TYPE [2:0]	Select the CRC input data type. * This value can be changed only when CRC_EN=0.	
	0xx	Not used
	100	Specified flash data
	110	Specified flash data, validate CRC value

CRC_H (CRC High Register): 1072H

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

Initial value: 00H

CRC_L (CRC Low Register): 1073H

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

Initial value: 00H

CRC[15:0]	CRC result

CRC_MNT_H (CRC Monitor High Register): 1074H

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

Initial value: 00H

CRC_MNT_L (CRC Monitor Low Register): 1075H

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

Initial value: 00H

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

CRC_ADDR_START_H (CRC Start Address High Register): 1079H

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	CRC_ADDR_START[16]
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

CRC_ADDR_START_M (CRC Start Address Middle Register): 107AH

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

Initial value: 00H

CRC_ADDR_START_L (CRC Start Address Low Register): 107BH

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

Initial value: 00H

CRC_ADDR_START[16:0] CRC start address

CRC_ADDR_END_H (CRC END Address High Register): 107CH

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	CRC_ADDR_END[16]
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

CRC_ADDR_END_M (CRC END Address Middle Register): 107DH

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

Initial value: 00H

CRC_ADDR_END_L (CRC END Address Low Register): 107EH

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

Initial value: 00H

CRC_ADDR_END[16:0] CRC end address

18.4 Polynomial

- CRC16, Polynomial representations Normal : 0x8C81

$$f(x) = 1 + x^7 + x^{10} + x^{11} + x^{15} + x^{16}$$

19 Power down operation

A96G150 has two power-down modes to minimize the power consumption of the device. In power down mode, power consumption is reduced considerably. A96G150 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.

19.1 Peripheral operation in IDLE/STOP mode

Table 42 shows operation status of each peripheral in IDLE mode and STOP mode.

Table 42. 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).
Watch Timer	Operates continuously.	Stops (can be operated with sub clock).
Timer0~4	Operates continuously.	Halts (only when the event counter mode is enabled, timer operates normally).
ADC	Operates continuously.	Stops.
BUZ	Operates continuously.	Stops.
USI0/1	Operates continuously.	Only operates with external clock.
Internal OSC (32MHz)	Oscillates.	Stops when the system clock (fx) is f_{HSIRC}.
WDTRC OSC (128kHz)	Can be operated with setting value.	Can be operated programmable.
Main OSC (0.4~12MHz)	Oscillates.	Stops when fx = f_{XIN}.
Sub OSC (32.768kHz)	Oscillates.	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 • USART2 by RX, WT (sub clock), WDT • USI0/1 by RX, I2C(Slave mode)

19.2 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.

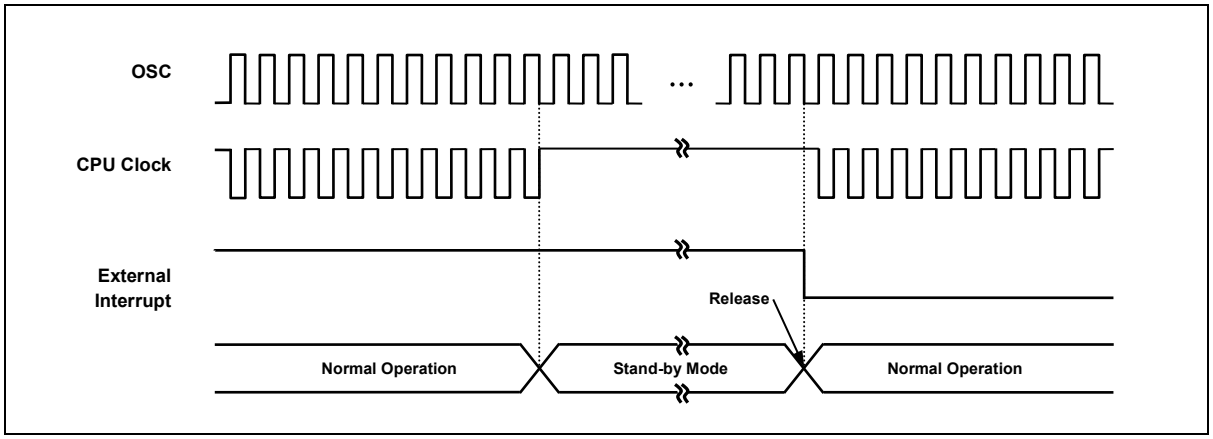


Figure 117. IDLE Mode Release Timing by an External Interrupt

19.3 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 (f_{IRC}) is selected for the system clock and the sub clock (f_{SUB}) is oscillated, the internal RC oscillator stops oscillation and the sub clock is continuously oscillated in stop mode. At that time, the watch timer can be operated with the sub clock.

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 118 shows the timing diagram.

As shown in the Figure 118, 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.

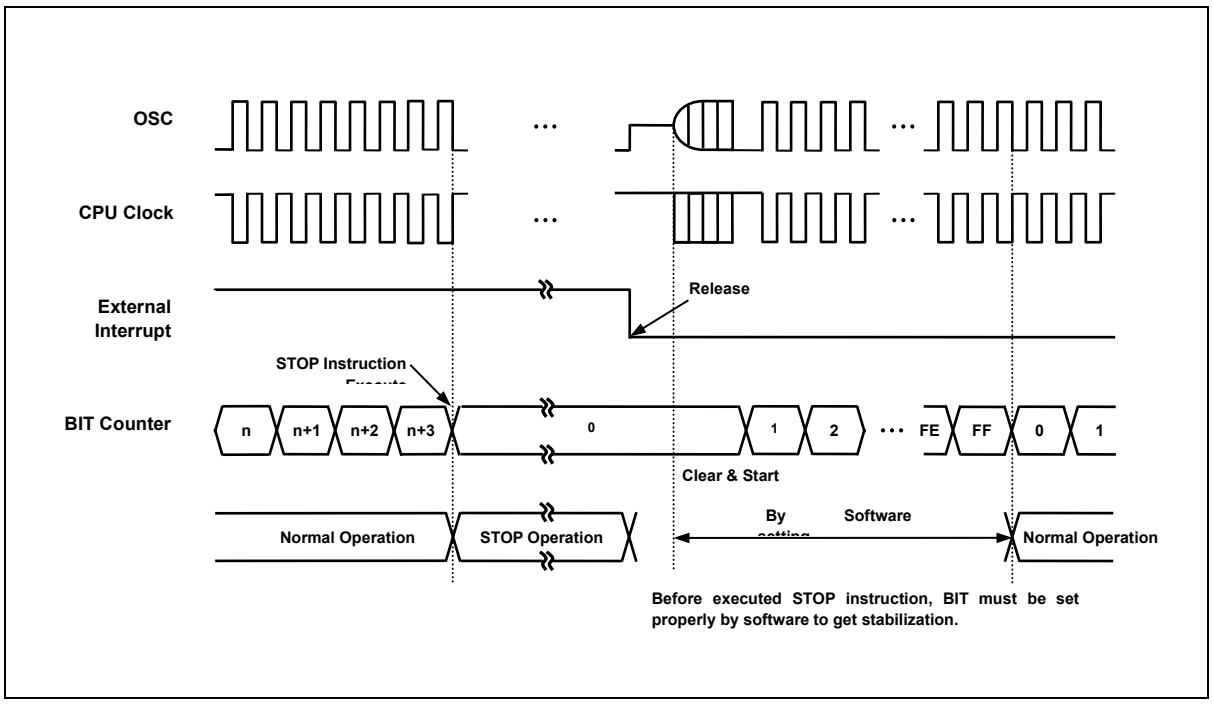


Figure 118. STOP Mode Release Timing by External Interrupt

19.4 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 119). 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'.

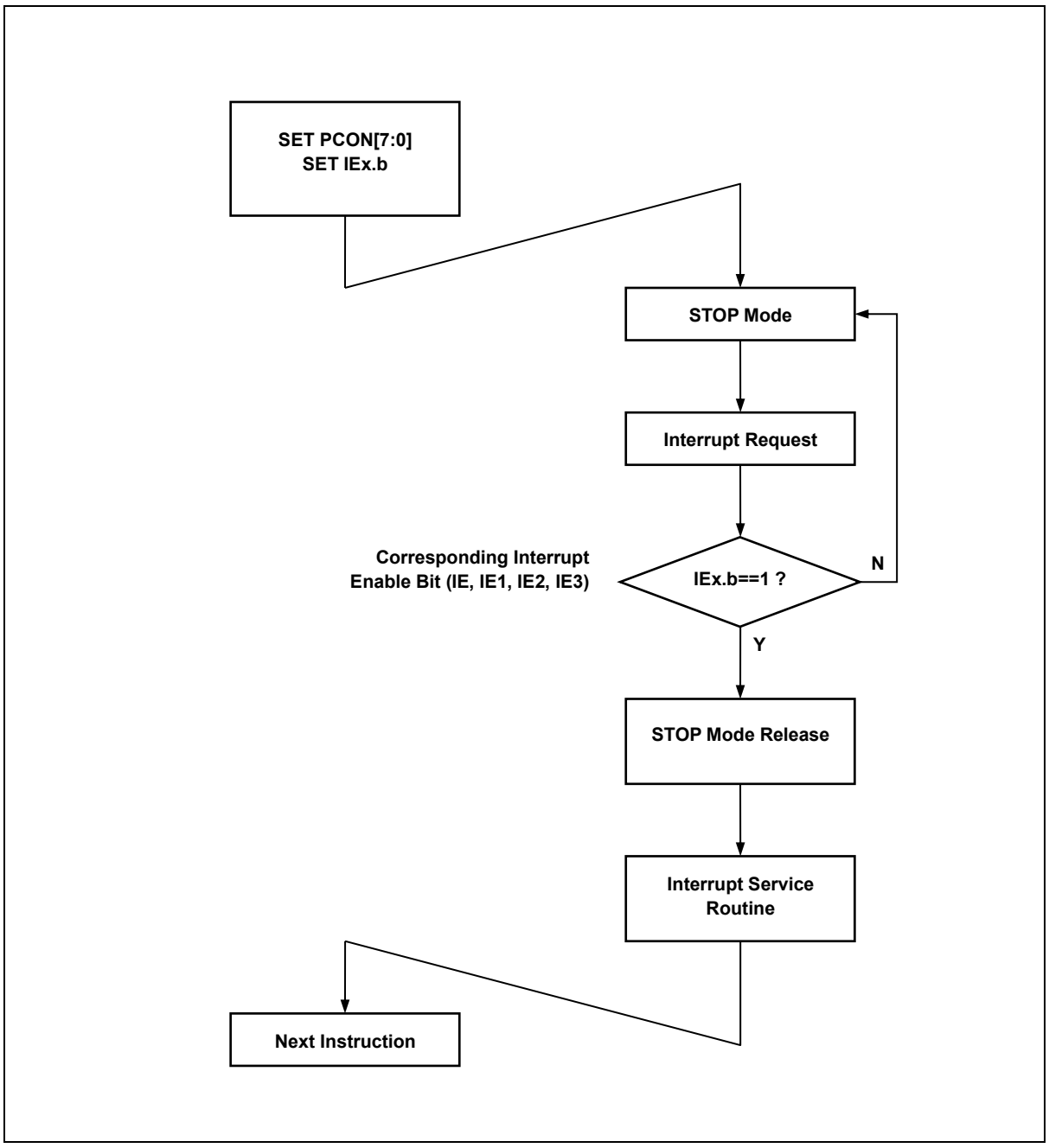


Figure 119. STOP Mode Release Flow

19.5 Register map

Table 43. Power Down Operation Register Map

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

19.6 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

```
MOV   PCON, #01H   ; IDLE mode
NOP
NOP
NOP
.
.
.
```

Example 2

```
MOV   PCON, #03H   ; STOP mode
NOP
NOP
NOP
.
.
.
```

20 Reset

Table 44 shows hardware setting values of main peripherals.

Table 44. 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

A96G150 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

20.1 Reset block diagram

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

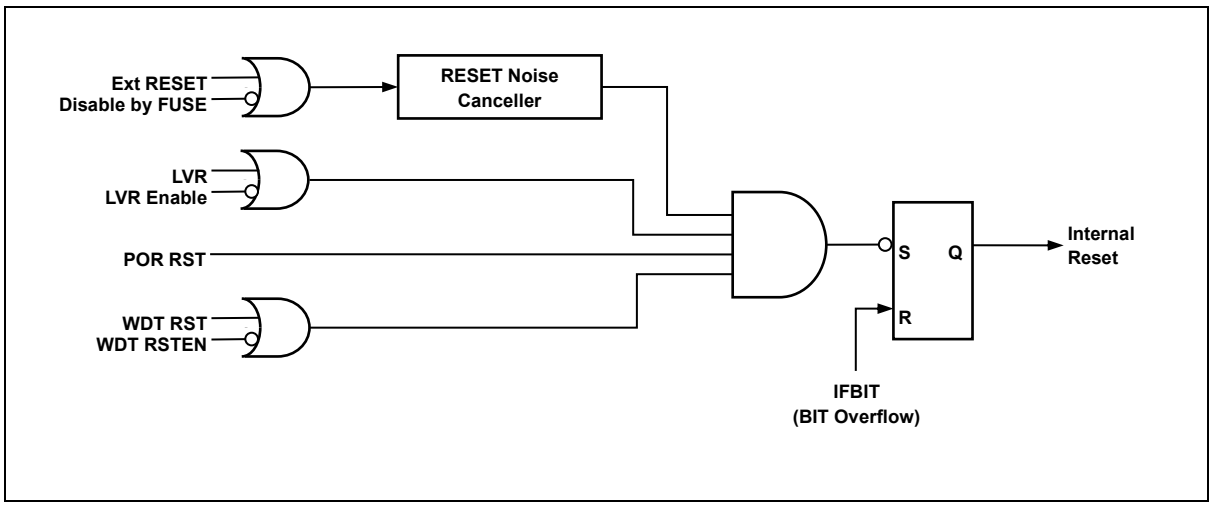


Figure 120. Reset Block Diagram

20.2 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.

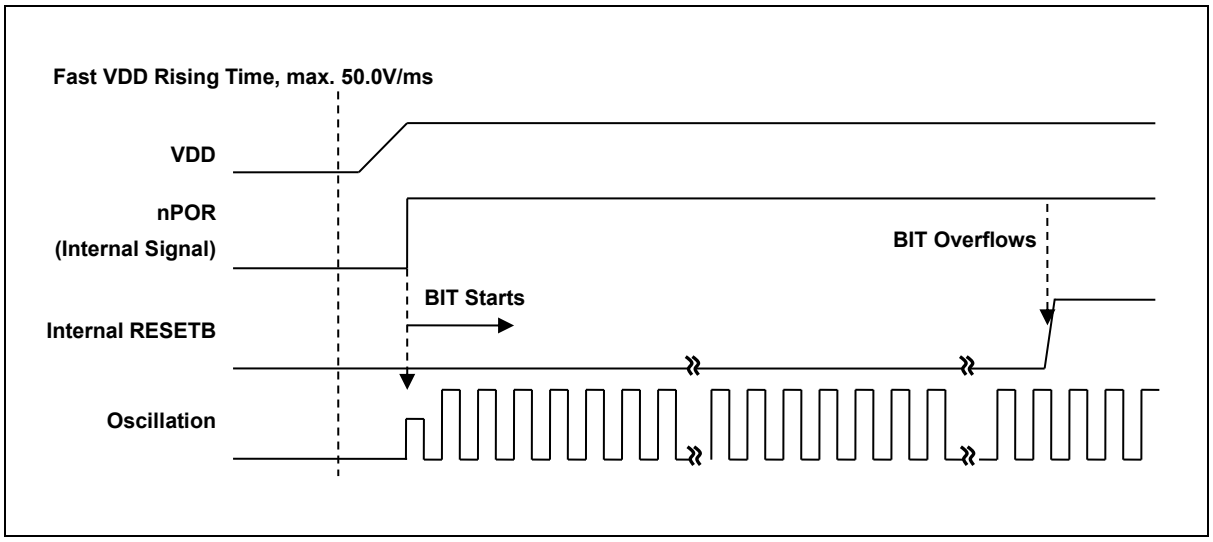


Figure 121. Fast VDD Rising Time

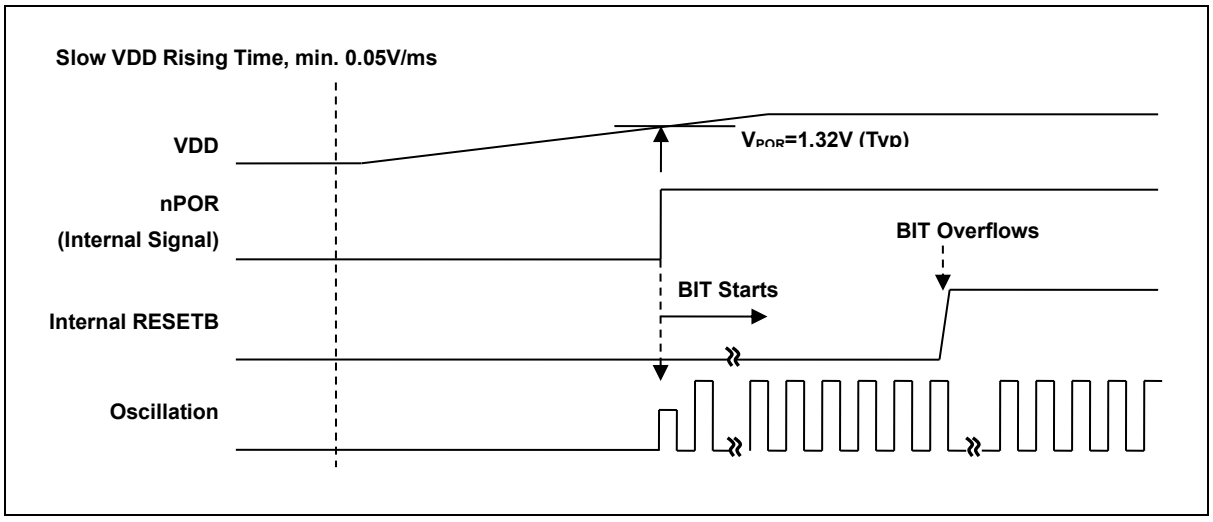


Figure 122. Internal RESET Release Timing on Power-up

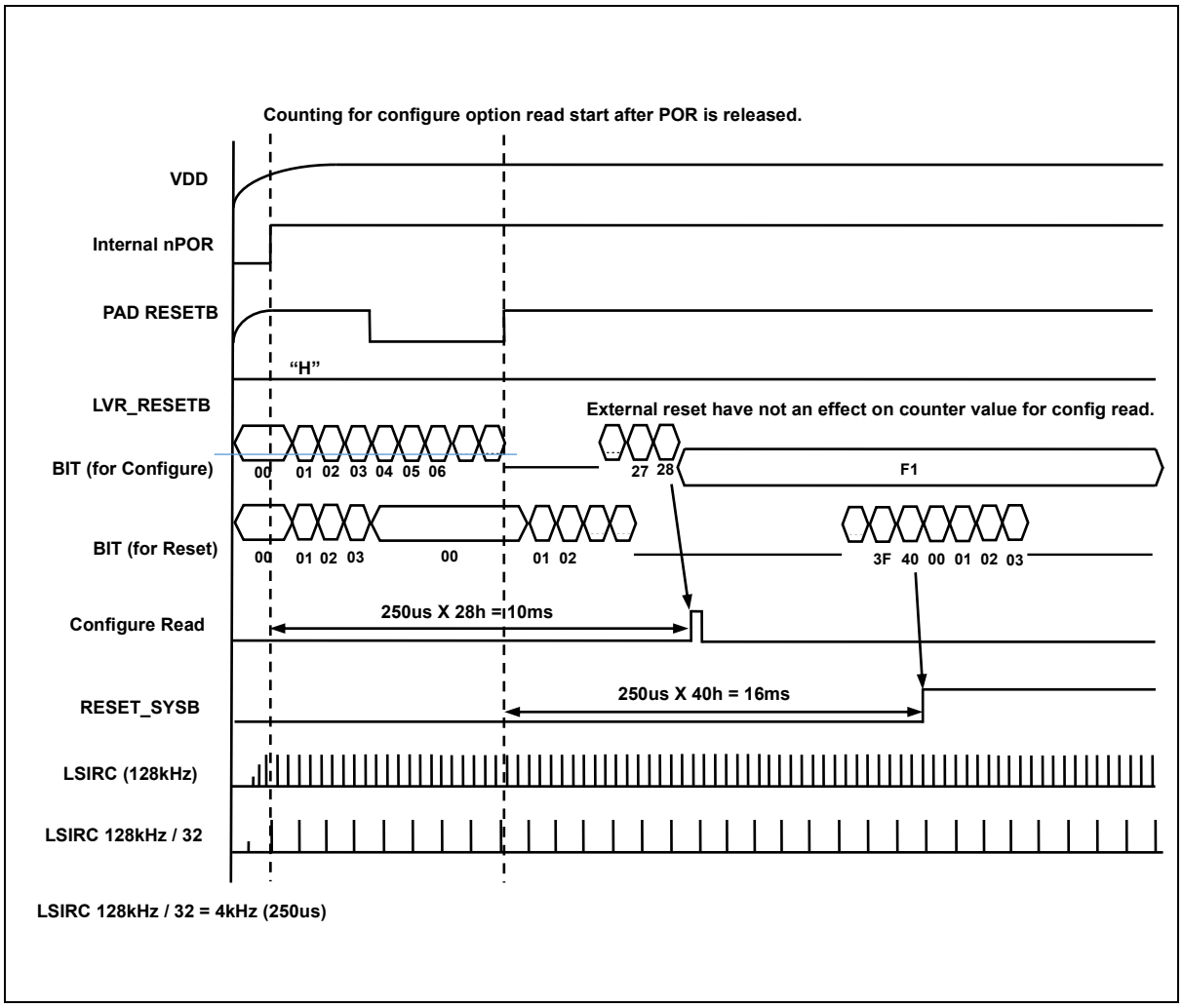


Figure 123. Configuration Timing when Power-on

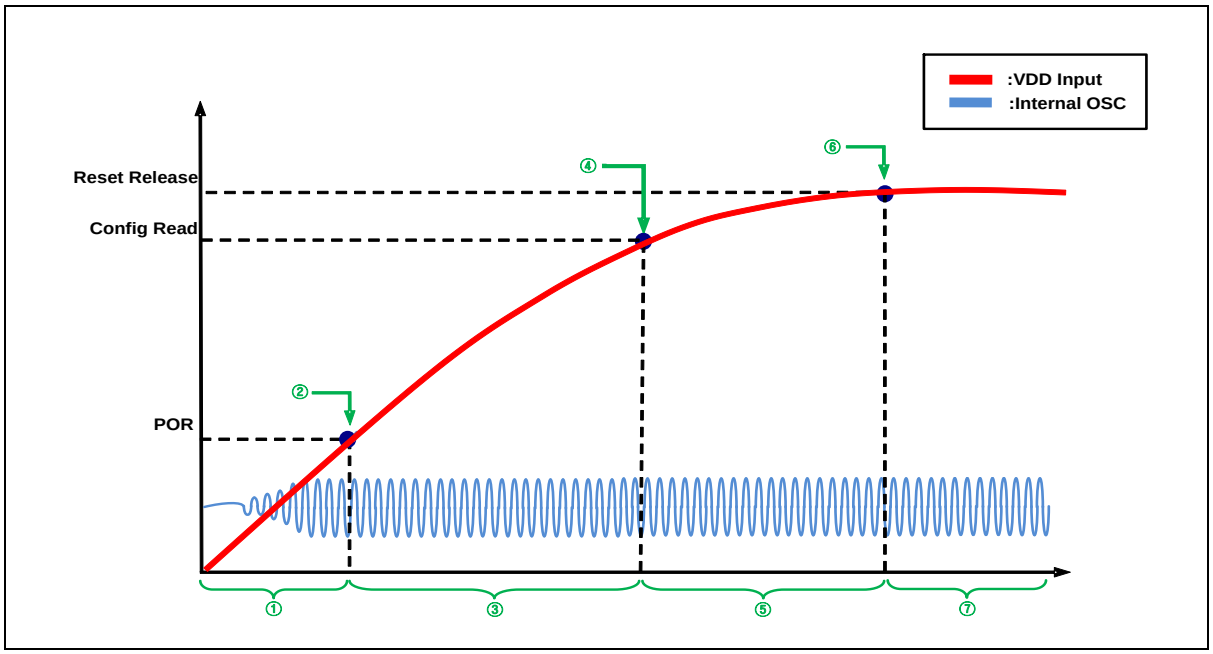


Figure 124. Boot Process Waveform

Table 45. 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) A. after16ms, after External Reset Release (External reset) B. 16ms point after POR (POR only)	BIT is used for Peripheral stability
⑦	Normal operation	

20.3 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.

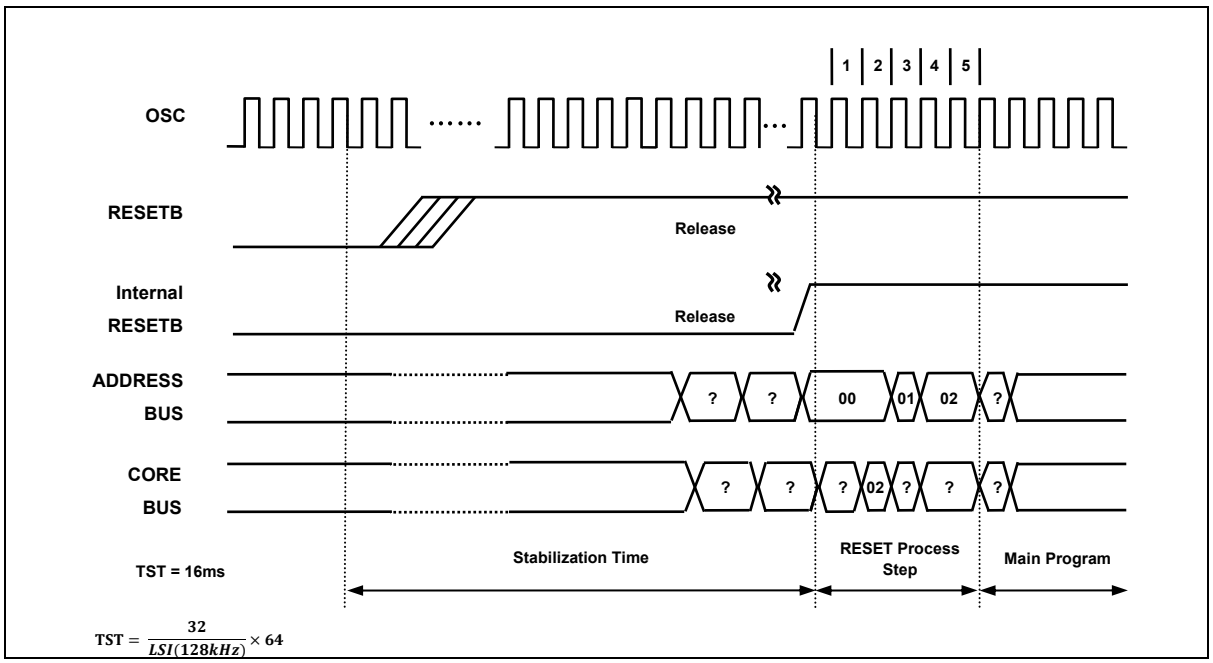


Figure 125. Timing Diagram after RESET

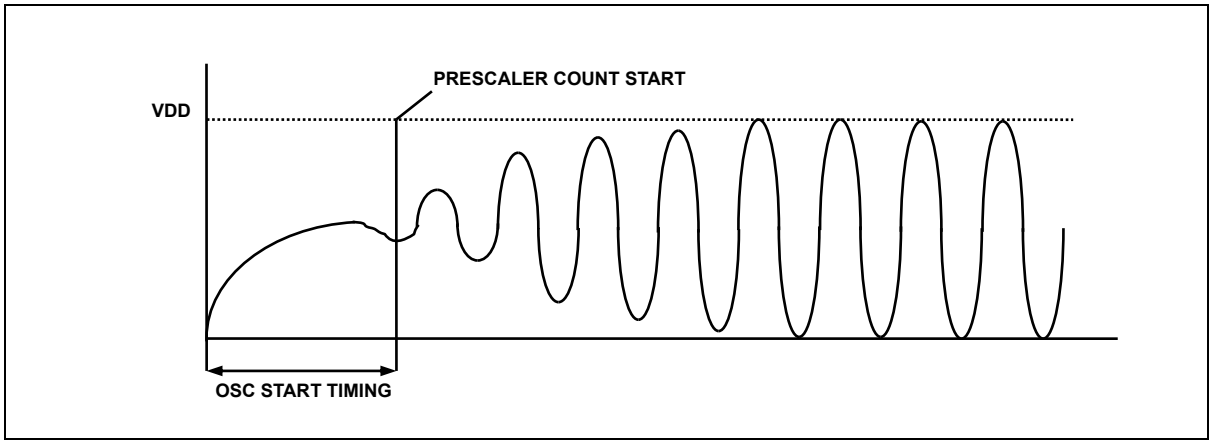


Figure 126. Oscillator Generating Waveform Example

As shown Figure 126, the stable generating time is not included in the start-up time. The resetb pin has a pull-up register by hardware

20.4 Low voltage reset process

A96G150 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[3:0] bits to be 1.61V, 1.68V, 1.77V, 1.88V, 2.00V, 2.13V, 2.28V, 2.46V, 2.68V, 2.81V, 3.06V, 3.21V, 3.56V, 3.73V, 3.91V, 4.25V.

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.

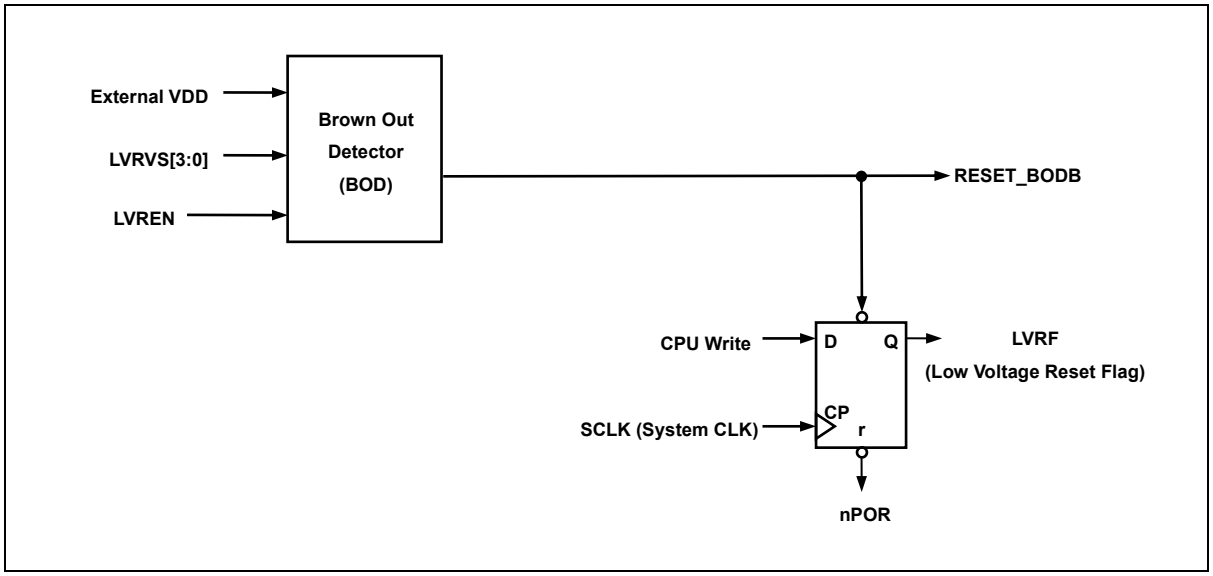


Figure 127. Block Diagram of LVR

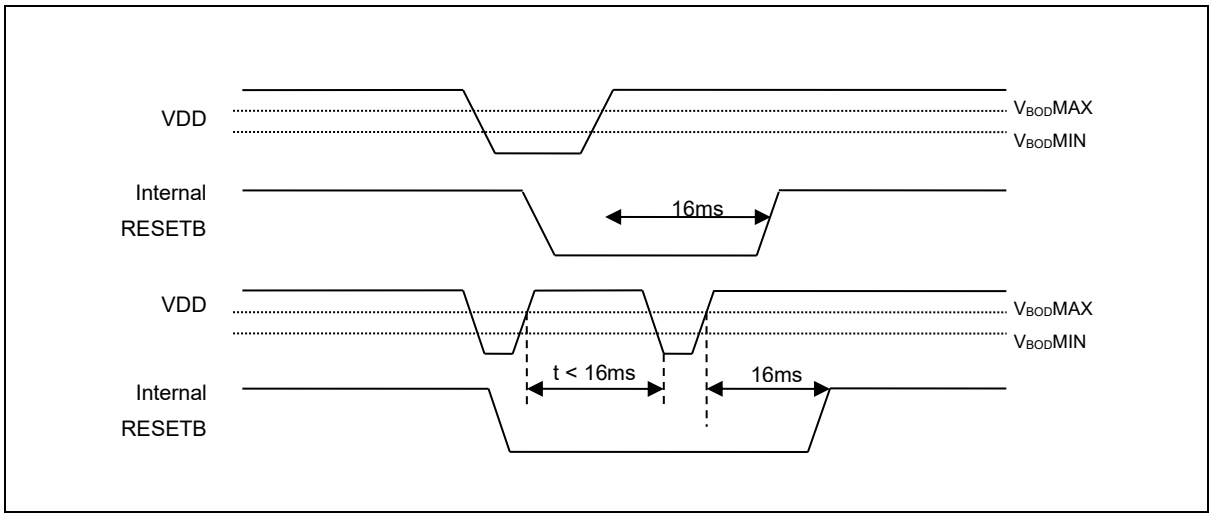


Figure 128. Internal Reset at Power Fail Situation

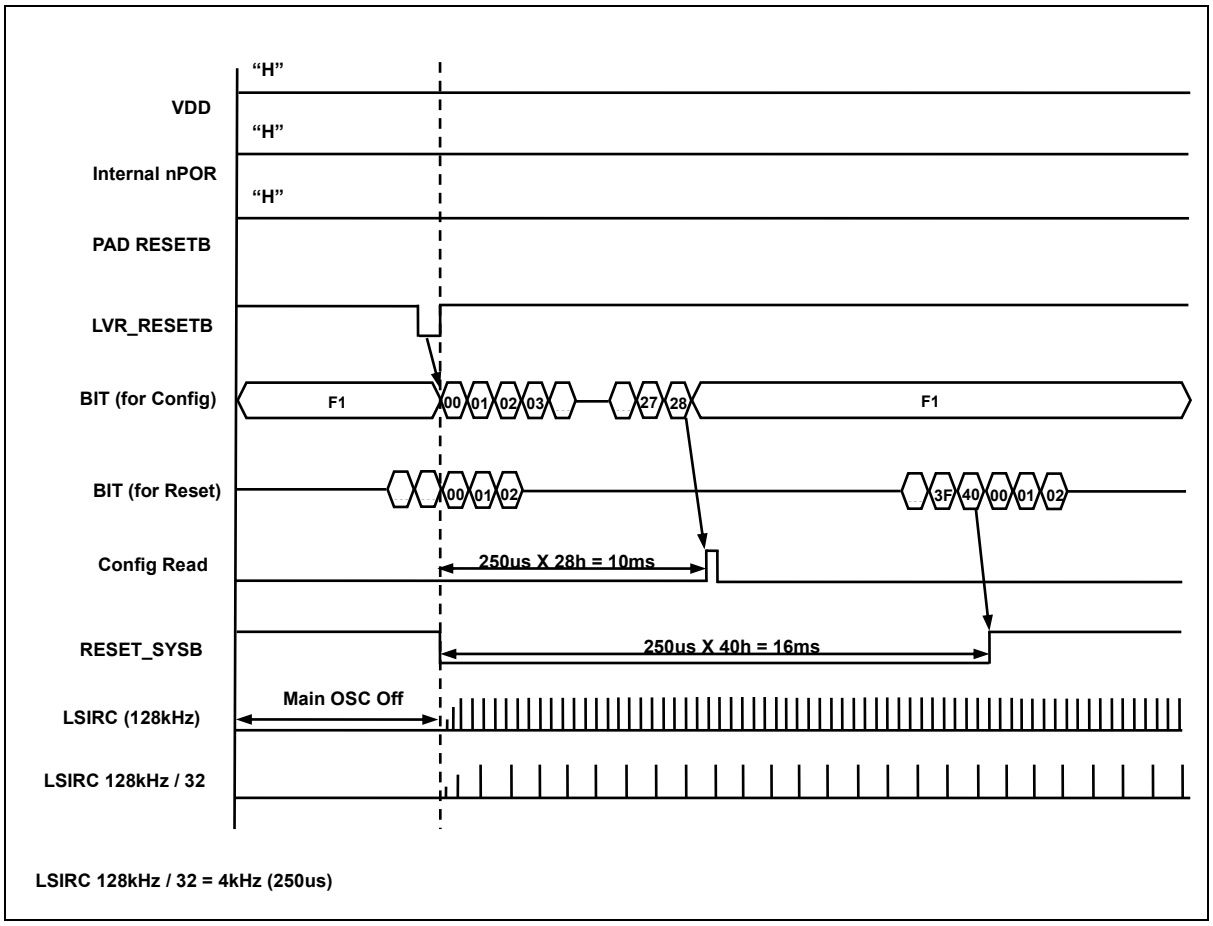


Figure 129. Configuration Timing When LVR RESET

20.5 LVI block diagram

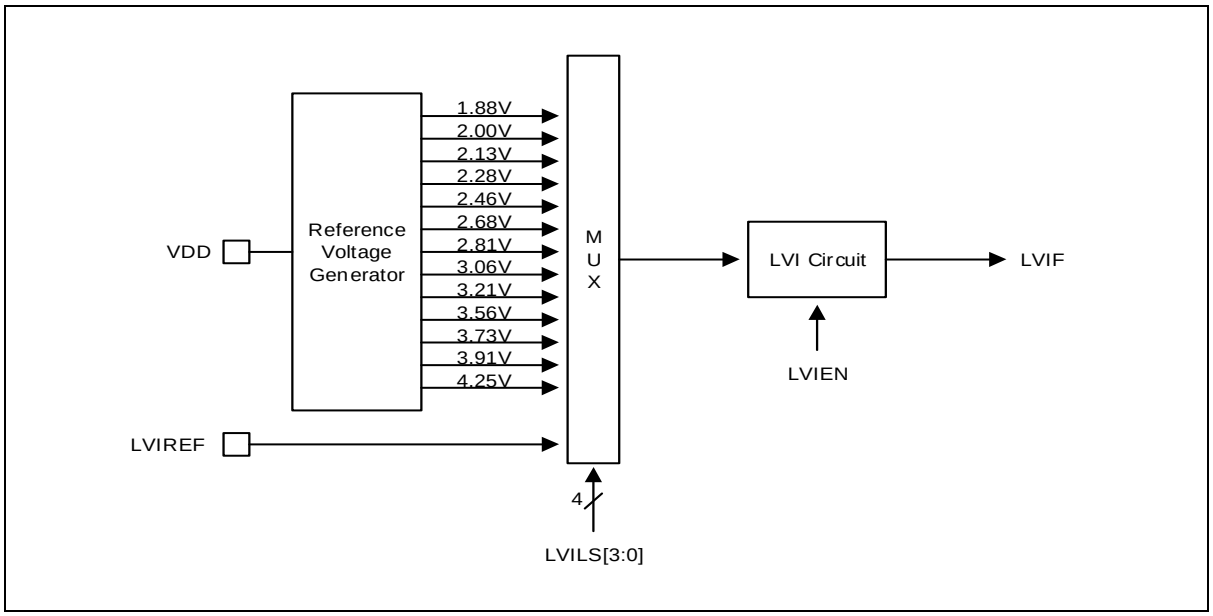


Figure 130. LVI Block Diagram

20.6 Register map

Table 46. 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

20.7 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
–	–	–	LVRVS3	LVRVS2	LVRVS1	LVRVS0	LVREN
–	–	–	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

LVRVS[3:0] LVR Voltage Select

LVRVS3	LVRVS2	LVRVS1	LVRVS0	Description
0	0	0	0	1.61V
0	0	0	1	1.68V
0	0	1	0	1.77V
0	0	1	1	1.88V
0	1	0	0	2.00V
0	1	0	1	2.13V
0	1	1	0	2.28V
0	1	1	1	2.46V
1	0	0	0	2.68V
1	0	0	1	2.81V
1	0	1	0	3.06V
1	0	1	1	3.21V
1	1	0	0	3.56V
1	1	0	1	3.73V
1	1	1	0	3.91V
1	1	1	1	4.25V

LVREN LVR Operation

0	LVR Enable
1	LVR Disable

NOTES:

1. The LVRVS[3:0] bits are cleared by a power-on reset but are retained by other reset signals.
2. The LVRVS[3:0] 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	LVILS3	LVILS2	LVILS1	LVILS0
–	–	R/W	R/W	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[3:0] LVI Level Select

LVIVS3	LVIVS2	LVIVS1	LVIVS0	Description
0	0	0	0	Not available
0	0	0	1	Not available
0	0	1	0	Not available
0	0	1	1	1.88V
0	1	0	0	2.00V
0	1	0	1	2.13V
0	1	1	0	2.28V
0	1	1	1	2.46V
1	0	0	0	2.68V
1	0	0	1	2.81V
1	0	1	0	3.06V
1	0	1	1	3.21V
1	1	0	0	3.56V
1	1	0	1	3.73V
1	1	1	0	3.91V
1	1	1	1	4.25V

21 Memory programming

A96G150 has flash and data EEPROM memory to which a program can be written, erased, and overwritten while mounted on the board. Serial ISP mode is supported.

Flash of A96G150 features the followings:

- Flash Size : 64Kbytes
- 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
- Up to 300,000 program/erase cycles at typical voltage and temperature for data EEPROM memory
- Security feature

21.1 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.

21.1.1 Register map

Table 47. Flash Control and Status Register Map

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

21.1.2 Register description

FEMR (Flash Mode Register): 1020H

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

Initial value: 00H

FSEL	Select flash memory.	
	0	Deselect flash memory
	1	Select flash memory
ESEL	Select data EEPROM.	
	0	Deselect data EEPROM.
	1	Select data EEPROM.
PGM	Enable program or program verify mode with VFY	
	0	Disable program or program verify mode
	1	Enable program or program verify mode
ERASE	Enable erase or erase verify mode with VFY	
	0	Disable erase or erase verify mode
	1	Enable erase or erase verify mode
PBUFF	Select page buffer	
	0	Deselect page buffer
	1	Select page buffer
OTPE	Select OTP area instead of program memory	
	0	Deselect OTP area
	1	Select OTP area
VFY	Set program or erase verify mode with PGM or ERASE	
	Program Verify: PGM=1, VFY=1	
	Erase Verify: ERASE=1, VFY=1	
FEEN	Enable program and erase of Flash. When inactive, it is possible to read as normal mode	
	0	Disable program and erase
	1	Enable program and erase

FECR (Flash Control Register): 1021H

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

Initial value: 03H

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

AEE — Enable data EEPROM bulk erase mode
- 0 Disable bulk erase mode of data EEPROM
- 1 Enable bulk erase mode of data EEPROM

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

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

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

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

nFERST — Reset Flash control logic. It is set automatically after 1 clock
- 0 Reset Flash control logic
- 1 No operation (default)

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

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

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	Test Only.
ROMINT	Flash and data EEPROM 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. FEARs 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.

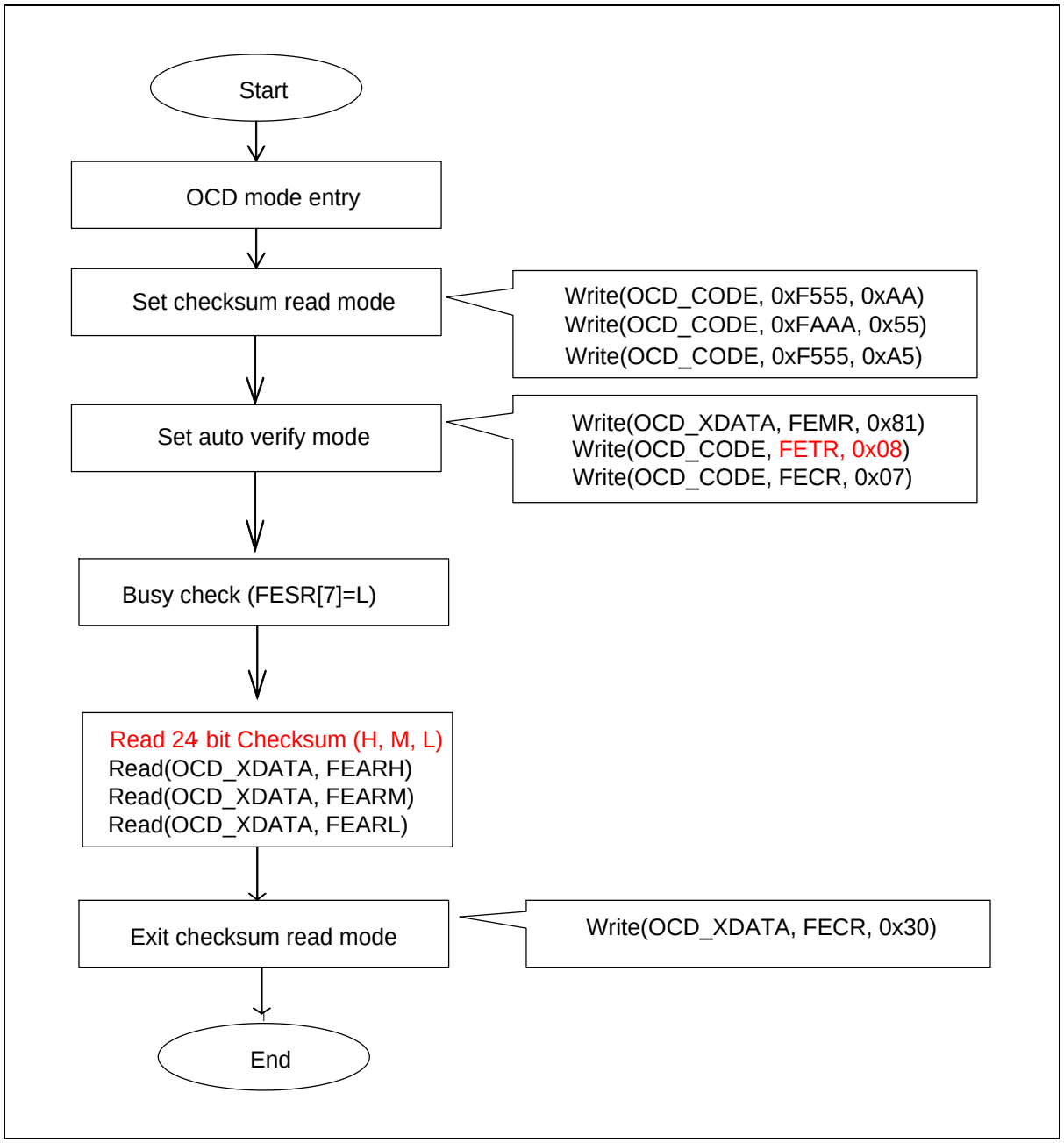


Figure 131. Read Device Internal Checksum (Full Size)

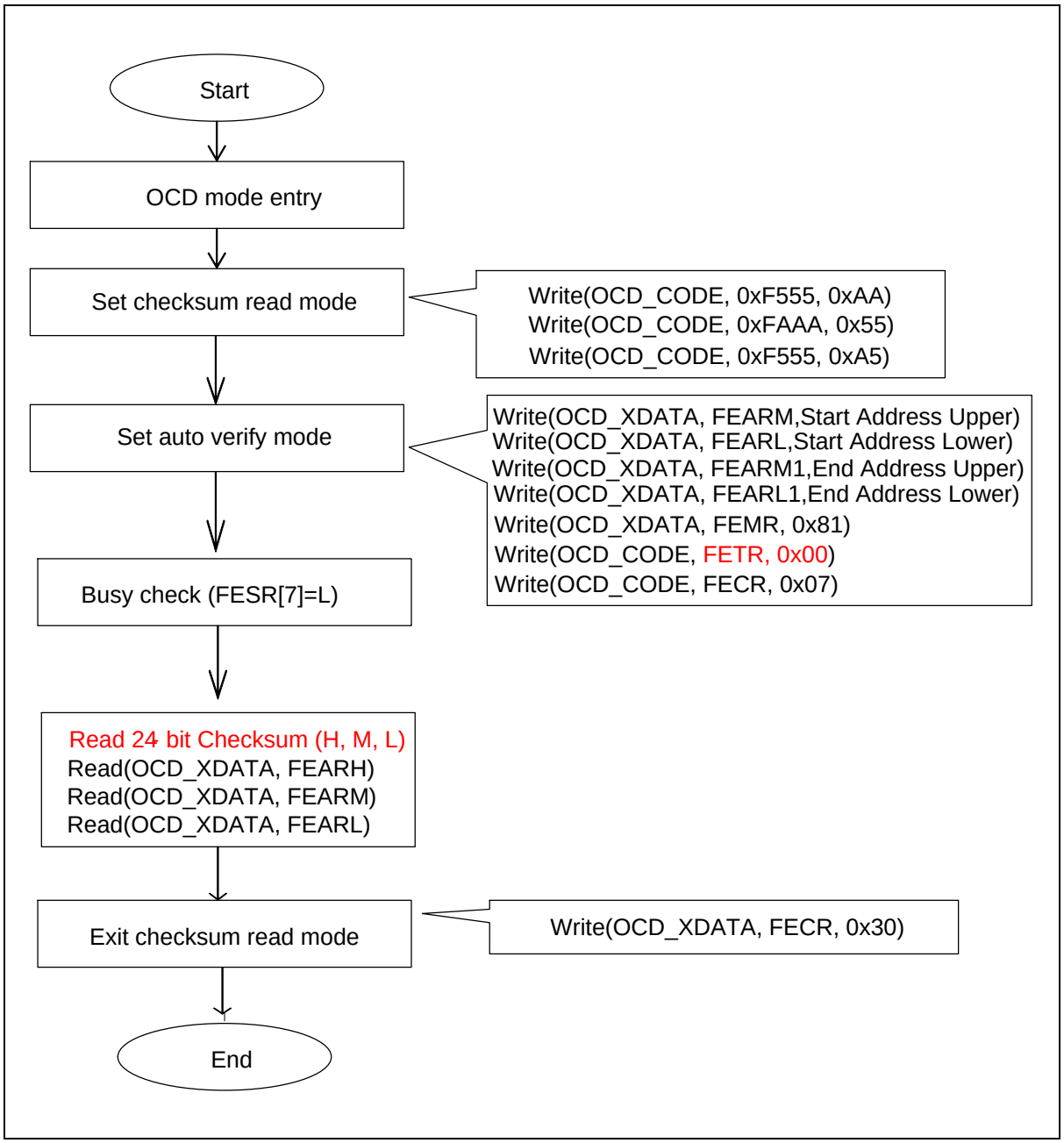


Figure 132. Read Device Internal Checksum (User Define Size)

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 48. Program and Erase Time

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

21.2 Memory map

21.2.1 Flash memory map

Program memory uses 64K bytes of flash memory. It is read by byte and written by byte or page. One page is 64-bytes

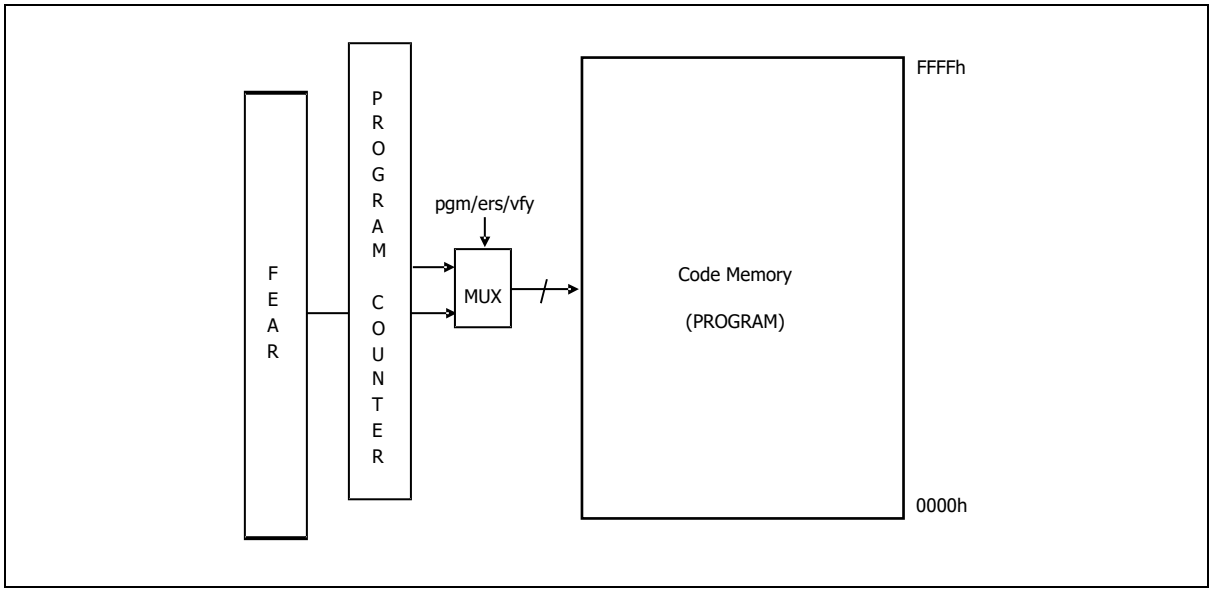


Figure 133. Flash Memory Map

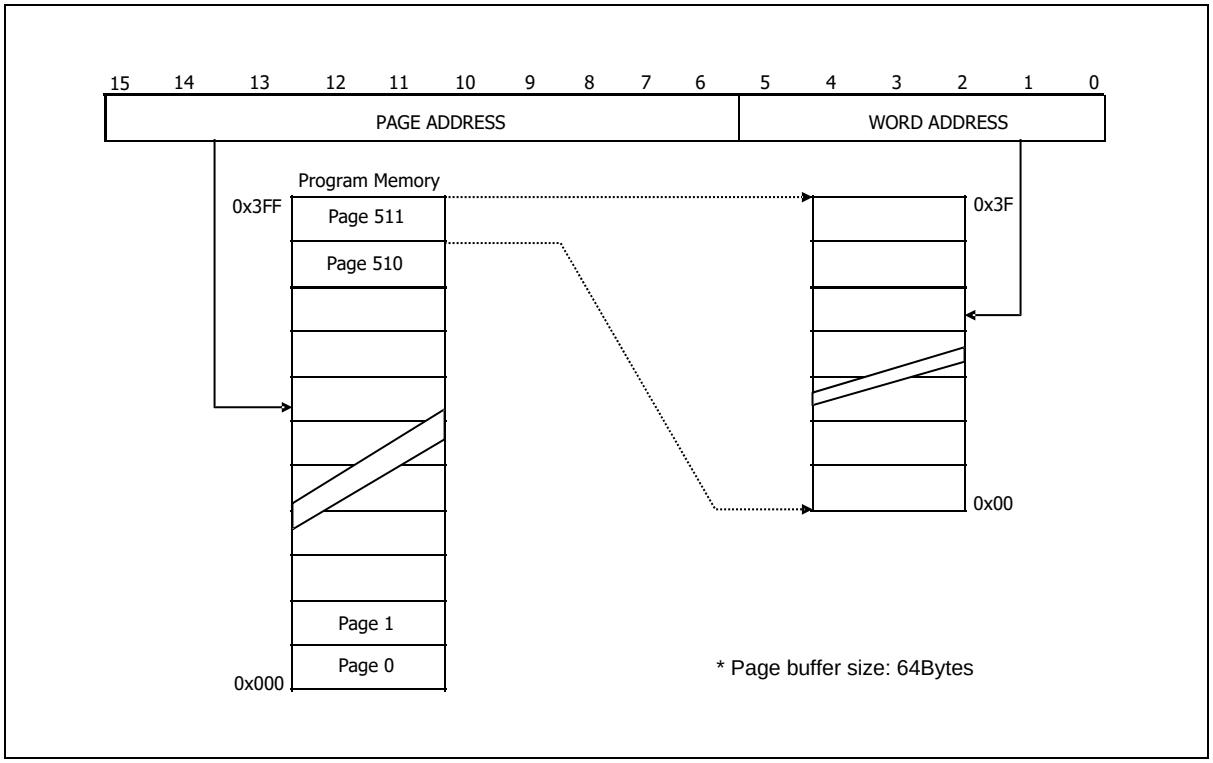


Figure 134. Address Configuration of Flash Memory

21.2.2 Data EEPROM memory map

Data EEPROM memory uses 2K bytes of flash memory. It is read by byte and written by byte or page. One page is 32-bytes

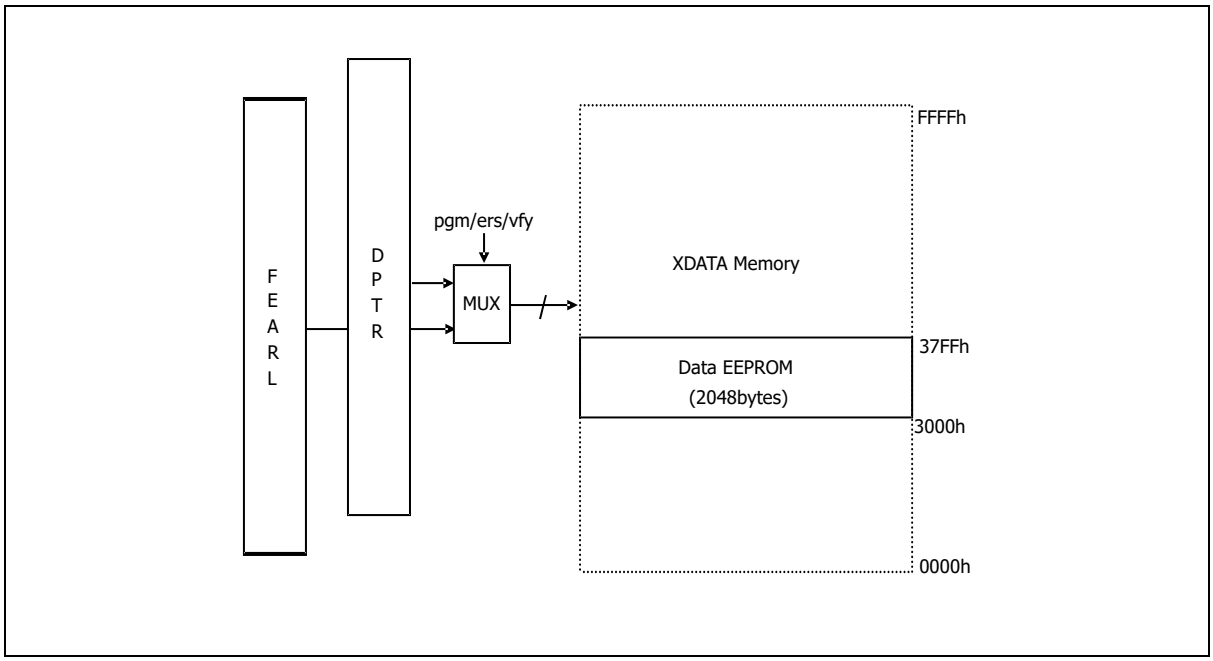


Figure 135. Data EEPROM Memory Map

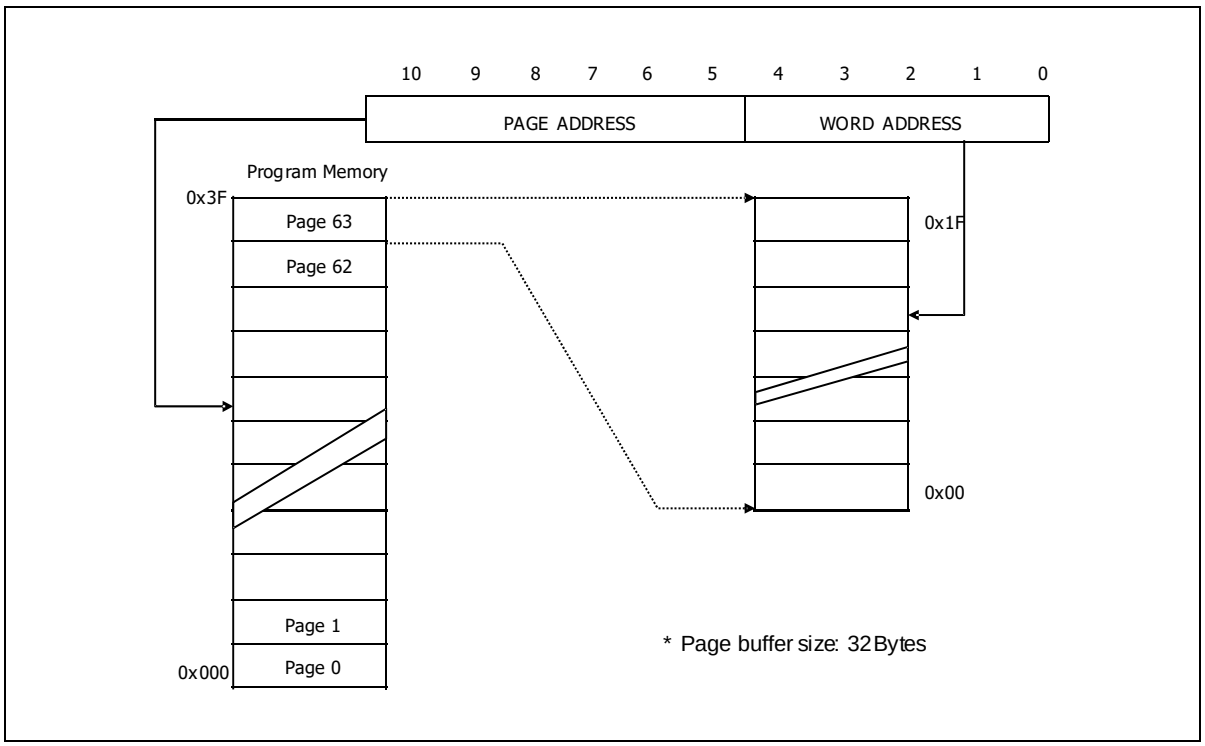


Figure 136. Address Configuration of Data EEPROM Memory

21.3 Serial In-system Program (ISP) mode

Serial In-system Program (ISP) uses the interface of debugger which uses two wires. Refer to Chapter 22.Development tools in details about debugger.

21.3.1 Flash operation

Configuration (This Configuration is just used for following 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

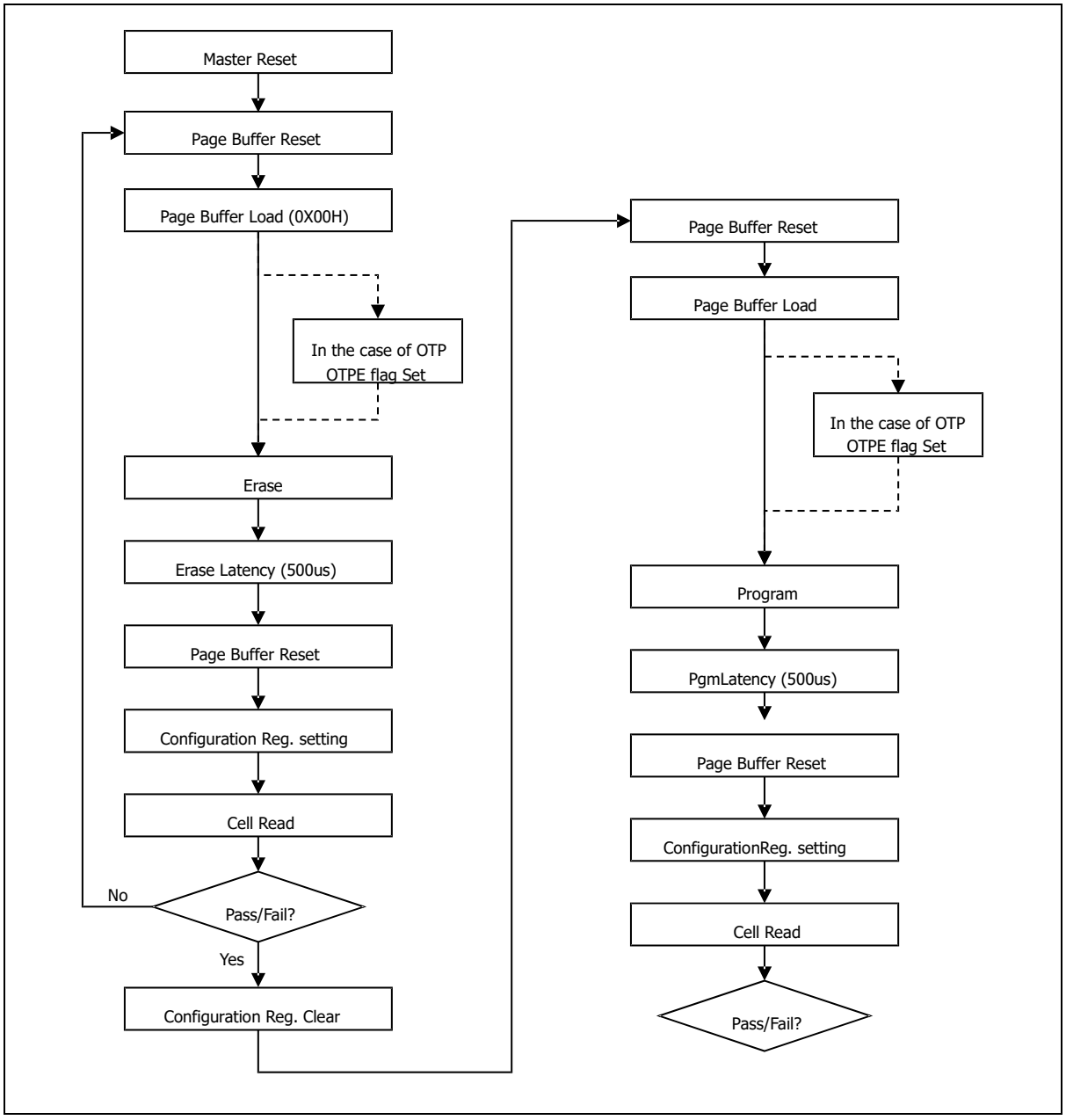


Figure 137. The Sequence of Page Program and Erase of Flash Memory

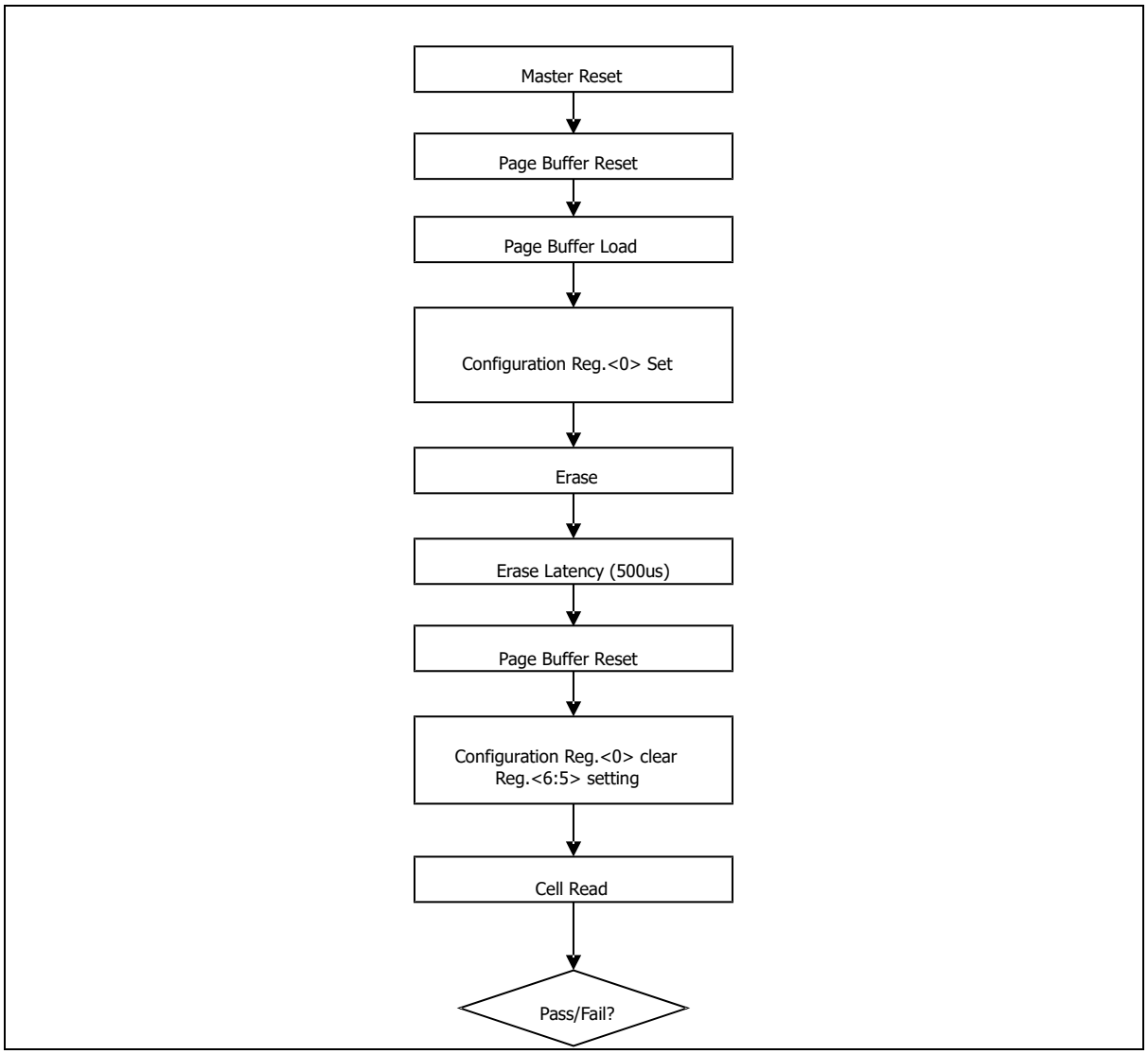


Figure 138. The Sequence of Bulk Erase of Flash Memory

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 49. 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.

21.3.2 Data EEPROM operation

Program and erase operation of Data EEPROM are executed by direct and indirect address mode.

Direct address mode uses external data area of 8051. Indirect address mode uses address register of SFR area.

Data EEPROM Read

① Enter OCD(=ISP) mode.

② Set ENBDM bit of BCR.

③ Enable debug and Request debug mode.

④ Read data from Data EEPROM.

Enable program mode

① Enter OCD(=ISP) mode.[1]

② Set ENBDM bit of BCR.

③ Enable debug and Request debug mode.

④ Enter program/erase mode sequence.[2]

A. Write 0xA5 to FEDR.

B. Write 0x5A to FEDR.

NOTES:

1. Refer to how to enter ISP mode..
2. Command sequence to activate data EEPROM write/erase mode. It is composed of sequentially writing to data register (FEDR).

EEPROM write mode

① Enable program mode.

② Reset page buffer. FEMR: 0100_0001 FECR:0000_0010

③ Select page buffer. FEMR:0100_1001

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

⑤ Set write mode. FEMR:0110_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 step2 to step 8 until all pages are written.

EEPROM page erase mode

① Enable program mode.

② Reset page buffer. FEMR: 0100_0001 FECR:0000_0010

③ Select page buffer. FEMR:0100_1001

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

⑤ Set erase mode. FEMR:0101_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 step2 to step 8 until all pages are erased.

EEPROM bulk erase mode

① Enable program mode.

② Reset page buffer. FEMR: 0100_0001 FECR:0000_0010

③ Select page buffer. FEMR:0100_1001

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

⑤ Set erase mode. FEMR:0101_0001.

⑥ Set FETCR

⑦ Start bulk erase. FECR:0100_1011

⑧ Insert one NOP operation

⑨ Read FESR until PEVBSY is 1.

Data EEPROM program verify mode

① Enable program mode.

② Set program verify mode. FEMR:0110_0011

③ Read data from Flash.

Data EEPROM erase verify mode

① Enable program mode.

② Set erase verify mode. FEMR:0101_0011

③ Read data from Flash.

Data EEPROM page buffer read

① Enable program mode.

② Select page buffer. FEMR:0100_1001

③ Read data from Flash.

21.4 Mode entrance method of ISP mode

21.4.1 Mode entrance method for ISP

Table 50. Mode Entrance Method for ISP

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

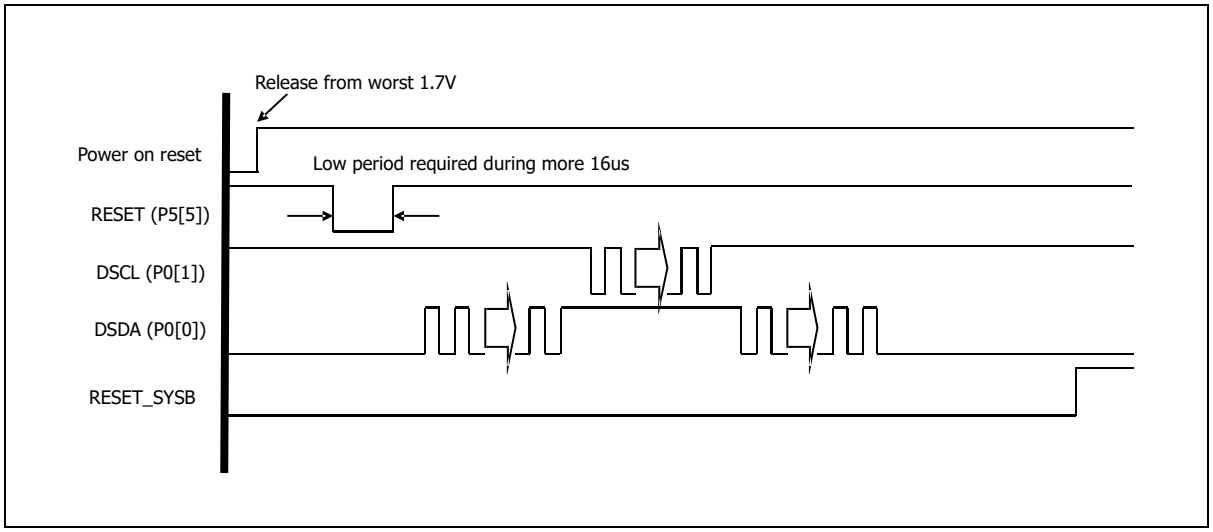


Figure 139. ISP Mode

21.5 Security

A96G150 provides Lock bits which can be left un-programmed (“0”) or can be programmed (“1”) to obtain the additional features listed in Table 51. Security Policy using Lock Bits. The Lock bit can only be erased to “0” with the bulk erase command and a value of more than 0x40 at FETCR.

Table 51. Security Policy using Lock Bits

LOCK mode		USER mode												ISP mode											
		FLASH				Data EEPROM				OTP				FLASH				Data EEPROM				OTP			
DR_P	CR_P	R	W	PE	BE	R	W	PE	BE	R	W	PE	BE	R	W	PE	BE	R	W	PE	BE	R	W	PE	BE
0	0	O	O	O	X	O	O	O	O	X	X	X	X	O	O	O	O	O	O	O	O	O	O	O	O
0	1	O	O	O	X	O	O	O	O	X	X	X	X	X	X	X	O	O	O	O	O	O	X	X	O
1	0	O	O	O	X	O	O	O	O	X	X	X	X	O	◇	◇	◇	X	X	X	O	O	◇	◇	◇
1	1	O	O	O	X	O	O	O	O	X	X	X	X	X	X	X	◇	X	X	X	O	O	X	X	◇

NOTES:

1. CR_P : Lock bit of Flash memory
2. DR_P : Lock bit of data EEPROM
3. R: Read
4. W: Write
5. PE: Page erase
6. BE: Bulk Erase
7. O: Operation is possible.
8. X: Operation is impossible.
9. ◇: When LOCKE is programmed, each operation can be done after data EEPROM is erased with the bulk erase command.

21.6 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
CR_P	HL	DR_P	VAPEN	–	–	–	RSTS

Initial value: 00H

CR_P	Code Read Protection	
	0	Disable
	1	Enable
HL	Code Write Protection	
	0	Disable
	1	Enable
DR_P	Data Read 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(P52)
	1	Enable RESETB pin

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

CONFIGURE OPTION 1: ROM Address 0000H (A96G150 64K 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	0.7Kbytes (Address 0100H – 03FFH)
0	0	1	1.7Kbytes (Address 0100H – 07FFH)
0	1	0	2.7Kbytes (Address 0100H – 0BFFH)
0	1	1	3.7Kbytes (Address 0100H – 0FFFH)
1	0	0	61.7Kbytes (Address 0100H – F7FFH)
1	0	1	62.7Kbytes (Address 0100H – FBFFH)
1	1	0	63.2Kbytes (Address 0100H – FDFFH)
1	1	1	63.5Kbytes (Address 0100H – FEFFH)

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

CONFIGURE OPTION 1: ROM Address 0000H (A96G158 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	0.7Kbytes (Address 0100H – 03FFH)
0	0	1	1.7Kbytes (Address 0100H – 07FFH)
0	1	0	2.7Kbytes (Address 0100H – 0BFFH)
0	1	1	3.7Kbytes (Address 0100H – 0FFFH)
1	0	0	29.7Kbytes (Address 0100H – 77FFH)
1	0	1	30.7Kbytes (Address 0100H – 7BFFH)
1	1	0	31.2Kbytes (Address 0100H – 7DFFH)
1	1	1	31.5Kbytes (Address 0100H – 7EFFH)

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

22 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 development ecosystem of the customers.

22.1 Compiler

ABOV semiconductor does not provide any compiler for A96G150. It is recommended to consult a compiler provider.

Since A96G150 has Mentor 8051 as its core, and ROM is smaller than 64Kbytes in size, a developer can use any standard 8051 compilers of other providers.

22.2 Core and debug tool information

ABOV's microcontroller uses OCD (On-Chip Debugger) interface for debugging, which is ABOV's own interface. The OCD monitors and controls the core and supports the read and write operations of external memory and devices. In addition, it supports memory monitoring and break functions.

Debug interfaces such as an OCD interface enable a microcontroller to write on internal programmable memory, thereby, allow it to support ISP (In-System Program) which making possible to write as a single chip or as an embedded chip in the system.

Table 52 provides information of the core and debug emulation interface.

Table 52. Core and Debug Information

	Value	Description
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. A96G150 has 96 series core and OCD 1 interface.
2. Also, A96G150 can be operated with OCD II dongle because OCD II dongle includes all of OCD1 function.
3. 95 series core is the old version of 96 series core.

22.2.1 Feature of 94/96/97 Series core

ABOV's microcontroller contains an M8051/CM8051 core that was made by improving the 8051. The M8051/ CM8051 core is compatible with the 8051. It reduces time of operation cycles and makes development easy by providing the OCD debug function.

ABOV's 8-bit microcontroller has a core of 94-series, 96-series, or 97-series, that is basically compatible with the 8051 series at the instruction set level. A core of each series uses different Debug Interface respectively as shown in Table 53.

Table 53. 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 54.

Table 54. 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		
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

NOTES:

1. A96G150 has 96 series core and OCD 1 interface.
2. Also, A96G150 can be operated with OCD II dongle because OCD II dongle includes all of OCD1 function.
3. EA means that All Interrupt Enable bit or Disable bit (Standard 8051).
4. Group: When a programmer selects a specific interrupt (e.g. Interrupt1), the whole interrupts: 0, 6, 12, and 18 have higher priorities.

ABOV's 8-bit microcontrollers maintain the compatibility of binary levels with 8051 cores; however, the cores and series have differences in performances, functions of the core, and debug interfaces.

Through the following sections, you can see the differences of each series.

22.2.2 OCD type of 94/96/97 Series core

96-series core uses OCD 1 for a debug interface, while the cores of 94-series and 97-series use OCD 2 for debug interfaces. The OCD 1 and OCD 2 use the same method in Hardware, but protocols are not compatible each other.

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

Table 55. OCD Type of Each Series

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

In Table 56, debug interfaces of the OCD 1 and OCD 2 are compared.

Table 56. 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

This series supports basic operations such as Run, Stop, Step, Break point, register reading/ writing, Memory reading/ writing, and SFR reading/ writing.

94 Series and 97 Series – OCD 2

This series supports the features of the OCD 1 and the features listed below (however, the protocol is not compatible 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.

22.2.3 Interrupt priority of 94/96/97 Series core

In the M8051, users can set interrupt priorities by group. Thus, 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 can set interrupt priorities to have more functions than the existing functions, and set individual priority over each interrupt source.

Table 57. 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 priorities by group is available only with IP/IP1 settings.
  - With IP/IP1 settings, users can set interrupt priorities by group unit.
  - The interrupt priority of a group unit (4 interrupts in a group) can be changed to the level between 0 and 3 according to the value of IP/IP1.

94, 97 Series

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

22.2.4 Extended Stack Pointer of 94/96/97 Series core

M8051 uses IRAM area for Stack Pointer. However, 94-series and 97-series microcontrollers use not only IRAM area but also XRAM area for Stack Pointer by configuring additional registers.

The setting procedure is as follows:

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

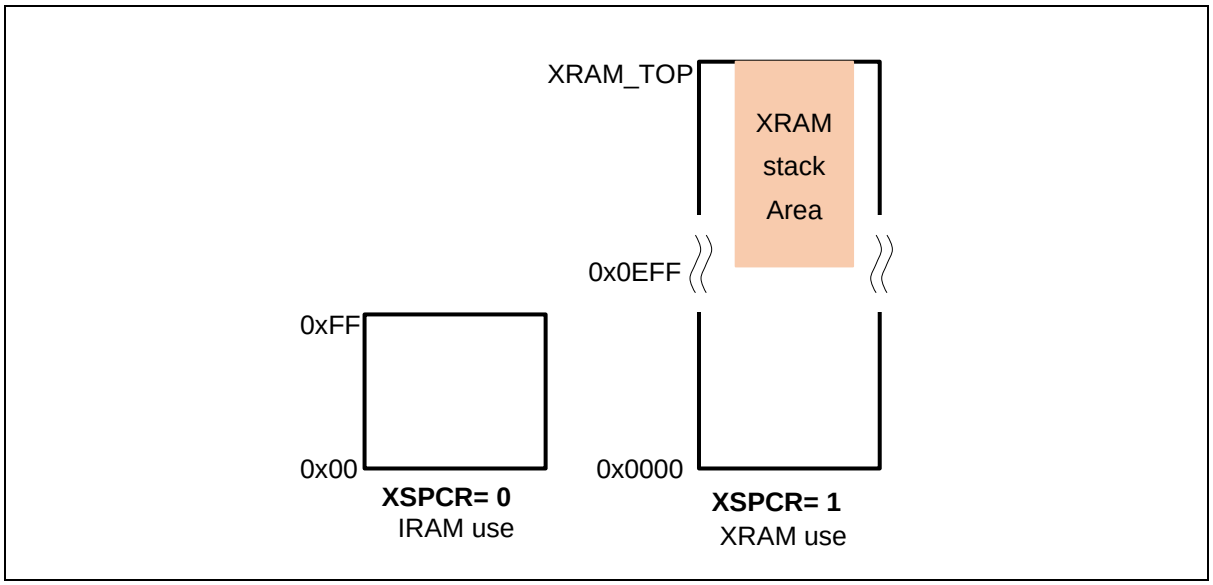


Figure 140. Configuration of the Extended Stack Pointer

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

22.3 OCD (On-chip debugger) emulator and debugger

Microcontroller series that uses an 8051 core has 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. DSCL is for a clock signal and DSDA is for a 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 58. Debug Feature by Series

	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 94-series and 97-series provides the RTM (Real Time Monitoring) function that enables to monitor 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 is the functions that have been extended from the OCD 2:

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

Figure 141 shows the standard 10-pin connector of OCD 1 and OCD 2.

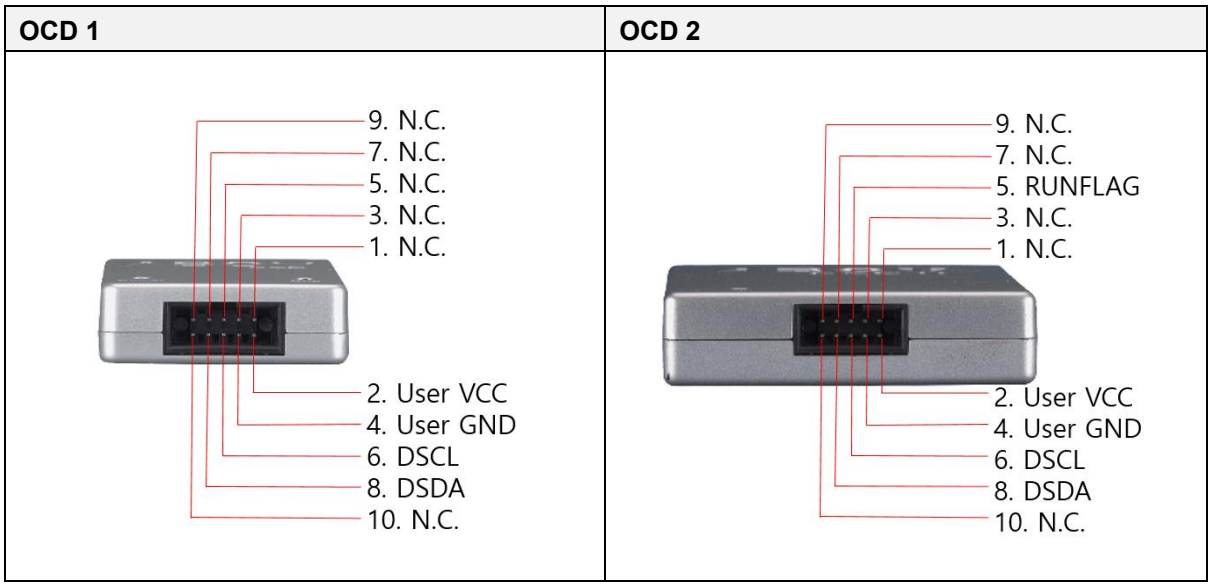


Figure 141. OCD 1 and OCD 2 Connector Pin Diagram

Table 59 introduces pins used for the OCD 1 and OCD 2.

Table 59. OCD 1 and OCD 2 Pin Description

	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 interfacing 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 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 a user wants to see more details, it is recommended to refer to OCD debugger manual by visiting ABOV's website (http://www.abovsemi.com) and downloading debugger S/W and corresponding manuals.

Connection:

- DSCL (A96G150 P13 port)
- DSDA (A96G150 P11 port)

Figure 142 shows pinouts of OCD connector.

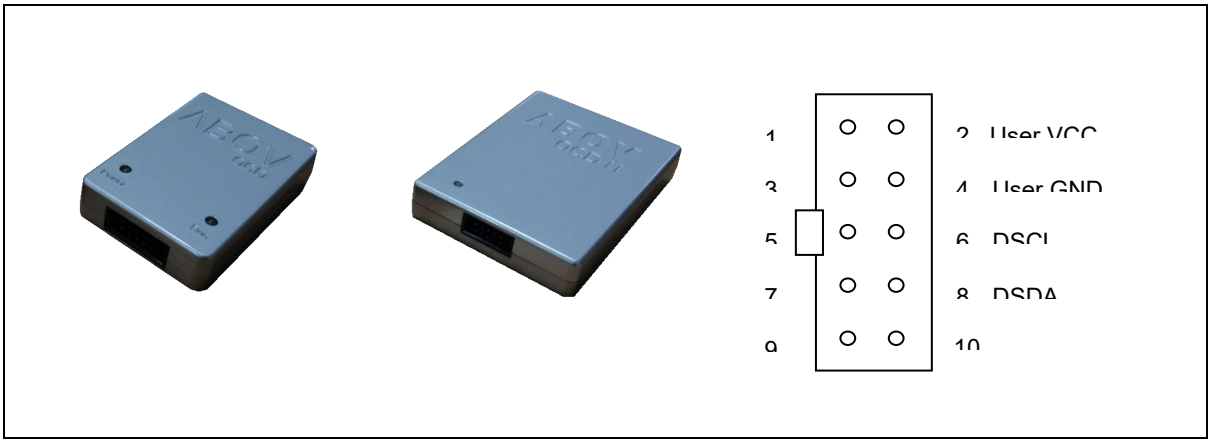


Figure 142. Debugger (OCD1/OCD2) and Pinouts

22.3.1 On-chip debug system

A96G150 supports On-chip debug (OCD) system. We recommend developing and debugging program with A96G1 series. On-chip debug system of A96G150 can be used for programming the non-volatile memories and on-chip debugging. Detail descriptions for programming via the OCD interface can be found in this section.

Table 60 introduces features of OCD.

Table 60. 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 143 shows a block diagram of the OCD interface and the On-chip Debug system.

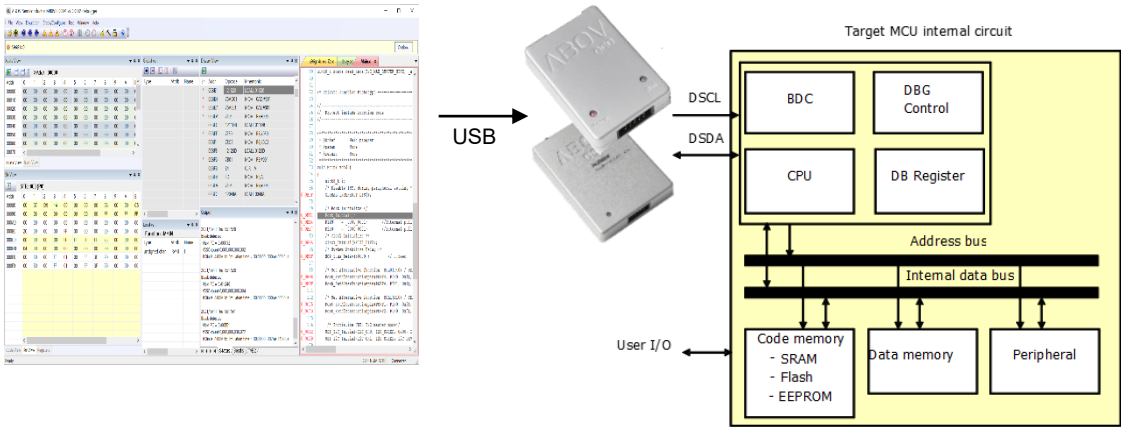


Figure 143. On-chip Debugging System in Block Diagram

22.3.2 Entering debug mode

While communicating through the OCD, users can enter the microcontroller into DEBUG mode by applying power to it. This means, the microcontroller enters DEBUG mode by placing specific signals to DSCL and DSDA at the moment of initialization when powering on the microcontroller. To do this, users should be able to control the power of the microcontroller (VCC or VDD) and need to be careful to place a capacitive load such as a large capacity condenser 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.

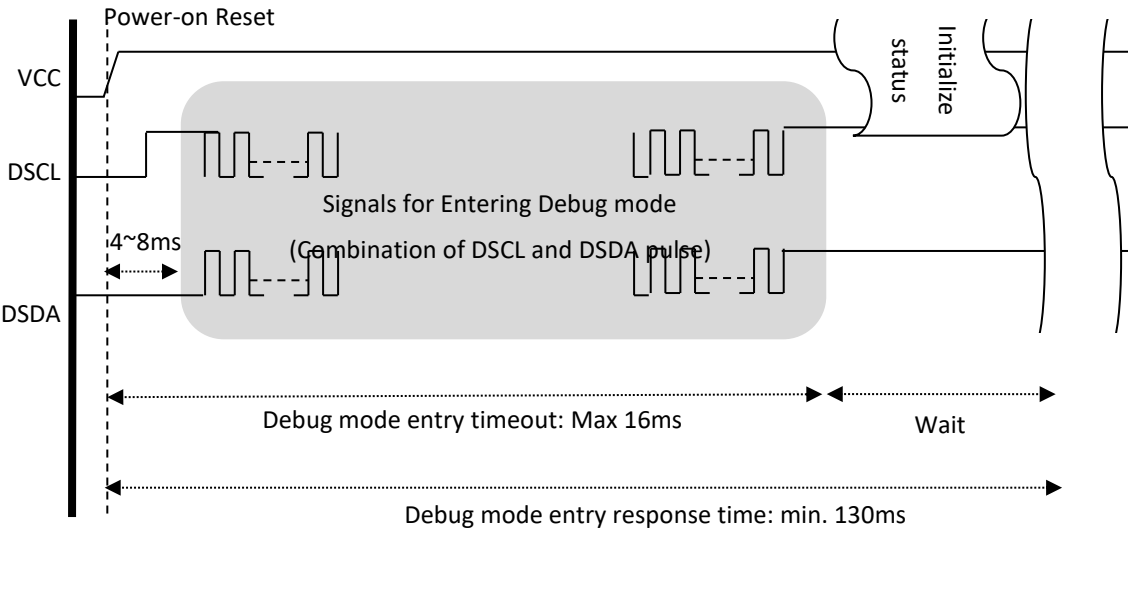


Figure 144. Timing Diagram of Debug Mode Entry

22.3.3 Two-wire communication protocol

For the OCD interface, the semi-duplex communication protocol is used through separate two wires such as DSCL and DSDA. The DSCL is a serial clock signal and the DSDA is a 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. In addition, a communication packet includes a start bit and an end bit to indicate a start and an 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 command which has error again.
- Background debugger command is composed of a bundle of packet.
- Start condition and stop condition notify the start and the stop of background debugger command respectively.

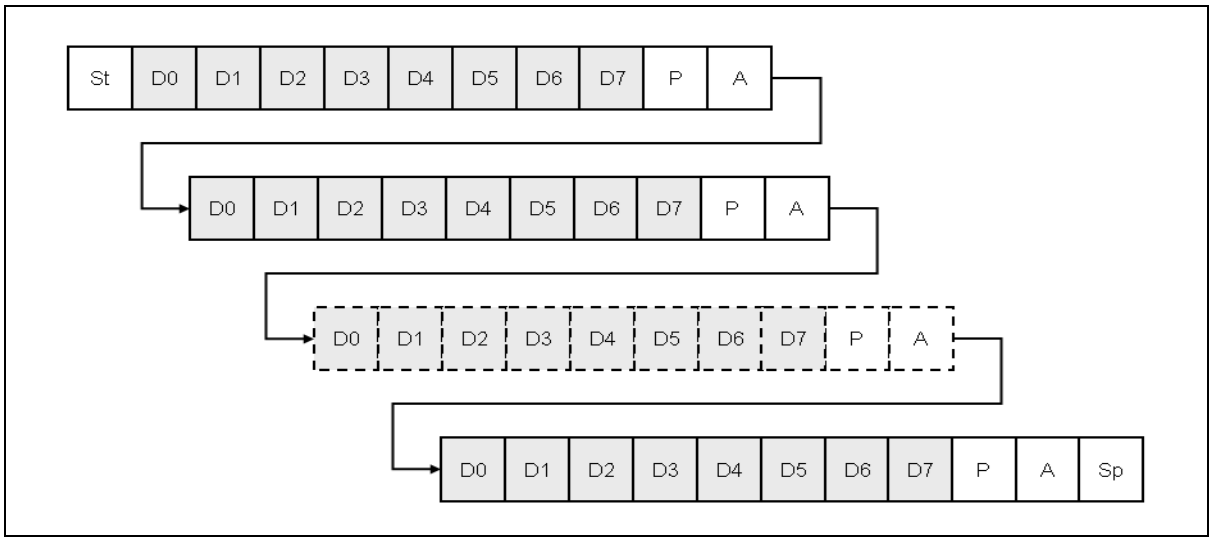


Figure 145. 10-bit Transmission Packet

Packet transmission timing

Figure 146 shows a timing diagram of a packet transmission using the OCD communication protocol. The start bit means a start of a packet and is valid when DSDA falls from 'H' to 'L' while External Host maintains DSCL to 'H'. After this, communication data is transferred and received between a Host and a microcontroller. The end bit means an end of the data transmission and is valid when DSDA changes from 'L' to 'H' while a Debugger maintains DSCL to 'H'. Next, the microcontroller places the bus in a wait state and processes the received data.

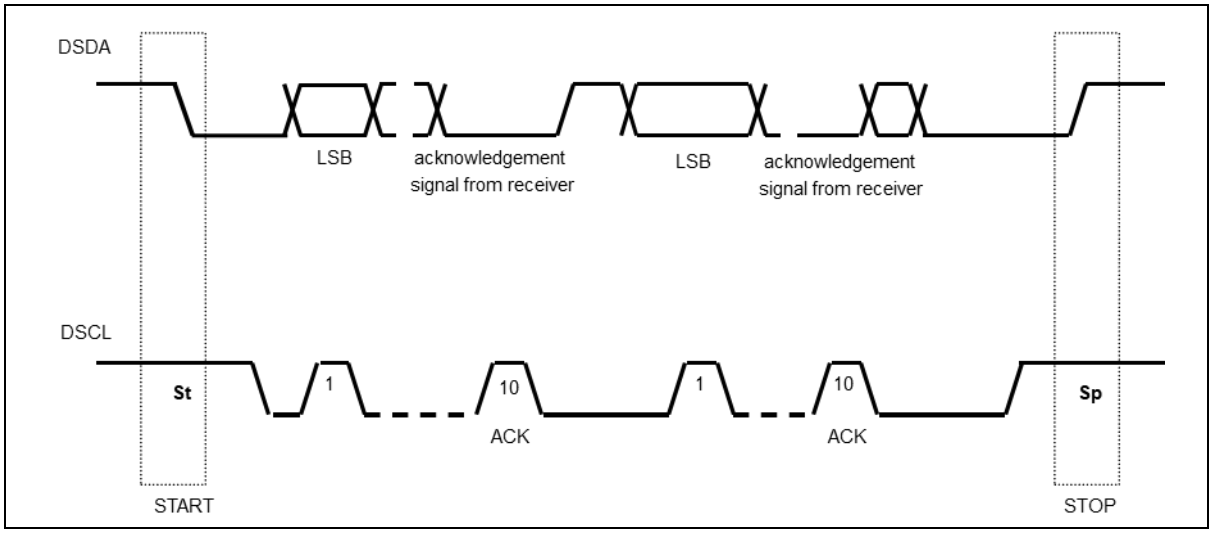


Figure 146. Data Transfer on OCD

Figure 147 shows a timing diagram of each bit based on the state of DSCL clock and DSDA data. Similar to I2C signal, DSDA data is allowed to change when DSCL is ‘L’. If the data changes when DSCL is ‘H’, the change means ‘START’ or ‘STOP’.

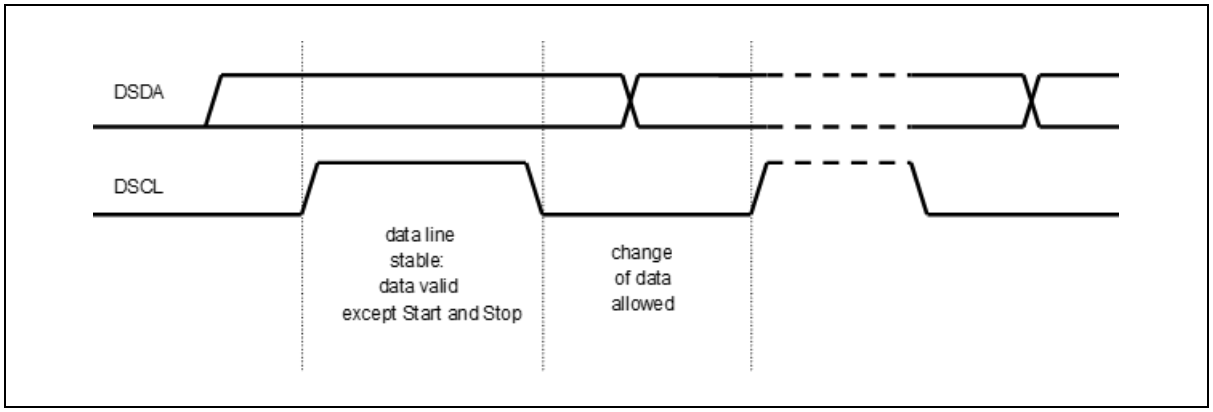


Figure 147. Bit Transfer on Serial Bus

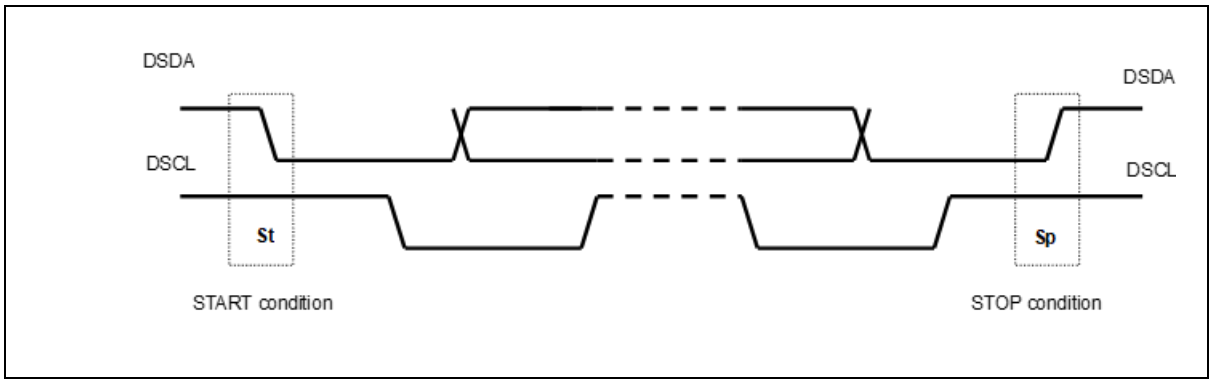


Figure 148. Start and Stop Condition

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 the reception.

As shown in Figure 149, when transferring data, a receiver outputs DSDA to ‘L’ to inform the normal reception of data. If a receiver outputs DSDA to ‘H’, it means error reception of data.

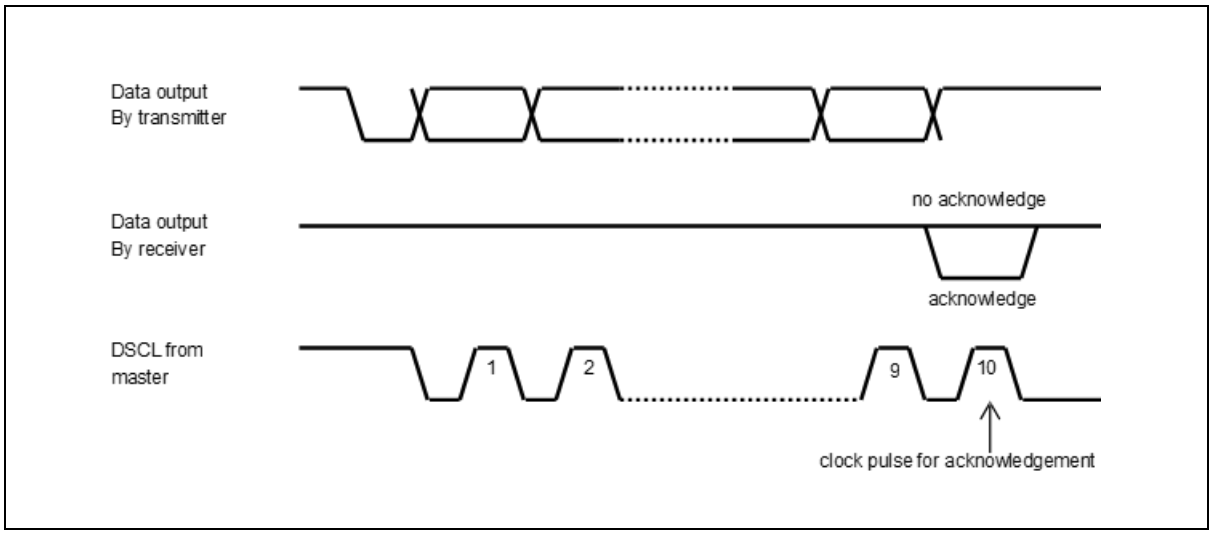


Figure 149. Acknowledge on Serial Bus

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 150 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 DSCL to ‘H’.

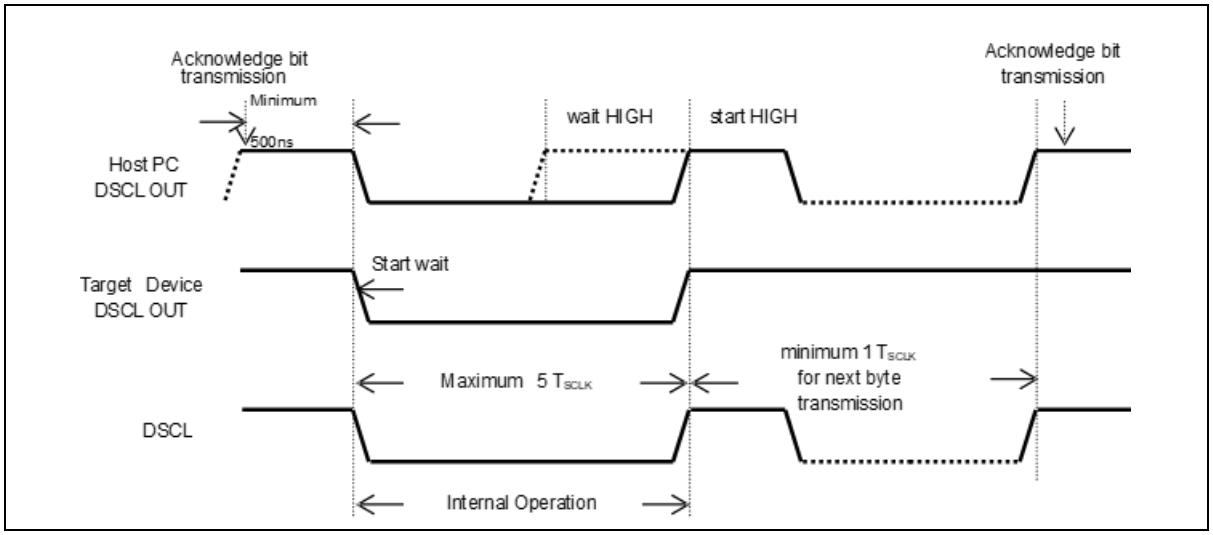


Figure 150. Clock Synchronization during Wait Procedure

22.4 Programmers

22.4.1 E-PGM+

E-PGM+ USB is a single programmer. A user can program A96G150 directly using the E-PGM+.

Figure 151. E-PGM+ (Single Writer) and Pinouts

22.4.2 OCD emulator

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

22.4.3 Gang programmer

E-Gang4 and E-Gang6 allows a user to program on multiple devices at a time. They run not only in PC controlled mode but also in standalone mode without PC control. USB interface is available and it is easy to connect to the handler.

Figure 152. E-Gang4 and E-Gang6 (for Mass Production)

22.5 Flash programming

Program memory of A96G150 is a flash type. This flash ROM is accessed through four pins such as DSCL, DSDA, VDD and VSS in serial data format. For more information about flash memory programming, please refer to Chapter 21. Memory programming

Table 61 introduces each pin and corresponding I/O status.

Table 61. Pins for Flash Programming

Pin name	Main chip pin name	During programming	
		I/O	Description
DSCL	P13	I	Serial clock pin. Input only pin.
DSDA	P11	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.

22.5.1 On-board programming

Microcontroller need only four signal lines including VDD and VSS pins for programming flash ROM with serial protocol. Therefore, the on-board programming is possible if the programming signal lines are considered when the PCB of application board is designed.

22.6 Connection of transmission

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

Normally, it is recommended to place a resister greater than 4.7kΩ for 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.

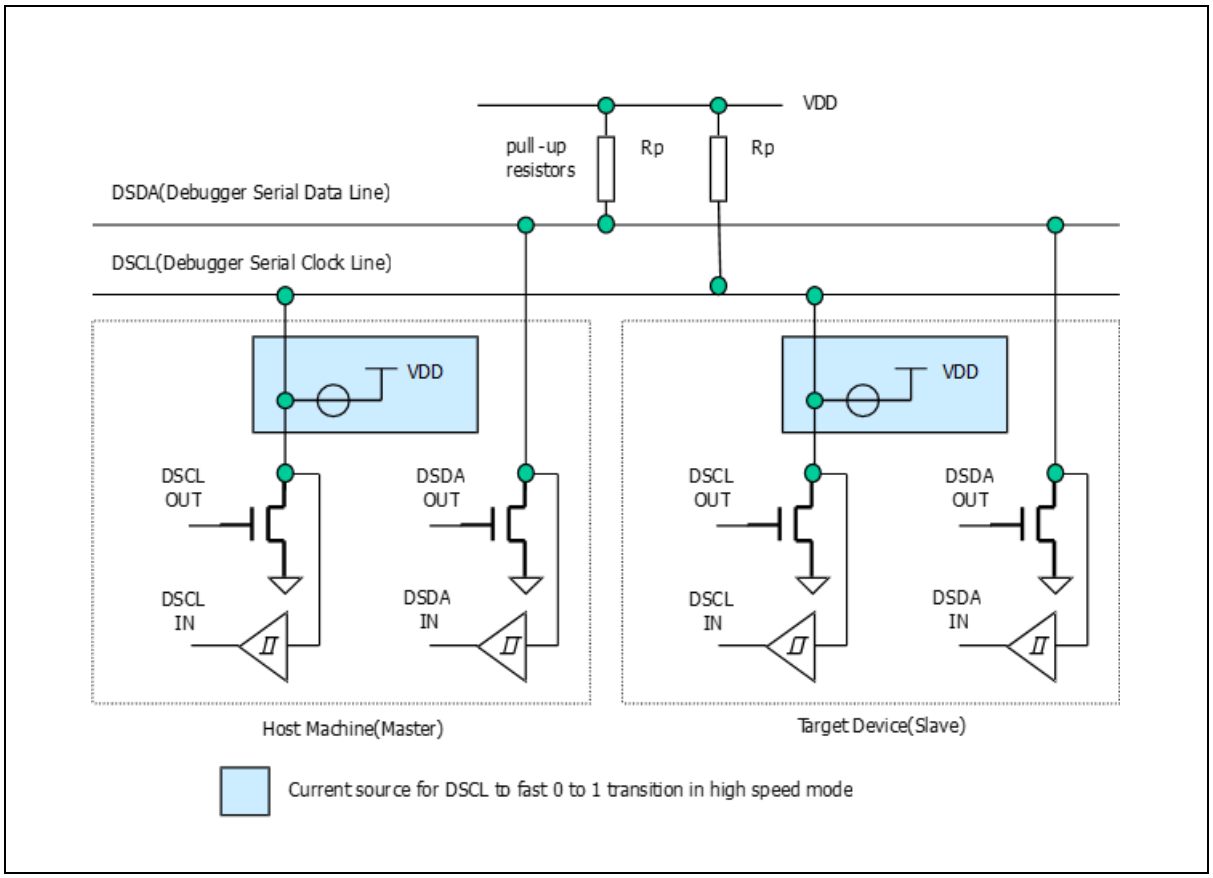


Figure 153. Connection of Transmission

22.7 Circuit design guide

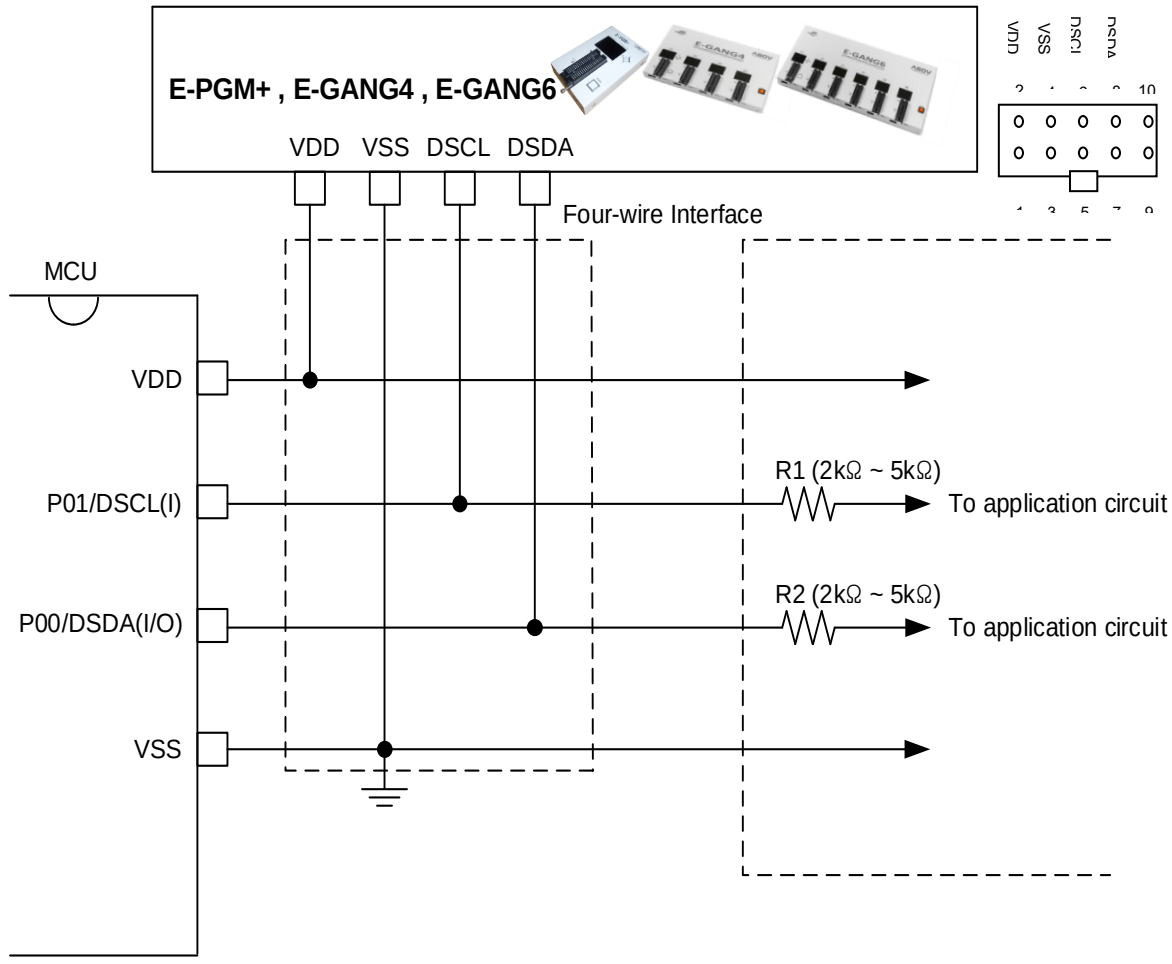
When programming flash memory, the programming tool needs 4 signal lines, DSCL, DSDA, VDD, and VSS. If a user designs a PCB circuit, the user should consider the usage of these 4 signal lines for the on-board programming.

Please be careful to design the related circuit of these signal pins because rising/falling timing of DSCL and DSDA is very important for proper programming.

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

Figure 154 shows an example circuit where the OCD pins (DSCL and DSDA) are shared with other functions. They are required to be connected when debugging or ISP (In System Program) is executed.

Normally, the OCD pins are connected to outside to execute the predefined functions. Even when they are connected for debugging or ISP directly, the OCD pins are shared with other functions by using resistors as shown in Figure 154. 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 154. 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 62. 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 62. 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 62. 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 62. 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

Table 62. Instruction Table (continued)

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

Table 62. 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

Revision	Date	Notes
1.00	2022. 06. 22	First creation
1.01	2022. 11. 01	Revision the font of this document

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
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
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