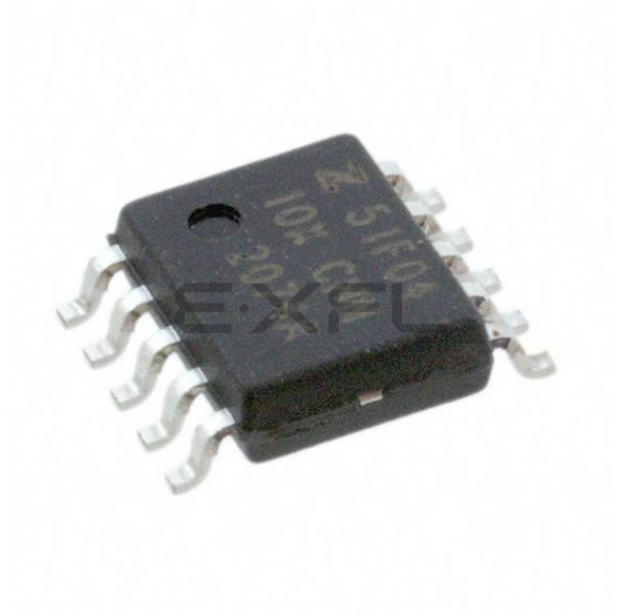
Zilog - Z51F0410HCX Datasheet



Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	8051
Core Size	8-Bit
Speed	8MHz
Connectivity	I²C, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	8
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 8x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	10-SOP (0.154", 3.90mm Width)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z51f0410hcx

Warning: DO NOT USE THIS PRODUCT IN LIFE SUPPORT SYSTEMS.

LIFE SUPPORT POLICY

ZILOG'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS PRIOR WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF ZILOG CORPORATION.

As used herein

Life support devices or systems are devices which (a) are intended for surgical implant into the body, or (b) support or sustain life and whose failure to perform when properly used in accordance with instructions for use provided in the labeling can be reasonably expected to result in a significant injury to the user. A critical component is any component in a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system or to affect its safety or effectiveness.

Document Disclaimer

 Information in this publication concerning the devices, applications, or technology described is intended to suggest possible uses and may be superseded. ZILOG, INC. DOES NOT ASSUME LIABILITY FOR OR PROVIDE A REPRESENTATION OF ACCURACY OF THE INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED IN THIS DOCUMENT. ZILOG ALSO DOES NOT ASSUME LIABILITY FOR INTELLECTUAL PROPERTY INFRINGEMENT RELATED IN ANY MANNER TO USE OF INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED HEREIN OR OTHERWISE. The information contained within this document has been verified according to the general principles of electrical and mechanical engineering.

Z8051 is a trademark or registered trademark of Zilog, Inc. All other product or service names are the property of their respective owners.

Table of Contents

1.11.7 P0[6]/P0[7] Port Structure

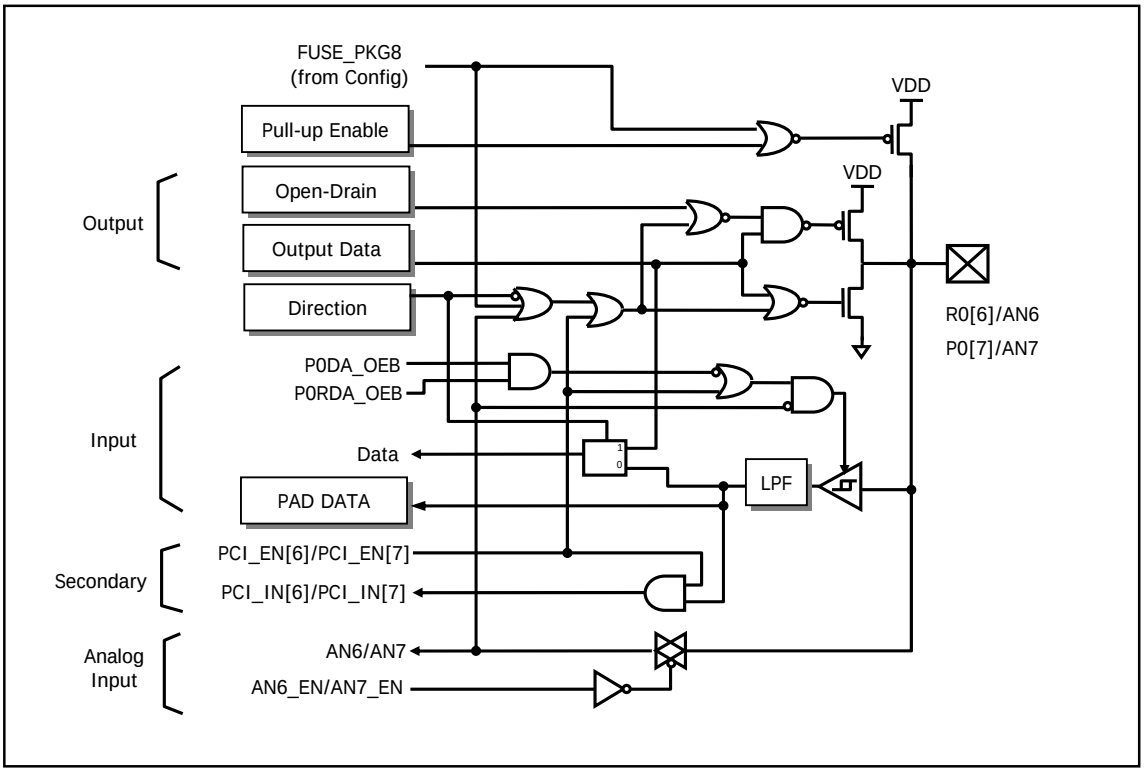


Figure 1.24 P0[6] / AN6, P0[7] / AN7 Port Structure

The analog channel selection bit enables the path of the AN6/AN7 and disable normal logic data path to prevent the input gate leakage current. When the direction register value is 0, the input data is always external PAD voltage.

The pull-up resister is directly controlled by the pull-up register bit regardless of current port direction. The open-drain control is also by open-drain register. On open-drain mode, the push-pull drives just N-MOS. When the direction is output (value 1), the output PAD voltage is controlled by push-pull driver for the current output data. The secondary input PCI_EN[6]/PCI_EN[7] enable the input data path continuously. On normal read mode (non secondary mode), the input data path is only enabled during the CPU OEB (active low). When the analog channel (AN6/AN7) is enabled, the first input gate from the PAD is disabled (highest priority) to prevent the input leakage current for the floating voltage status. At read operation, the input data is selected by PAD direction register. If its value is ‘1’, it reads the current output register value. Otherwise, it reads the current PAD voltage directly (just during OEB active). In addition, always the current PAD voltage is read by PAD DATA register.

1.12.10 AC Characteristics

(VDD=5.0V±10%, VSS=0V, TA=-40~+85℃)

Table 1.13 AC Characteristics

Parameter	Symbol	PIN	MIN	TYP	MAX	Unit
Operating Frequency	fMCP	XIN	1	-	8	MHz
System Clock Cycle Time	tSYS	-	125	-	1000	ns
Oscillation Stabilization Time (8MHz)	tMST1	XIN, XOUT	-	-	10	ms
External Clock “H” or “L” Pulse Width	tCPW	XIN	90	-	-	ns
External Clock Transition Time	tRCP,tFCP	XIN	-	-	10	ns
Interrupt Input Width	tIW	INT0~INT1	2	-	-	tSYS
RESETB Input Pulse “L” Width	tRST	RESETB	8	-	-	tSYS
External Counter Input “H” or “L” Pulse Width	tECW	EC0	2	-	-	tSYS
Event Counter Transition Time	tREC,tFEC	EC0	-	-	20	ns

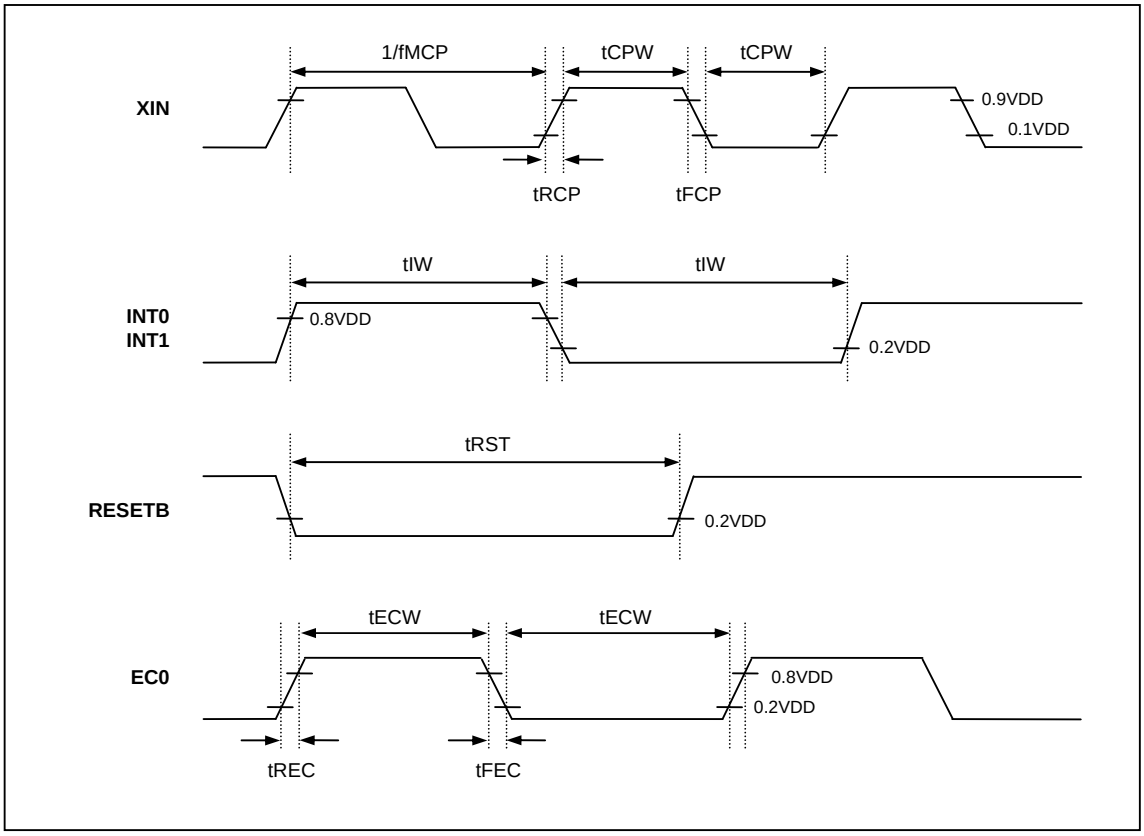


Figure 1.25 AC Timing

1.12.11 Typical Characteristics

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

2.2.2 SFR Map

Table 2.2 SFR Map

Address	Function	Symbol	R/W	@Reset							
				7	6	5	4	3	2	1	0
80H	**Port 0 Data Register**	**P0**	R/W	0	0	0	0	0	0	0	0
84H	Reserved										
84H	Reserved										
86H	**BOD Control Register**	**BODR**	R/W	1	0	0	0	0	0	0	1
			R	-	-	-	-	-	-	-	-
89H	**Port 0 Pull-up Resistor Option Register**	**P0PU**	R/W	0	0	0	0	0	0	0	0
8AH	**System Clock Control Register**	**SCCR**	R/W	0	0	0	0	0	1	0	0
8BH	**BIT Clock Control Register**	**BCCR**	R/W	0	-	-	-	0	1	0	1
8CH	**Basic Interval Timer Register**	**BITR**	R/W	0	0	0	0	0	0	0	0
8DH	**Watch Dog Timer Mode Register**	**WDTMR**	R/W	0	0	0	0	0	0	0	0
8EH	**Watch Dog Timer Register**	**WDTR**	W	1	1	1	1	1	1	1	1
	Watch Dog Timer Counter Register	**WDTCR**	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	Reserved	-	-	-	-	-	-	-	-	-	-
91H	Reserved	-	-	-	-	-	-	-	-	-	-
92H	**Pin Mux Control Register**	**PINMCR**	**R/W**	-	-	-	0	0	0	0	0
93H	Reserved	-	-	-	-	-	-	-	-	-	-
94H	Reserved	-	-	-	-	-	-	-	-	-	-
95H	Reserved	-	-	-	-	-	-	-	-	-	-
96H	**Buzzer Control Register**	**BUZCR**	R/W	-	-	-	-	-	0	0	0
97H	**TIMER 0,1,4 Interrupt Flag Register**	**TFLG**	R/W	-	0	0	0	-	-	-	-
98H	**Port 0 Direction Register**	**P0IO**	**R/W**	0	0	0	0	0	0	0	0
99H	Reserved	-	-	-	-	-	-	-	-	-	-
9AH	**A/D Converter Mode Register**	**ADCM**	R/W	1	0	0	0	1	1	1	1
9BH	**A/D Converter Result High Register**	**ADCRH**	R	X	X	X	X	X	X	X	X
9CH	**A/D Converter Result Low Register**	**ADCRL**	R/W	0	1	0	0	-	-	X	X
9DH	**Watch Timer Mode Register**	**WTMR**	R/W	0	-	-	0	0	0	0	0
9EH	**Watch Timer Register**	**WTR**	W	0	1	1	1	1	1	1	1
	Watch Timer Counter Register	**WTCR**	R	0	0	0	0	0	0	0	0

BEH	Reserved	-	-	-	-	-	-	-	-	-	-
BFH	Reserved	-	-	-	-	-	-	-	-	-	-
C0H	**Port 0 Debounce Register**	**P0DB**	R/W	0	0	0	0	0	0	0	0
C1H	**Port 0 Pin Change Interrupt**	**P0PC**	R/W	0	0	0	0	0	0	0	0
C2H	Reserved	-	-	-	-	-	-	-	-	-	-
C3H	Reserved	-	-	-	-	-	-	-	-	-	-
C4H	Reserved	-	-	-	-	-	-	-	-	-	-
C5H	Reserved	-	-	-	-	-	-	-	-	-	-
C6H	Reserved		-	-	-	-	-	-	-	-	-
C7H	Reserved		-	-	-	-	-	-	-	-	-
C8H	Reserved	-	-	-	-	-	-	-	-	-	-
C9H	Reserved	-	-	-	-	-	-	-	-	-	-
CAH	Reserved		-	-	-	-	-	-	-	-	-
CBH	Reserved		-	-	-	-	-	-	-	-	-
CCH	Reserved		-	-	-	-	-	-	-	-	-
CDH	**Timer 4 Data High Register**	**T4H**	**R**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
CEH	Timer 4 Mode Control Register	T4CR	R/W	0	0	0	0	0	0	0	0
CFH	Timer 4 Data Low Register	T4L	R/W	0	0	0	0	0	0	0	0
D7H	**I2C Slave Address Register**	**I2CAR**	R/W	0	0	0	0	0	0	0	0
D8H	Reserved	-	-	-	-	-	-	-	-	-	-
D9H	Reserved	-	-	-	-	-	-	-	-	-	-
DAH	**I2C Mode Control Register**	**I2CMR**	R/W	0	0	0	0	0	0	0	0
DBH	**I2C Status Register**	**I2CSR**	R	0	0	0	0	0	0	0	0
DCH	**I2C SCL Low Period Register**	**I2CSCLLR**	R/W	0	0	1	1	1	1	1	1
DDH	**I2C SCL High Period Register**	**I2CSCLHR**	R/W	0	0	1	1	1	1	1	1
DEH	**I2C SDA Hold Time Register**	**I2CSDAHR**	R/W	0	0	0	0	0	0	1	1
DFH	**I2C Data Register**	**I2CDR**	R/W	1	1	1	1	1	1	1	1
E1H	Reserved	-	-	-	-	-	-	-	-	-	-
E2H	**USART Control Register 1**	**UCTRL1**	R/W	0	0	0	0	0	0	0	0
E3H	**USART Control Register 2**	**UCTRL2**	R/W	0	0	0	0	0	0	0	0
E4H	**USART Control Register 3**	**UCTRL3**	R/W	0	0	0	0	0	0	0	0
E5H	**USART Status Register**	**USTAT**	R/W	1	0	0	0	-	0	0	0
E6H	**USART Baud Rate Generation Register**	**UBAUD**	R/W	1	1	1	1	1	1	1	1
E7H	**USART Data Register**	**UDATA**	R/W	1	1	1	1	1	1	1	1

4.4 WT

4.4.1 Overview

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

4.4.2 Block Diagram

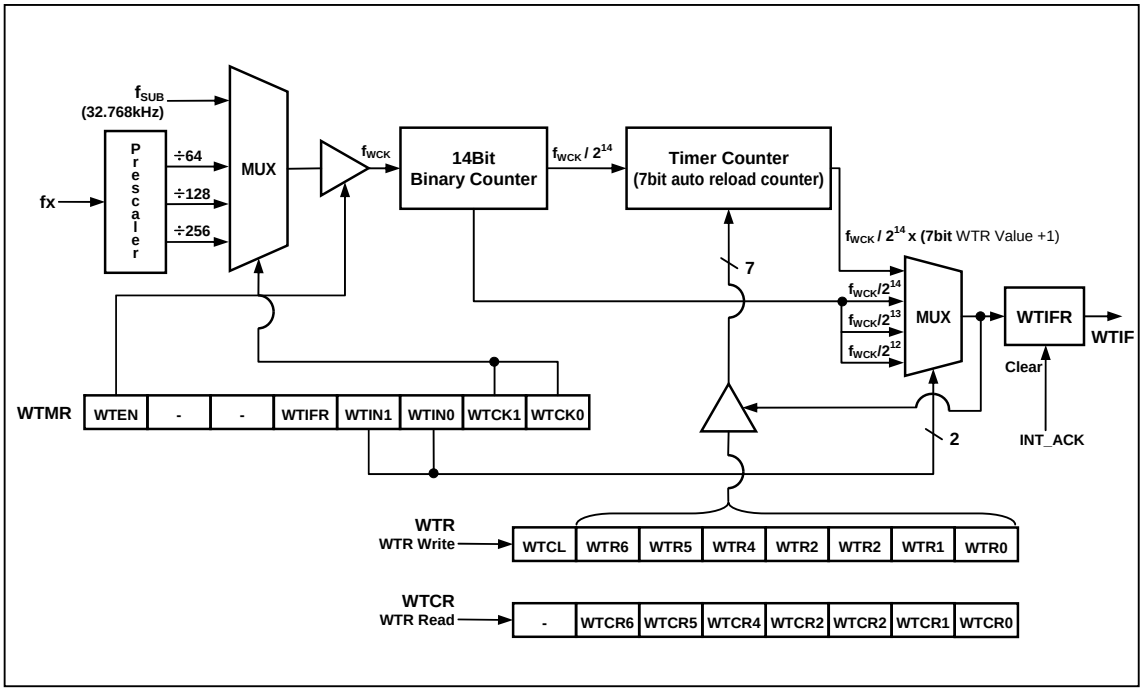


Figure 4.5 Watch Timer Block Diagram

4.4.3 Register Map

Table 4.4 WT Register Map

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

4.4.4 Watch Timer Register description

The watch timer register (WT) consists of Watch Timer Mode Register (WTMR), Watch Timer Counter Register (WTCR) and Watch Timer Register (WTR). As WTMR is 6-bit writable/readable

4.5.1.2 8-Bit Timer/Counter Mode

The 8-bit Timer/Counter Mode is selected by control registers as shown in Figure 4.6.

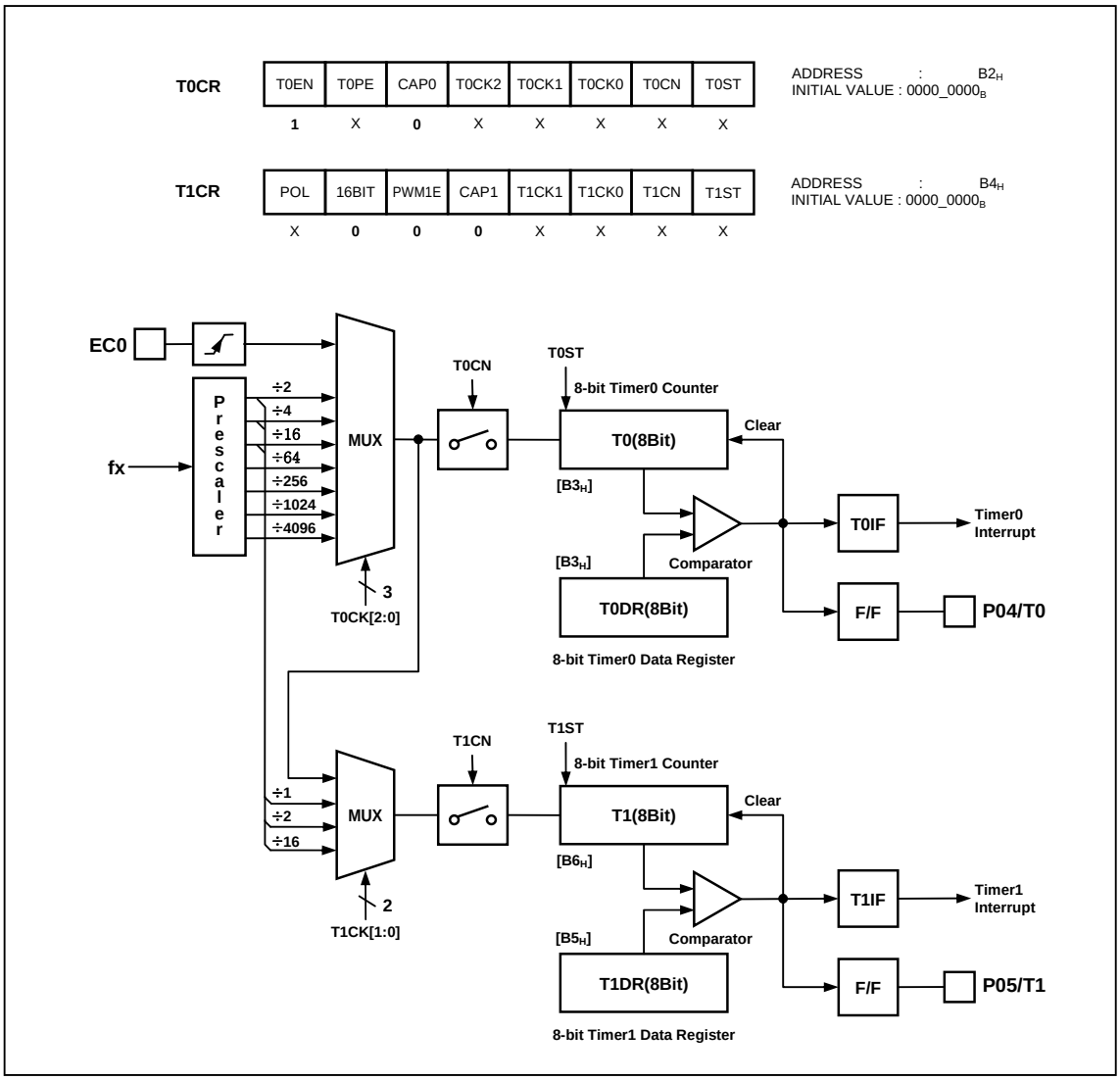


Figure 4.6 8 Bit Timer/Event Counter0, 1 Block Diagram

The two 8-bit timers have each counter and data register. The counter register is increased by internal or external clock input. The timer 0 can use the input clock with 2, 4, 16, 64, 256, 1024, 4096 prescaler division rates (T0CK[2:0]). The timer 1 can use the input clock with 1, 2, 16 and timer 0 overflow clock (T1CK[1:0]). When the value of T0, 1 value and the value of T0DR, T1DR are respectively identical in Timer 0, 1, the interrupt of timer 0, 1 occurs. The external clock (EC0) counts up the timer at the rising edge. If EC0 is selected from T0CK[2:0], EC0 port becomes input port. The timer 1 can't use the external EC0 clock.

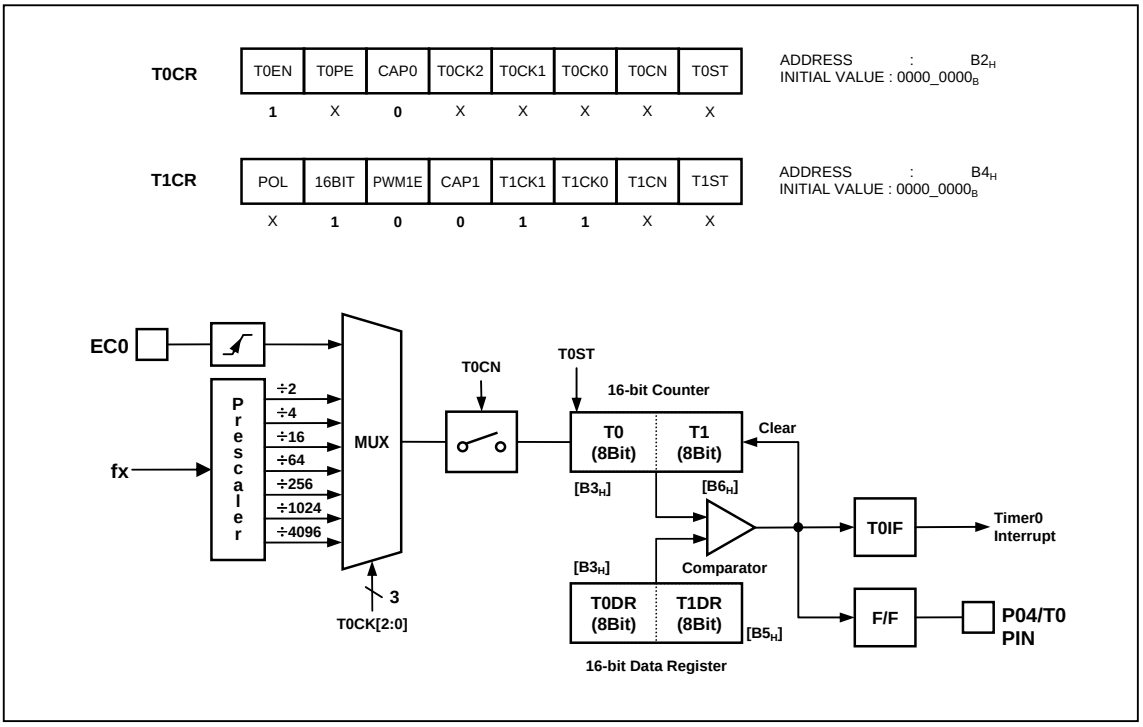


Figure 4.9 16 Bit Timer/Event Counter0, 1 Block Diagram

Note: Do not set T0DR to 0x00 in 16-bit mode. If T0DR is set to 0x00, Timer interrupt or count match occur after T1DR+0x01. If you set T0DR to be 0x00, T1DR must have one fewer number of count than the number of count which you want.

Example: If T1DR=0x01 and T0DR=0x00, counter match occurs when T1=0x02 and T0=0x00.

4.5.1.4 8-Bit Capture Mode

The timer 0, 1 capture mode is set by CAP0, CAP1 as '1'. The clock source can use the internal/external clock. Basically, it has the same function of the 8-bit timer/counter mode and the interrupt occurs at T0, T1 and T0DR, T1DR matching time, respectively. The capture result is loaded into CDR0, CDR1. The T0, T1 value is automatically cleared by hardware and restarts counter.

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

As the EIEDGE and EIPOLA register setting, the external interrupt INT0, INT1 function is chosen.

The CDR0, T0 and T0DR are in same address. In the capture mode, reading operation is read the CDR0, not T0DR because path is opened to the CDR0. The CDR1 has the same function.

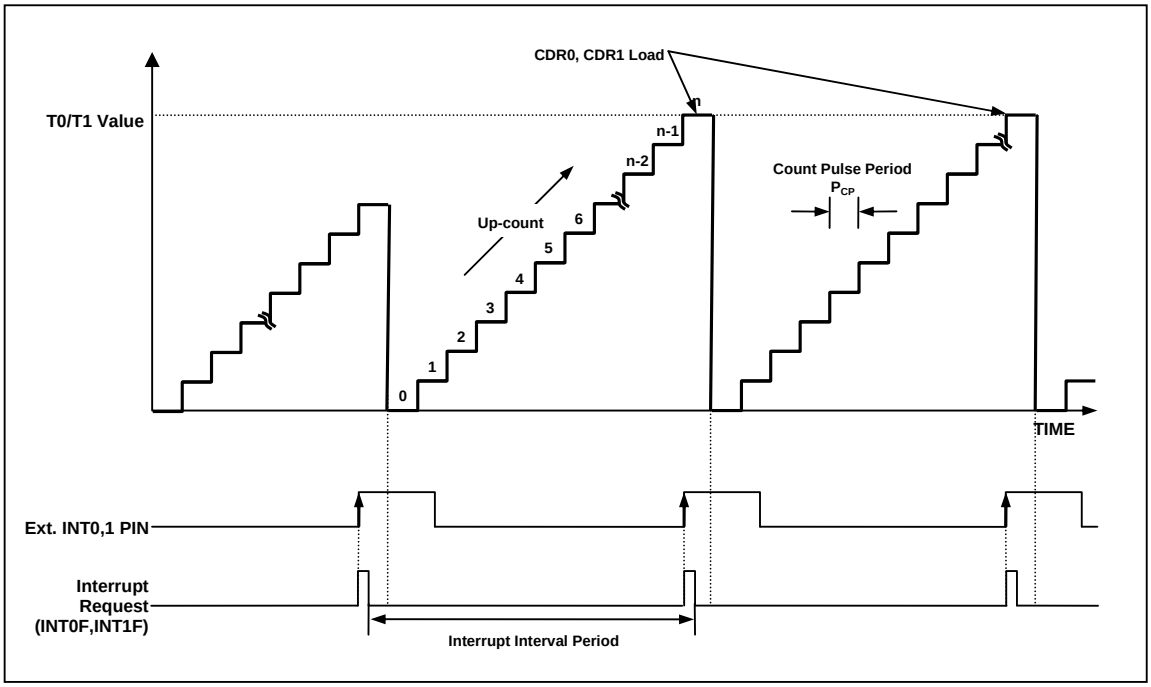


Figure 4.11 Input Capture Mode Operation of Timer 0, 1

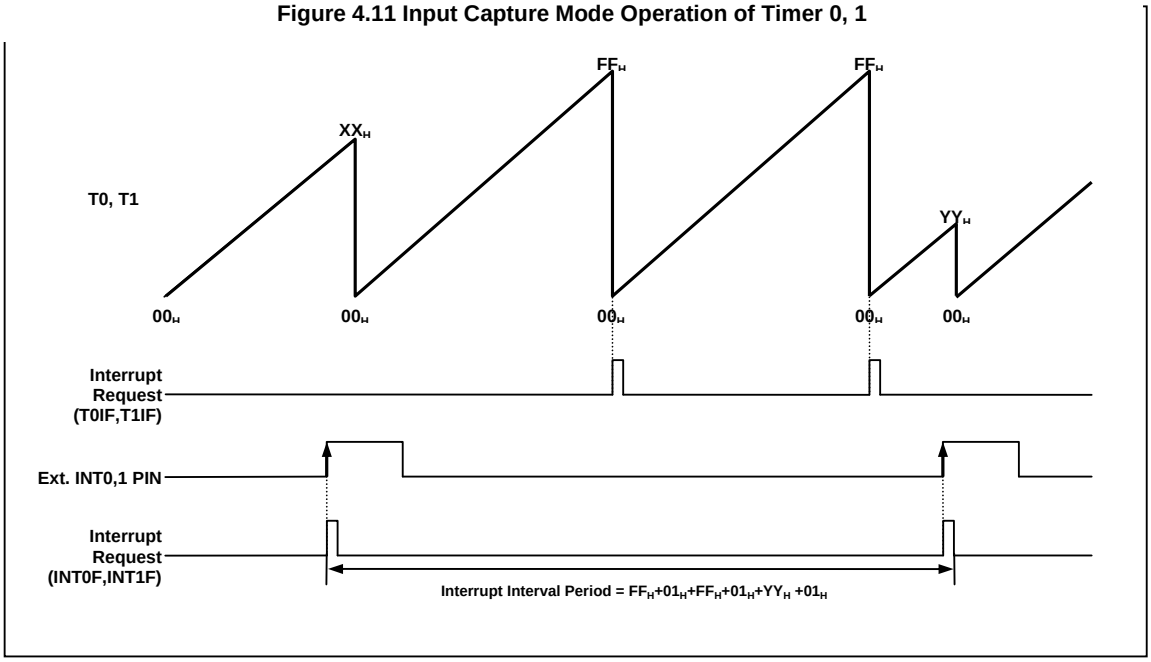


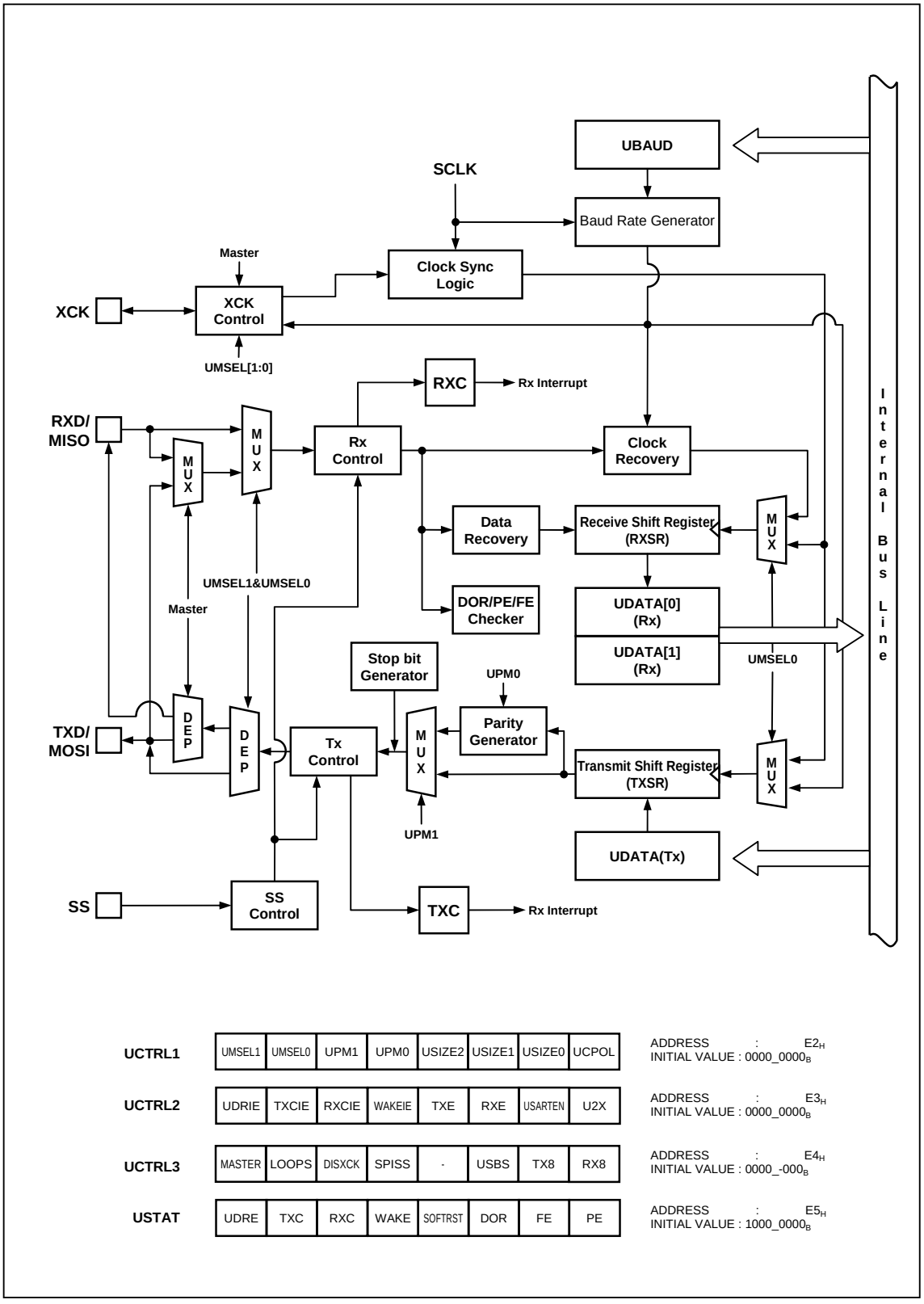
Figure 4.12 Express Timer Overflow in Capture Mode

4.5.1.5 16-Bit Capture Mode

The 16-bit capture mode is the same operation as 8-bit capture mode, except that the timer register uses 16 bits.

The clock source is selected from T0CK[2:0] and T1CK[1:0] must set 11b and 16BIT0 bit must set to '1'. The 16-bit mode setting is shown as Figure 4.13.

4.6.2 Block Diagram



UCTRL1	UMSEL1	UMSEL0	UPM1	UPM0	USIZE2	USIZE1	USIZE0	UCPOL	ADDRESS : $E2_H$ INITIAL VALUE : 0000_0000_B
UCTRL2	UDRIE	TXCIE	RXCIE	WAKEIE	TXE	RXE	USARTEN	U2X	ADDRESS : $E3_H$ INITIAL VALUE : 0000_0000_B
UCTRL3	MASTER	LOOPS	DISXCK	SPISS	-	USBS	TX8	RX8	ADDRESS : $E4_H$ INITIAL VALUE : 0000_-000_B
USTAT	UDRE	TXC	RXC	WAKE	SOFTRST	DOR	FE	PE	ADDRESS : $E5_H$ INITIAL VALUE : 1000_0000_B

Figure 4.18 USART Block Diagram

Note: In Tx mode, the length of start bit can be shorter than one or two clock of the lengh of the other data bits.

4.6.8.1 Sending Tx data

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

4.6.8.2 Transmitter flag and interrupt

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

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

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

The Transmit Complete (TXC) flag bit is set when the entire frame in the transmit shift register has been shifted out and there are no more data in the transmit buffer. The TXC flag is automatically cleared when the Transmit Complete Interrupt service routine is executed, or it can be cleared by writing ‘0’ to TXC bit in UCTRL2 register.

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

4.6.8.3 Parity Generator

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

4.6.8.4 Disabling Transmitter

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

4.6.9 USART Receiver

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

4.9.4 Register Map

Table 4.17 Analog Comparator Register Map

Name	Address	Dir	Default	Description
ACCSR	E9H	R/W	00H	Analog Comparator Control & Status Register

4.9.5 Analog Comparator Register description

Analog Comparator Register has one control register, Analog Comparator Control & Status Register (ACCSR). Note that AMUXENB is the inverted signal of AMUXEN bit which comes from ADC's ADCM2 register

4.9.6 Register description for Analog Comparator

ACCSR (Analog Comparator Control & Status Register) : E9H

7	6	5	4	3	2	1	0
ACE	ACBG	ACO	ACIF	ACIE	-	ACISM1	ACISM0
R/W	R/W	R	R	R/W	-	R/W	R/W

Initial value : 00H

ACE Enable Analog Comparator (AC).

0 Disable AC (power down)

1 Enable AC

ACBG Select (-) input source of AC, Band Gap Reference Voltage or AN4.

0 (-) input is from AN4

1 (-) input is from Band Gap Reference Voltage

ACO This bit represents the value of ACOUT (Output of Analog Comparator). ACO bit is sampled by SCLK, system clock, twice. When ACE is '0', this bit is also cleared.

0 Comparator output is LOW

1 Comparator output is HIGH

ACIF This bit is set when an Analog Comparator Interrupt is generated according to the ACISM[1:0] bits. This bit is cleared when Analog Comparator Interrupt is executed or '0' is written to this bit field.

0 No interrupt generated or cleared

1 Interrupt generated

ACIE Enable Analog Comparator Interrupt.

0 Disable Interrupt, Polling mode operation

1 Enable Interrupt

ACISM[1:0] Select Interrupt Mode of Analog Comparator.

ACISM1	ACISM0	Description
0	0	Reserved
0	1	Interrupt on falling edge of ACOUT
1	0	Interrupt on rising edge of ACOUT
1	1	Interrupt on both edge of ACOUT

PSR1 (Comparator Pin Selection Register) : A0H

7	6	5	4	3	2	1	0
							ACO_EN

7.2 Two-pin external interface

7.2.1 Basic transmission packet

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

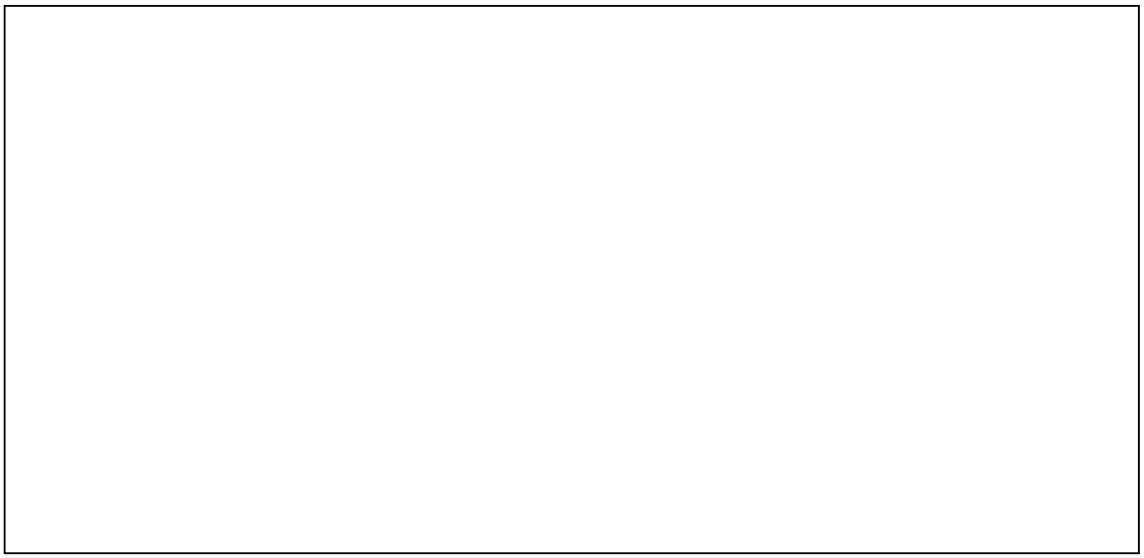


Figure 7.2 10-bit transmission packet

7	6	5	4	3	2	1	0
ARL7	ARL6	ARL5	ARL4	ARL3	ARL2	ARL1	ARL0
R/W	R.W	R.W	R.W	R.W	R.W	R.W	R.W

Initial value : 00H

ARL[7:0] Flash and EEPROM address low

FEARM (Flash and EEPROM address middle Register) : F3H

7	6	5	4	3	2	1	0
ARM7	ARM6	ARM5	ARM4	ARM3	ARM2	ARM1	ARM0
R/W	R.W	R.W	R.W	R.W	R.W	R.W	R.W

Initial value : 00H

ARM[7:0] Flash and EEPROM address middle

FEARH (Flash and EEPROM address high Register) : F4H

7	6	5	4	3	2	1	0
ARH7	ARH6	ARH5	ARH4	ARH3	ARH2	ARH1	ARH0
R/W	R.W	R.W	R.W	R.W	R.W	R.W	R.W

Initial value : 00H

ARH[7:0] Flash and EEPROM address high

CHKSUM[23:16] Checksum Result from auto verify mode

FEAR registers are used for program, erase and auto-verify. In program and erase mode, it is page address and ignored the same least significant bits as the number of bits of page address. In auto-verify mode, address increases automatically by one.

FEARs are write-only register. Reading these registers returns 24-bit checksum result

FEDR (Flash and EEPROM data control Register) : F5H

7	6	5	4	3	2	1	0
FEDR7	FEDR6	FEDR5	FEDR4	FEDR3	FEDR2	FEDR1	FEDR0
W	W	W	W	W	W	W	W

Initial value : 00H

FEDR[7:0] Flash and EEPROM data

Data register. In no program/erase/verify mode, READ/WRITE of FECR read or write data from EEPROM or Flash to this register or from this register to Flash or EEPROM.

The sequence of writing data to this register is used for EEPROM program entry. The mode entrance sequence is to write 0xA5 and 0x5A to it in order.

FETCR (Flash and EEPROM Time control Register) : EDH

7	6	5	4	3	2	1	0
TCR7	TCR6	TCR5	TCR4	TCR3	TCR2	TCR1	TCR0

8.5.3 Parallel Mode timing diagram

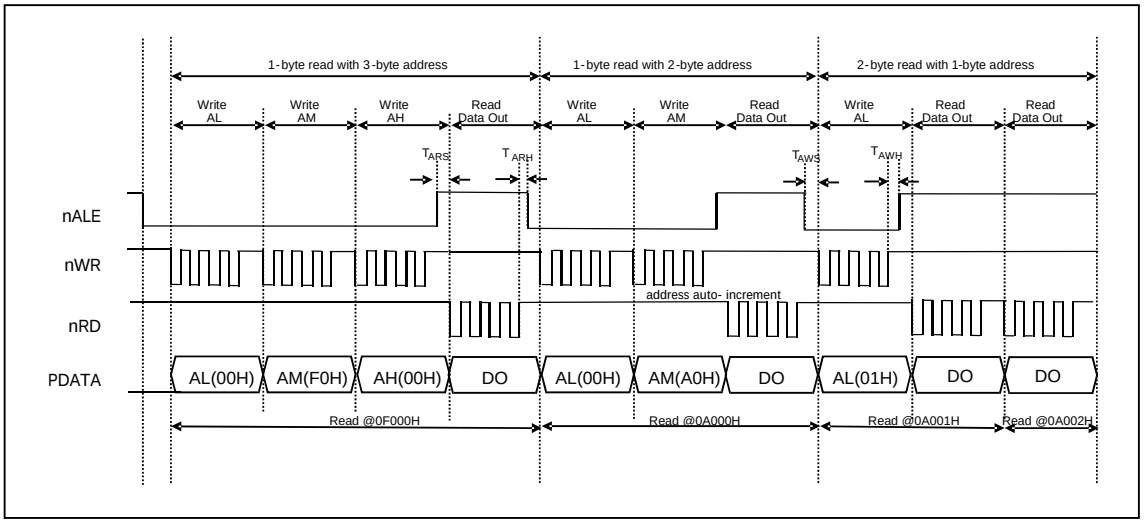


Figure 8.8 Parallel Byte Read Timing of Program Memory

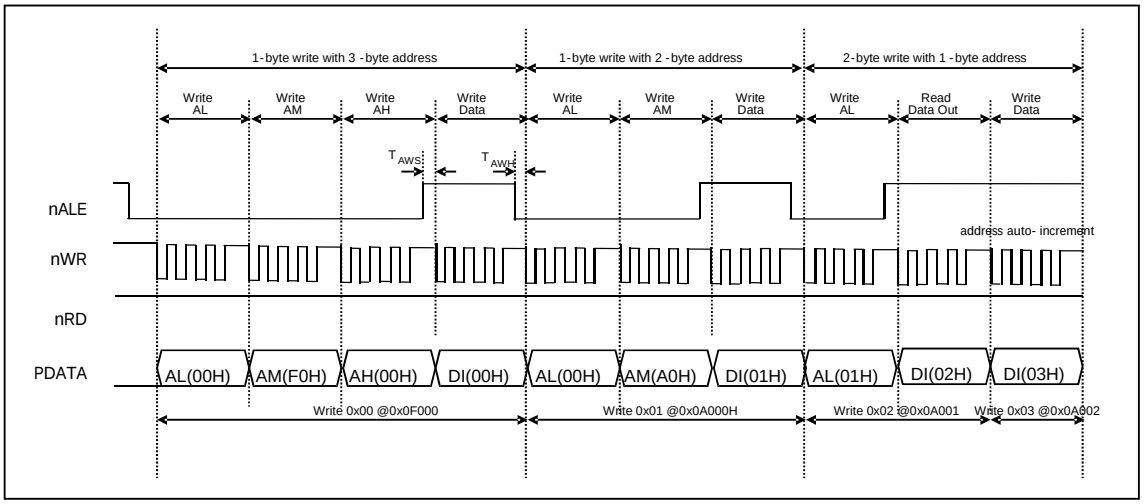


Figure 8.9 Parallel Byte Write Timing of Program Memory

Table 8-5 Control Pin Description

Pin	P01	P02	P03	P04	P05
Function	PDATA[0]	PDATA[1]	nALE	nWR	nRD

8.6 Mode entrance method of ISP and byte-parallel mode

8.6.1 Mode entrance method for ISP

TARGET MODE	R03	R05	R03

10. APPENDIX

10.1 Instruction Table

Instructions are either 1, 2 or 3 bytes long as listed in the 'Bytes' column below.
Each instruction takes either 1, 2 or 4 machine cycles to execute as listed in the following table. 1 machine cycle comprises 2 system clock cycles.

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

LOGICAL				
Mnemonic	**Description**	**Bytes**	**Cycles**	**Hex code**
ANL A,Rn	AND register to A	1	1	58-5F
ANL A,dir	AND direct byte to A	2	1	55
ANL A,@Ri	AND indirect memory to A	1	1	56-57
ANL A,#data	AND immediate to A	2	1	54
ANL dir,A	AND A to direct byte	2	1	52
ANL dir,#data	AND immediate to direct byte	3	2	53
ORL A,Rn	OR register to A	1	1	48-4F
ORL A,dir	OR direct byte to A	2	1	45
ORL A,@Ri	OR indirect memory to A	1	1	46-47
ORL A,#data	OR immediate to A	2	1	44
ORL dir,A	OR A to direct byte	2	1	42
ORL dir,#data	OR immediate to direct byte	3	2	43
XRL A,Rn	Exclusive-OR register to A	1	1	68-6F
XRL A,dir	Exclusive-OR direct byte to A	2	1	65
XRL A, @Ri	Exclusive-OR indirect memory to A	1	1	66-67
XRL A,#data	Exclusive-OR immediate to A	2	1	64
XRL dir,A	Exclusive-OR A to direct byte	2	1	62
XRL dir,#data	Exclusive-OR immediate to direct byte	3	2	63
CLR A	Clear A	1	1	E4
CPL A	Complement A	1	1	F4
SWAP A	Swap Nibbles of A	1	1	C4
RL A	Rotate A left	1	1	23
RLC A	Rotate A left through carry	1	1	33
RR A	Rotate A right	1	1	03

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	3	2	D8-DF
DJNZ dir,rel	Decrement direct byte, jnz relative	3	2	D5

MISCELLANEOUS				
Mnemonic	**Description**	**Bytes**	**Cycles**	**Hex code**
NOP	No operation	1	1	00

ADDITIONAL INSTRUCTIONS (selected through EO[7:4])				
Mnemonic	**Description**	**Bytes**	**Cycles**	**Hex code**
MOVC @(DPTR++),A	M8051W/M8051EW-specific instruction supporting software download into program memory	1	2	A5
TRAP	Software break command	1	1	A5

In the above table, an entry such as E8-EF indicates a continuous block of hex opcodes used for 8 different registers, the register numbers of which are defined by the lowest three bits of the corresponding code. Non-continuous blocks of codes, shown as 11→F1 (for example), are used for absolute jumps and calls, with the top 3 bits of the code being used to store the top three bits of the destination address.

The CJNE instructions use the abbreviation #d for immediate data; other instructions use #data.

10.2 Instructions on how to use the input port.

- Error occur status
  - Using compare jump instructions with input port, it could cause error due to the timing conflict inside the MCU.
  - Compare jump Instructions which cause potential error used with input port condition:

```
JB      bit, rel   ; jump on direct bit=1
JNB     bit, rel   ; jump on direct bit=0
JBC     bit, rel   ; jump on direct bit=1 and clear
CJNE    A, dir, rel        ; compare A, direct jne relative
DJNZ    dir, rel   ; decrement direct byte, jnz relative
```

  - It is only related with Input port. Internal parameters, SFRs and output bit ports don't cause an y error by using compare jump instructions.
  - If input signal is fixed, there is no error in using compare jump instructions.

- Error status example

```
while(1){
  if (P00==1){ P10=1; }
  else { P10=0; }
  P11^=1;
}
```

```
zzz:  JNB     080.0, xxx  ; it can cause an error
      SETB    088.0
      SJMP    yyy
xxx:  CLR     088.0
yyy:  MOV     C,088.1
      CPL     C
      MOV     088.1,C
      SJMP    zzz
```

```
unsigned char ret_bit_err(void)
{
  return !P00;
}
```

```
      MOV     R7, #000
      JB      080.0, xxx  ; it can cause an error
      MOV     R7, #001
xxx:  RET
```

- Preventative measures (2 cases)