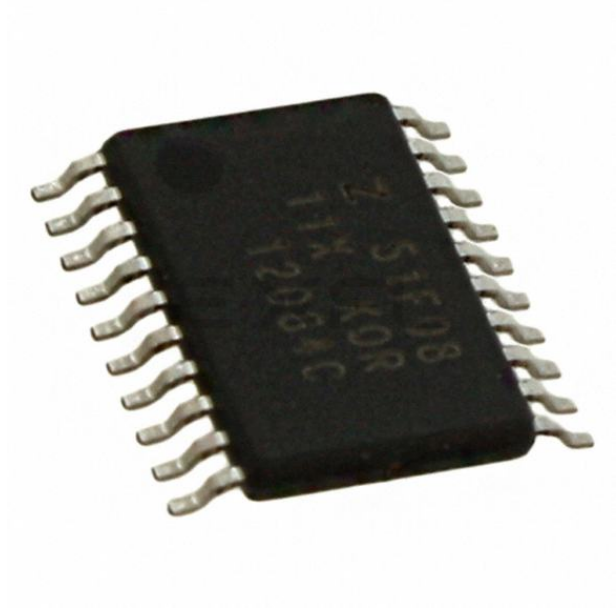

Welcome to E-XFL.COM

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	12MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	18
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 9x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-TSSOP (0.173", 4.40mm Width)
Supplier Device Package	20-TSSOP
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z51f0811rhx

8.2 Data Memory

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

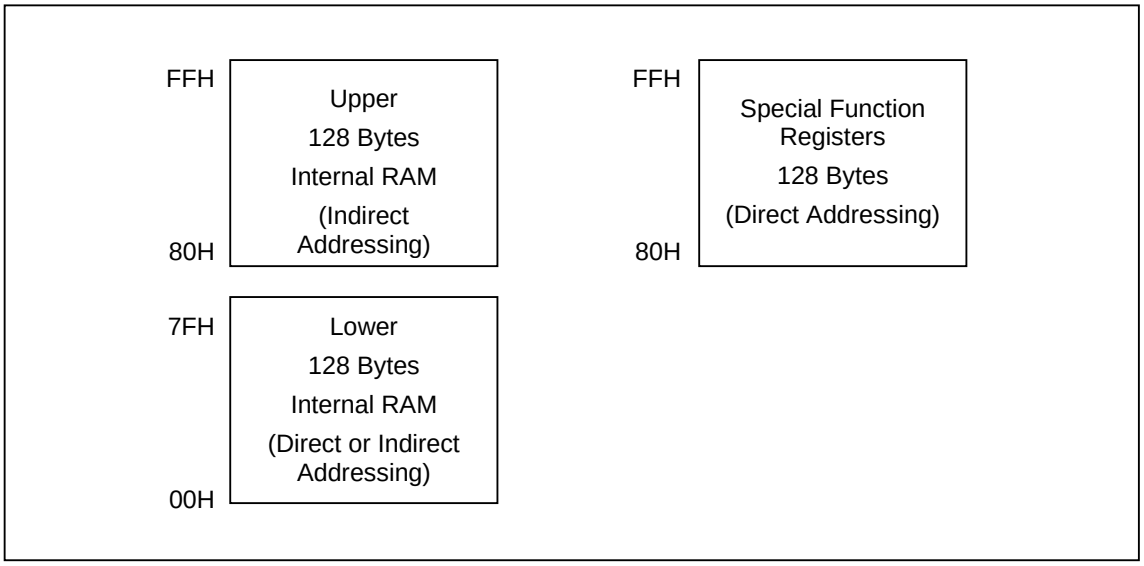

Figure 8-2 Data memory map

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

Internal Data memory addresses are always one byte wide, which implies an address space of only 256 bytes. However, the addressing modes for internal RAM can in fact accommodate 384 bytes, using a simple trick. Direct addresses higher than 7FH access one memory space and indirect addresses higher than 7FH access a different memory space. Thus Figure 8-2 shows the upper 128 and SFR space occupying the same block of addresses, 80H through FFH, although they are physically separate entities.

The lower 128 bytes of RAM are present in all 8051 devices as mapped in Figure 8-3. The lowest 32 bytes are grouped into 4 banks of 8 registers. Program instructions call out these registers as R0 through R7. Two bits in the Program Status Word select which register bank is in use. This allows more efficient used of code space, since register instructions are shorter than instructions that use direct addressing.

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

All of the bytes in the lower 128 can be accessed by either direct or indirect addressing. The upper 128 bytes RAM can only be accessed by indirect addressing. These spaces are used for user RAM and stack pointer.

10.4 Interrupt Vector Table

The interrupt controller supports 32 interrupt sources as shown in the Table 10-2 below. When interrupt becomes service, long call instruction (LCALL) is executed in the vector address. Interrupt request 32 has a decided priority order.

Table 10-2 Interrupt Vector Address Table

Interrupt Source	Symbol	Interrupt Enable Bit	Priority	Mask	Vector Address
Hardware Reset	RESETB	0	0	Non-Maskable	0000H
External Interrupt 0	INT0	IE0.0	1	Maskable	0003H
External Interrupt 1	INT1	IE0.1	2	Maskable	000BH
External Interrupt 2	INT2	IE0.2	3	Maskable	0013H
External Interrupt 3	INT3	IE0.3	4	Maskable	001BH
Pin Change Interrupt (P0)	INT4	IE0.4	5	Maskable	0023H
Reserved	INT5	-	6	-	002BH
USART0 Rx	INT6	IE1.0	7	Maskable	0033H
USART0Tx	INT7	IE1.1	8	Maskable	003BH
SPI0	INT8	IE1.2	9	Maskable	0043H
I^2C	INT9	IE1.3	10	Maskable	004BH
USART1 Rx	INT10	IE1.4	11	Maskable	0053H
USART1 Tx	INT11	IE1.5	12	Maskable	005BH
T0	INT12	IE2.0	13	Maskable	0063H
T1	INT13	IE2.1	14	Maskable	006BH
T2	INT14	IE2.2	15	Maskable	0073H
T3	INT15	IE2.3	16	Maskable	007BH
T4	INT16	IE2.4	17	Maskable	0083H
EEPROM	INT17	IE2.5	18	Maskable	008BH
ADC	INT18	IE3.0	19	Maskable	0093H
Comparator	INT19	IE3.1	20	Maskable	009BH
WT	INT20	IE3.2	21	Maskable	00A3H
WDT	INT21	IE3.3	22	Maskable	00ABH
BIT	INT22	IE3.4	23	Maskable	00B3H
Reserved	INT23	-	24	-	00BBH
Reserved	INT24	-	25	-	00C3H
Reserved	INT25	-	26	-	00CBH
Reserved	INT26	-	27	-	00D3H
Reserved	INT27	-	28	-	00DBH
External Interrupt 4	INT28	IE4.4	29	Maskable	00E3H
External Interrupt 5	INT29	IE4.5	30	Maskable	00EBH
External Interrupt 6	INT30	IE5.0	31	Maskable	00F3H
External Interrupt 7	INT31	IE5.1	32	Maskable	00FBH

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

10.5 Interrupt Sequence

An interrupt request is held until the interrupt is accepted or the interrupt latch is cleared to ‘0’ by a reset or an instruction. Interrupt acceptance always generates at last cycle of the instruction. So instead of fetching the current instruction, CPU executes internally LCALL instruction and saves the PC stack. For the interrupt service routine, the interrupt controller gives the address of LJMP instruction to CPU. After finishing the current instruction, at the next instruction to go interrupt service routine needs 5~8 machine cycle and the interrupt service task is terminated upon execution of an interrupt return instruction [RETI]. After generating interrupt, to go to interrupt service routine, the following process is progressed

INT27E Reserved
0 Disable
1 Enable

INT26E Reserved
0 Disable
1 Enable

INT25E Reserved
0 Disable
1 Enable

INT24E Reserved
0 Disable
1 Enable

IE5 (Interrupt Enable Register 5) : ADH

7	6	5	4	3	2	1	0
-	-	INT35E	INT34E	INT33E	INT32E	INT31E	INT30E
R	R	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

INT35E Reserved
0 Disable
1 Enable

INT34E Reserved
0 Disable
1 Enable

INT33E Reserved
0 Disable
1 Enable

INT32E Reserved
0 Disable
1 Enable

INT31E Enable or disable External Interrupt 7
0 Disable
1 Enable

INT30E Enable or disable External Interrupt 6
0 Disable
1 enable

11.3.4 Watch Dog Timer Register description

The Watch dog timer (WDT) Register consists of Watch Dog Timer Register (WDTR), Watch Dog Timer Counter Register (WDTCR) and Watch Dog Timer Mode Register (WDTMR).

11.3.5 Register description for Watch Dog Timer

WDTR (Watch Dog Timer Register: Write Case) : 8EH

7	6	5	4	3	2	1	0
WDTR7	WDTR6	WDTR5	WDTR4	WDTR3	WDTR2	WDTR1	WDTR0
W	W	W	W	W	W	W	W

Initial value : FFH

WDTR[7:0] Set a period
WDT Interrupt Interval=(BIT Interrupt Interval) x(WDTR Value+1)

Note) To guarantee proper operation, the data should be greater than 01H.

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

7	6	5	4	3	2	1	0
WDTCR7	WDTCR6	WDTCR5	WDTCR4	WDTCR3	WDTCR2	WDTCR1	WDTCR0
R	R	R	R	R	R	R	R

Initial value : 00H

WDTCR[7:0] WDT Counter

WDTMR (Watch Dog Timer Mode Register) : 8DH

7	6	5	4	3	2	1	0
WDTEN	WDTRSON	WDTCL	-	-	-	-	WDTIFR
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)

WDTIFR When WDT Interrupt occurs, this bit becomes '1'. For clearing bit, write '0' to this bit or auto clear by INT_ACK signal.
0 WDT Interrupt no generation
1 WDT Interrupt generation

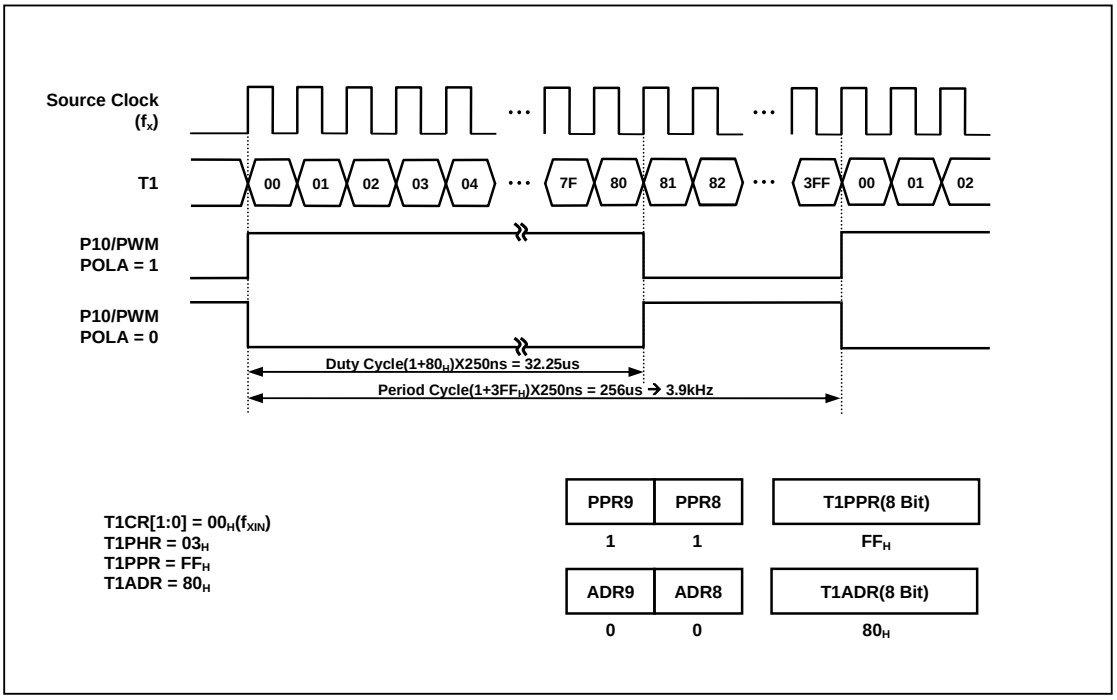

Figure 11-16 Example of PWM at 4MHz

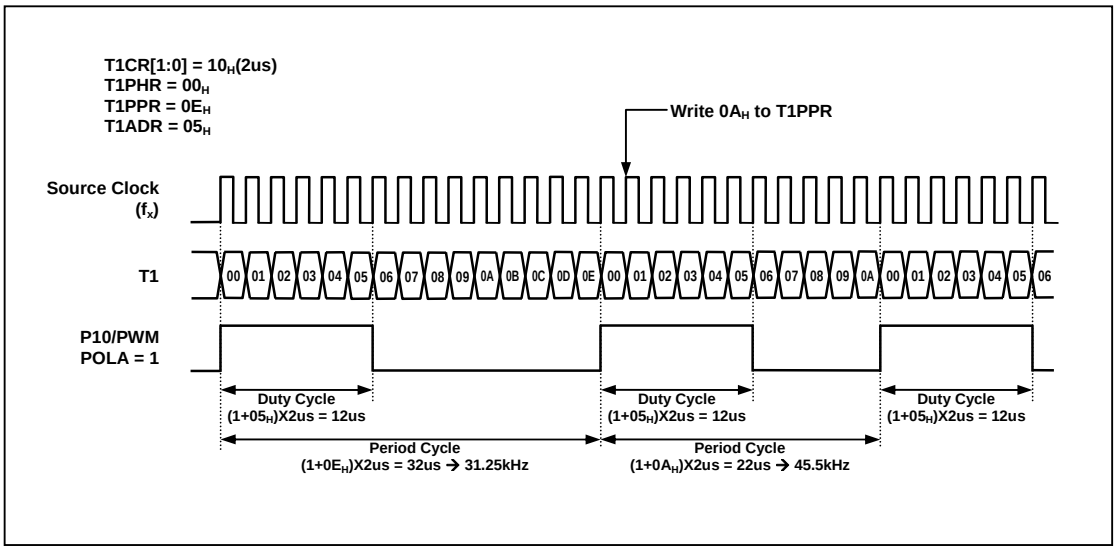

Figure 11-17 Example of Changing the Period in Absolute Duty Cycle at 4Mhz

Update period & duty register value at once

The period and duty of PWM comes to move from temporary registers to PPR (PWM Period Register) and PDR (PWM Duty Register) when always period match occurs. If you want that the period and duty is immediately changed, the UPDT bit in the T1PCR register must set to '1'. It should be noted that it needs the 3 cycle of timer clock for data transfer in the internal clock synchronization circuit. So the update data is written before 3 cycle of timer clock to get the right output waveform.

Phase correction & Frequency correction

Figure 11-21 Example of PWM External Synchronization with BLNKB Input

FORCE Drive ALL ch with A-ch mode

If FORCA bit sets to '1', it is possible to enable or disable all PWM output pins through PWM outputs which occur from A-ch duty counter. It is noted that the inversion outputs of A, B, C channel have the same A-ch output waveform. According to POLA/B/C, it is able to control the inversion of outputs.

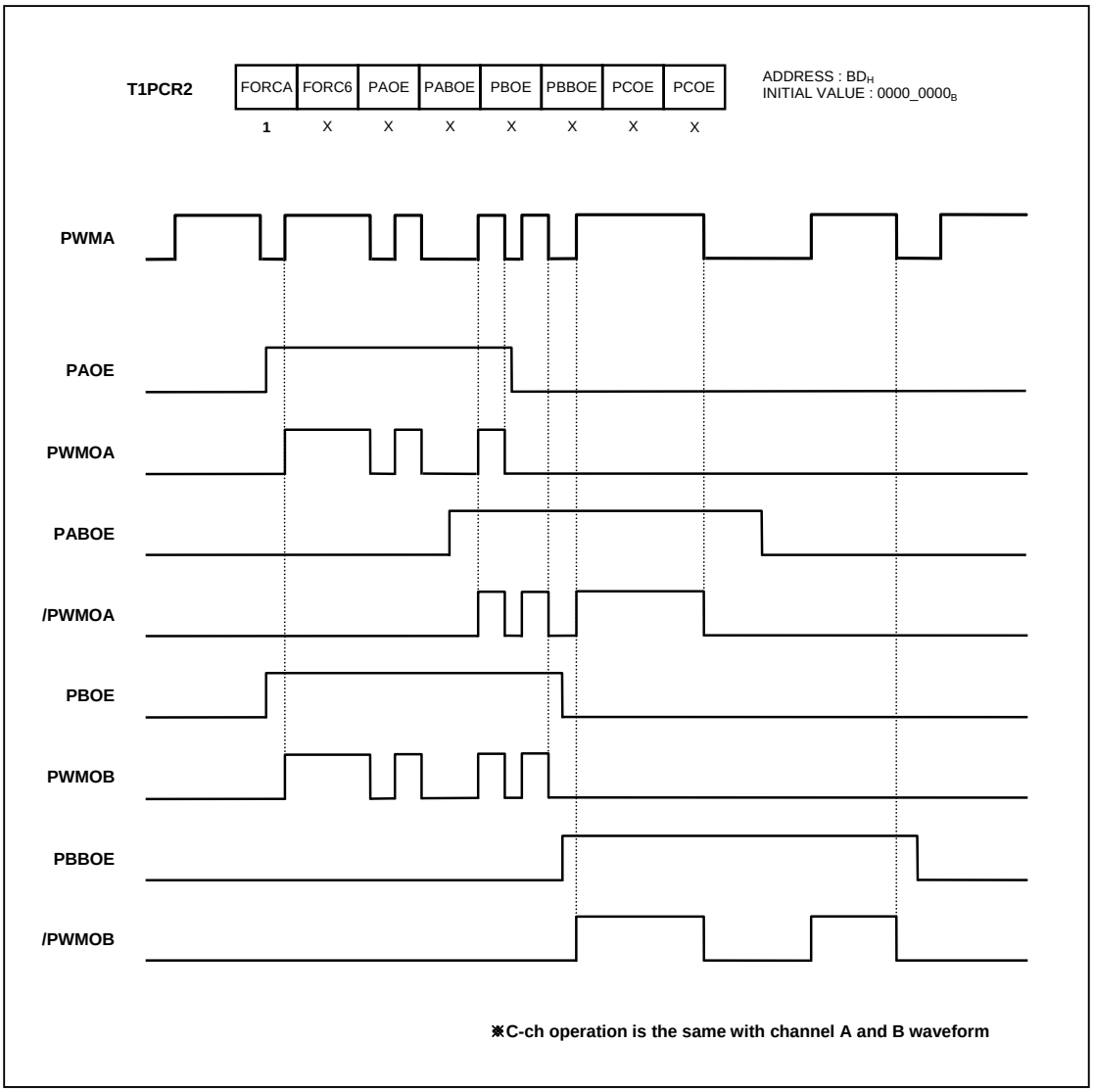

Figure 11-22 Example of Force Drive All-ch with A-ch

generates the Timer 2 interrupt (no timer 3 interrupt). The clock source is selected from T2CK[2:0] and T3CK[1:0] must set 11b and 16BIT bit must set to '1'. The timer 2 is LSB 8-bit, the timer 3 is MSB 8-bit. T2DR must not be 0x00(0x01~0xFF). The 16-bit mode setting is shown as Figure 11-29.

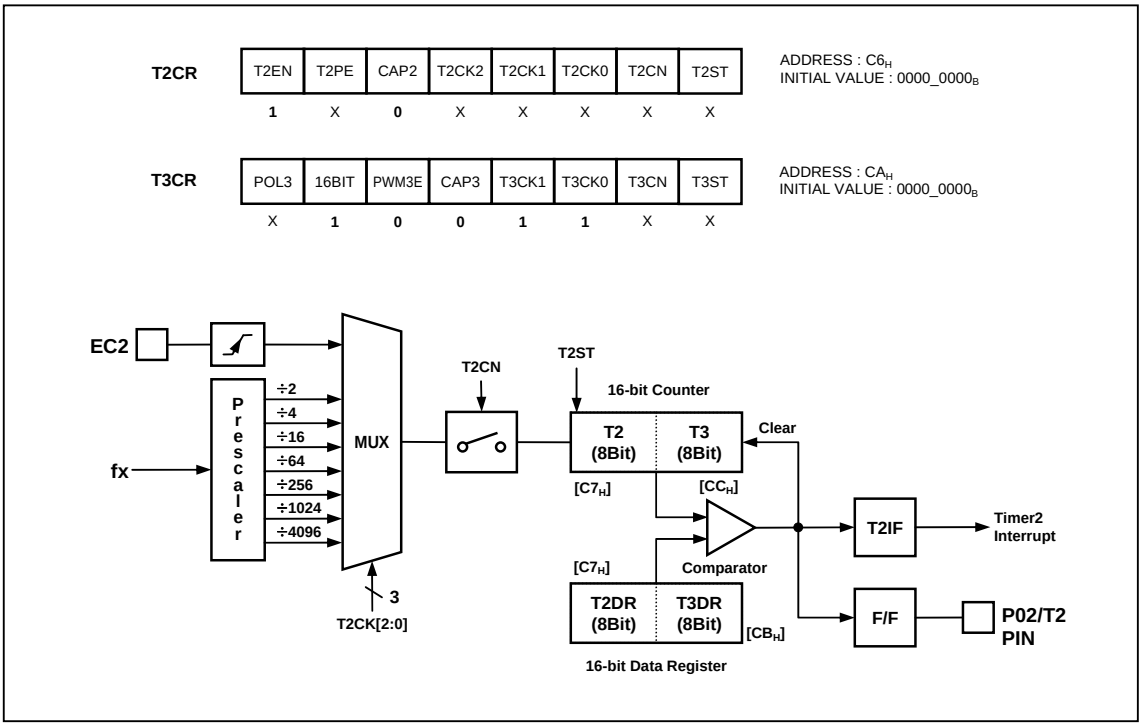

Figure 11-29 16 Bit Timer/Event Counter2, 3 Block Diagram

11.5.2.4 8-Bit Capture Mode

The timer 2, 3 capture mode is set by CAP2, CAP3 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 T 2, 3 and T2DR, T3DR matching time, respectively. The capture result is loaded into CDR2, CDR3. The T2, T3 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 INT2, INT3 function is chosen.

The CDR2, T2 and T2DR are in same address. In the capture mode, reading operation is read the CDR2, not T2DR because path is opened to the CDR2. The CDR3 has the same function.

0	Counter stop
1	Clear Counter and start

T4L (Timer 4 Low Register: Read Case) : CFH

7	6	5	4	3	2	1	0
T4L7	T4L6	T4L5	T4L4	T4L3	T4L2	T4L1	T4L0
R	R	R	R	R	R	R	R

Initial value : 00H

T4L[7:0] T4L Counter

T4LDR (Timer 4 Low Data Register: Write Case) : CFH

7	6	5	4	3	2	1	0
T4LD7	T4LD6	T4LD5	T4LD4	T4LD3	T4LD2	T4LD1	T4LD0
W	W	W	W	W	W	W	W

Initial value : FFH

T4LD[7:0] T4L Compare

LCDR4 (Low Capture 4 Data Register: Read Case) : CFH

7	6	5	4	3	2	1	0
LCDR47	LCDR46	LCDR45	LCDR44	LCDR43	LCDR42	LCDR41	LCDR40
R	R	R	R	R	R	R	R

Initial value : 00H

LCDR4[7:0] T4L Capture data

T4H (Timer 4 High Register: Read Case) : D5H

7	6	5	4	3	2	1	0
T4H7	T4H6	T4H5	T4H4	T4H3	T4H2	T4H1	T4H0
R	R	R	R	R	R	R	R

Initial value : 00H

T4H[7:0] T4H Counter Period

T4HDR (Timer 4 High Data Register: Write Case) : D5H

7	6	5	4	3	2	1	0
T4HD7	T4HD6	T4HD5	T4HD4	T4HD3	T4HD2	T4HD1	T4HD0
W	W	W	W	W	W	W	W

Initial value : FFH

T4HD[7:0] T4H Compare

HDR4 (High Capture 4 Data Register: Read Case) : D5H

7	6	5	4	3	2	1	0
HCDR47	HCDR46	HCDR45	HCDR44	HCDR43	HCDR42	HCDR41	HCDR40
R	R	R	R	R	R	R	R

Initial value : 00H

11.7.3 Clock Generation

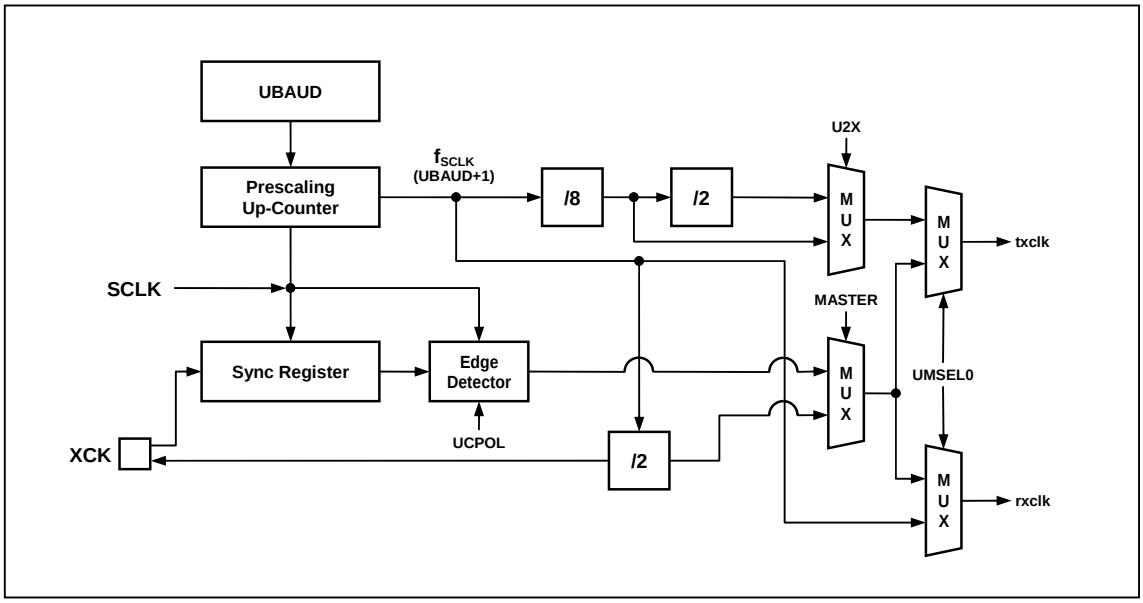

Figure 11-40 Clock Generation Block Diagram

The Clock generation logic generates the base clock for the Transmitter and Receiver. The USART supports four modes of clock operation and those are Normal Asynchronous, Double Speed Asynchronous, Master Synchronous and Slave Synchronous. The clock generation scheme for Master SPI and Slave SPI mode is the same as Master Synchronous and Slave Synchronous operation mode. The UMSELn bit in UCTRLx1 register selects between asynchronous and synchronous operation. Asynchronous Double Speed mode is controlled by the U2X bit in the UCTRLx2 register. The MASTER bit in UCTRLx2 register controls whether the clock source is internal (Master mode, output port) or external (Slave mode, input port). The XCK pin is only active when the USART operates in Synchronous or SPI mode.

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

Table 11-14 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	$\text{Baud Rate} = \frac{\text{fSCLK}}{2(\text{UBAUDx} + 1)}$

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

11.7.8.3 Parity Generator

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

11.7.8.4 Disabling Transmitter

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

11.7.9 USART Receiver

The USART Receiver is enabled by setting the RXE bit in the UCTRLx1 register. When the Receiver is enabled, the normal pin operation of the RXD pin is overridden by the USART as the serial input pin of the Receiver. The baud-rate, mode of operation and frame format must be set before serial reception. If synchronous or spi operation is used, the clock on the XCK pin will be used as transfer clock. If USART operates in spi mode, SS pin is used as SS input pin in slave mode or can be configured as SS output pin in master mode. This can be done by setting SPISS bit in UCTRLx3 register.

11.7.9.1 Receiving Rx data

When USART is in synchronous or asynchronous operation mode, the Receiver starts data reception when it detects a valid start bit (LOW) on RXD pin. Each bit after start bit is sampled at pre-defined baud-rate (asynchronous) or sampling edge of XCK (synchronous), and shifted into the receive shift register until the first stop bit of a frame is received. Even if there's 2^{nd} stop bit in the frame, the 2^{nd} stop bit is ignored by the Receiver. That is, receiving the first stop bit means that a complete serial frame is present in the receiver shift register and contents of the shift register are to be moved into the receive buffer. The receive buffer is read by reading the UDATAx register.

If 9-bit characters are used (USIZE[2:0] = 7) the ninth bit is stored in the RX8 bit position in the UCTRLx3 register. The 9^{th} bit must be read from the RX8 bit before reading the low 8 bits from the UDATAx register. Likewise, the error flags FE, DOR, PE must be read before reading the data from UDATAx register. This is because the error flags are stored in the same FIFO position of the receive buffer.

11.7.9.2 Receiver flag and interrupt

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

The Receive Complete (RXC) flag indicates whether there are unread data present in the receive buffer. This flag is set when there are unread data in the receive buffer and cleared when the receive

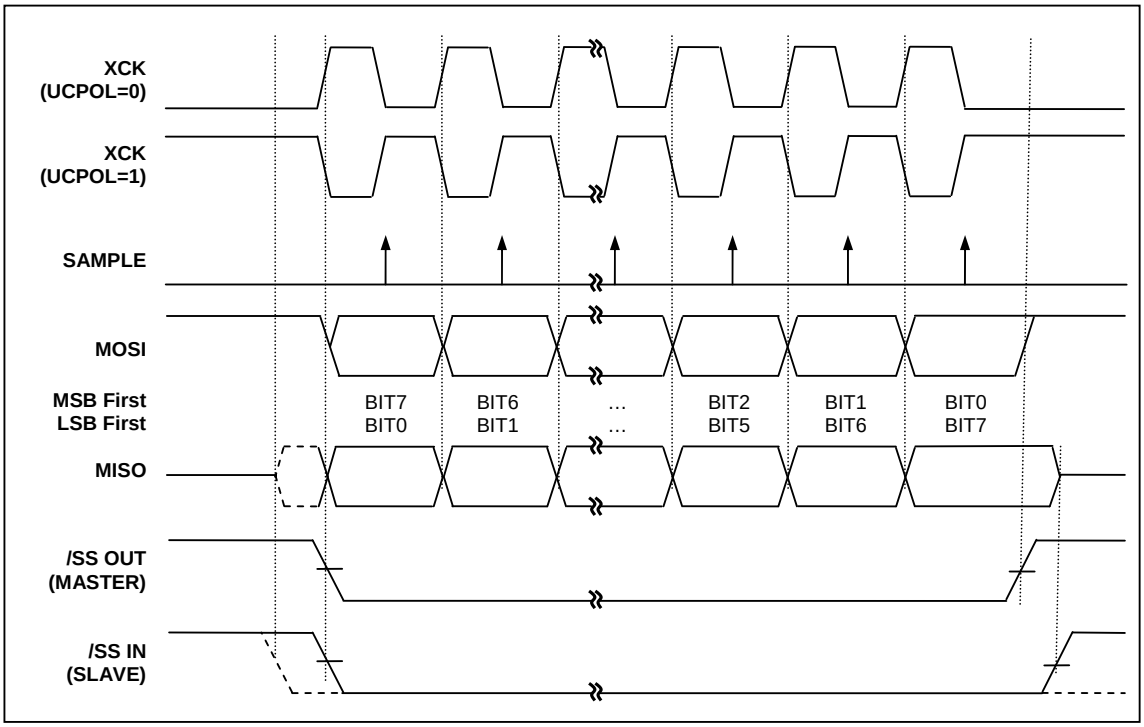

Figure 11-47 SPI Clock Formats when UCPHA=1

When UCPHA=1, the slave begins to drive its MISO output when SS goes active low, but the data is not defined until the first XCK edge. The first XCK edge shifts the first bit of data from the shifter onto the MOSI output of the master and the MISO output of the slave. The next XCK edge causes both the master and slave to sample the data bit value on their MISO and MOSI inputs, respectively. At the third XCK edge, the USART shifts the second data bit value out to the MOSI and MISO output of the master and slave respectively. When UCPHA=1, the slave's SS input is not required to go to its inactive high level between transfers.

Because the SPI logic reuses the USART resources, SPI mode of operation is similar to that of synchronous or asynchronous operation. An SPI transfer is initiated by checking for the USART Data Register Empty flag (UDRE=1) and then writing a byte of data to the UDATA Register. In master mode of operation, even if transmission is not enabled (TXE=0), writing data to the UDATA register is necessary because the clock XCK is generated from transmitter block.

11.7.11 Register Map

Table 11-16 Register Map

Name	Address	Dir	Default	Description
UCTRL01	E2H	R/W	00H	USART Control 1 Register 0
UCTRL02	E3H	R/W	00H	USART Control 2 Register 0
UCTRL03	E4H	R/W	00H	USART Control 3 Register 0
USTAT0	E5H	R	80H	USART Status Register 0
UBAUD0	E6H	R/W	FFH	USART Baud Rate Generation Register 0
UDATA0	E7H	R/W	FFH	USART Data Register 0
UCTRL11	FAH	R/W	00H	USART Control 1 Register 1
UCTRL12	FBH	R/W	00H	USART Control 2 Register 1
UCTRL13	FCH	R/W	00H	USART Control 3 Register 1
USTAT1	FDH	R	80H	USART Status Register 1
UBAUD1	FEH	R/W	FFH	USART Baud Rate Generation Register 1
UDATA1	FFH	R/W	FFH	USART Data Register 2

11.7.12 USART Register description

USART module consists of USART Control 1 Register (UCTRLx1), USART Control 2 Register (UCTRLx2), USART Control 3 Register (UCTRLx3), USART Status Register (USTATx), USART Data Register (UDATAx), and USART Baud Rate Generation Register (UBAUDx).

11.7.13 Register description for USART

UCTRLx1 (USART Control 1 Register) E2H, FAH

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

Initial value : 00_H

UMSEL[1:0] Selects operation mode of USART

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

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

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

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

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

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

0	LSB First
1	MSB First

UCPOL Selects polarity of XCK in synchronous or spi mode

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

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

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

1 XCK is active while any frame is on transferring.

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

0 SS pin is normal GPIO or other primary function

1 SS output to other slave device

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

0 1 Stop Bit

1 2 Stop Bit

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

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

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

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

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

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

USTAT (USART Status Register) E5H

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

I2CSR (I^2C Status Register) : DBH

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

Initial value : 00H

GCALL This bit has different meaning depending on whether I^2C is master or slave. Note 1)
When I^2C is a master, this bit represents whether it received AACK (Address ACK) from slave.
When I^2C is a slave, this bit is used to indicate general call.
0 No AACK is received (Master mode)
1 AACK is received (Master mode)
0 Received address is not general call address (Slave mode)
1 General call address is detected (Slave mode)

TEND This bit is set when 1-Byte of data is transferred completely. Note 1)
0 1 byte of data is not completely transferred
1 1 byte of data is completely transferred

STOP This bit is set when STOP condition is detected. Note 1)
0 No STOP condition is detected
1 STOP condition is detected

SSEL This bit is set when I^2C is addressed by other master. Note 1)
0 I^2C is not selected as slave
1 I^2C is addressed by other master and acts as a slave

MLOST This bit represents the result of bus arbitration in master mode. Note 1)
0 I^2C maintains bus mastership
1 I^2C has lost bus mastership during arbitration process

BUSY This bit reflects bus status.
0 I^2C bus is idle, so any master can issue a START condition
1 I^2C bus is busy

TMODE This bit is used to indicate whether I^2C is transmitter or receiver.
0 I^2C is a receiver
1 I^2C is a transmitter

RXACK This bit shows the state of ACK signal.
0 No ACK is received
1 ACK is generated at ninth SCL period

Note 1) These bits can be source of interrupt.

When an I^2C interrupt occurs except for STOP interrupt, the SCL line is hold LOW. To release SCL, write arbitrary value to I2CSR. When I2CSR is written, the TEND, STOP, SSEL, LOST, RXACK bits are cleared.

	0	No detection
	1	Detection

EXTRF External Reset flag bit. The bit is reset by writing ‘0’ to this bit or by Power ON reset.

0	No detection
1	Detection

WDTRF Watch Dog Reset flag bit. The bit is reset by writing ‘0’ to this bit or by Power ON reset.

0	No detection
1	Detection

OCDRF On-Chip Debug Reset flag bit. The bit is reset by writing ‘0’ to this bit or by Power ON reset.

0	No detection
1	Detection

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

0	No detection
1	Detection

BODLS[1:0] BOD level Voltage

BODLS1	BODLS0	Description
0	0	1.6V
0	1	2.5V
1	0	3.6V
1	1	4.2V

BODEN BOD operation

0	BOD disable
1	BOD enable

FESR (Flash and EEPROM Status Register) : ECH

7	6	5	4	3	2	1	0
PEVBSY	VFYGOOD	-	-	ROMINT	WMODE	EMODE	VMODE
R	R/W	R	R	R/W	R	R	R

Initial value : 80H

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

0 Busy (Operation processing)

1 Complete Operation

VFYGOOD Auto-verification result flag.

0 Auto-verification fails

1 Auto-verification successes

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

FEARL (Flash and EEPROM address low Register) : F2H

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

Initial value : 00H

ARL[7:0] Flash 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
W	W	W	W	W	W	W	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
W	W	W	W	W	W	W	W

Initial value : 00H

ARH[7:0] Flash and EEPROM address high

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

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

15.3.2 Data EEPROM Memory Map

Data EEPROM memory uses 512-byte of EEPROM. It is read by byte and written by byte or page. One page is 16-byte. It is mapped to external data memory of 8051

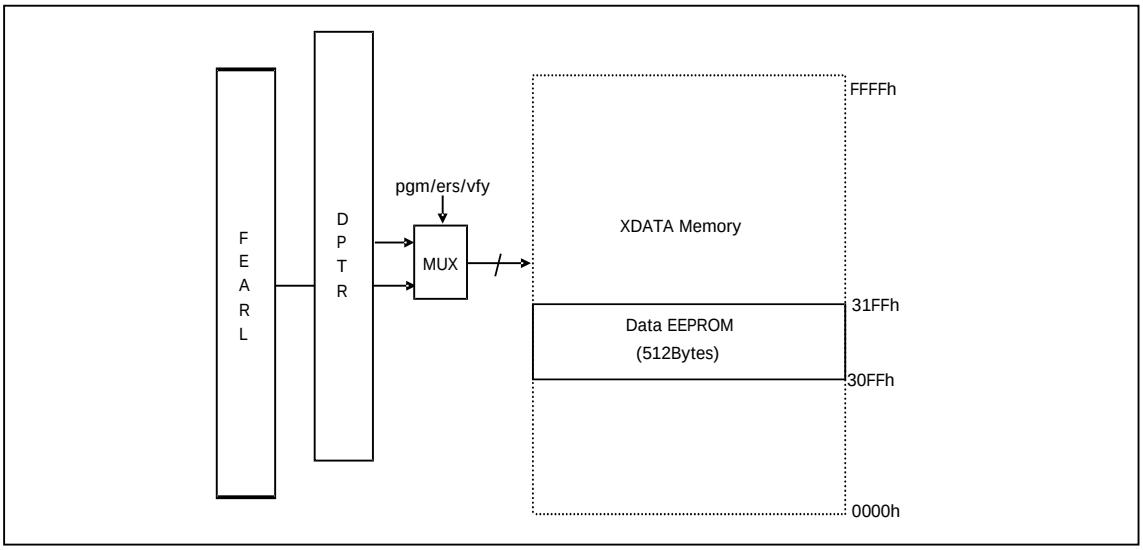

Figure 15-3 Data EEPROM memory map

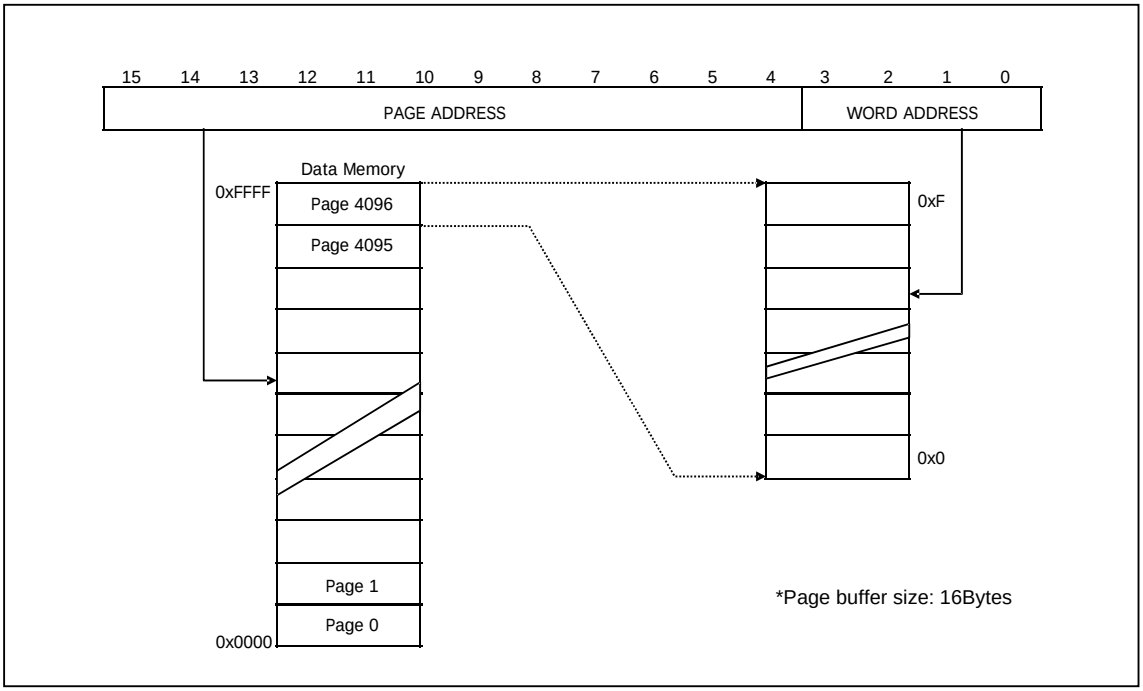

Figure 15-4 Address configuration of data EEPROM

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

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

B. Instructions on how to use the input port.

- Error occur status