**Preview:**

**CHAPTER 01: 8051- MICROCONTROLLER**

In this chapter students will know MCS51 microcontroller family, the architecture of 8051, the functions of different pins of 8051, Special function registers, and memory organization of 8051. Also students can learn different I/O ports, functions of timers and counters, interrupt structure. Then serial communication that is transfer of data serially bit by bit manner using SBUF, TXD, RXD. Power consumption using different modes like Ideal mode, Power Down Operation.

**1.1 Introduction to 8051 family**

In 1981, Intel Corporation introduced an 8-bit microcontroller called the 8051. This microcontroller had 128 bytes of RAM, 4K bytes of on-chip ROM, two timers, one serial port, and four ports (each 8-bits wide) all on a single chip. Intel refers 8051 family of devices as MCS-51. This family is characterized by block diagram given below.

****

Fig 1.1: Block diagram of 8051

The UV-EPROM version of the 8051 is the 8751. The flash ROM version is marketed by many companies including Atmel Corp. and Dallas Semiconductor.

The Atmel Flash 8051 is called AT89C51, while Dallas Semiconductor calls theirs DS89C4xO (DS89C420/430/440). The NV-RAM version of the 8051 made by Dallas Semiconductor is called DS5000.

There are also OTP (one-time-programmable) versions of the 8051 available from different sources. Philips family of 8051 microcontroller include features such as A-to-D converters, D-to-A converters, extended I/O, and both OTP and flash.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Device** | **RAM (bytes)** | **ROM(bytes)** | **16-Bit Timer/Conter** | **No. Vectord Int** | **Full Duplex I/O** |
| 8031 | 128 | None | 2 | 5 | 1 |
| 8032 | 256 | None | 2 | 6 | 1 |
| 8051 | 128 | 4k ROM | 2 | 5 | 1 |
| 8052 | 256 | 8k ROM | 3 | 6 | 1 |
| 8751 | 128 | 4k EPROM | 2 | 5 | 1 |
| 8752 | 256 | 8k EPROM | 3 | 6 | 1 |
| AT89C51 | 128 | 4k Flash Memory | 2 | 5 | 1 |
| AT89C52 | 256 | 8k Flash Memory | 3 | 6 | 1 |

## Table 1.1: Different microcontroller along with their configuration

## 1.2 8051 Microcontroller

## 1.2.1 Salient Features:

## It requires 5V power supply and has in built oscillator circuit.

## 8051 microcontroller has 8 bit ALU and hence 8 bit data bus.

## It has 16 bit address bus and hence can interface(216) 64 KB of RAM and ROM.

## It has on chip RAM of 128bytes

## It has on chip ROM of 4 KB used to store program.

## It has on chip programmable serial port UART

## It has 2 timers each of 16 bits.

## It has 8 bit bi-directional ports used for interfacing Input/ Output devices.

## It has two power saving modes of operation

## It has built in Boolean processor used for bit level logical operations.

## 1.2.2 Pin Diagram

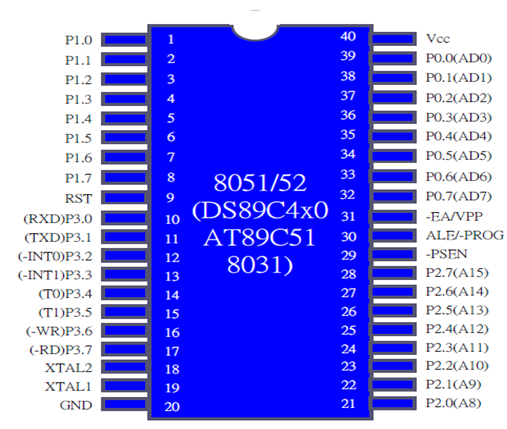


Fig1.2: Pin Diagram

##### **Description of Pin Diagram :**

A total of 32 pins are rserved for the four ports and remaining pins are Vcc, GND, XTAL1, XTAL2, RST, -EA are used by all members of 8051 and 8031 families.

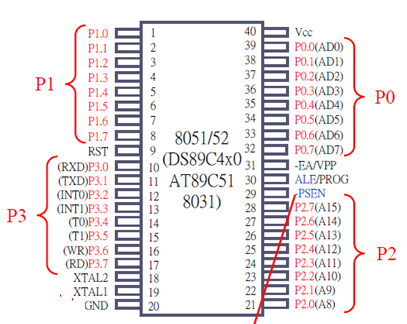


Fig1.3: Port Pins

A total of 32 pins are reserved for the four ports P0, P1, P2, P3, where each port takes 8 pins. Port Latches specify the value to be output on the specific output port or the value read from the specific input port.

Ports are bit addressable i.e each bit can be changed. First bit has the same address as the register. Example: P1 has address 90H in the SFR, so P1.7 or address 97H refer to the same bit. The 8051 contains four

I/O ports. All four ports are bidirectional i.e can act as input as well as output. Each port is SFR (Special Function Registers P0 through P3) which works like a latch, an output driver and an input buffer.

Both output driver and input buffer of Port 0 and output driver of Port 2 are used for accessing external memory.

**Port 0:-** Port 0 is also designated as AD0-AD7, allowing it to be used for both address and data.

**Port 1:-** Port 1 is dedicated solely for I/O interfacig.

**Port 2:-** Port 2 has 8 pins (P2.1-P2.7). Port-2 is used for accesing higher address byte from external memory or a normal input/output port.

**Port 3:-** Port 3 has 8 pin (P3.0-P3.7) . Port-3 pins have alternate functions. Port three can be used forserial communication, timers, interrupts interfacing.

###### **Other pins:**

###### **XTAL1 And XTAL2(Pin 18 and 19)**

The 8051 has an on-chip oscillator but requires an external clock to run it A quartz crystal oscillator is connected to inputs XTAL1 (pin19) and XTAL2 (pin18). The quartz crystal oscillator also needs two capacitors of 30 pF value.

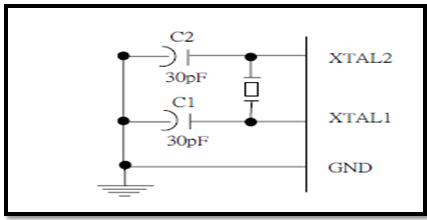


Figure 1.4: Using On Chip Oscillator

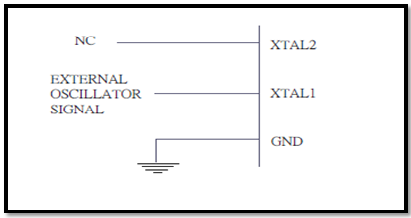


Figure 1.5 Using External oscillator

If you use a frequency source other than a crystal oscillator, such as a TTL oscillator. It will be connected to XTAL1and XTAL2 is left unconnected as shown in fig.

The speed of 8051 is the maximum oscillator frequency connected to XTAL ex. A 12-MHz chip must be connected to a crystal with 12 MHz frequency or less. We can observe the frequency on the XTAL2 pin using oscilloscope.

##### **8051 Clock and Instruction Cycle**

In 8051, one instruction cycle consists of twelve (12) clock cycles. Instruction cycle is sometimes called as Machine cycle in some books.

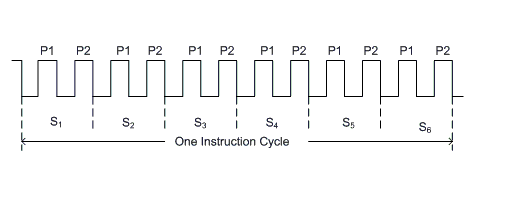
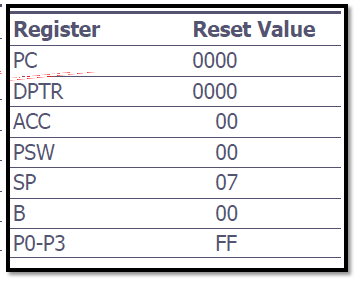


Fig1.6: One instruction Cycle

###### **RST**

RESET pin is an input and is active high (normally low). Upon applying a high pulse to this pin, the microcontroller will reset and terminate all activities. This is often referred to as a power-on reset. Activating a power-on reset will cause all values in the registers to be lost. Some registers values on reset operation are shown here.



###### Table 1.2: Values in the registers after reset

###### **EA “external access’’:**

EA “external access’’ is an input pin and must be connected to Vcc or GND The 8051 family members all come with on-chip ROM to store programs -EA pin is connected to V cc The 8031 and 8032 family

members do not have on-chip ROM, so code is stored on an external ROM and is fetched by 8031/32 -EA pin must be connected to GND to indicate that the code is stored externally **PSEN:-** “program store enable’’ is an output pin.

**ALE:-**“address latch enable”, is an output pin and is active high. ALE indicates if P0 has address or data. When ALE=0, it provides data D0-D7. When ALE=1, it has address A0-A7.

**Vcc:-**Provides +5V supply voltage to the chip

## Basic 8051 Architecture

## 

## Fig 1.7: Architecture of 8051

### Memory Organization

* + - 1. **Internal RAM**

##### 128 bytes of Internal RAM Structure (lower address space)

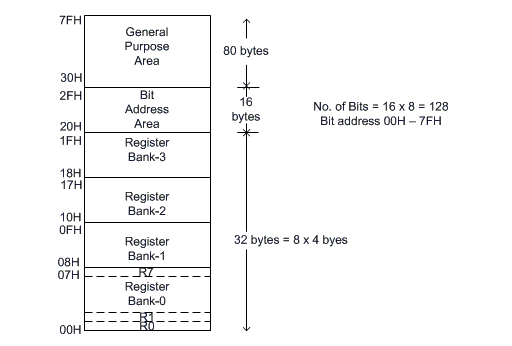


Fig 1.8: Internal RAM Structure

The lower 32 bytes are divided into 4 separate banks. Each register bank has 8 registers of one byte each. A register bank is selected depending upon two bank select bits in the PSW register. Next 16bytes are bit addressable. In total, 128bits (16X8) are available in bitaddressable area. Each bit can be accessed and modified by suitable instructions. The bit addresses are from 00H (LSB of the first byte in 20H) to 7FH (MSB of the last byte in 2FH). Remaining 80bytes of RAM are available for general purpose.

* **Register Bank in detail (R0-R7)**

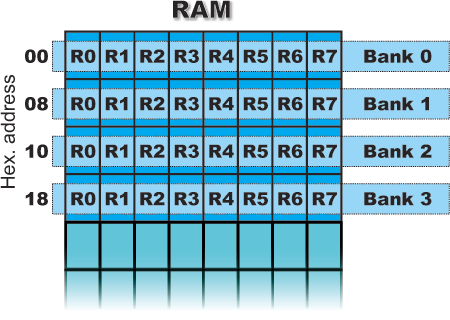


Fig 1.8.1: Register bank (R0- R7)

This is a common name for 8 general-purpose registers (R0, R1, R2 ...R7). Even though they are not true SFRs, they deserve to be discussed here because of their purpose. They occupy 4 banks within RAM. Similar to the accumulator, they are used for temporary storing variables and intermediate results during operation. Which one of these banks is to be active depends on two bits of the PSW Register. Active bank is a bank the registers of which are currently used.

The following example best illustrates the purpose of these registers. Suppose it is necessary to perform some arithmetical operations upon numbers previously stored in the R registers: (R1+R2) - (R3+R4). Obviously, a register for temporary storing results of addition is needed.

**This is how it looks in the program:**

**MOV A,R3 ;** Means: move number from R3 into accumulator

**ADD A,R4 ;** Means: add number from R4 to accumulator

(result remains in accumulator)

**MOV R5,A ;** Means: temporarily move the result from

accumulator into R5

**MOV A,R1 ;** Means: move number from R1 to accumulator

**ADD A,R2 ;** Means: add number from R2 to accumulator

**SUBB A,R5 ;** Means: subtract number from R5 (there are

R3+R4)

##### **Internal Data Memory and Special Function Register (SFR) Map**

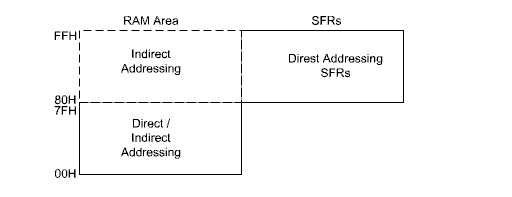


Fig 1.9: Internal Data Memory Map

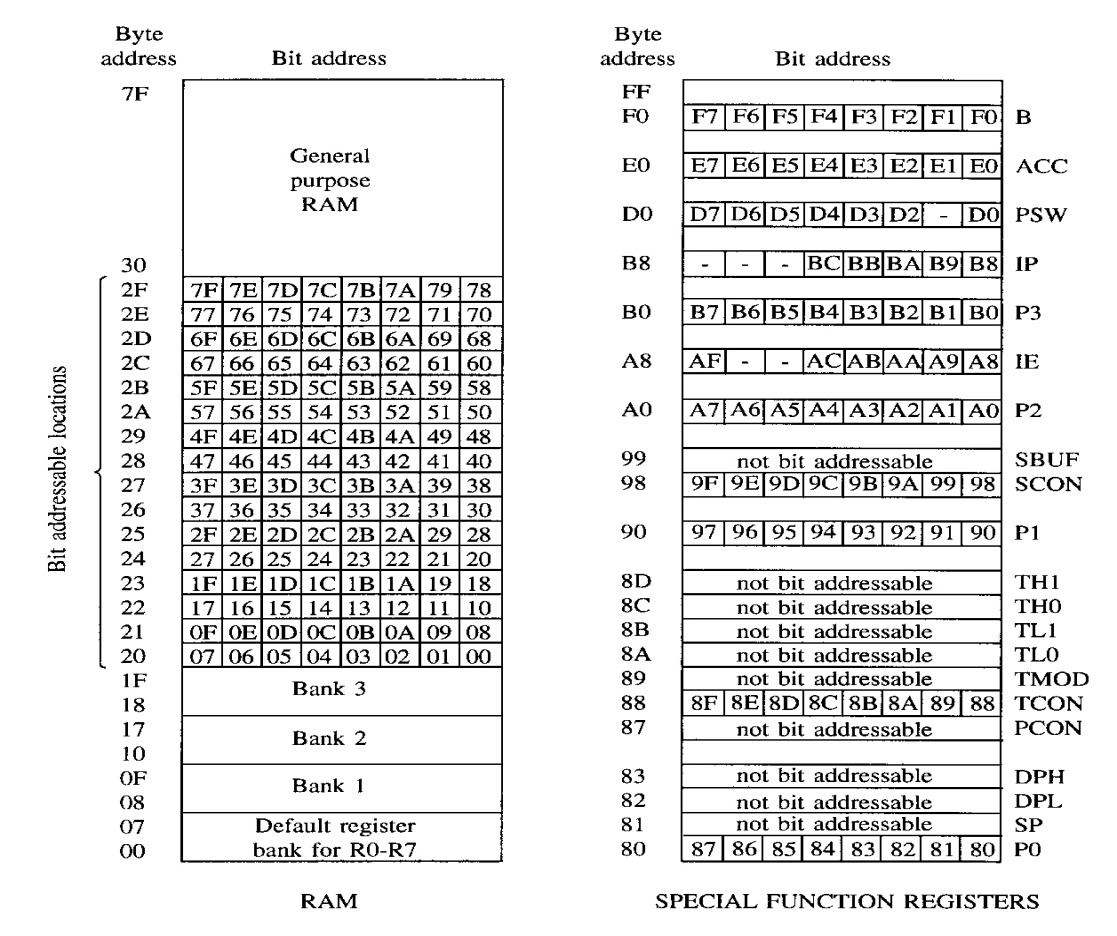


Fig 1.9.1 Internal and External 128 Bytes of RAM

The special function registers (SFRs) are mapped in the upper 128 bytes of internal data memory address.

Hence there is an address overlap between the upper 128 bytes of data RAM and SFRs. Please note that the upper 128 bytes of data RAM are present only in the 8052 family. The lower128 bytes of RAM (00H - 7FH) can be accessed both by direct or indirect addressing while the upper 128 bytes of RAM (80H - FFH) are accessed by indirect addressing.The SFRs (80H - FFH) are accessed by direct addressing only. This feature distinguishes the upper 128 bytes of memory from the SFRs, as shown in fig 1.4.

##### **SFR Map**

The set of Special Function Registers (SFRs) contains important registers such as Accumulator, Register B, I/O Port latch registers, Stack pointer, Data Pointer, Processor Status Word (PSW) and various control registers. Some of these registers are bit addressable (they are marked with a \* in the diagram below). The detailed map of various registers is shown in the following table.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| F8H |  |  |  |  |  |  |  |  |
| F0H | B\* |  |  |  |  |  |  |  |
| E8H |  |  |  |  |  |  |  |  |
| E0H | ACC\* |  |  |  |  |  |  |  |
| D8H |  |  |  |  |  |  |  |  |
| D0H | PSW\* |  |  |  |  |  |  |  |
| C8H | (T2CON)\* |  | (RCAP2L) | (RCAP2H) | (TL2) | (TH2) |  |  |
| C0H |  |  |  |  |  |  |  |  |
| B8H | IP\* |  |  |  |  |  |  |  |
| B0H | P3\* |  |  |  |  |  |  |  |
| A8H | IE\* |  |  |  |  |  |  |  |
| A0H | P2\* |  |  |  |  |  |  |  |
| 98H | SCON\* | SBUF |  |  |  |  |  |  |
| 90H | P1\* |  |  |  |  |  |  |  |
| 88H | TCON\* | TMOD | TL0 | TL1 | TH0 | TH1 |  |  |
| 80H | P0\* | SP | DPL | DPH |  |  |  | PCON |

Table 1.3: SFRs along with their address

###### **ACC – Accumulator**

Commonly used for move and arithmetic instructions. Can be referred to in several ways: Implicitly in op-codes. Referred to as ACC (or A) for instructions that allow specifying a register. By its SFR address 0E0H. Operates in a similar manner to the 8085 accumulator.

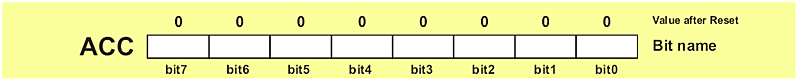


Fig 1.10: ACC Register

**Bit addressable.** ACC.2 means bit 2 of the ACC register.

###### **B Register:-**

Commonly used as a temporary register, much like a 9th R register. Used by two op-codes MUL AB, div AB B register holds the second operand and will hold part of the result Upper 8 bits of the multiplication result Remainder in case of division.

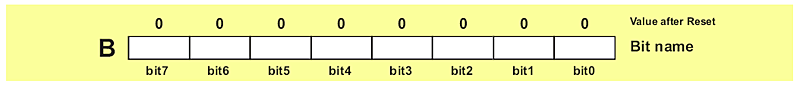


Fig 1.11 B register

Can also be accessed through its SFR address of 0F0H. Bit addressable

###### **DPL and DPH Registers:-**

2 8-bit registers that can be combined into a 16-bit DPTR – Data Pointer. Used by commands that access external memory Also used for storing 16bit values mov DPTR, #data16 ; setup DPTR with 16bit ext address movx A, @DPTR ; copy mem[DPTR] to A

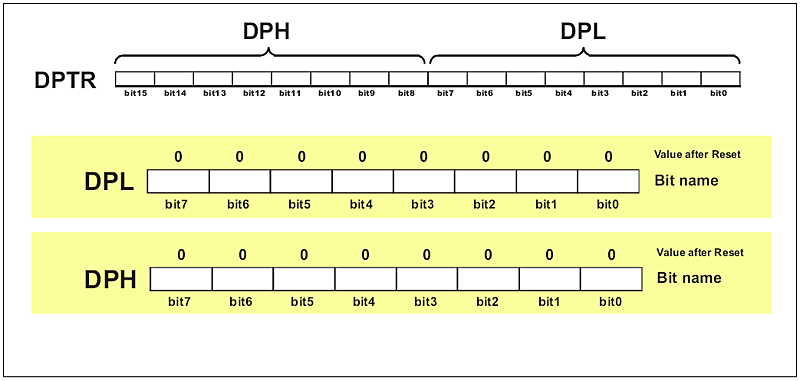


Fig 1.12: DPTR Register

Can be accessed as 2 separate 8-bit registers if needed. DPTR is useful for string operations and Look-Up-Table (LUT) operations.

###### **SP Register:-**

SP is the stack pointer. SP points to the last used location of the stack. Push operation will first increment SP and then copy data. Pop operation will first copy data and then decrement SP. In 8051, stack grows

upwards (from low memory to high memory) and can be in the internal RAM only. On power-up, SP points to 07H. Register banks 2,3,4 (08H to 1FH) form the default stack area.

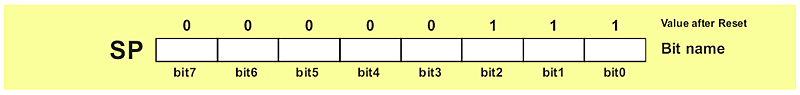


Fig 1.13: SP Register

Stack can be relocated by setting SP to the upper memory area in 30H to 7FH. mov SP, #32H

###### **PSW Register:-**

Program Status Word is a “bit addressable” 8-bit register that has all the flags.

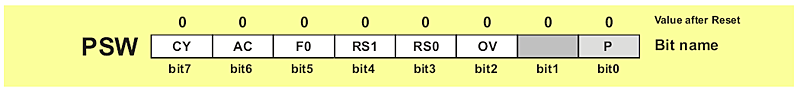


Fig 1.14: PSW register

|  |  |  |
| --- | --- | --- |
| **Symbol** | **Position** | **Function** |
| CY | PSW.7 | Carry Flag |
| AC | PSW.6 | Auxiliary Carry Flag. For BCD Operations |
| F0 | PSW.5 | Flag 0. Available to the user for general purposes. |
| RS1 | PSW.4 | Register bank select bits. Set by software to determine which register bank is being used. |
| RS2 | PSW.3 |
| OV | PSW.2 | Overflow Flag |
| - | PSW.1 | Not used |
| P | PSW.0 | Parity Flag. Even Parity. |

Table 1.4: Bitwise description of PSW

Following table shows how this register bank can be selected using RS1 and RS2 values.

|  |  |  |
| --- | --- | --- |
| **RS1** | **RS2** | **SPACE IN RAM** |
| 0 | 0 | Bank0 00h-07h |
| 0 | 1 | Bank1 08h-0Fh |
| 1 | 0 | Bank2 10h-17h |
| 1 | 1 | Bank3 18h-1Fh |

Table 1.5 RS1 and RS2 value to select register bank

### I/O PORTS, Timer/Counter, Interrupt Structure

###### **1.2.6.1 I/O PORTS**

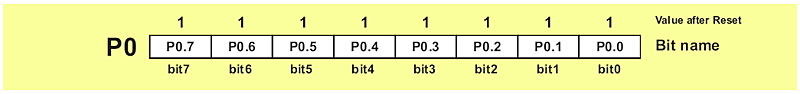


Fig 1.15: port with its bit value

If neither external memory nor serial communication system are used then 4 ports within total of 32 input/output pins are available for connection to peripheral environment. Each bit within these ports affects the state and performance of appropriate pin of the microcontroller. Thus, bit logic state is reflected on appropriate pin as a voltage (0 or 5 V) and vice versa, voltage on a pin reflects the state of appropriate port bit.

As mentioned, port bit state affects performance of port pins, i.e. whether they will be configured as inputs or outputs. If a bit is cleared (0), the appropriate pin will be configured as an output, while if it is set (1), the appropriate pin will be configured as an input. Upon reset and power-on, all port bits are set (1), which means that all appropriate pins will be configured as inputs.

###### **1.2.6.2 Timer/Counters:-**

The 8051 has two timers/counters, they can be used either as, Timers ¯ to generate a time delay or as Eventcounters ¯ to count events happening outside the microcontroller . Both Timer 0 and Timer 1 are 16 bits wide. Since 8051 has an 8-bit architecture, each 16-bits timer is accessed as two separate registers of low byte and high byte. The low byte register is called TL0/TL1 and the high byte register is called TH0/TH1 and accessed like any other register.

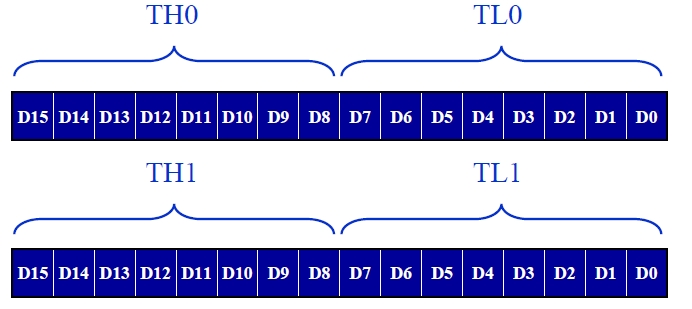


Fig 1.16: Timer THx/TLx

The high and low bytes of the 16-bit counting register(TL0/TH0) for timer/counter T0 and for timer T1 it is TH1 / TL1. In the 8052, one more pair exists (TH2) / (TL2) for the T2 timer. (RCAP2H) and (RCAP2L) exist only in the 8052 and they are copies of the TH2 and TL2 registers. Addres of these are

###### **TMOD: Timer/Counter Mode Control Register (not bit addressable)**

Both timers 0 and 1 use the same register, called TMOD (timer mode), to set the various timer operation modes. TMOD is a 8-bit register. The lower 4 bits are for Timer 0. The upper 4 bits are for Timer 1.

In each case, the lower 2 bits are used to set the timer mode, the upper 2 bits to specify the operation

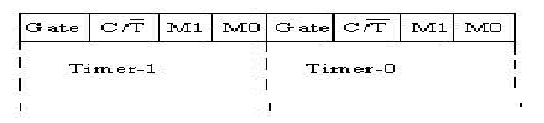


Fig1.16: TMOD register

Gate: This is an OR Gate enabled bit which controls the effect of on START/STOP of Timer. It is set to one ('1') by the program to enable the interrupt to start/stop the timer.

If TR1/0 in TCON is set and signal on pin is high (i.e GATE=1) then the timer starts counting using either internal clock (timer mode) or external pulses (counter mode). If GATE=0 and TR1/0 in TCON is set then it is software control.

**:-** It is used for the selection of Counter/Timer mode.

* Mode Select Bits:

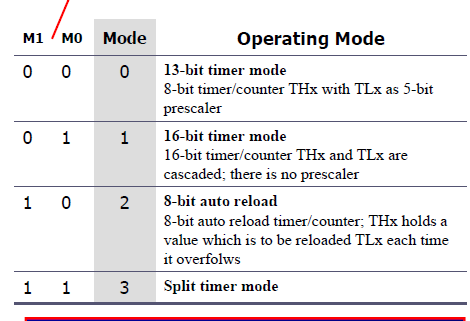


Fig 1.16 Timer mode bits

###### **Timer control (TCON)**

TCON is bit addressable. The address of TCON is 88H. It is partly related to Timer and partly to interrupt. The bits of TCON are as shown in fig below.

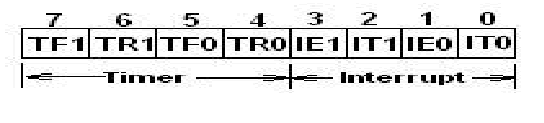


Fig 1.17: TCON bits

**The meaning various bits of TCON are as follows.**

* **TF1:-**Timer1 overflow flag. It is set when timer rolls from all 1s to 0s. It is cleared when processor vectors to execute ISR located at address 001BH.
* **TR1: -**Timer1 run control bit. Set to 1 to start the timer / counter.
* **TF0: -** Timer0 overflow flag. (Similar to TF1)
* **TR0:-**Timer0 run control bit. IE1 : Interrupt1 edge flag. Set by hardware when an external interrupt edge is detected. It is cleared when interrupt is processed.
* **IE0:-** Interrupt0 edge flag. (Similar to IE1)
* **IT1:-** Interrupt1 type control bit. Set/ cleared by software to specify falling edge / low level triggered external interrupt.
* **IT0:**- Interrupt0 type control bit. (Similar to IT1)

As mentioned earlier, Timers can operate in four different modes. They are as follows

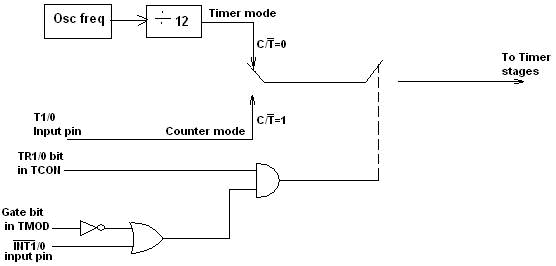


Fig 1.17 Timer/Counter Control Logic

###### **Timer Modes :**

###### **Timer Mode-0:**

In this mode, the timer is used as a 13-bit UP counter as follows.

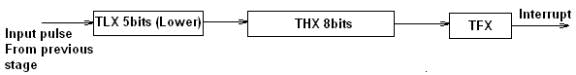


Fig. 1.18 Operation of Timer on Mode-0

The lower 5 bits of TLX and 8 bits of THX are used for the 13 bit count. Upper 3 bits of TLX are ignored. When the counter rolls over from all 0's to all 1's, TFX flag is set and an interrupt is generated. The input pulse is obtained from the previous stage. If TR1/0 bit is 1 and Gate bit is 0, the counter continues counting up. If TR1/0 bit is 1 and Gate bit is 1, then the operation of the counter is controlled by input. This mode is useful to measure the width of a given pulse fed to input.

###### **Timer Mode-1:**

This mode is similar to mode-0 except for the fact that the Timer operates in 16-bit mode.

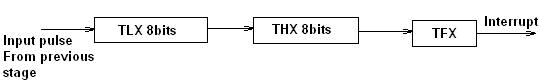


Fig. 1.19 Operation of Timer on Mode-1

###### **Timer Mode-2: (Auto-Reload Mode) :**

This is counter/timer operation. Counting is performed in TLX while THX stores a constant value. In this mode when the timer overflows a 8 bit i.e. TLX becomes FFH, it is fed with the value stored in THX. For example if we load THX with 50H then the timer in mode 2 will count from 50H to FFH. After that 50H is again reloaded. This mode is useful in applications like fixed time sampling.

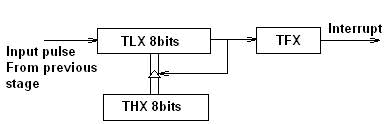


Fig. 1.20 Operation of Timer on Mode-2

###### **Timer Mode-3:**

Timer 1 in mode-3 simply holds its count. The effect is same as setting TR1=0. Timer0 in mode-3 establishes TL0 and TH0 as two separate counters.

Control bits TR1 and TF1 are used by Timer-0 (higher 8 bits) (TH0) in Mode-3 while TR0 and TF0 are available to Timer-0 lower 8 bits(TL0).

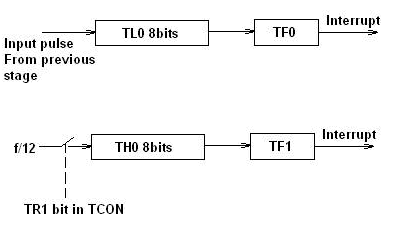


Fig. 1.21 Operation of Timer on Mode-3

###### **1.2.6.3 Interrupt Structure:**

* **8051 provides 5 vectored interrupts. They are -**

2. TF0
3. TF1
4. RI/TI
5. RESET

* Out of these, are external interrupts whereas
* Timer and Serial port interrupts are generated internally.
* The external interrupts could be negative edge triggered or low level triggered.
* All these interrupt, when activated, set the corresponding interrupt flags.
* Except for serial interrupt, the interrupt flags are cleared when the processor branches to the Interrupt Service Routine (ISR).
* The external interrupt flags are cleared on branching to Interrupt Service Routine (ISR), provided the interrupt is negative edge triggered.
* For low level triggered external interrupt as well as for serial interrupt, the corresponding flags have to be cleared by software by the programmer.

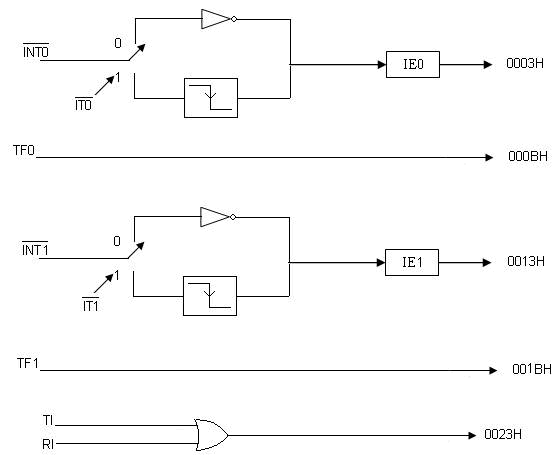


Fig 1.22: 8051 Interrupt Details

Each of these interrupts can be individually enabled or disabled by 'setting' or 'clearing' the corresponding bit in the IE.

(Interrupt Enable Register) SFR. IE contains a global enable bit EA which enables/disables all interrupts at once.

###### **Interrupt Enable register (IE): Address: A8H**

http://nptel.ac.in/courses/117104072/micro/lecture9/images/lec9_1_clip_image002_0002.gif

Fig 1.23: Interrupt Enable register

EX0 http://nptel.ac.in/courses/117104072/micro/lecture9/images/lec9_1_clip_image014.gifhttp://nptel.ac.in/courses/117104072/micro/lecture9/images/lec9_1_clip_image002.gif    interrupt (External)   enable bit

ET0 http://nptel.ac.in/courses/117104072/micro/lecture9/images/lec9_1_clip_image014.gifTimer-0 interrupt enable bit

EX1 http://nptel.ac.in/courses/117104072/micro/lecture9/images/lec9_1_clip_image014.gifhttp://nptel.ac.in/courses/117104072/micro/lecture9/images/lec9_1_clip_image004.gif   interrupt (External) enable bit

ET1 http://nptel.ac.in/courses/117104072/micro/lecture9/images/lec9_1_clip_image014.gif Timer-1 interrupt enable bit

ES http://nptel.ac.in/courses/117104072/micro/lecture9/images/lec9_1_clip_image014.gif Serial port interrupt enable bit

ET2 http://nptel.ac.in/courses/117104072/micro/lecture9/images/lec9_1_clip_image014.gif Timer-2 interrupt enable bit

EA http://nptel.ac.in/courses/117104072/micro/lecture9/images/lec9_1_clip_image014.gif Enable/Disable

**Setting '1' http://nptel.ac.in/courses/117104072/micro/lecture9/images/lec9_1_clip_image014.gif Enable the corresponding interrupt**

**Setting '0' http://nptel.ac.in/courses/117104072/micro/lecture9/images/lec9_1_clip_image014.gif Disable the corresponding interrupt**

* **Priority level structure:**

Each interrupt source can be programmed to have one of the two priority levels by setting (high priority) or clearing (low priority) a bit in the IP (Interrupt Priority) Register .

A low priority interrupt can itself be interrupted by a high priority interrupt, but not by another low priority interrupt.

* **Interrupt Priority register  (IP)**

###### http://nptel.ac.in/courses/117104072/micro/lecture9/images/lec9_1_clip_image002_0005.gif

Fig 1.24: Interrupt priority register

**PT2:-** Used for future

**PS:** Serial Port interrupt priority bit

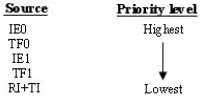
**PT1:-** Timer 1 priority interrupt

**PX1:-** External interrupt 1 priority

**PT0:** -Timer 0 priority interrupt

**PX0:** -External interrupt 0 priority

* Each interrupt source can be programmed to have one of the two priority levels by setting (high priority) or clearing (low priority) a bit in the IP (Interrupt Priority) Register .
* A low priority interrupt can itself be interrupted by a high priority interrupt, but not by another low priority interrupt.
* If two interrupts of different priority levels are received simultaneously, the request of higher priority level is served.
* If the requests of the same priority level are received simultaneously, an internal polling sequence determines which request is to be serviced.
* Thus, within each priority level, there is a second priority level determined by the polling sequence, as follows.



    '0'    http://nptel.ac.in/courses/117104072/micro/lecture9/images/lec9_1_clip_image014.gif   low priority

    '1'   http://nptel.ac.in/courses/117104072/micro/lecture9/images/lec9_1_clip_image014.gif   high priority

### 1.2.7 Serial Port Interface

###### **SBUF Register**

Serial Port Data Buffer. 2 registers at the same location. It is an 8-bit register used solely for serial communication. For a byte data to be transferred via the TxD line, it must be placed in the SBUF register first. The moment a byte is written into SBUF, it is framed with the start and stop bits and transferred serially via the TxD line.

SBUF holds the byte of data when it is received by 8051 RxD line. When the bits are received serially via RxD, the 8051 deframes it by eliminating the stop and start bits, making a byte out of the data received, and then placing it in SBUF.

* **SCON:-**

SCON is an 8-bit register used to program the start bit, stop bit, and data bits of data framing, among other things.

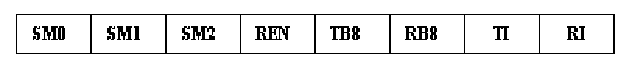


Fig 1.25: SCON register

SM0 : SCON.7 Serial port mode specifier

SM1 : SCON.6 Serial port mode specifier

SM2 : SCON.5 Used for multiprocessor communication

REN : SCON.4 Set/cleared by software to enable/disable reception

TB8 : SCON.3 Not widely used

RB8 : SCON.2 Not widely used

TI : SCON.1 Transmit interrupt flag. Set by HW at the begin of the stop bit mode 1. And cleared by SW

RI : SCON.0 Receive interrupt flag. Set by HW at the begin of the stop bit mode 1. And cleared by SW

Note: Make SM2, TB8, and RB8 =0

* **SM0, SM1**

They determine the framing of data by specifying the number of bits per character, and the start and stop bits

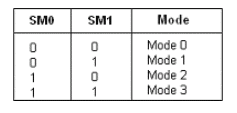


Fig 1.26: SM0 & SM1 for selecting serial mode

* **SM2: -**This enables the multiprocessing capability of the 8051
* **REN: -** (receive enable) - It is a bit-a dressable register. When it is high, it allows 8051 to receive data on Rxd pin. If low, the receiver is disable.
* **TI: -** (transmit interrupt) - When 8051 finishes the transfer of 8-bit character. It raises TI flag to indicate that it is ready to transfer another byte. TI bit is raised at the beginning of the stop bit.
* **RI: -** (receive interrupt) - When 8051 receives data serially via RxD, it gets rid of the start and stop bits and places the byte in SBUF register. It raises the RI flag bit to indicate that a byte has been received and should be picked up before it is lost. RI is raised halfway through the stop bit.

###### **Power Mode control Register (PCON):-**

Register PCON controls processor power down, sleep modes and serial data baud rate. Only one bit of PCON is used with respect to serial communication. The seventh bit (b7)(SMOD) is used to generate the baud rate of serial communication.

**Address: 87H .**

http://nptel.ac.in/courses/117104072/micro/lecture11/images/pcon.gif

Fig 1.26 PCON register

SMOD**:-** Serial baud rate modify bit   
GF1**:-** General purpose user flag bit 1   
GF0**:-** General purpose user flag bit 0   
 PD**:-** Power down bit   
 IDL**:-** Idle mode bit

**1.2.8 Serial Data Transmission Modes:**

**1] Serial Mode-0:**

In this mode, the serial port works like a shift register and the data transmission works synchronously with a clock frequency of fosc /12. Serial data is received and transmitted through RXD. 8 bits are transmitted/ received at a time. Pin TXD outputs the shift clock pulses of frequency fosc /12, which is connected to the external circuitry for synchronization. The shift frequency or baud rate is always 1/12 of the oscillator frequency

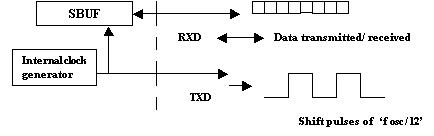


Fig 1.27: Serial Mode 0 working mechanism

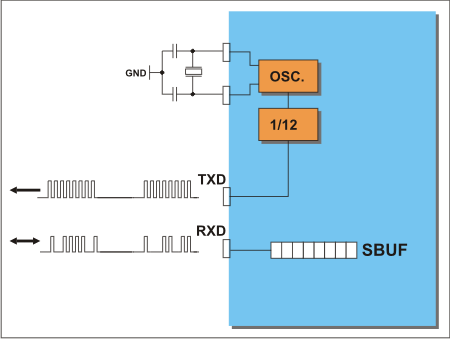


Fig 1.28: Serial mode 0 TXD and RXD connection

**2] Serial Mode-1 (standard UART mode) :**

In mode-1, the serial port functions as a standard Universal Asynchronous Receiver Transmitter (UART) mode. 10 bits are transmitted through TXD or received through RXD. The 10 bits consist of one start bit (which is usually '0'), 8 data bits (LSB is sent first/received first), and a stop bit (which is usually '1'). Once received, the stop bit goes into RB8 in the special function register SCON. The baud rate is variable.

The following figure shows the way the bits are transmitted/ received.

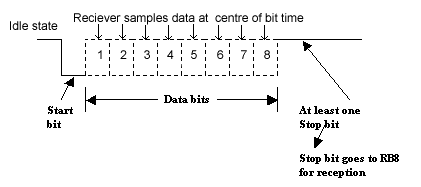


Fig1.29: Transmission of bits with start and stop bits

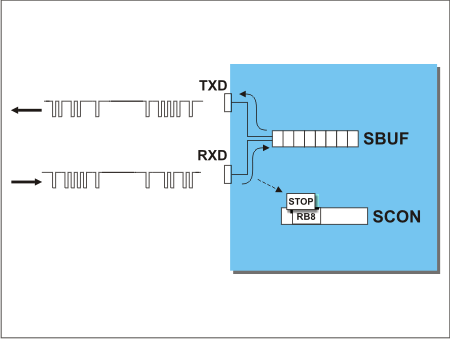


Fig 1.30 Serial mode 1 TXD and RXD connection

**TRANSMIT** - Data transmit is initiated by writing data to the SBUF register. End of data transmission is indicated by setting the TI bit of the SCON register.

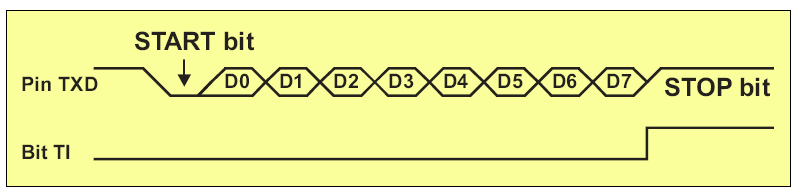


Fig 1.31: Transmission of bits with start and stop bits

**Receive: -** The START bit (logic zero (0)) on the RXD pin initiates data receive. The following two conditions must be met: bit REN=1 and bit RI=0. Both of them are stored in the SCON register. The RI bit is automatically set upon data reception is complete.

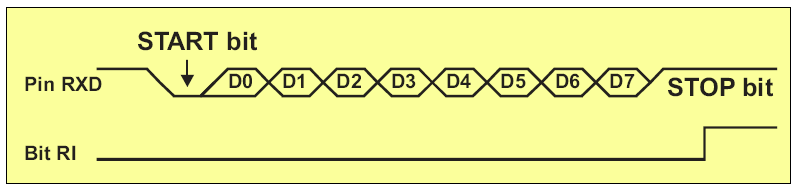


Fig 1.32: Reception of bits with start and stop bits

Bit time = 1/fbaud

In receiving mode, data bits are shifted into the receiver at the programmed baud rate. The data word (8-bits) will be loaded to SBUF if the following conditions are true. 1. RI must be zero. (i.e., the previously received byte has been cleared from SBUF) 2. Mode bit SM2 = 0 or stop bit = 1. After the data is received and the data byte has been loaded into SBUF, RI becomes one.

Mode:**-**1 baud rate generation:

Timer-1 is used to generate baud rate for mode-1 serial communication by using overflow flag of the timer to determine the baud frequency. Timer-1 is used in timer mode-2 as an auto-reload 8-bit timer. The data rate is generated by timer-1 using the following formula.

http://nptel.ac.in/courses/117104072/micro/lecture11/images/lec11_1_clip_image002_0000.gif

Where,

SMOD is the 7th bit of PCON register fosc is the crystal oscillator frequency of the microcontroller It can be noted that fosc/ (12 X [256- (TH1)]) is the timer overflow frequency in timer mode-2, which is the auto-reload mode. If timer-1 is not run in mode-2, then the baud rate is,

http://nptel.ac.in/courses/117104072/micro/lecture11/images/lec11_1_clip_image002_0001.gif

**Timer:-**1 can be run using the internal clock, fosc/12 (timer mode) or from any external source via pin T1 (P3.5) (Counter mode).

**Example:-** If standard baud rate is desired, then 11.0592 MHz crystal could be selected. To get a standard 9600 baud rate, the setting of TH1 is calculated as follows. Assuming SMOD to be '0'

http://nptel.ac.in/courses/117104072/micro/lecture11/images/lec11_1_clip_image002_0002.gif

Or

http://nptel.ac.in/courses/117104072/micro/lecture11/images/lec11_1_clip_image002_0003.gif

\Or

http://nptel.ac.in/courses/117104072/micro/lecture11/images/lec11_1_clip_image002_0004.gif

###### **3] Serial Data Mode-2 - Multiprocessor Mode:-**

In this mode 11 bits are transmitted through TXD or received through RXD. The various bits are as follows: a start bit (usually '0'), 8 data bits (LSB first), a programmable 9 th (TB8 or RB8)bit and a stop bit (usually '1'). While transmitting, the 9th data bit (TB8 in SCON) can be assigned the value '0' or '1'. For example, if the information of parity is to be transmitted, the parity bit (P) in PSW could be moved into TB8. On reception of the data, the 9th bit goes into RB8 in 'SCON', while the stop bit is ignored. The baud rate is programmable to either 1/32 or 1/64 of the oscillator frequency.

Baud Rate

f baud = (2 SMOD /64) fosc

###### **4] Serial Mode-3 - Multi processor mode with variable baud rate:-**

In this mode 11 bits are transmitted through TXD or received through RXD. The various bits are: a start bit (usually '0'), 8 data bits (LSB first), a programmable 9 th bit and a stop bit (usually '1').

Mode-3 is same as mode-2, except the fact that the baud rate in mode-3 is variable (i.e., just as in mode-1)

f baud = (2 SMOD /32) \* ( fosc / 12 (256-TH1)) .

**1.2.9 8051 Microcontroller Power Consumption Control:-**

Generally speaking, the microcontroller is inactive for the most part and just waits for some external signal in order to takes its role in a show. This can cause some problems in case batteries are used for power supply. In extreme cases, the only solution is to set the whole electronics in sleep mode in order to minimize consumption. A typical example is a TV remote controller: it can be out of use for months but when used again it takes less than a second to send a command to TV receiver. The AT89S53 uses approximately 25mA for regular operation, which doesn't make it a power-saving microcontroller. Anyway, it doesn’t have to be always like that, it can easily switch the operating mode in order to reduce its total consumption to approximately 40uA. Actually, there are two power-saving modes of operation: Idle and Power Down.

* **Idle mode**

Upon the IDL bit of the PCON register is set, the microcontroller turns off the greatest power consumer- CPU unit while peripheral units such as serial port, timers and interrupt system continue operating normally consuming 6.5mA. In Idle mode, the state of all registers and I/O ports remains unchanged.

In order to exit the Idle mode and make the microcontroller operate normally, it is necessary to enable and execute any interrupt or reset. It will cause the IDL bit to be automatically cleared and the program resumes operation from instruction having set the IDL bit. It is recommended that first three instructions to execute now are NOP instructions. They don't perform any operation but provide some time for the microcontroller to stabilize and prevents undesired changes on the I/O ports.

* **Power Down mode**

By setting the PD bit of the PCON register from within the program, the microcontroller is set to Power down mode, thus turning off its internal oscillator and reduces power consumption enormously. The microcontroller can operate using only 2V power supply in power- down mode, while a total power consumption is less than 40uA. The only way to get the microcontroller back to normal mode is by reset.

While the microcontroller is in Power Down mode, the state of all SFR registers and I/O ports remains unchanged. By setting it back into the normal mode, the contents of the SFR register is lost, but the content of internal RAM is saved. Reset signal must be long enough, approximately 10mS, to enable stable operation of the quartz oscillator.

**QUESTION BANK**

**2Marks Questions:**

1. In the 8051μc, how many pins are designed as I/O ports?
2. What are the features of 8051μc?
3. Which are the different interrupts present in 8051μc?

###### What are the features of RS-232?

1. What is significance of SMOD in deciding baud rate?
2. What is the role of SBUF register in serial data communication?
3. Under what conditions are TI & RI bits raised?

**4M Questions**

**MSBTE Asked Questions**

1. List the ports available in 8051 state alternate functions of port 3.[**Summer 13-14**]
2. Draw the structure of port φ and explain its operation in input mode[**Summer 13-14**]
3. What are the steps to program a timer in mode 1? Explain.
4. What are the steps to program a timer in mode 2? Explain
5. Which are the different powers saving modes available in 8051μc?
6. Explain the steps taken by microcontroller 8051 upon occurrence of an interrupt.
7. Draw and explain IE & IP SFRs of 8051 microcontroller.
8. Why do you need serial communication? What is significance of SMOD in deciding baud rate?
9. Give an overview of 8051 family.
10. Draw the format of SCON and describe its bits functionality
11. How many timer/counters are available in 8051? When they are used? Write its initialization instruction if any.
12. Draw PSW of 8051. State the function of each bit.
13. Draw the architecture of 8051.
14. Enlist the various interrupts of 8051 with their priorities, vector locations and cause of interrupts.
15. State the functions of pins VPP, PSEN, PROG, ALE.

**06 M Question**

1. Draw and describe internal memory organization of 8051 mc.
2. Draw and describe Pin diagram of 8051 mc.[**Summer 13-14**]