UNIT 5
8051 MICROCONTROLLER

Introduction:

In microprocessor based system design, one may note that a standalone microprocessor is not self-sufficient. It requires other components like memory, and input/output devices to form a minimum workable system configuration. Hence the overall microprocessor based system cost is high. So the microprocessor based system is not reliable in some real-time applications, where the cost and size of the system is critical. Hence the Intel Company has introduced microcontroller with on chip memory, and i/o ports.

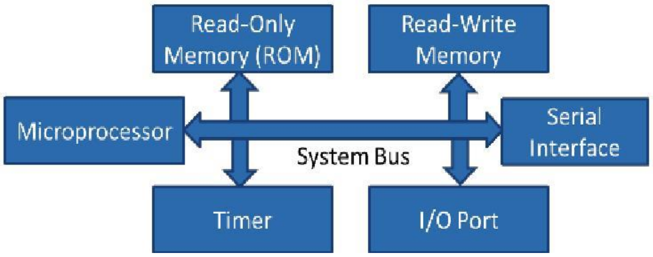
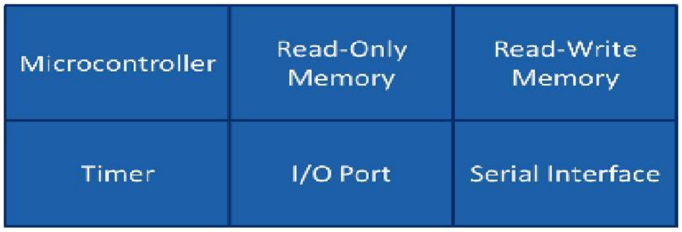
Microcontroller: The microcontroller is a programmable electronic semiconductor chip which has integrated with on chip **CPU, memory, I/O ports**, **serial ports** and **timers/counters**. Hence the microcontroller is referred to as the **system on chip (SOC)** is called microcontroller. And the features of the microcontrollers are given below.

5.1 List the Features of microcontrollers?

1. It is an 8bit microcontroller.
2. 8bit accumulator, 8bit Register and 8bit ALU.
3. On chip RAM 128 bites (data memory).
4. On chip ROM 4 Kbytes (program memory).
5. Two 16bit counter/ timer.
6. A 16 bit DPTR(data pointer)
7. Two levels of interrupt priority.
8. 4 byte bi-directional input/ output port.
9. Power saving mode (on some derivatives).
10. 16bit address bus:-it can access 2^16 memory locations:-64kb (65536) each of RAM and ROM.
11. It is an inclusion of Boolean processing system, have an ability to allow logic operations to be carried out on registers and RAM.
12. 8bit data bus:-it can access 8bit of data in one operation.
13. UART (this serial communication port makes chip to use simply as a serial communication interface).
14. It has four separate Register set. (Each contains 8 Registers (R0 to R7)).
15. Central Processing Unit - Ranging from small and simple 4-bit processors to complex 32- or 64-bit processors.
16. Volatile memory (RAM) for data storage.
17. ROM, EPROM, EEPROM or Flash memory for program and operating parameter storage.
18. Discrete input and output bits, allowing control or detection of the logic state of an individual package pin.
19. Peripherals such as timers, event counters, PWM generators, and watchdog.
20. Clock Generator - often an oscillator for a quartz timing crystal, resonator or RC circuit.
21. Analog-to-digital converters, some include digital-to-analog converters.
22. In-circuit programming and debugging support.

5.2 Compare Microcontroller and Microprocessor :

Microprocessor	Micro Controller
Microprocessor is heart of Computer system.	Micro Controller is a heart of embedded system.
A microprocessor contains ALU, registers, control unit, interrupts and clock circuits.	A microcontroller contains the microprocessor, timers, counters, ADC,DAC,RAM,ROM,UART and clock etc.
Memory, I/O ports, timers, interrupts are not available inside the chip. Hence the circuit becomes large.	Here all are integrated inside the microcontroller chip, hence the circuit is small.

Microprocessors consist of many opcodes for moving data from external memory to the CPU.	Microcontroller contains one (or) two opcodes for moving data from external memory to the CPU.
Cost of the entire system increases	Cost of the entire system is low
No. of bit manipulation instructions are limited.	It has many bit manipulation instructions.
Most of the microprocessors do not have power Saving features.	Most of the micro controllers have power saving Modes like idle mode and power saving mode. This Helps to reduce power consumption even further.
Access time for memory and I/O devices are more	Requires less access time for memory & I/O devices
It is used for general purpose	It is used for specific task.
More flexible in design point of view	Less flexible in design point of view.
Mainly used in personal computers	Used mainly in washing machine, MP3 players
Examples: INTEL 8086,INTEL Pentium series	**Examples:** INTEL8051,89960,PIC16F877
 Read-Only Memory (ROM) Read-Write Memory Microprocessor Serial Interface System Bus Timer I/O Port	 Microcontroller Read-Only Memory Read-Write Memory Timer I/O Port Serial Interface

5.4 Draw the block diagram of a microcomputer and explain the each block:

MICRO COMPUTER:

The basic blocks of a computer are the CPU, I/O devices & memory. Due to the advances in semiconductor technology, it is possible to fabricate CPU on single chip. There are 4 levels of chips SSI,MSI,LSI and VLSI. These levels are based on number of gates provided on the chip.

A microcomputer is a computer that uses a microprocessor as its CPU along with the I/O devices and the memory. All the blocks are wired together through a structure known as bus. A bus contains one or more wires which carries collection of various signals among the devices. It can be unidirectional (or) bidirectional depending on whether the signal flows in one direction or both directions.

"Microcomputer is a small computer that contains a microprocessor as its central processor"

Features of Microcomputer:

- Microcomputers have smaller instruction sets
- less expensive
- smaller size

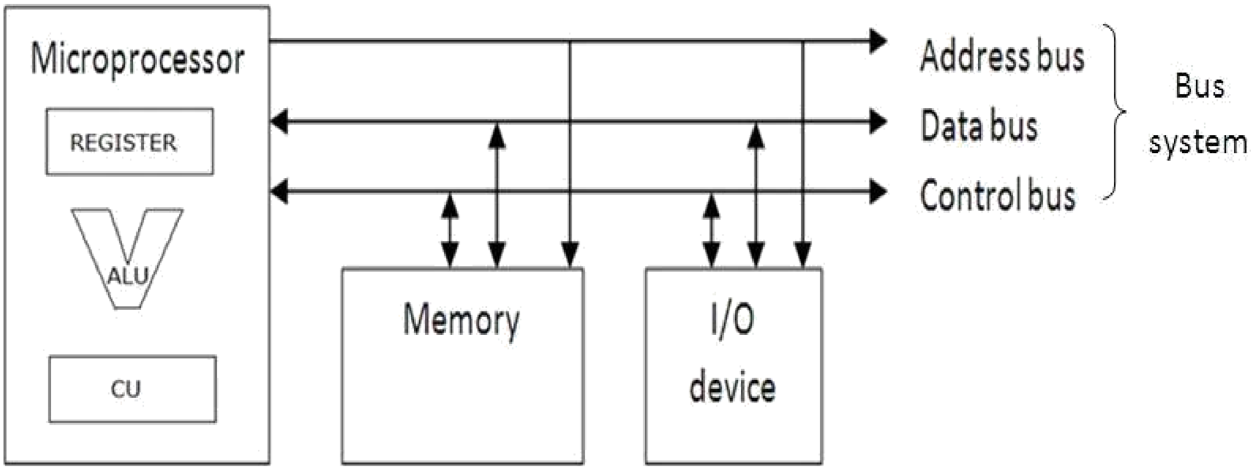


Figure: Block diagram of a Microcomputer system

1. Microprocessor:

(i). Arithmetic Logic Unit (ALU):

` The arithmetic and logical unit (ALU) performs arithmetic operations such as addition, subtraction, multiplication, and or division, and logical operation such as AND, OR, NOT and XOR needed to carry out the instructions.

(ii). Control Unit (CU):

The control unit (CU) is responsible for fetching instructions from main memory and determining their type.

(iii). Registers:

Those are used to store data. After processing the instructions output is again stored in registers for further processing of that output.

2. Memory Unit (MU):

The memory unit (MU) is used to store information such as number or character data. By store we mean that the memory has the ability to hold this information for processing or for outputting at a later time.

Types of Memory:

The memory unit is divided into **primary storage memory** and **secondary storage memory**. Typically, **Primary storage memory** is implemented with semiconductor memories: read-only memory **(ROM)** and random access read/write memory **(RAM)** integrated circuits.

Secondary storage memory is used for long-term storage of information that is used in **disk** and **CD ROM**.

3. I/O device:

(i). Input unit (IU) :

The input unit (IU) is used to given the information to be processed from external input device such as a card reader, keyboard, or switch.

(ii) Output Unit (OU) :

The output unit (OU) is used to output the processed results of computer to the external output devices such as a printer, monitor, 7- segment display, and LED.

4. System Bus:

A bus is a collection of wires used to transmit signals in parallel. According to the purpose, the buses of a microcomputer can be divided into three types: address bus, data bus, and control bus. Three buses are shown are shown in Figure.

(i) Address Bus:

The unidirectional address bus transmits the address signals emitted from CPU to memory and
I/O port.

ii) Data Bus: The signal on the bidirectional data bus is the data either from CPU to memory and I/O or
from memory and I/O to CPU.

iii).Control Bus:

The control bus is used to transmit the control signals such as read, write, and interrupt control signal.

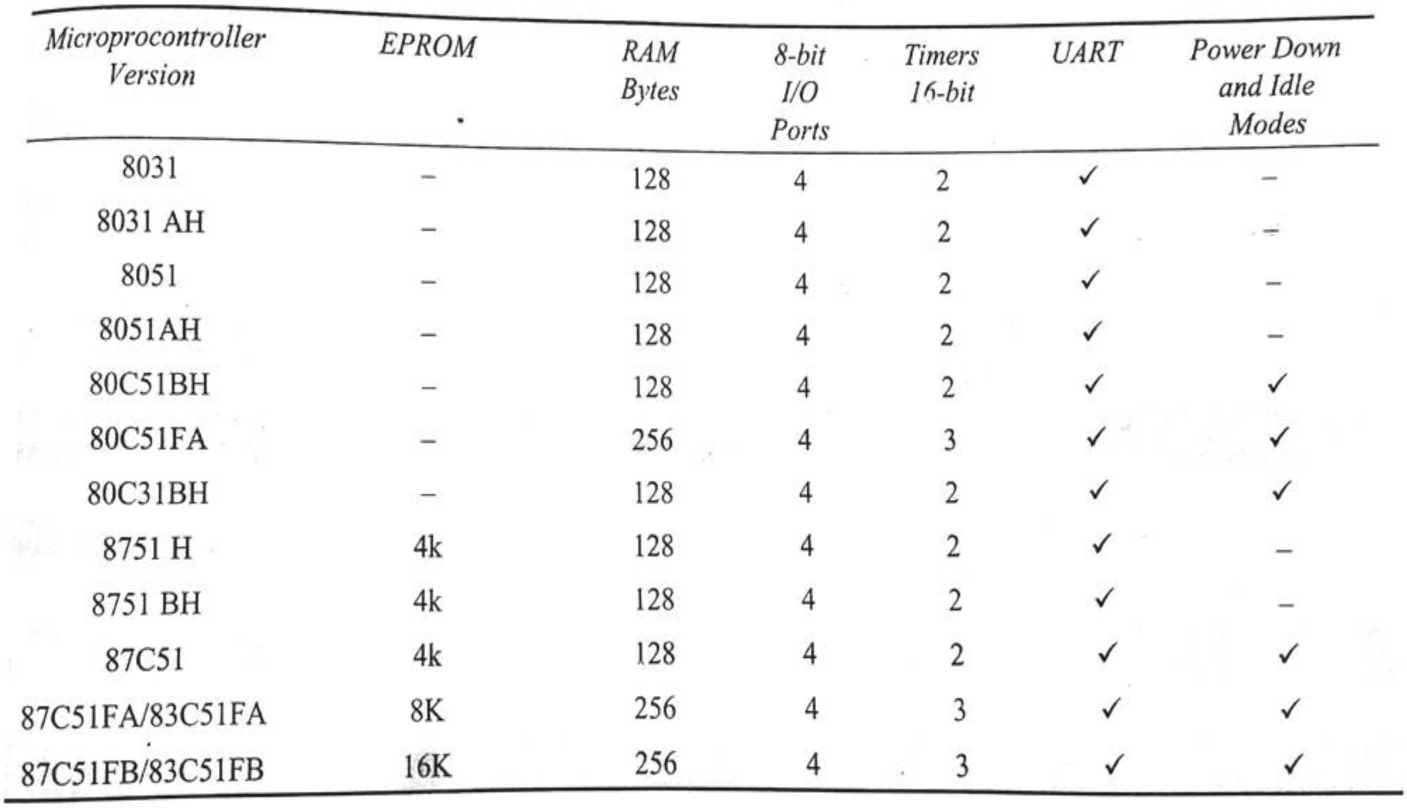
Examples: Accumulator, Program Counter, Instruction Register.

5.4 Give the details of 8051 Microcontroller family chips.

Several manufacturers introduced variety of microcontrollers in the field of digital computers. INTEL was introduced number of families like 8048, 8051, 8096 etc. INTEL 4004 was the first 4-bit processor which appeared in 1971. In 1974 Texas instruments introduced the first microcontroller TMS1000. INTEL 8080 was then introduced in 1974, as the advanced version of 8008 which also appeared in 1972.

TMS 1000	8-bit microcontroller	1974
Intel 8048	8-bit microcontroller	1976
Intel 8031	8 bit (ROM-less)	
Intel 8051	8 bit (Mask ROM)	1980
8096(MCS-96)	16-bit microcontroller	1982
Microchip PIC16C64	8 bit	1985
Motorola 68HC11	8 bit (on chip ADC)	
Intel 80C196	16 bit	1982
Atmel AT89C51	8 bit (Flash memory)	
Microchip PIC 16F877	8 bit (Flash memory + ADC)	

Since 1975 Intel has been producing many different types of microcontrollers. In general each MCU that Intel has made falls into one of the below families. The families differ mostly in instruction sets and architecture. Within each family there are many different CPUs. While each CPU may have the same core the feature set can vary a lot.



Microprocontroller Version	*EPROM*	*RAM Bytes*	*8-bit I/O Ports*	*Timers 16-bit*	*UART*	*Power Down and Idle Modes*
8031	–	128	4	2	✓	–
8031 AH	–	128	4	2	✓	–
8051	–	128	4	2	✓	–
8051AH	–	128	4	2	✓	–
80C51BH	–	128	4	2	✓	✓
80C51FA	–	256	4	3	✓	✓
80C31BH	–	128	4	2	✓	✓
8751 H	4k	128	4	2	✓	–
8751 BH	4k	128	4	2	✓	–
87C51	4k	128	4	2	✓	✓
87C51FA/83C51FA	8K	256	4	3	✓	✓
87C51FB/83C51FB	16K	256	4	3	✓	✓

5.5 Give the functional block diagram of 8051 Microcontroller

Functional block diagram of 8051 micro controller

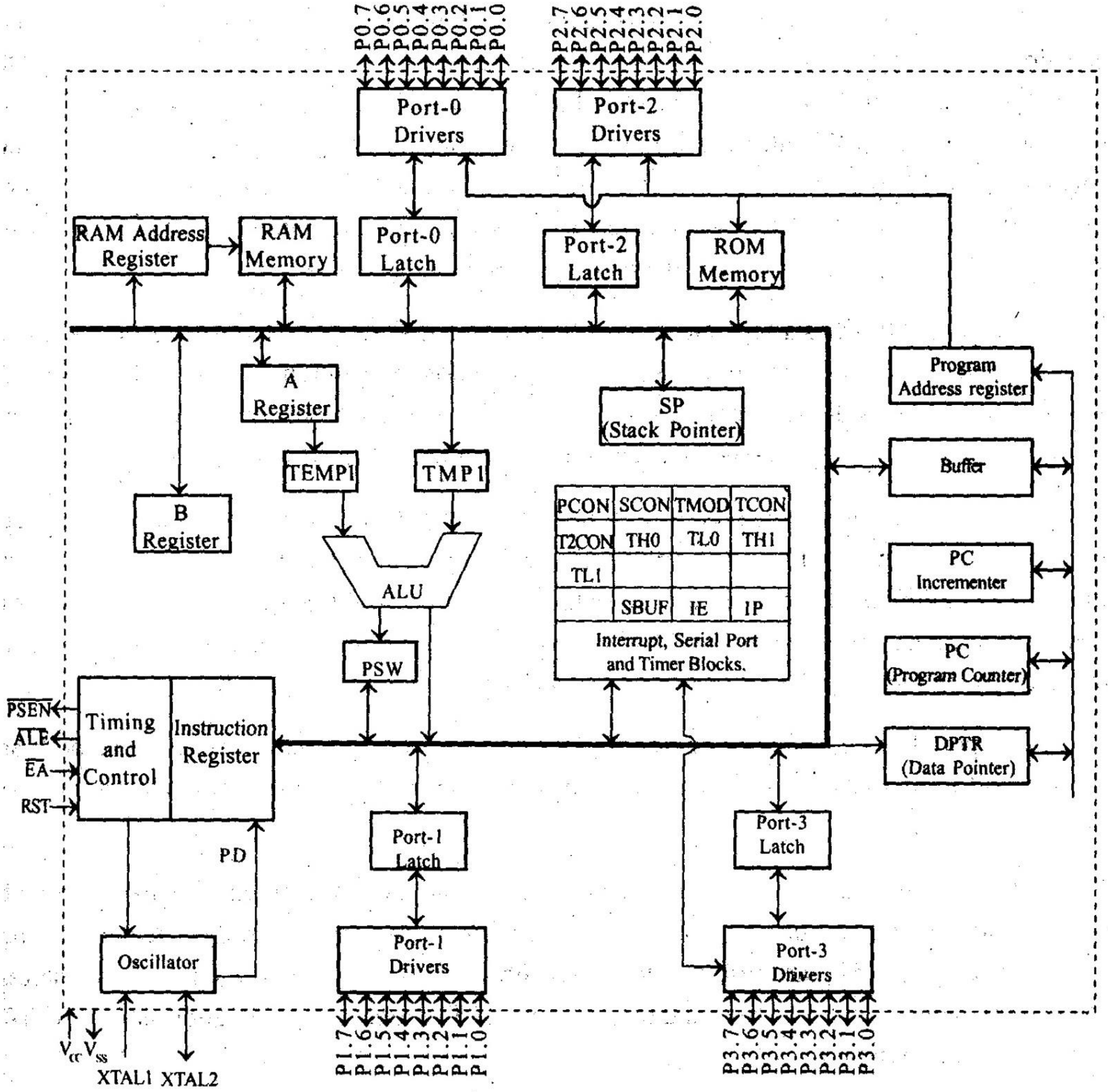
Major components of Intel 8051 microcontroller:

The 8051 microcontroller is an 8-bit microcontroller. Let us see the major components of 8051 microcontroller and their functions.

An 8051 microcontroller has the following 12 major components:

1. ALU (Arithmetic and Logic Unit)
2. PC (Program Counter)
3. Registers
4. PSW (Program Status Word)
5. Timers and counters
6. Serial data communication
7. System bus

8. Four general purpose parallel input/output ports
9. Interrupt control logic with five sources of interrupt
10 Internal RAM and ROM
11. Timing and control unit & oscillator
12. Stack Pointer (SP)
13. Data Pointer (DPTR)



PCON	SCON	TMOD	TCON
T2CON	TH0	TL0	TH1
TL1			
	SBUF	IE	IP

Interrupt, Serial Port and Timer Blocks.

Figure: 8051 Block diagram

Now let us see the functions of each of these components

1.ALU:

- All arithmetic and logical functions are carried out by the ALU.
- Addition, subtraction with carry, and multiplication come under arithmetic operations.
- Logical AND, OR and exclusive OR (XOR) come under logical operations.

2.Program Counter (PC):

- A program counter is a 16-bit register and it has no internal address.
- The basic function of program counter is to fetch from memory the address of the next instruction to be executed.
- The PC holds the address of the next instruction residing in memory and when a command is encountered, it produces that instruction. This way the PC increments automatically, holding the address of the next instruction.

3. Registers :

- Registers are usually known as data storage devices. 8051 microcontroller has 2 registers, namely Register A and Register B.
- Register A serves as an accumulator while Register B functions as a general purpose register.
- These registers are used to store the output of mathematical and logical instructions.

8051 microcontroller also has **7 Special Function Registers (SFRs).** They are:

- Serial Port Data Buffer (SBUF)
- Timer/Counter Control (TCON)
- Timer/Counter Mode Control (TMOD)
- Serial Port Control (SCON)
- Power Control (PCON)
- Interrupt Priority (IP)
- Interrupt Enable Control (IE)

4.Timing & control unit and oscillator:

- Synchronization among internal operations can be achieved with the help of clock circuits which are responsible for generating clock pulses.
- During each clock pulse a particular operation will be carried out, thereby, assuring synchronization among operations.
- For the formation of an oscillator, we are provided with two pins XTAL1 and XTAL2 which are used for connecting a resonant network in 8051 microcontroller device.

The Timing and Control pins are :

i. ALE (Address Latch Enable) - Latches the address signals on Port P0

ii. EA (External Address) - Holds the 4K bytes of program memory

iii. PSEN (Program Store Enable) - Reads external program memory

iv. RST (Reset) - Reset the ports and internal registers upon start up Quartz crystal is used to generate periodic clock pulses.

5. Timers and counters:

A microcontroller has another feature of its internal timer/counter register. Timer registers are used in timing applications for producing time delay to control the external devices in specified times and counter registers are used for counting the external clock events. 8051 consists of two 16-bit timers, T0 & T1.

6. Internal RAM and ROM:

ROM:

- A code of 4K memory is incorporated as on-chip ROM in 8051.
- The 8051 ROM is a non-volatile memory meaning that its contents cannot be altered.

RAM:

- The 8051 microcontroller is composed of 128 bytes of internal RAM.
- This is a volatile memory since its contents will be lost if power is switched off.
- These 128 bytes of internal RAM are divided into 32 working registers which in turn constitute 4 register banks (Bank 0-Bank 3) with each bank consisting of 8 registers (R0 - R7).

7. Four General Purpose Parallel Input/ Output Ports:

The 8051 microcontroller has four 8-bit input/output ports. These are:

i)PORT P0:

When there is no external memory present, this port acts as a general purpose input/output port. In the presence of external memory, it functions as a multiplexed address and data bus(AD_0-AD_7). It performs a dual role.

ii)PORT P1:

This port is used for various interfacing activities. This 8-bit port is a normal I/O port i.e. it does not perform dual functions.

iii)PORT P2:

Similar to PORT P0, this port can be used as a general purpose port when there is no external memory but when external memory is present it works in conjunction with PORT 0 as an address bus. This is an 8-bit port and performs dual functions.

iv) PORT P3:

PORT P3 used for special functions like RXD, TXD, INT0, INT1, T0, T1, RD, WR.

8. Interrupt Control:

- An event which is used to suspend or halt the normal program execution for a temporary period of time in order to serve the request of another program or hardware device is called an interrupt.
- An interrupt can either be an internal or external event which suspends the microcontroller for a while and thereby obstructs the sequential flow of a program.

Types of interrupts:

There are two ways of giving interrupts to a microcontroller:

- one is by sending software instructions
- Other is by sending hardware signals.

The interrupt mechanism keeps the normal program execution in a "put on hold" mode and executes a subroutine program and after the subroutine is executed, it gets back to its normal program execution. This subroutine program is also called an interrupt handler. A subroutine is executed when a certain event occurs.

In **8051, 5 sources of interrupts** are provided. They are:

a) 2 external interrupt sources connected through INT0 and INT1

b) 3 external interrupt sources- serial port interrupt, Timer Flag 0(TF0) and Timer Flag 1(TF1).

9. Serial Data Communication:

Serial Communication means a method of establishing communication among computers is by transmitting and receiving data bits is a serial connection network. In 8051,

- The SBUF (Serial Port Data Buffer) register holds the data
- The SCON (Serial Control) register manages the data communication
- The PCON (Power Control) register manages the data transfer rates. Further, two pins - RXD and TXD, establish the serial network.

The SBUF register has 2 parts – one for storing the data to be transmitted and another for receiving data from outer sources. The first function is done using TXD pin and the second function is done using RXD pin. There are 4 programmable modes in serial data communication. They are:

i) Serial Data mode 0 (shift register mode)

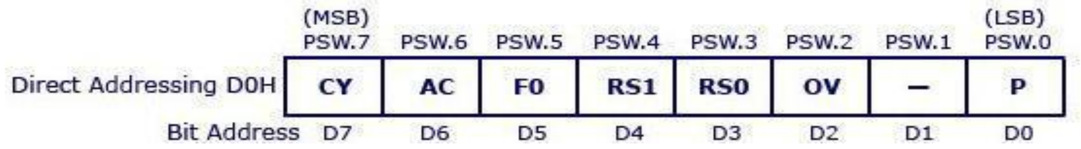
ii) Serial Data mode 1 (standard UART)

iii) Serial Data mode 2 (multiprocessor mode)
iv) Serial Data mode 3

10. PSW (Program Status Word) :

Flags are one bit registers provided to store results of certain program instructions. Some instructions can test the condition of the flag and make decision based on the flag status. Many instruction implicitly (or) explicitly effect the several status flags, which are grouped together to form the program status word.

- Program Status Word or
- Program status word or simply PSW is an 8-bit register.
- It consists of carry, auxiliary carry, and overflow and parity flags.
- There are bits RS1 and RS0 for register bank selection.
- PSW is a bit addressable register. Each of the PSW bits is referred as PSW.X (PSW.0 is the LSB,which is parity & PSW.7 is the MSB, which is carry flag)



	(MSB) PSW.7	PSW.6	PSW.5	PSW.4	PSW.3	PSW.2	PSW.1	(LSB) PSW.0
Direct Addressing D0H	CY	AC	F0	RS1	RS0	OV	–	P
Bit Address	D7	D6	D5	D4	D3	D2	D1	D0

10. Data Pointer (DPTR) :

- The data pointer or DPTR is a 16-bit register.
- It is made up of two 8-bit registers called DPH and DPL.
- Separate addresses are assigned to each of DPH and DPL.
- The data pointer is used for addressing the off-chip data and code with the MOVX,MOVC commands, respectively.
- With 16- bit DPTR , a maximum of 64K off chip data memory and a maximum of 64K off chip program memory can be addressed.

11. Stack Pointer (SP) :

- The stack pointer (SP) in 8051 is an 8-bit register. The main purpose of SP is to access the stack.
- It is incremented during PUSH or CALL operations and is decremented during POP or RET operations.
- The SP acts as a pointer for an address that points to the top of the stack.
- After the RESET operation, the stack pointer is initialized to 07H, causing the stack begin at 08H.

12. System bus:

A bus is group of wires using which data transfer takes place from one location to another within a system. Buses reduce the number of paths or cables needed to set up connection between components. There are mainly two kinds of buses. Those are

(i)Data Bus:

- The purpose of data bus is to transfer data. It acts as an electronic channel using which data travels. Wider the width of the bus, greater will be the transmission of data.

(ii)Address Bus:

- The purpose of address bus is to transfer information but not data. The information tells from where within the components, the data should be sent to or received from. The capacity or memory of the address bus depends on the number of wires that transmit a single address bit.

iii)Control Bus:

- The control bus is used to transmit the control signals such as read, write, and interrupt control signal.

5.6 Draw the register structure of 8051and explain.

<u>REGISTER STRUCTURE</u>:

1. PC (Program Counter) :

- The PC is 16 bits in length and identifies the location of the next instruction to be executed.
- During normal operation, the 8051 fetches instructions one after the other from the program memory.
- Every time an instruction byte is fetched from memory, the 8051 increments the value in PC such that it points to the next sequential byte of code.
- After reset, PC=0000H (PCH=00, PCL=00).
- The contents of PC are automatically modified by hardware when a call or branch instruction is executed.

2. ACC (Accumulator):

- The ACC register contains the source and result of arithmetic and logical operations.
- The mnemonics for accumulator-specific instructions refer to the accumulator simply as A.

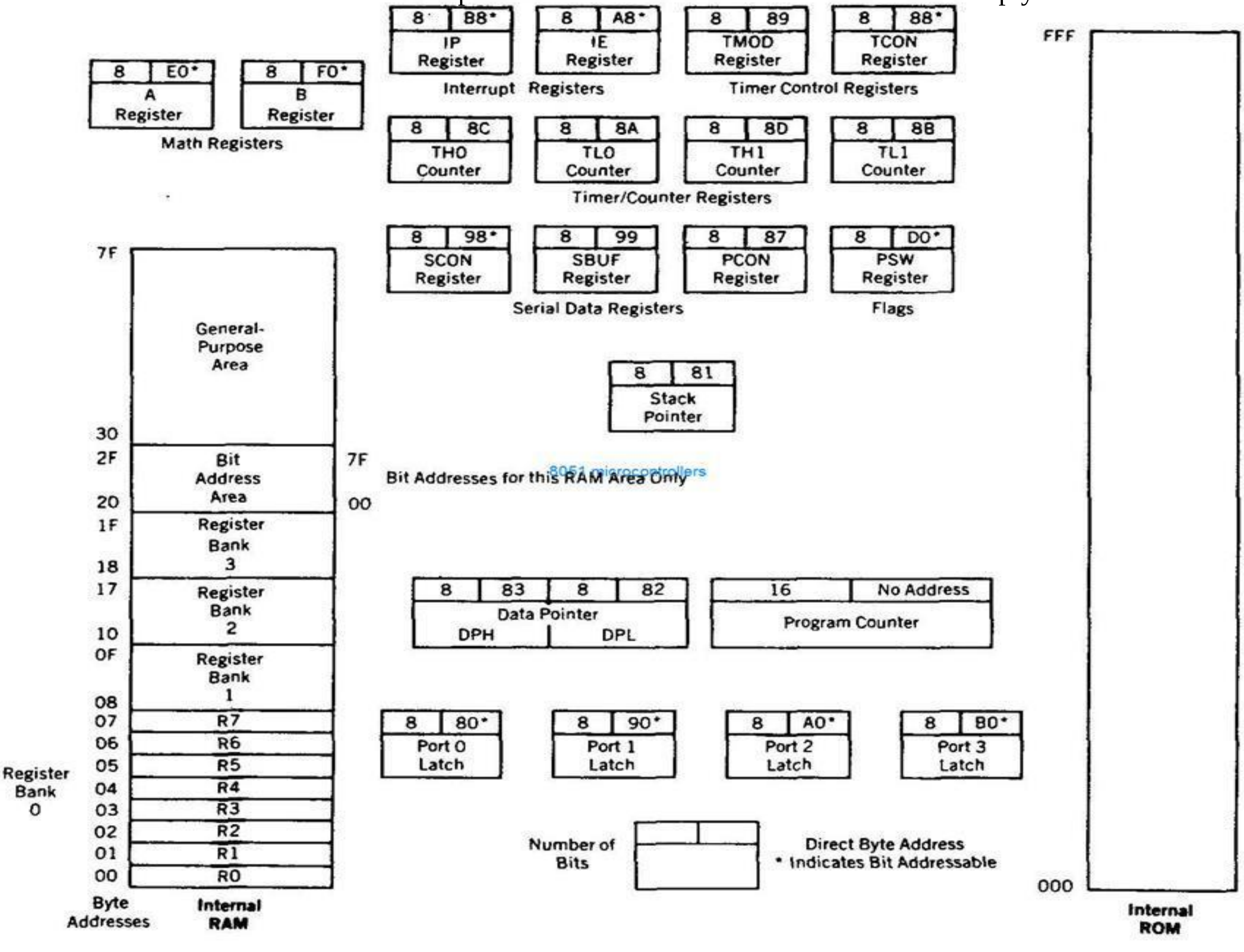
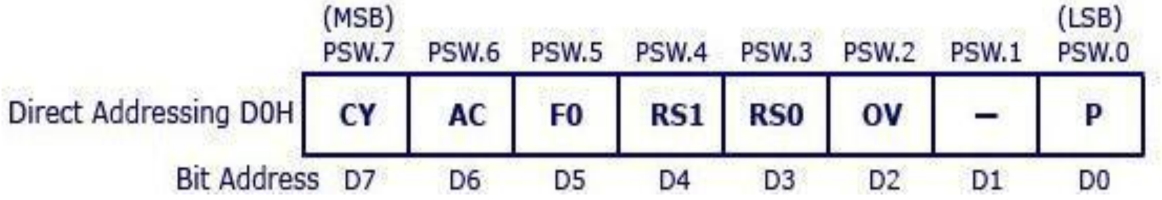


Figure: Register structure of 8051

3. PSW (Program Status Word) :

- The PSW register contains program status information as detailed in Figure.
- The PSW register is bit addressable.



	(MSB) PSW.7	PSW.6	PSW.5	PSW.4	PSW.3	PSW.2	PSW.1	(LSB) PSW.0
Direct Addressing D0H	CY	AC	F0	RS1	RS0	OV	–	P
Bit Address	D7	D6	D5	D4	D3	D2	D1	D0

Figure: PSW bit diagram

CY: Carry flag. The CY is set if there is a carry-out or borrow-in for the MSB of the result during the execution of an arithmetic instruction. Otherwise, CY is reset.

AC: Auxiliary Carry flag. The AC is set if there is a carry-out from the low nibble into the high nibble or a borrow-in from the high nibble into the low nibble. Otherwise, AC is reset. This flag can be used in the adjustment of binary to BCD.

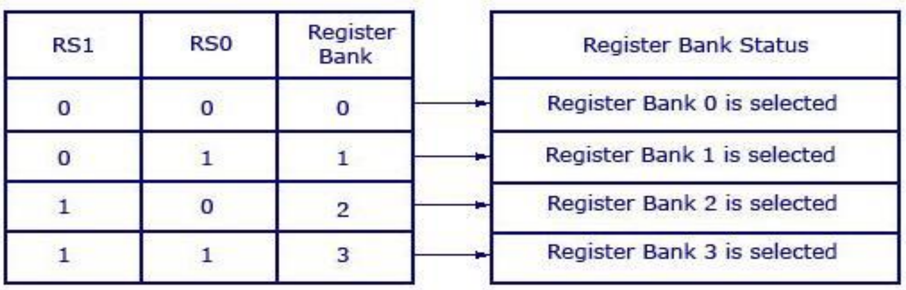
F0: Flag 0 is available to the user for general purposes.

OV: Overflow flag. When OV is set, It indicates that the signed result is out of range. If the result is not out of range, OV remains reset.

P: Parity bit. The P flag is set if the result produce by the instruction has odd parity, that is, if it contains an odd number of bits at the 1 logic level. If parity is even, P flag is reset.

RS0: Register bank select
bit 0 **RS1:**Register bank
select bit 1

Depending upon RS0 and RS1 the Register Bank Status is given as:



RS1	RS0	Register Bank	Register Bank Status
0	0	0	Register Bank 0 is selected
0	1	1	Register Bank 1 is selected
1	0	2	Register Bank 2 is selected
1	1	3	Register Bank 3 is selected

4. B(B register) :

- The B register is used during multiply and divide executions. For other instructions it can be treated as another general purpose register.
- While multiplying, it holds one of the 8-bit operands and after the execution, it stores the higher byte of the result.
- While dividing, it holds an 8- bit divisor and after the execution, the remainder is stored in B-register.

5. IP (Interrupt priority register):

The IP register is used to define the priority levels (high and low) of interrupt resources.

6. IE(Interrupt enable register) :

The IE register is used to enable or disable the interrupt resources.

7. SBUF (Serial data buffer) :

The SBUF is actually two separate registers: one transmit buffer register and one receive buffer register. When a data byte is moved to SBUF, it goes to the transmit buffer where it is held for serial transmission. When a data byte is moved from SBUF, it comes from the receiver buffer.

8. SCON(Serial control register) :

The serial control register and status register contains not only the mode selection bits, but also the 9th data bit for transmit and receive (TB8 and RB8), and the serial port interrupt bits (T1 and R1).

9. TH1, TL1 (Timer/counter 1 register high low bytes)**:**

The two 8-bit registers are used to store the count value of
timer/counter 1. **10. TH0, TL0** (Timer/counter 0 register)**:**

The two 8-bit registers are used to store the count value of timer/counter 0.

11. TMOD (Timer/counter mode control register):

It is used to define the operating modes of timer/ counters and timer or counter selection.

12. TCON (Timer /counter control register) :

It contains the overflow flags and run control bits of timer counters, and the edge flag and type control bits of interrupts.

13. DPH, DPL (data pointer high and low bytes):

8051 has two 16-bit registers: one is the program counter (PC) and the other is data pointer register

(DPTR). The data pointer consists of a high byte (DPH) and a low byte (DPL). Its intended function is to hold a 16-bit address. It may be manipulated as a 16-bit register or as two independent 8-bit registers.

14. PCON (Power control Register) :

It contains the double baud rate bit (SMOD), idle mode (IDL) bit and power down mode (PD) bit.

15. SP (Stack pointer):

It is 8-bit wide. It is incremented before data is stored during PUSH and CALL executions. While the stack may reside anywhere in on-chip RAM, the stack pointer is initialized to 07H after a reset. This causes the stack to begin at location 08H.

16. P0, P1, P2, and P3 (I/O portsn0 to 3):

The P0, P1, P2, and P3 are the SFR latches of 8051 I/O ports 0, 1, 2, and 3, respectively.

5.7 Explain the function of various special function registers.

<u>Special Function Registers (SFR):</u>

The 8051 is a flexible microcontroller with a relatively large number of modes of operations. We may inspect and/or change the operating mode of the 8051 by manipulating the values of the 8051's Special Function Registers (SFRs).

- **SFRs are accessed as if they were normal Internal RAM.** The only **difference** is that **Internal RAM** is from **address 00h through 7Fh** whereas **SFR registers** exist in the **address range of 80h through FFh**. Each SFR has an address (80h through FFh) and a name.
- As you can see, although the address range of 80h through FFh offer 128 possible addresses, there are only 21 SFRs in a standard 8051. All other addresses in the SFR range (80h through FFh) are considered invalid. Writing to or reading from these registers may produce undefined values or behavior.

Types of SFR's:

i)Serial Port Data Buffer (SBUF)

ii) Timer/Counter Control (TCON)

iii) Timer/Counter Mode Control (TMOD)

iv) Serial Port Control (SCON)

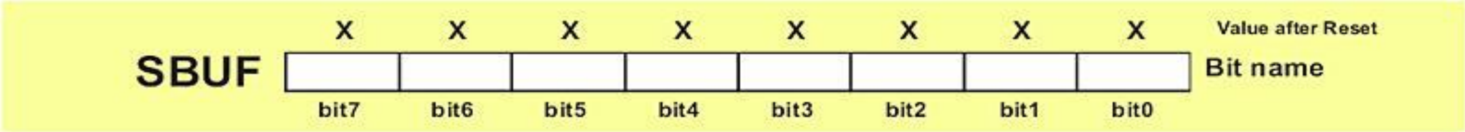
v) Power Control (PCON)

vi)Interrupt Priority (IP)

vii) Interrupt enable control (IE)

i). SBUF (Serial data buffer) :

The SBUF is actually two separate registers: one transmit buffer register and one receive buffer register. When a data byte is moved to SBUF, it goes to the transmit buffer where it is held for serial transmission. When a data byte is moved from SBUF, it comes from the receiver buffer.

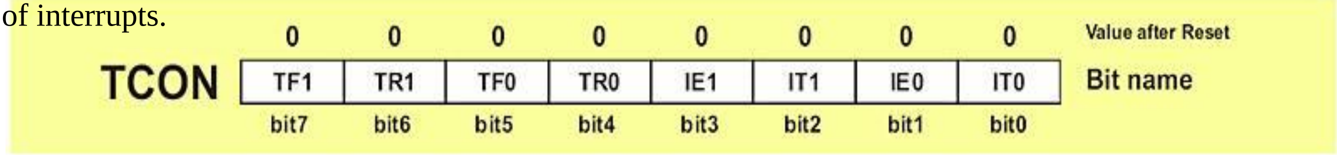


	X	X	X	X	X	X	X	X	Value after Reset
SBUF									Bit name
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	

- A microcontroller has an integrated UART (Universal Asynchronous Receiver and Transmitter) better known as a serial port. It is a full-duplex port, thus being able to transmit and receive data simultaneously and at different baud rates.
- Serial data transmit is nothing but writing to the SBUF register, while data receive represents reading the same register. The microcontroller takes care of not making any error during data transmission.
- The physical address of SBUF is **99**H.

ii). TCON(Timer /counter control register) :

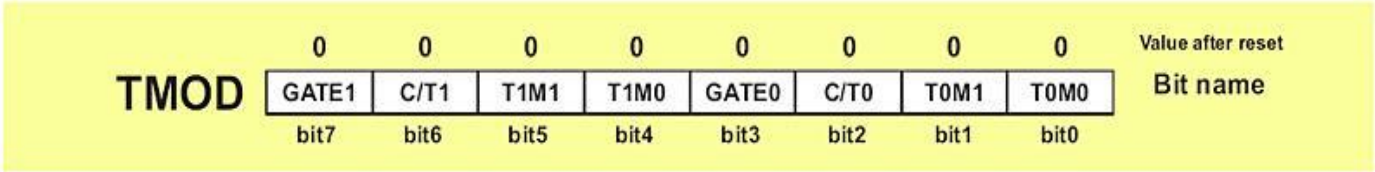
It contains the overflow flags and run control bits of timer counters, and the edge flag and type control bits of interrupts.



	0	0	0	0	0	0	0	0	Value after Reset
TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	Bit name
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	

- TCON is a byte addressable special function register with address 88H.
- TCON register is used to control the operation of the timers and also contain flags used to indicate the
  timer overflow.
- In TCON register, higher nibble bits are used to control the timers and the lower nibble bits are used to
  control the interrupts.

iii. TMOD(Timer/counter mode control register) :
It is used to define the operating modes of timer/ counters. Timer or counter selection.



TMOD								
Value after reset	0	0	0	0	0	0	0	0
Bit name	GATE1	C/T1	T1M1	T1M0	GATE0	C/T0	T0M1	T0M0
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0

- This is an 8-bit register which is used control the mode of operation of both timers T0 and T1 to set the various timer modes.
- In this TMOD register, lower 4 bits are for Timer 0 and the upper 4 bits are for Timer 1.
- **TMOD is not bit addressable**. The physical address of TMOD register is **89**H.

TH$_0$&TL$_0$ (Timer0 registers):

- Timer0 T0 is a 16-bit register and accessed as lower byte and higher byte. The lower byte is referred as a
  TL$_0$ and the higher byte is referred to as TH$_0$.
- The physical address of TL$_0$ is **8A**H and TH$_0$ is **8C**H. These are only byte addressable registers.

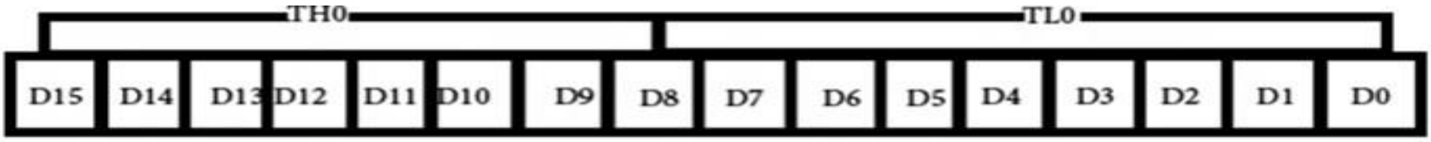


Fig. Timer0

TH$_1$&TL$_1$ (Timer1 registers):

- Timer1 (T1) is also a 16-bit register and is split into two bytes, referred to as TL1 and TH1.
- These registers can be accessed like any other registers.
- The physical address of **TL1** is **8B**H and TH1 is **8D**H. These are only byte addressable registers.

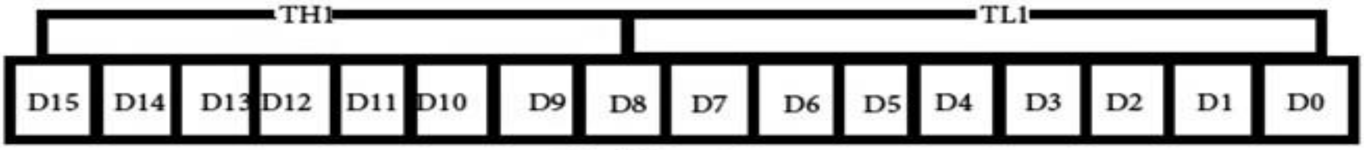
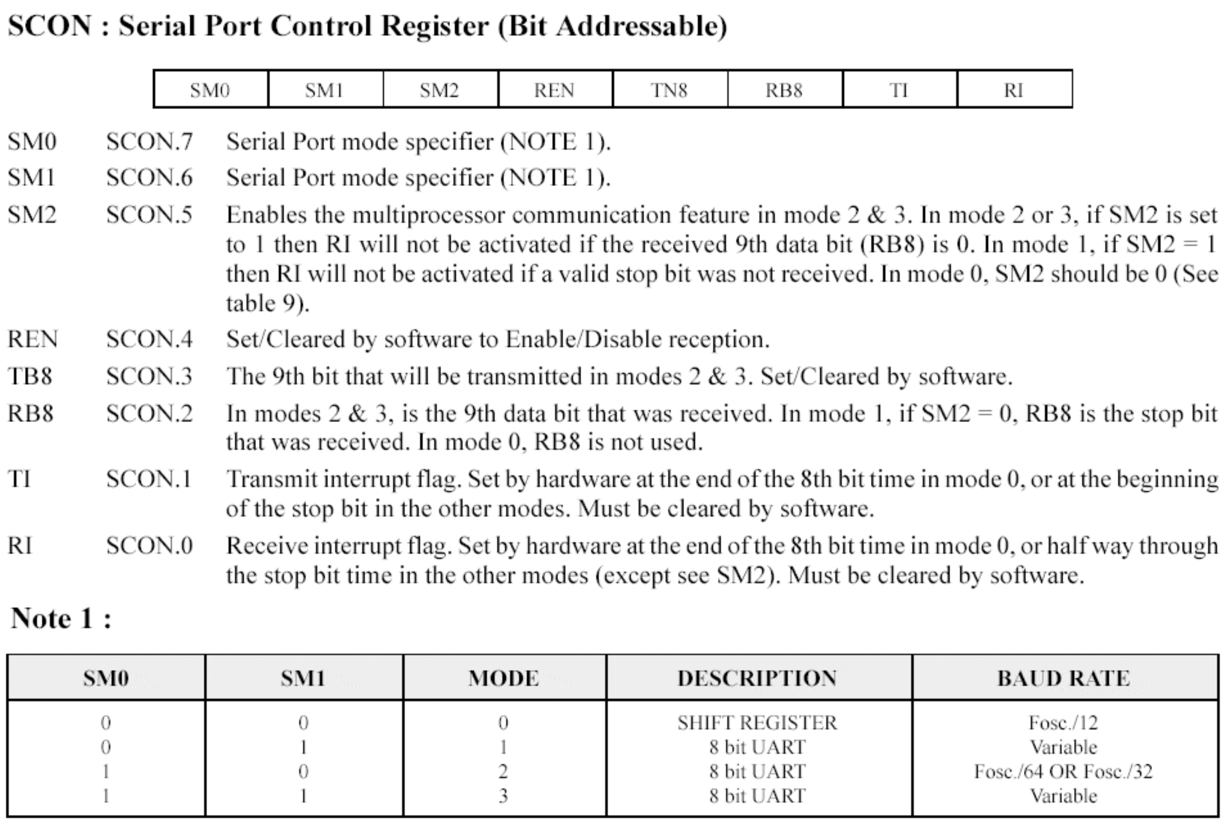


Fig. Timer1

iv). SCON (Serial control register):
The serial control register and status register contains not only the mode selection bits, but also the 9th data bit for transmit and receive (TB8 and RB8), and the serial port interrupt bits (T1 and R1).

- It controls the serial communication.
- The SCON bits can be used to select the serial communication mode, such as 8-bit UART, 9-bit UART.
- The physical address of SCON is 98H. It is bit/Byte addressable register.



SCON : Serial Port Control Register (Bit Addressable)

SM0	SM1	SM2	REN	TN8	RB8	TI	RI

SM0 SCON.7 Serial Port mode specifier (NOTE 1).

SM1 SCON.6 Serial Port mode specifier (NOTE 1).

SM2 SCON.5 Enables the multiprocessor communication feature in mode 2 & 3. In mode 2 or 3, if SM2 is set to 1 then RI will not be activated if the received 9th data bit (RB8) is 0. In mode 1, if SM2 = 1 then RI will not be activated if a valid stop bit was not received. In mode 0, SM2 should be 0 (See table 9).

REN SCON.4 Set/Cleared by software to Enable/Disable reception.

TB8 SCON.3 The 9th bit that will be transmitted in modes 2 & 3. Set/Cleared by software.

RB8 SCON.2 In modes 2 & 3, is the 9th data bit that was received. In mode 1, if SM2 = 0, RB8 is the stop bit that was received. In mode 0, RB8 is not used.

TI SCON.1 Transmit interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or at the beginning of the stop bit in the other modes. Must be cleared by software.

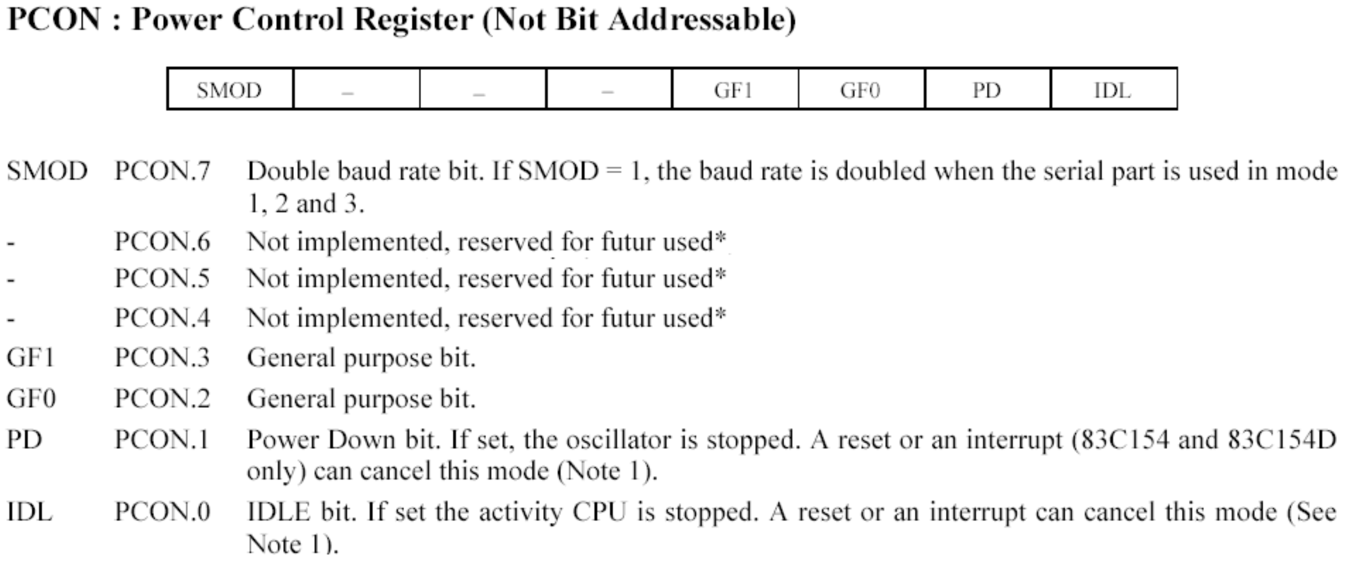
RI SCON.0 Receive interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or half way through the stop bit time in the other modes (except see SM2). Must be cleared by software.

Note 1 :

SM0	SM1	MODE	DESCRIPTION	BAUD RATE
0	0	0	SHIFT REGISTER	Fosc./12
0	1	1	8 bit UART	Variable
1	0	2	8 bit UART	Fosc./64 OR Fosc./32
1	1	3	8 bit UART	Variable

PCON (Power control Register) :

- As the name indicates, this register is used for efficient power management of 8051 micro controller.
- Commonly referred to as PCON register, this is a dedicated SFR for power management alone. Power control register is used to control the power control modes in 8051.
- There are 2 modes that we can operate microcontroller i.e., ***Idle mode and Power down mode.*** These modes are used for low power dissipation.
- The physical address of PCON is **87**H. PCON is not bit addressable.



PCON : Power Control Register (Not Bit Addressable)

SMOD	–	–	–	GF1	GF0	PD	IDL

SMOD PCON.7 Double baud rate bit. If SMOD = 1, the baud rate is doubled when the serial part is used in mode 1, 2 and 3.

- PCON.6 Not implemented, reserved for futur used*

- PCON.5 Not implemented, reserved for futur used*

- PCON.4 Not implemented, reserved for futur used*

GF1 PCON.3 General purpose bit.

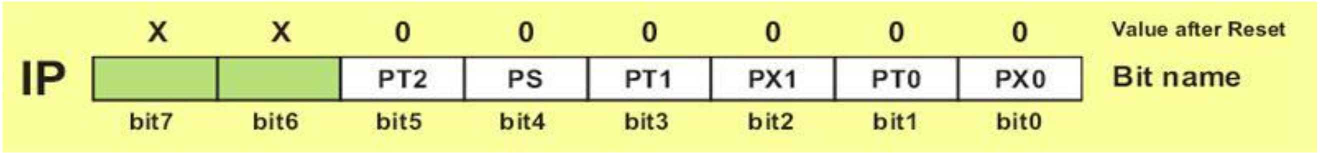
GF0 PCON.2 General purpose bit.

PD PCON.1 Power Down bit. If set, the oscillator is stopped. A reset or an interrupt (83C154 and 83C154D only) can cancel this mode (Note 1).

IDL PCON.0 IDLE bit. If set the activity CPU is stopped. A reset or an interrupt can cancel this mode (See Note 1).

vi). IP (Interrupt priority register) :

The IP register is use to define the priority levels (high and low) of interrupt resources.

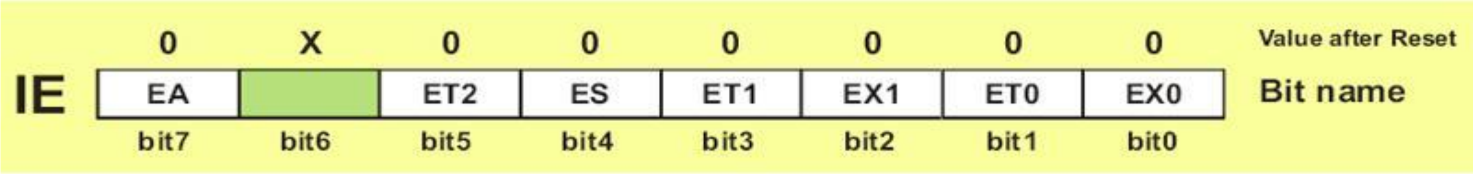


	X	X	0	0	0	0	0	0	Value after Reset
IP			PT2	PS	PT1	PX1	PT0	PX0	Bit name
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	

- It is possible to change the priority levels of the interrupts by setting or clearing the corresponding bit in the Interrupt Priority (IP) register.
- If these interrupt priorities are not programmed, the microcontroller executes in predefined manner and its order is $\overline{\textbf{INT0}}$**)**, TF0, $\overline{\textbf{INT1}}$**)** ,TF1 and RI & TI.
- The physical address of IE registers is **B8**H. It is bit / byte addressable register.

vii). IE(Interrupt enable register) :

The IE register is used to enable or disable the interrupt resources.



	0	X	0	0	0	0	0	0	Value after Reset
IE	EA		ET2	ES	ET1	EX1	ET0	EX0	Bit name
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	

- This 8-bit register is used for enabling and disabling the interrupts.
- It is a bit addressable register in which EA must be set to 1 for enabling interrupts. The whole interrupt system can be disabled by clearing the EA bit of the IE register. The corresponding bit in this register enables particular interrupt like timer interrupt, external and serial interrupt inputs.
- The physical address of IE registers is A8H.

Accumulator: (ACC)

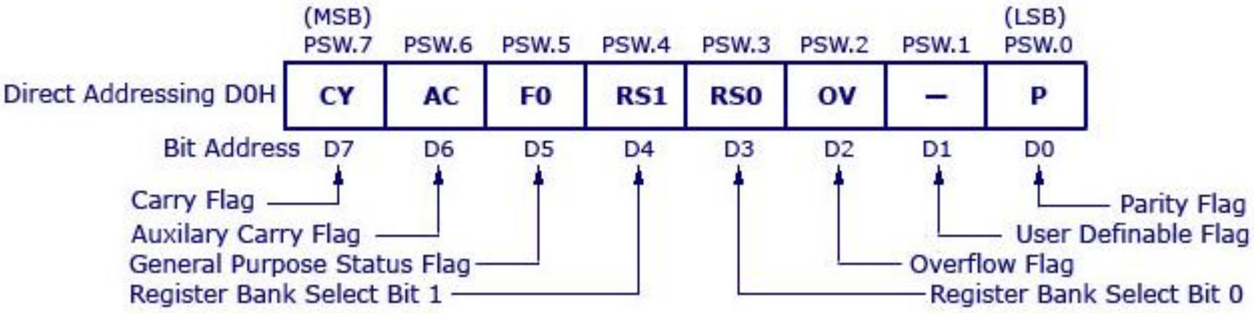
- The Accumulator is an 8- bit register (A or ACC also) holds which hold one of the operand and the result of most of arithmetic and logic operations.
- It is used for all data transfers between 8051 and external memory
- A is usually accessed by direct addressing and its physical address is **E0H.**
- Accumulator is ***both byte and bit addressable.***

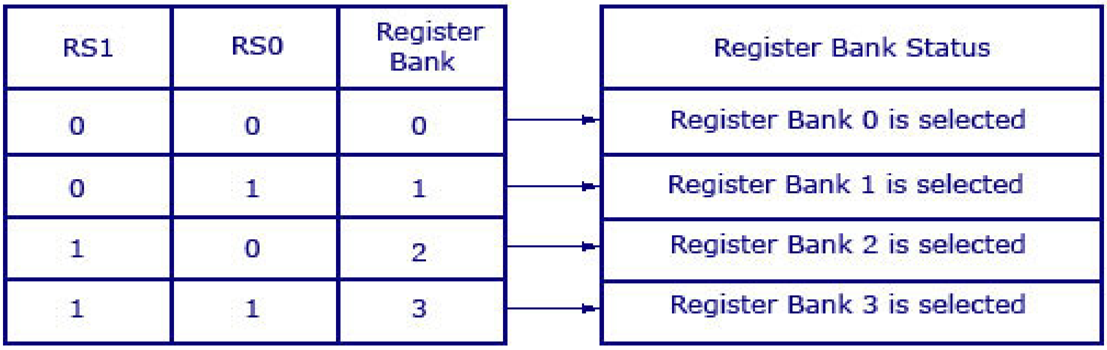
B register:

- It is an 8-bit register, with address **F0H**.
- The B-register is used to store one of the operands for multiply and division instructions.
- It can be used to store data temporarily.
- It is bit addressable as well as byte addressable.

Program Status Word (PSW) (Address D0):

- PSW register reflects the status of the operation of the processor.
- The picture below shows PSW register. It has Five flags.
- PSW register is both bit and byte addressable.
- The physical address of PSW is D0H. The individual bits are then accessed using D1, D2 … D7.
- It has Carry bit, Auxiliary Carry, two register bank select bits, Overflow flag, parity bit and user-definable status flag. RS1 and RS0 bits are used to select the Register Banks. After resetting the microcontroller, the bank0 will automatically selected.





RS1	RS0	Register Bank	Register Bank Status
0	0	0	Register Bank 0 is selected
0	1	1	Register Bank 1 is selected
1	0	2	Register Bank 2 is selected
1	1	3	Register Bank 3 is selected

Stack pointer (**SP**):

- Stack pointer is 8-bit byte addressable register with an address of 81H in SFR address space.
- The stack pointer is used to store top address of the Stack memory.
- The stack pointer will be initialized to 07H after resetting the microcontroller.

Port0-Port3: (P0-P3)

- Generally these registers are port latches. These SFR's are used to read the input port data and write data
  to the output port.
- All these registers are bit and byte addressable with addresses of 80H, 90H, A0H and B0H for P0, P1, P2 and P3 respectively.

DPL/DPH: (Data pointer lower byte/higher byte)

- DPL and DPH are byte addressable registers with addresses 82H and 83H respectively.
- The DPL and DPH are combined to form 16-bit data pointer register. The data pointer is used to specify the external RAM addresses.

Some of the 8051 SFR's are

S.No	**Symbol**		**Name of SFR**	**Address (Hex)**
1	ACC*		Accumulator	**0E0**
2	B*		B-Register	**0F0**
3	PSW*		Program Status word register	**0DO**
4	SP		Stack Pointer Register	**81**
5	DPTR	DPL	Data pointer low byte	**82**
		DPH	Data pointer high byte	**83**
6	P0*		Port 0	**80**
	P1*		Port 1	**90**
8	P2*		Port 2	**0A**
9	P3*		Port 3	**0B**
10	IP*		Interrupt Priority control	**0B8**
11	IE*		Interrupt Enable control	**0A8**
12	TMOD		Tmier mode register	**89**
13	TCON*		Timer control register	**88**
14	TH0		Timer 0 Higher byte	**8C**
15	TL0		Timer 0 Lower byte	**8A**
16	TH1		Timer 1Higher byte	**8D**
17	TL1		Timer 1 lower byte	**8B**
18	SCON*		Serial control register	**98**
19	SBUF		Serial buffer register	**99**
20	PCON		Power control register	**87**

The * indicates the bit addressable SFRs Table: SFRs of 8051 Microcontroller

5.8 Give the pin diagram of 8051 Microcontroller and specify the purpose of each pin

Pin diagram of 8051 microcontroller:

The 8051 is a 8-bit microcontroller. It is a 40 pin IC. Then the function of each pin is described as follows :

(1) Pins 1-8: P1.0-P1.7, I/O port 1

- An 8-bit bidirectional I/O port 1, P1.
- With internal pull up.

(2) Pin 9: RESET input

(3) Pins 10-17: P3.0-P3.7, I/O port 3

- An 8-bit bidirectional I/O port 3
- With internal pull up
- P3 also serves the special features:

P3.0 (RXD) Serial input port
P3.1 (TXD) Serial output port
P3.2 (INT0) External interrupt input 0
P3.3 (INT1) External interrupt input 1
P3.4 (T0) Timer/Counter 0 external input
P3.5 (T1) Timer/Counter 1 external input
P3.6 (WR) External data memory write strobe
P3.7 (RD) External data memory read strobe

The CPU generates write and read signals, WR and RD, as needed during external data memory accesses (MOVX instruction).

(4) Pins 18-19:(XTAL2, XTAL1)

XTAL1 is the input to the inverting oscillator amplifier and input to internal clock generator circuits. XTAL2 is the output from the inverting oscillator amplifier. To generate the required clock signal, a crystal is connected across XTAL1 and XTAL2 pins, and two loading capacitors (20 pF) are required from XTAL1 and XTAL2 to ground.

(5) Pin 20: Vss, ground, 0V reference

(6) Pins 21-28: P2.0-P2.7, I/O port 2

- An 8-bit bidirectional I/O port with internal pull-ups.
- P2 emits the high-order address byte (A8-A15) during fetches from external program memory and during accesses to external data memory that uses 16-bit addresses.

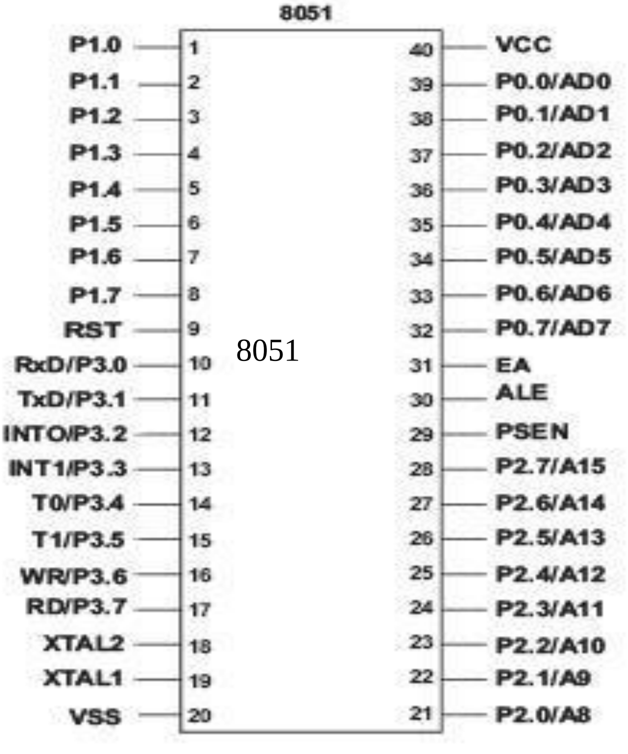


Figure 7 : Pin diagram of 8051

(7) Pin 29:(PSEN, Program Store Enable)

- Program store enable(PSEN) is the output control signal, activated every six oscillator periods, while fetching the external program memory.

- It is the read strobe to external program memory. In practice, PSEN pin is wired to the output enable (OE) pin of external ROM.
- PSEN is not activated during fetches from internal program memory.

(8) Pin 30: ALE/PROG, Address Latch Enable/programming pulse input

- Output pulse for latching the low byte address (A0-A7) during accesses to external memory.
- In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency, and can be used for external timing or clocking.

(9) Pin 31: EA /Vpp, External Access Enable/Programming Supply Voltage

- This pin must be externally held low to enable the 8051 to fetch code from external program memory. For ROM less devices, such as the 8031, this pin should be wired to ground for using external ROM.
- If EA is held high, the device executes from internal program memory unless the program counter (PC) contains an address greater than the maximum internal program memory space.
- If the 8051 on-chip program memory is not used, this pin is directly wired to ground.
- In programming mode, this pin must be connected to the programming power supply generally 12V.

(10) Pins 32-39: P0.0-P0.7, I/O port 0

- An 8-bit bidirectional I/O port, P0.
- Similar to port 2 and 3, these pins can be used as input output pins when we don't use any external memory. When ALE or Pin 30 is at 1 then this port is used as data bus, when ALE pin at 0, then this port is used as lower order address bus(A0 to A7).

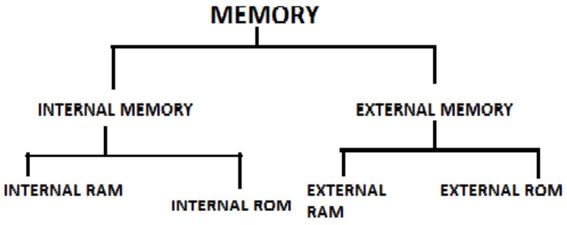
(11) Pin 40: Vcc, positive power supply, +5V.

5.9 Describe internal Memory, external Memory and ports of 8051 :

Memory organization of 8051:

The 8051 uses separated memory architecture that is referred to as Harvard architecture whereas Von Neumann architecture defines a system where code and data can share common memory.

Types of Memory :

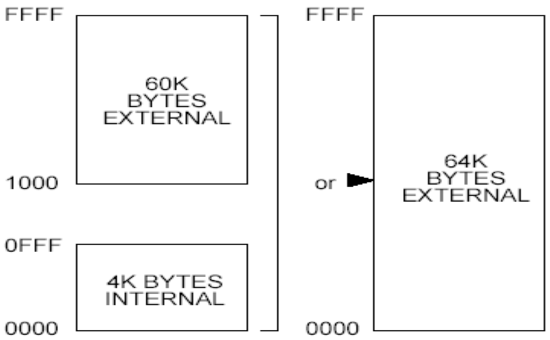


(A)External ROM (Code Memory):

The executable program code is stored in this code memory. The code memory size is limited to 64KBytes (in a standard 8051). The 64K program memory space of 8051 is divided into internal and external memory.

- If the EA pin is high, then 8051 executes from the internal program memory until the address exceeds 0FFFh. After that, locations 1000h through FFFFh are executed from the external memory portion.
- If EA pin is held low, the 8051 executes instructions from external memory only.

The 80C51 Program Memory.



(a) **If EA =1** (b) **if EA =0**

(B)External RAM (Data Memory):

This is read-write memory and is available for storage of data. Up to 64KBytes of external RAM data memory is supported (in a standard 8051).

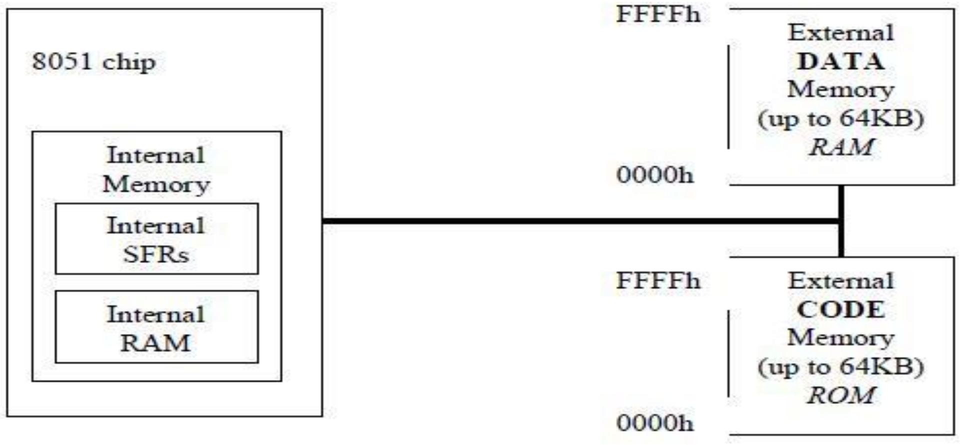

8051 chip
Internal Memory
Internal SFRs
Internal RAM
FFFFh
0000h
External DATA Memory (up to 64KB) RAM
FFFFh
0000h
External CODE Memory (up to 64KB) ROM

Figure : Memory structure of 8051

(C)On-Chip Memory (or)Internal Memory:

The 8051's on-chip memory consists of 256 memory bytes. Locations available to the user occupy addressing space from 0 to 7Fh, i.e. first 128 registers and this part of RAM is divided in several blocks. The first 128 bytes of internal data memory are both directly and indirectly addressable. The upper 128 bytes of data memory (from 0x80 to 0xFF) can be addressed only indirectly.

First 128 bytes: 00h to 1Fh Register Banks, 20h to 2Fh Bit Addressable RAM, 30 to 7Fh General Purpose RAM. **Next 128 bytes:** 80h to FFh Special Function Registers.

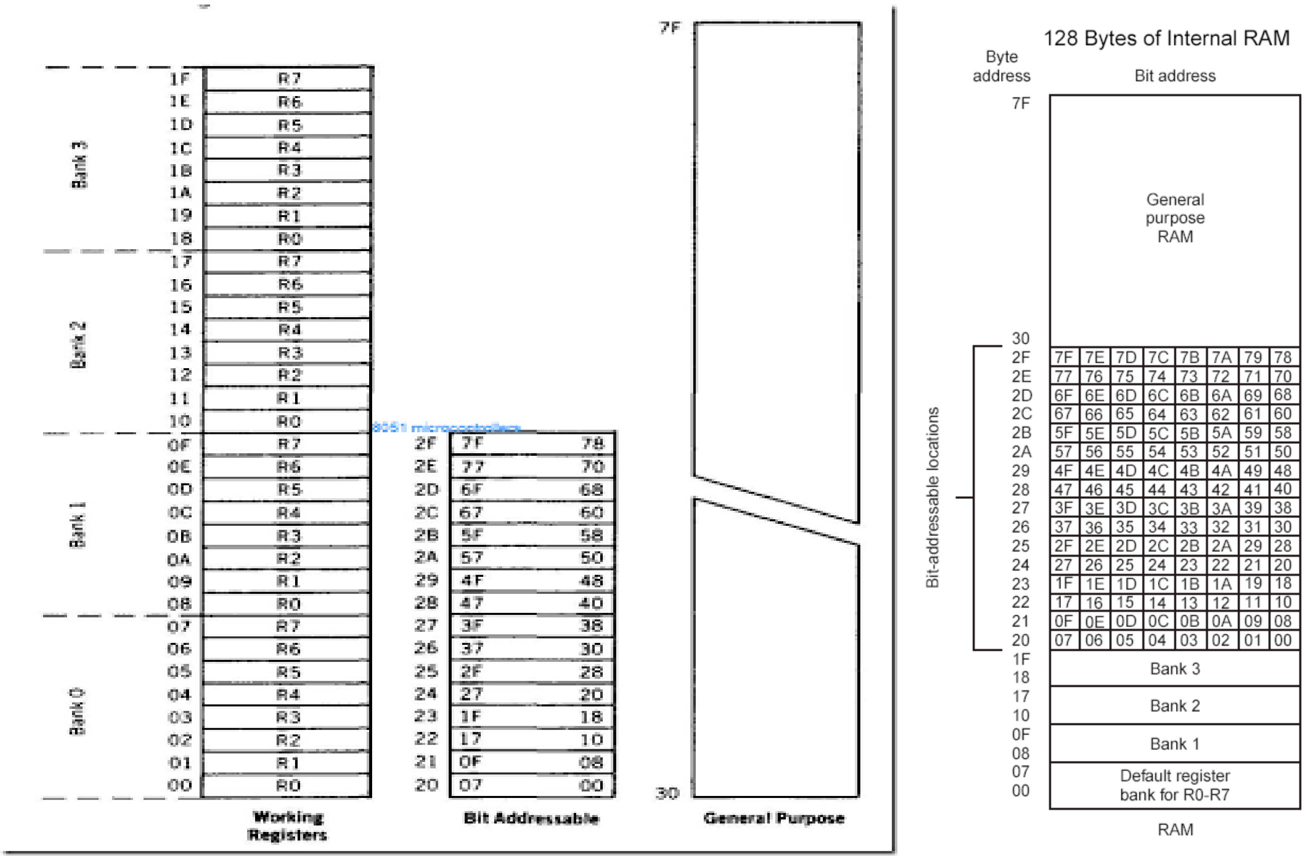
128 bytes of internal RAM are divided into:

(1)Register banks(bank0 – bank3)(32 bytes & addressing from 00h -1FH)
(2)Bit/byte addressable area(16bytes & 20H -2Fh)
(3)General purpose area(80 bytes &30H -7FH)

- These 128 bytes of internal RAM are divided into 32 working registers which in turn constitute 4 register banks (Bank 0-Bank 3) with each bank consisting of 8 registers (R0 - R7). Each register can be addressed by a name (or) by its RAM address. Bits RS1 and RS0 in PSW determines which bank of registers is currently in use. When RESET the 8051, bank0 is selected by default.

- Memory block in the range of 20h to 2Fh is bit-addressable, which means that each bit being there has its own address from 0 to 7Fh. Since there are 16 such registers, this block contains in total of 128 bits with separate addresses (Bit 0 of byte 20h has the bit address 0, and bit 7 of byte 2Fh has the bit address 7Fh). A bit address is used in bit operations with Boolean processor and bit transfers.

- The RAM between 30H to 7FH has 80 bytes, which is only accessible by the address of the byte and is not used in bit operations.



Byte address								
2F	7F	7E	7D	7C	7B	7A	79	78
2E	77	76	75	74	73	72	71	70
2D	6F	6E	6D	6C	6B	6A	69	68
2C	67	66	65	64	63	62	61	60
2B	5F	5E	5D	5C	5B	5A	59	58
2A	57	56	55	54	53	52	51	50
29	4F	4E	4D	4C	4B	4A	49	48
28	47	46	45	44	43	42	41	40
27	3F	3E	3D	3C	3B	3A	39	38
26	37	36	35	34	33	32	31	30
25	2F	2E	2D	2C	2B	2A	29	28
24	27	26	25	24	23	22	21	20
23	1F	1E	1D	1C	1B	1A	19	18
22	17	16	15	14	13	12	11	10
21	0F	0E	0D	0C	0B	0A	09	08
20	07	06	05	04	03	02	01	00

Fig: Internal RAM organization of 8051

(D) Onchip ROM or internal ROM:

The 8051 has 4K bytes of internal ROM with addresses from 0000H to 0FFFH. It is programmed by the manufacturer. This chip cannot be erased or altered after fabrication.

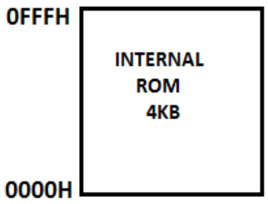


Fig: on chip ROM of 8051

input/output ports of 8051:

The 8051 Microcontroller input/output ports are working in two modes of operation those are

1. Single chip mode
2. Expanded mode

1.Single chip Mode:

In this mode of operation all 4 input/output ports are working as input and output pins.

2. Expanded Mode:

In this mode of operation each port has their own functionality as shown below:

i).PORT P0:

When there is no external memory present, this port acts as a general purpose input/output port. In the presence of external memory, it functions as a multiplexed address and data bus. It performs a dual role.

ii).PORT P1:

This port is used for various interfacing activities. This 8-bit port is a normal I/O port i.e. it does not perform dual functions.

iii).PORT P2:

Similar to PORT P0, this port can be used as a general purpose port when there is no external memory but when external memory is present it works in conjunction with PORT PO as an address bus. This is an 8-bit port and performs dual functions.

iv).PORT P3:
It is an 8-bit bidirectional I/O port 3 with internal pull up.PORT P3 used for special functions.

PORT 0 :
Port-0 can be used as a normal bidirectional I/O port or it can be used for address/data interfacing for accessing external memory. When control is '1', the port is used for address/data interfacing. When the control is '0', the port can be used as a bidirectional I/O port.

PORT 0 as an Input Port :
Let us assume that control is '0'. When the port is used as an input port, '1' is written to the latch. In this situation both the output MOSFETs are 'off'. Hence the output pin have floats hence whatever data written on pin is directly read by read pin.

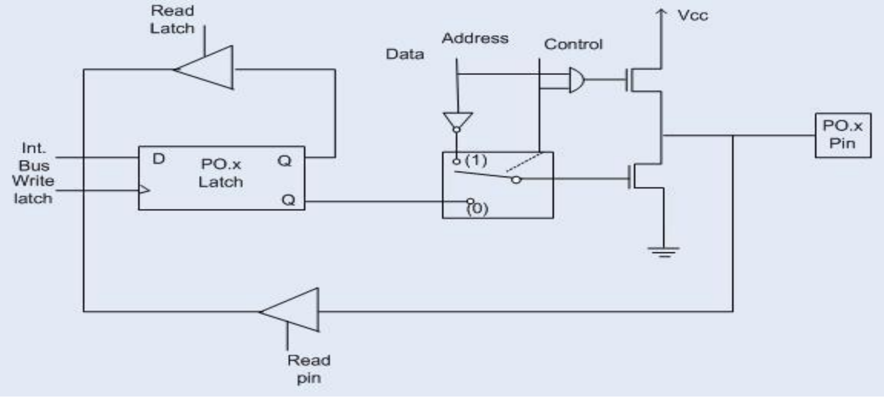


Figure: A True bidirectional port (Port 0 of 8051)

PORT 0 as an Output Port :
Suppose we want to write 1 on pin of Port 0, a '1' written to the latch which turns 'off' the lower FET while due to '0' control signal upper FET also turns off as shown in fig. above. Here we wants logic '1' on pin but we getting floating value so to convert that floating value into logic '1' we need to connect the pull up resistor parallel to upper FET . This is the reason **why we needed to connect pull up resistor to port 0 when we want to initialize port 0 as an output port** .

- If we want to write '0' on pin of port 0 , when '0' is written to the latch, the pin is pulled down by the lower FET. Hence the output becomes zero.
- When the control is '1', address/data bus controls the output driver FETs. If the address/data bus (internal) is '0', the upper FET is 'off' and the lower FET is 'on'. The output becomes '0'. If the address/data bus is '1', the upper FET is 'on' and the lower FET is 'off'. Hence the output is '1'. Hence for normal address/data interfacing (for external memory access) no pull-up resistors are required.Port-0 latch is written to with 1's when used for external memory access.

PORT 1: The structure of a port-1 pin is shown in figure below. It has 8 pins (P1.1-P1.7) .

Port-1 dedicated only for I/O interfacing. When used as output port, not needed to connect additional pull-up resistor like port 0.

- It have provided internally pull-up resistor as shown in figure above.
- The pin is pulled up or down through internal pull-up when we want to initialize as an output port.
- To **use port-1** as **input port, '1'** has to be **written to the latch.** In this input mode when '1' is written to the pin by the external device then it read fine.
- But when **'0' is written** to the pin by the external device then the external source must sink current due to internal pull-up. If the external device is not able to sink the current the pin voltage may rise, leading to a possible wrong reading.

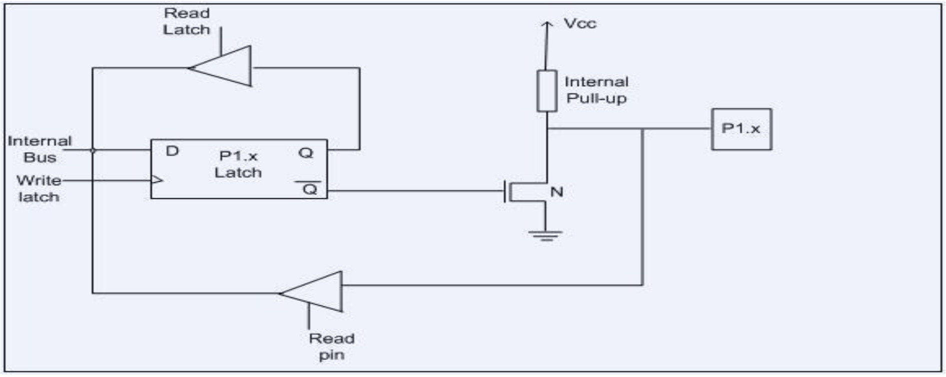
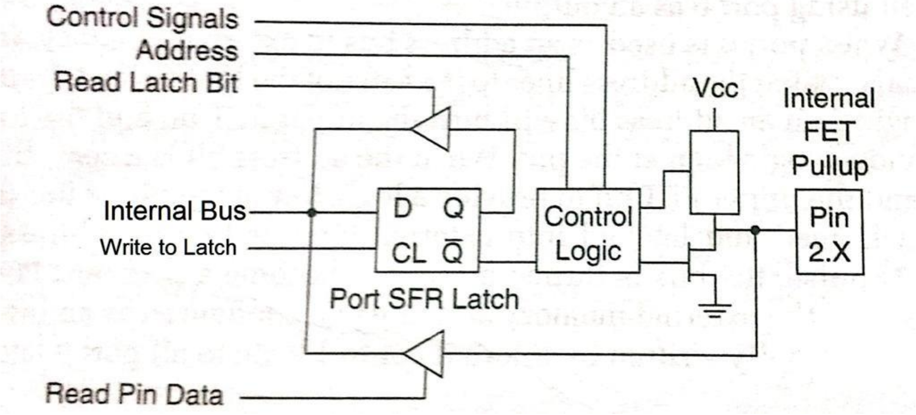


Figure: A Quasi Bidirectional port

PORT 2:

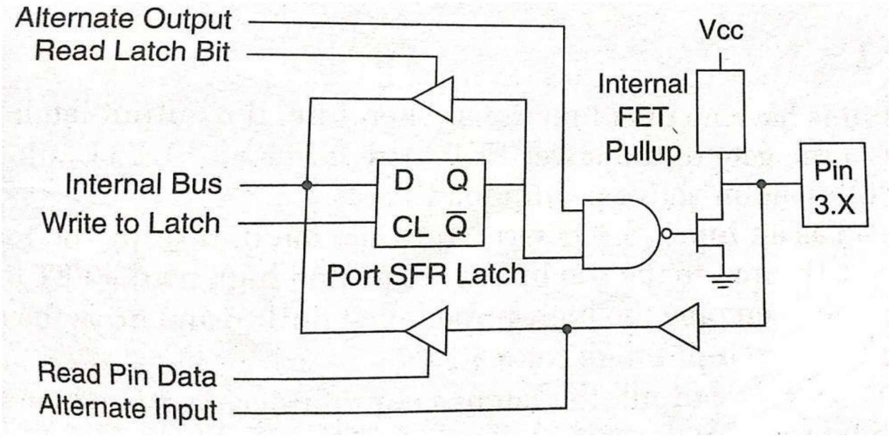
The structure of a port-2 pin is shown in figure above. It has 8-pins (P2.0-P2.7).

- Port-2 we use for higher external address byte or a normal input/output port. The I/O operation is similar to Port-1.
- Port-2 latch remains stable when Port-2 pin are used for external memory access. Here again due to internal pull-up there is limited current driving capability.



PORT 3:

Port-3 (P3.0-P3.7) having alternate functions to each pin. The internal structure of a port-3 pin is shown in figure below.



Following are the alternate functions of port 3:

- P3.0 (RXD) Serial input port
- P3.1 (TXD) Serial output port
- P3.2 (INT0) External interrupt input 0
- P3.3 (INT1) External interrupt input 1
- P3.4 (T0) Timer/Counter 0 external input
- P3.5 (T1) Timer/Counter 1 external input
- P3.6 (WR) External data memory write strobe
- P3.7 (RD) External data memory read strobe

The CPU generates write and read signals, WR and RD, as needed during external data memory accesses (MOVX instruction).

It work as an IO port same like Port 2. only alternate function of port 3 makes its architecture different than other ports.

5.10 Explain Timers and counters in 8051:

Timers and counters in 8051:

The 8051 comes equipped with two timers, both of which may be controlled, set, read, and configured individually. The 8051 timers have three general functions:

1) Keeping time and/or calculating the amount of time between events,

2) Counting the events themselves,

3) Generating baud rates for the serial port.

The 8051 micro controller has two 16- bit Counters/Timers named as T0 and T1. Each counter may be programmed to count internal clock pulses, acting as a timer, or programmed to count external pulses as a counter.

8051 microcontroller has two Timers designated as Timer0 and Timer1. Each of these timers is assigned a 16-bit register. The value of a Timer register is incremented by one every time a timer counts. Timer takes a time period of one machine cycle to count one. This means that the **maximum number of times** a timer can count 2^{16}, i.e. **65536**. So the maximum allowed counts in value of Timer registers can be from 0000H to FFFFH. Since **8051 is an 8 bit controller,** the registers of 8051 Timers are accessed as **two different registers**; one for **lower byte** and other for **higher byte**. For example, register of Timer0 is accessed as **TL0** for lower byte and **TH0** for higher byte.

- The only difference between counting and timing is the source of the clock pulses to the counters. When used as a **timer**, the clock pulses are **sourced** from the **oscillator** through the **divide-by-12d circuit**.
- When used as a **counter**, pin T0 (P3.4) supplies pulses to counter 0 and pin T1 (P3.5) to counter 1. The C/ $\overline{T}$ bit in TMOD must be set to 1 to enable pulses from the Tx pin to reach the control circuit shown in Figure.

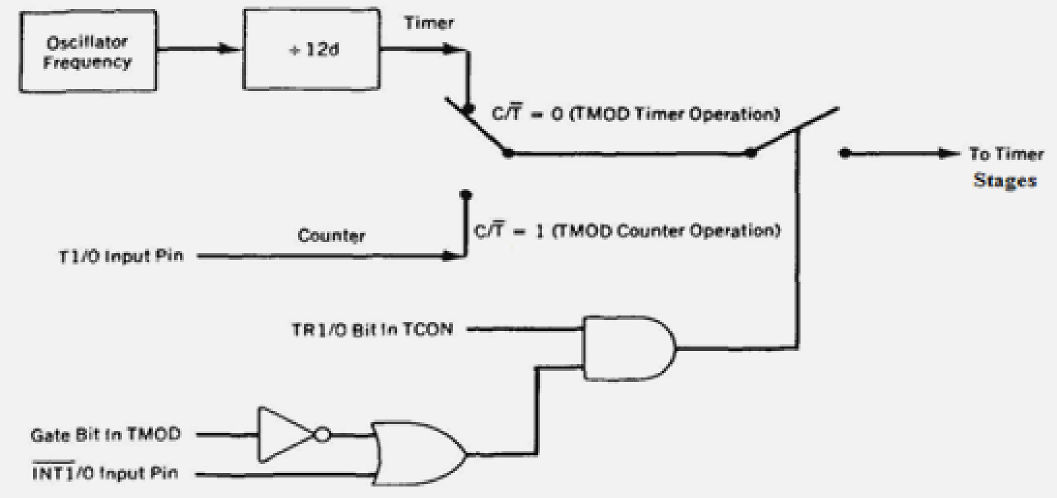


Fig. Timer/Counter control logic

How does a timer count?

A timer always counts up. It doesn’t matter whether the timer is being used as a timer, a counter, or a baud rate generator: A timer is always incremented by the microcontroller.

TMOD SFR:

Timer mode register is a 8-bit register. The **lower nibble** of the **TMOD** is to **control Timer0** and the **upper nibble** is to **control the Timer1**. The timer mode register is used to **select the timer mode** and it is used to select the **Timer operation or Counter operation.**

- This is an 8-bit register which is used by both timers T0 and T1 to set the various timer modes.
- In this TMOD register, lower 4 bits are for Timer0 and the upper 4 bits are for Timer1.
- **TMOD is not bit addressable**. The physical address of TMOD register is 89H.

GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0	**TMOD**
For Timer 1				For Timer 0				

- **GATE:** The hardware way of starting and stopping the timer by an external source (INT1/INT0) is achieved by making GATE=1.If GATE=0 then no need of external hardware to start and stop the timers.

- **$C/\overline{T}$:** Timer / Counter operation. If $C/\overline{T}$= 0 then Tx is used as a timer and if it is 1 then Tx is used as a counter.

M1&M0: These are used to select the timer mode.

M1	M0	Operating Mode
0	0	Mode 0 - 13-bit timer mode, 8-bit timer/counter THx and TLx as 5-bit pre scalar.
0	1	Mode 1 - 16-bit timer mode, 16-bit timer/counters THx and TLx are cascaded;
1	0	Mode 2- 8-bit Auto reload mode, THx holds a value which is to be reloaded into TLx each time it overflows.
1	1	Mode 3 - Spilt timer mode.

1).13-bit Time Mode (mode 0) :

- When the timer is in 13-bit mode, TLx will count from 0 to 31.
- When TLx is incremented from31, it will "reset" to 0 and increment THx.
- Thus, effectively, only 13 bits of the two timer bytes are being used: bits 0-4 of TLx and bits 0-7 of THx.

2).16-bit Time Mode (mode 1) :

- It functions just like 13-bit mode except that all 16 bits are used. TLx is incremented from 0 to 255.
- When TLx is incremented from 255, it resets to 0 and causes THx to be incremented by 1.
- Since this is a full 16-bit timer, the timer may contain up to 65,536 distinct values.
- If you set a 16-bit timer to 0, it will overflow back to 0 after 65,536 machine cycles.

3).8-bit Time Mode (mode 2) :

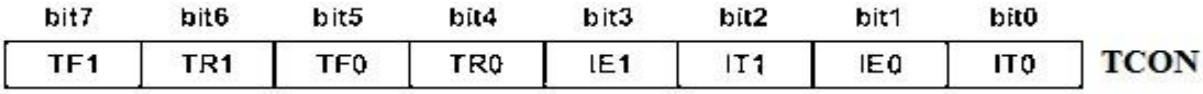
- When a timer is in mode 2, THx holds the "reload value" and TLx is the timer itself.
- Thus, TLx starts counting up.
- When TLx reaches 255 and issue sequentially incremented, instead of resetting to0 (as in the case of modes 0 and 1), it will be reset to the value stored in THx.

4).Split Timer Mode (mode 3) :

- When Timer 0 is placed in mode 3, it essentially becomes two separate 8-bit timers.
- That is to say,Timer 0 is TL0 and Timer 1 is TH0. Both timers count from 0 to 255 and overflow back to 0.
- All the bits that are related to Timer 1 will now be tied to TH0.While Timer 0 is in split mode, the real Timer 1 (i.e. TH1 and TL1) can be put into modes 0 or 1 or 2.

TCON SFR: Timer Control

In addition to the above two registers, the TCON register specifies the type of external interrupt to the 8051 microcontroller. The two external interrupts, whether edge or level triggered, specify by this register by a set, or cleared by appropriate bits in it. And, it is also a bit addressable register.



bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	TCON

TF1: Timer1 over flow flag bit.

- It is Set when timer1 **exceeds its maximum countable value**.
- It can be cleared when the processor vectors to execute **Interrupt Service Routine** of Timer 1 interrupt.

TR1: Timer 1 Run control bit.

- Set/cleared by software to on/off the Timer 1.

TF0: Timer 0 over flow flag.

- It is Set when Timer 0 **exceeds its maximum countable value.**
- It can be cleared when the processor vectors to execute Interrupt Service Routine of Timer 0 interrupt.

TR0: Timer 0 Run control bit.

- Set/cleared by software to on/off the Timer 0.

IE1: External interrupt Edge flag.

- It is set to 1, when a negative (falling) edge triggered signal is received on $\overline{\mathbf{INT1}}$.
- It is clear when the processor vectors to ISR.

IT1: External interrupt1 signal type control bit.

- Set to 1 by the program to enable $\overline{\mathbf{INT1}}$ interrupt to be triggered by the falling edge signal.
- Set to 0 by program to enable $\overline{\mathbf{INT1}}$ interrupt to be triggered by low level signal.

IE0: External interrupt0 Edge flag.

- It is set to 1, when a low level triggered signal is received on $\overline{\mathbf{INT0}}$
- It is clear when the processor vectors to ISR.

IT0: External interrupt 0 signal type control bit.

- Set to 1 by the program to enable $\overline{\mathbf{INT0}}$ interrupt to be triggered by the falling edge signal.
- Set to 0 by program to enable $\overline{\mathbf{INT0}}$ interrupt to be triggered by low level signal.

5.11 Explain Serial Input / Output of 8051?

Computers must be able to communicate with other computers in modern multiprocessor distributed systems. One cost- effective way of communication is to send and receive data bits serially.

The 8051 has a serial data communication circuit that uses register **SBUF** to **hold data**. Register **SCON controls data communication**, register **PCON controls data rates**, and pins **RXD** and **TXD** connect to the serial data network.

SBUF (Address 99H):

- A microcontroller has an integrated UART (Universal Asynchronous Receiver and Transmitter),better known as a **serial port**. It is a full-duplex port, thus being able to transmit and receive data simultaneously and at different baud rates..
- Serial data transmit is nothing but writing to the SBUF register, while data receive represents reading the same register. The microcontroller takes care of not making any error during data transmission.
- The physical address of SBUF is 99H.

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
								SBUF

SBUF has physically two registers. They are **transmitting buffer** and **receiving buffer**, the **transmitting buffer** is used to **hold data** to be **transmitted out** of the 8051 via **TXD**. The **receiving buffer holds received data** from external sources via **RXD**. Both mutually exclusive registers and use the address 99h.

SCON (address 98h):

- It controls the serial communication. (Serial port must be configured prior to being used, determine how many bits is contained in one serial “word”, baud rate and synchronization clock source.)
- The physical address of SCON is 98H. It is bit/Byte addressable.
- The functions of various bits of SCON register are

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
SM0	SM1	SM2	REN	TB8	RB8	TI	RI	SCON

- **SM0** - Serial port mode bit 0 is used for serial port mode selection.
- **SM1** - Serial port mode bit 1.

Serial port mode is selected by combining the SM0 and SM1 bits:

SM0	SM1	MODE	DESCRIPTION	BAUD RATE
0	0	0	8-bit Shift Register	f/12 the quartz frequency
0	1	1	8-bit UART	Variable
1	0	2	9-bit UART	f/32 of f//64
1	1	3	9-bit UART	Variable

SM2 - Serial port mode 2 bit, also known as multiprocessor communication enable bit.

- If it is set to 1, enables the micro controller is worked in multiprocessor system
- If this bit is 0 the microcontroller disable multiprocessor communication.

REN - Reception Enable bit.

- If this bit equal to 1 it enables serial reception.
- If this bit is 0 Serial reception is disabled.

TB8 - Transmitter bit 8.

- This is the 9th data bit, that is transmitted automatically after the 8 bit(D_0-D_7) of data have been transmitted in mode 2 and mode 3.
- It is actually used as a parity bit.

RB8 - Receiver bit 8.

- This is the 9th bit received after reception of 8 bits of data from transmitter.
- This is the parity bit received from transmitter.

TI - Transmit Interrupt flag:

- It is set when a byte has been completely transmitted.

RI - Receive Interrupt flag:

- Set when a byte has been completely received.

PCON: Power Control

Power control register is used to doubling the baud rate for serial communication and it is also used to operatethe microcontroller in power down mode and idle mode. The power down mode and idle mode features are present in later versions of 8051 i.e., 8052.

SMOD	---	---	---	GF1	GF0	PD	IDL

- SMOD - Double baud rate bit. If Timer1 is used to generate baud rate, the baud rate is doubled.
- ---- Reserved for future use
- GF1 – general purpose flag bit 1
- GF0 – general purpose flag bit 0
- PD – power down bit – setting this bit activates the power down mode
- IDL – idle bit – setting this bit activates the idle mode operation.

5.12 Explain interrupts in 8051?

The interrupt is a disturbance to the microcontroller while executing the main line program to perform specific task. There are five sources of interrupts in 8051 microcontroller, they are

- External Interrupt0 ($\overline{\textbf{INT0}}$)
- External interrupt1 ($\overline{\textbf{INT1}}$)
- Timer0 interrupt (TF0)
- Timer1 interrupt (TF1)
- Serial Port Interrupt (RI OR TI)

The Timer and Serial interrupts are internally generated by the microcontroller, whereas the external interrupts are generated by additional interfacing devices or switches that are externally connected to the microcontroller.

The interrupts of 8051 can be handled or controlled by the two SFR registers, they are IE and IP.

IE: Interrupt Enable register

Interrupt enable register is to enable or disable the particular interrupt, either external or internal.

EA	---	ET2	ES	ET1	EX1	ET0	EX0

- EA – This enables or disable the all the interrupts. If EA = 0, all interrupts are disabled. If EA =1, we can control each interrupt individually.
- ---- not implemented, reserved for future use.
- ET2 – this enables or disables Timer2 interrupt (in 8052) ES – this enables or disables Serial port interrupt
- ET1 - this enables or disables Timer1 interrupt
- EX1- - this enables or disables External interrupt1
- ET0 – this enables or disables Timer0 interrupt
- EX0 – this enables or disables External interrupt0

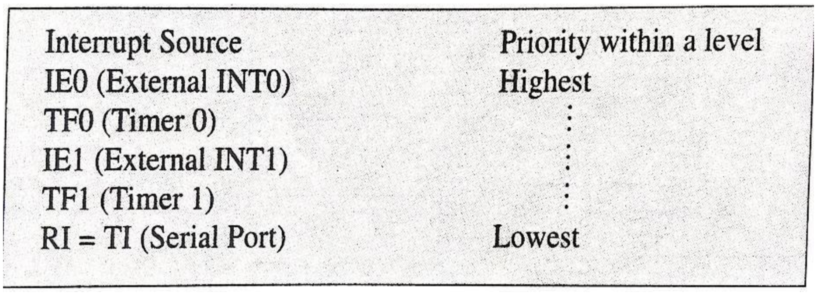
IP: Interrupt priority

The interrupt structure of 8051 provides two levels of the interrupt priorities for its sources of interrupts. Each interrupt source can be programmed to have one of these two levels using **Interrupt priority register (IP).**

---	---	PT2	PS	PT1	PX1	PT0	PX0

- ---- not used, reserved for future use.
- PT2 – defines the priority level of the Timer2 interrupt (in 8052).
- PS – defines the priority level of the Serial port interrupt.
- PT1 – defines the priority level of the Timer1 interrupt
- PX1 – defines the priority level of the External interrupt1 (**$\overline{\text{INT1}}$**)
- PT0 – defines the priority level of the Timer0 interrupt0.
- PX0 – defines the priority level of the External interrupt0 (**$\overline{\text{INT0}}$**).

The different sources of interrupts programmed with same level of priority, further follow a sequence of priority as shown in below



Interrupt Source	Priority within a level
IE0 (External INT0)	Highest
TF0 (Timer 0)	⋮
IE1 (External INT1)	⋮
TF1 (Timer 1)	⋮
RI = TI (Serial Port)	Lowest

5.13 Describe different Modes of operation in 8051?

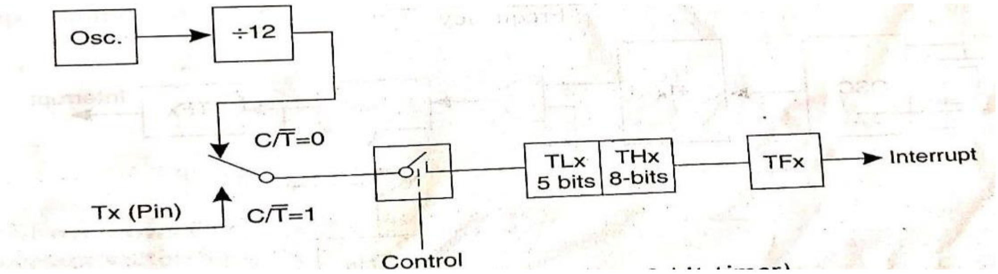
Timer modes:

The timers may operate in any one of four modes that are determined by the mode bits, M1 and M0 in the TMOD register. Figure shows the four timer modes.

M1	**M0**	**Timer Mode**	**Description of Mode**
0	0	Mode 0	13-bit Timer
0	1	Mode 1	16-bit Timer
1	0	Mode 2	8-bit auto-reload timer
1	1	Mode 3	8-bit split timer

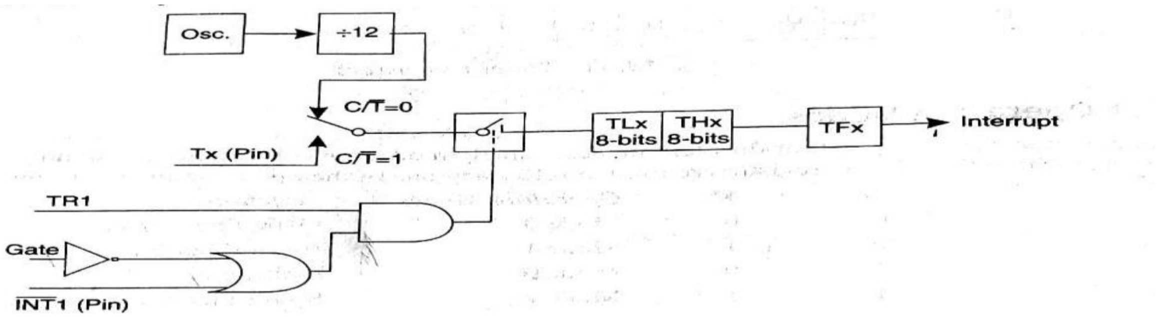
Mode 0 (13-bit timer):

Setting timer mode bits M1= 0 and M0= 0 in the TMOD register results in using the THx register as an 8-bit counter and TLx as a 5-bit counter (13-bit timer). The 13-bit counter can count values between 0000H to 1FFFH. Therefore, when the timer reaches its maximum of 1FFFH, it rolls over to 0000, and TF bit is set in TCON register. The TFx bit can be cleared by servicing Timer'x' interrupt.



Mode 1(16-bit timer):

Setting timer mode bits M1=0 and M0=1, Mode 1 in TMOD register results in a 16-bit timer. In timer mode 1 it acts as 16-bit timer, therefore it allows values from 0000H to FFFFH to be loaded into the timer's registers TLx and THx. The 16-bit timer can count values from 0000H – FFFFH. Therefore, when the timer reaches its maximum of FFFFH, it rolls over to 0000H, and TFx bit is set in TCON register. The TFx bit can be cleared by servicing Timer'x' interrupt.

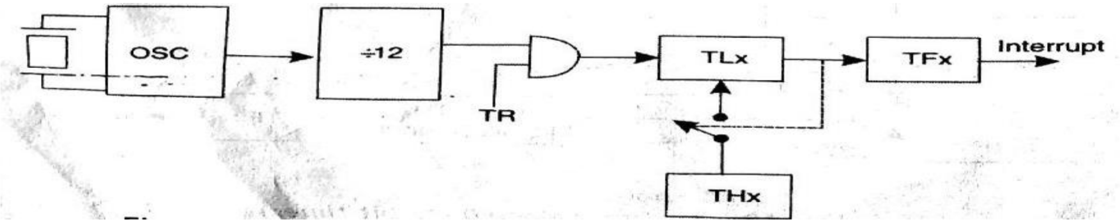


Mode 1(16-bit timer)

Mode 2 (8-bit auto reload timer):

By setting timer mode bits, M1=1 and M0=0, TLx (timer lower byte) acts as 8-bit timer and THx (Timer higher byte) is used to hold a value that is loaded into TLx every time when TLx overflows from FFH to 00H. The timer overflow flag (TFx) is also set when TLx overflows. The TFx bit can be cleared by servicing Timer'x' interrupt.

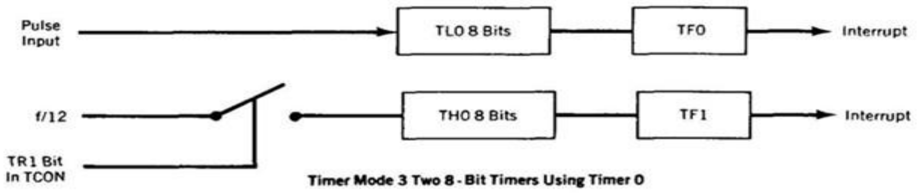
After timer is started, it starts to count up by incrementing the TL register. It counts up until it reaches its limit of FFH. When it exceeds the maximum value FFH, the TFx bit is set and TLx is reloaded automatically with the original value kept by the THx register. So the content of THx is automatically loaded to TLx.



Mode 3: (8-bit split timer)

Mode 3 is also known as a split timer mode. By setting timer mode bits, M1=1 and M0=1, mode 3 is selected. In mode 3 16-bit Timer0 is acts as two independent 8-bit timers, i.e., TL0 register is used as 8-bit timer and TH0 register acts as another 8-bit timer. In mode 3 Timer1 is not used.

In TL0 timer, when it exceeds its maximum limit FFH, Timer0 overflow flag bit (TF0) is set in TCON register. In TH0 timer, when it exceeds its maximum limit FFH, Timer1 overflow flag bit (TF1) is set in TCON register. Thus, Timer0 control bits are used to control TL0 8-bit timer and Timer1 control bits are used to control TH0 8- bit timer.



Timer Mode 3 Two 8 - Bit Timers Using Timer 0

Serial Transmission modes:

8051 has four modes of serial data transmission. Modes are selected by the programmer by setting the mode bits SM0 and SM1 in SCON.

SM0	SM1	Mode	Description
0	0	0	Shift register; baud = f/12
0	1	1	8-bit UART; baud = variable
1	0	2	9-bit UART; baud = 1/32 or f/64
1	1	3	9-bit UART; baud = variable

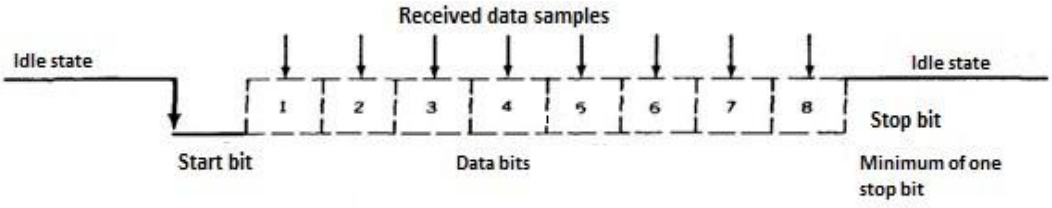
Mode 0: Shift register

Setting bits **SM0 = 0** and **SM1= 0** in SCON register, Mode0 is selected. In Mode0 the serial port acts shift register. In mode0, we can receive or transmit eight data bits using pin **RXD** only. In this, only **Half-duplex communication** is possible with fixed Baud rate 1/12 of the oscillator frequency.

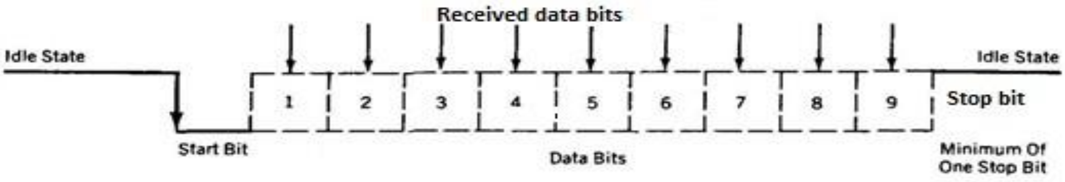
Mode 1: 8-bit UART

By setting Serial mode select bits **SM1 = 0** and **SM0 = 1**, we can select **Mode1.** In Mode1, it acts as 8-bit UART with variable baud rate. In this mode SBUF becomes a **10-bit full-duplex** receiver/ transmitter that may receive and transmit data at the same time. Pin RXD is used to receive data, and pin TXD is used to transmit data.

If we want to transmit 8-bit data, we have to transmit two additional bits called **Start bit** and **Stop bit.** Start bit (0) is added at the starting of 8-bit data and Stop bit (1) is added at the end of the 8-bit data.



Mode 2: 9-bit UART :- By setting Serial mode select bits **SM1 = 1** and **SM0 = 0**, we can select **Mode2.** In Mode2, it acts as 9-bit UART. In this mode 11-bits are transmitted, one start bit, 9 data bits and one stop bit. In this mode the serial communication is done with baud rate of F$_{OSC}$/32 (or) F$_{OSC}$/64. When transmitting the 9-bit data, the 9th bit is stored as TB8 in TCON register and while receiving 9-bit data, the 9th bit stored in RB8 bit position. Both the start and stop bits are discarded after receive data.



Mode 3: 9-bit UART with variable baud rate

By setting Serial mode select bits **SM1 = 1** and **SM0 = 0**, we can select **Mode3**. The mode3 is same as mode2, but mode3 we can use variable baud rate by using Timer1.

Mode2 and Mode3 are used for multiprocessor communication.