MODULE 2

- **8051 Architecture**

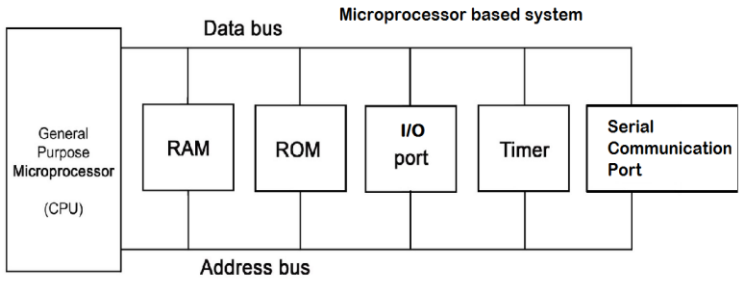
Architecture – Block diagram of 8051, Pin configuration, Registers, Internal Memory, Timers, Port Structures, Interrupts, Addressing Modes, Instruction set (Brief study of 8051 instruction set is sufficient)

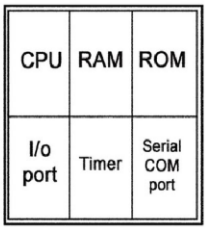
Course Outcome

- After completion of the module the student will be able to
  - Familiarize the instruction set of 8051 and perform assembly language programming

Comparison between Microprocessor and Microcontroller

MP	MC
Chip contains CPU only. There is no on chip RAM, ROM , Timer, I/O etc.	Chip contains RAM, ROM, I/O, Timers etc. in addition to CPU
Usually used in general purpose computers	Usually used for control oriented applications
Mp based systems are bulkier and costlier since RAM, ROM etc have to be interfaced externally	Mc based systems are compact since RAM ROM etc are built in the chip
Mp based systems are versatile since RAM, ROM, I/O etc are decided by the programmer.	Mc based systems have fixed amount of RAM, ROM, I/O etc. If the programmer changes these it loses the advantage of being compact and cheaper
Access times for I/O and memory is more since they are external	Access times for I/O and memory is less since they are internal
Instruction set focuses on processing intensive operations	Instruction set focuses on control intensive operations
Large number of instructions with flexible addressing mode	Less number of instructions with few addressing mode
Few pins have multiple function	More pins are multi functioned
Very few instructions have bit handling capacity	Many instructions have bit handling capacity
Eg: Intel 8085, 8086, Pentium etc	Eg: Intel 8051, Microchip PIC etc





Microcontroller

Microcontroller Families

Types of Microcontrollers

Microcontrollers are divided into various categories based on memory, architecture, bits and instruction sets. Following is the list of their types –

Based on bit configuration

- 4-**bit microcontroller** – generally used for toys ,watches, time pieces and other small appliances Eg Epson S1C60 , Texas instruments TMS 1000
- **8-bit microcontroller** − This type of microcontrollers can perform arithmetic and logical operations like addition, subtraction, multiplication division, etc on byte sized data. They are also suitable for serial communications. For example, Intel 8031 and 8051 , Microchips PIC16C56 and ATMega32 are 8 bits microcontroller.
- **16-bit microcontroller** − This type of microcontroller is used to perform arithmetic and logical operations where higher accuracy and speed is required. For example, Intel 8096/80196 and Microchip PIC 24 F are 16-bit microcontrollers.
- **32-bit microcontroller** − generally used in robotics, avionics, automobiles etc. Can run application programs under an operating system. Microchip PIC 32, Intel 80960, Intel Quark, 32 bit ARM

Based on the instruction set configuration

- **CISC** − CISC stands for Complex Instruction Set Computer. It allows the user to insert a single instruction as an alternative to many simple instructions. The CISC approach attempts to minimize the number of instructions per program. Eg Motorola 68HC11, 8051
- **RISC** − RISC stands for Reduced Instruction Set Computers. RISC processors have simple instructions taking about one clock cycle. Eg ARM, AVR, PIC

Families:

8051 (MCS – 51) - is an 8-bit microcontroller. 8051 microcontroller has 32 I/O pins, timers/counters, interrupts and UART's. This family uses Harvard architecture..

PIC- (Programmable interface controller) microcontrollers are available 8-bit, 16-bit and 32-bit versions. This family uses Harvard architecture. PIC has nearly 40 instructions which all are take 4 clock cycles to execute. It has on-chip peripherals like SPI, ADC, I2C, UART, analog comparator, internal RC oscillator, in-system programmability, etc.

AVR- is an 8-bit RISC architecture microcontroller. There are 16-bit and 32-bit microcontrollers also available in the same family. AVR has 140 instructions which are all 1 cycle based instructions. By default AVR microcontrollers operate with the 1 MHz clock cycle. AVR family microcontroller has on-chip boot-loader. By this we can program our microcontroller easily without any external programmer. AVR controllers has number of I/O ports, timers/counters, interrupts, A/D converters, USART, I2C interfaces, PWM channels, on-chip analog comparators.

ARM- ARM is 32 bit Microcontroller whose core is designed by ARM Limited with RISC architecture. ARM has von Neumann architecture (program and RAM in the same space). ARM Microcontrollers are extremely used in power saving and operate in very low power consumption. ARM executes most instructions in a single clock cycle. ARM Microcontrollers are widely used in mobile phones.

8051- Introduction

8051 is one of the first most popular microcontroller also known as MCS-51. It was introduced by Intel in the year 1981. Initially it came out as N-type metal-oxide-semiconductor (NMOS) based microcontroller, but later versions were based on complementary metal-oxide-semiconductor (CMOS) technology. These microcontrollers were named as 80C51, where C in the name tells that it is based on CMOS technology.

8051 microcontroller was designed by Intel in 1981. It is an 8-bit microcontroller. It is built with 40 pins DIP (dual inline package), 4kb of ROM storage and 128 bytes of RAM storage, 2 16-bit timers. It consists of are four parallel 8-bit ports, which are programmable as well as addressable as per the requirement. An on-chip crystal oscillator is integrated in the microcontroller having crystal frequency of 12 MHz.

It is an 8-bit microcontroller which means data bus is of 8-bits. Therefore, it can process 8-bits at a time. It is used in wide variety of embedded systems like robotics, remote controls, automotive industry, telecom applications, power tools etc.

Features –

There are some key features of 8051 which works as a foundation for students to learn microcontrollers. These features include:-

- 4 KB on-chip ROM (Program memory).
- 128 bytes on-chip RAM (Data memory).
- 8 bit data bus (bidirectional).
- 16 bit address bus (unidirectional).
- Two 16-bit timers.
- Instruction cycle of 1 microsecond with 12 MHz crystal.
- Four 8-bit ports.
- 128 user defined flags.
- Four register banks of 8 bit each.
- 16-byte bit-addressable RAM.

FUNCTIONAL BLOCK DIAGRAM OF 8051 MICROCONTROLLER

- The Intel 8051 contains two separate buses for both program and data. So, it has two distinctive memory spaces of 64K size for both program and data.

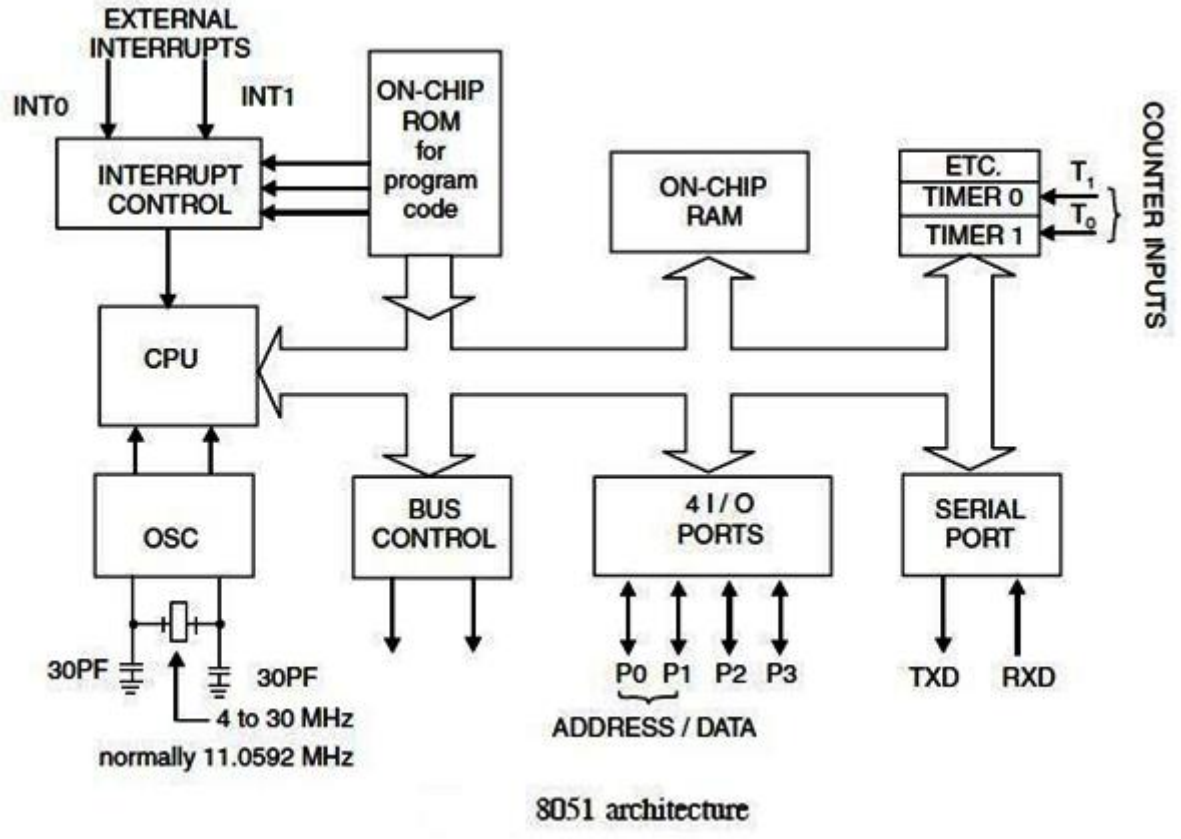


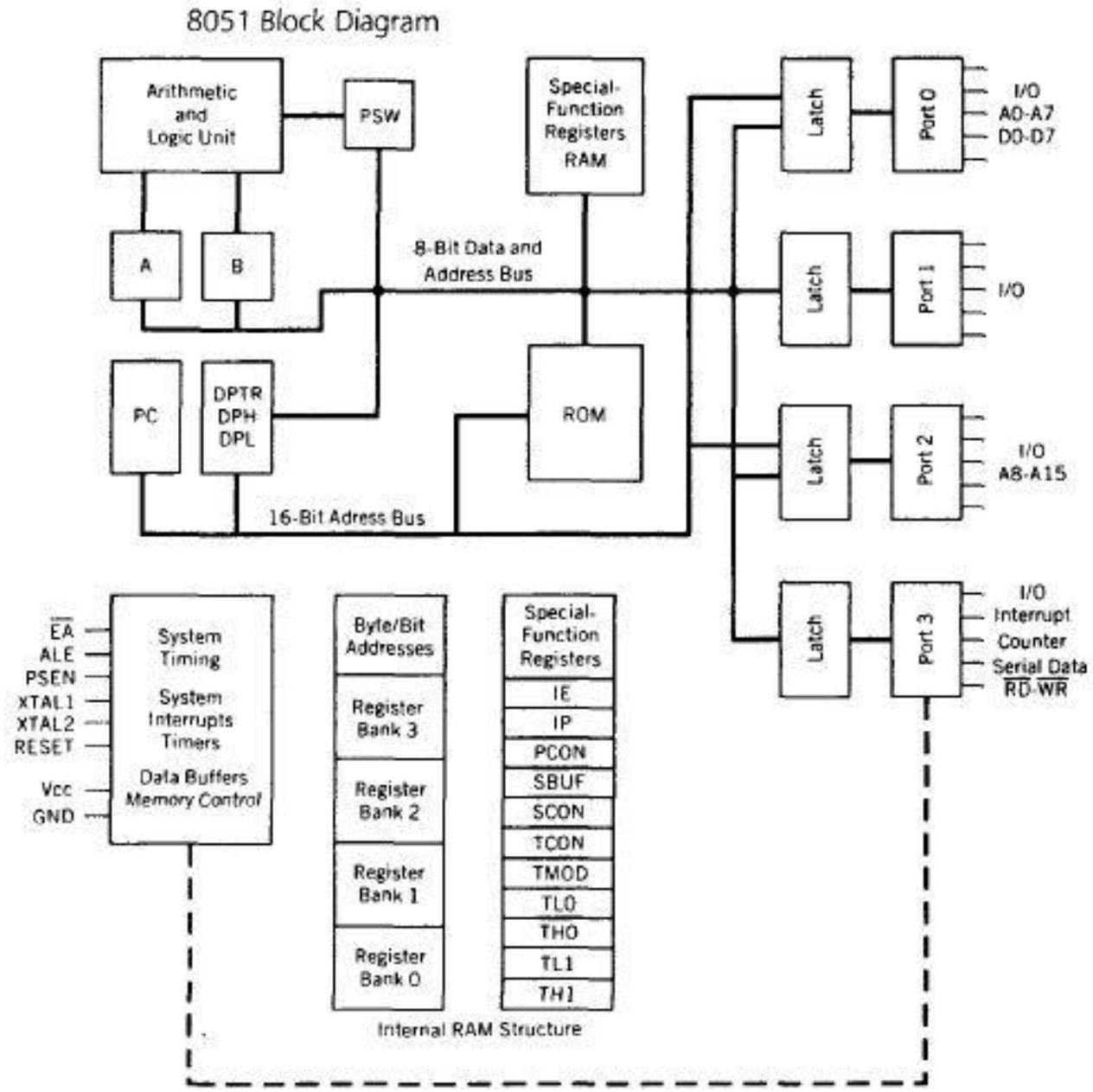
Fig: Block Diagram of 8051

- It is based on an 8 bit central processing unit with an 8 bit accumulator and another 8-bit B register as main processing blocks.
- Other portions of the architecture include few 8 bit and 16b it registers and 8-bit memory locations.
- It has some amount of data RAM built in the device for internal processing. This area is used for stack operations and temporary storage of data.
- It has a TIMING AND CONTROL UNIT which derives all the necessary timing and control signals recovered for the internal operation of the circuit. It also derives control signals recovered for controlling the system bus.
- OSCILLATOR circuit generates the basic timing clock signal for the operation of the circuit

using crystal oscillator

- 8051 is supported with on-chip peripheral functions like I/O ports, Timers/ Counters, Serial communication port.

Architecture - 8051



CENTRAL PROCESSING UNIT

- The CPU is the brain of the microcontrollers expected task reading user's programs and executing the as per instructions stored there in.

- Its primary elements are an Accumulator (AC), Stack Pointer (SP) Program Counter (PC), Program Status Word (PSW), Data Pointer (DTPR) and few more 8-bit register.

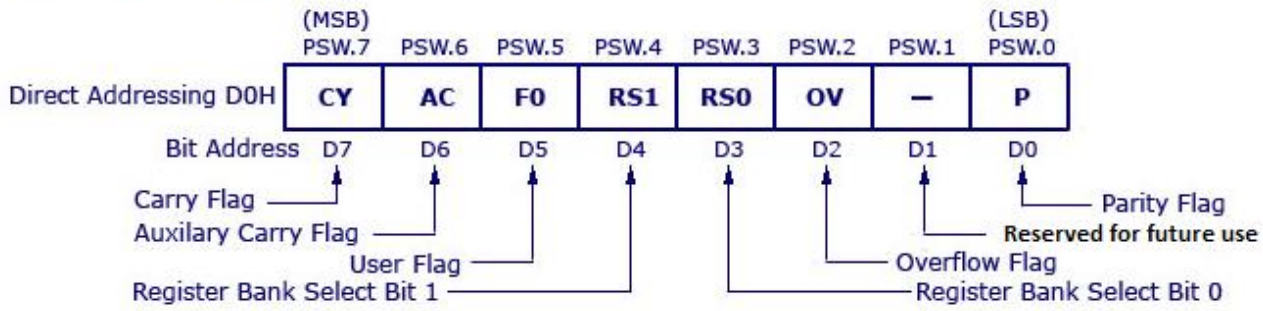
- **ARITHMETIC LOGIC UNIT (ALU)**
  - The arithmetic / logic unit Performs arithmetic and logic operations like addition, subtraction, AND, OR etc. It includes the accumulator, temporary register, arithmetic and logic circuits.
  - The temporary register is used to hold data during an arithmetic / logic operation.
  - The result is stored in the accumulator register.

CPU registers A and B

- **A –accumulator**
  - The accumulator register (ACC or A) act as an operand register, in case of some instructions.
  - It is an 8 bit register.
  - It is used to store data.
  - Used in all ALU operations
  - Used for all data transfer between 8051 and external memory
- **B Register**
  - It is an 8 bit register.
  - It is used to store data.
  - Used for division and multiplication operations
  - Also used for 8 bit data storage when not used by multiplication and division instructions
- **PSW**

The program status word (PSW) register, also referred to as the flag register, is an 8 bit register.

Program Status Word



Only 6 bits are used

- Among these four are CY (carry), AC (auxiliary carry), P (parity), and OV (overflow)
    - They are called conditional flags, meaning that they indicate some conditions that resulted after an instruction was executed
    - Other two are PSW3 and PSW4, designed as RS0 and RS1, and are used to change the bank
    - The two unused bits are user-definable.

CY- Carry Flag: This flag is set if there is a carry out from the D7 bit. This flag is affected by addition and subtraction operations and instructions that set or clear it directly.

AC- Auxiliary Carry Flag: This flag is set if there is a carry out from the D3 to D4; otherwise it is cleared. It is used for BCD operations

F0- User Flag 0- User defined flag

RS1- Register bank select bit 1

RS0- Register bank select bit 0

RS1	RS0	
0	0	Select register bank 0
0	1	Select register bank 1
1	0	Select register bank 2
1	1	Select register bank 3

These bits are used to select the current register bank from among the 4 register banks of 8051. Their default value is 00 ***ie Bank0 is the default register bank.***

OV- overflow flag: This flag is set if there is an overflow into the sign bit. This occurs when the result of a signed operation is too large causing the higher order bit to overflow into the sign bit.

P- Parity Flag: If the accumulator contains **odd number of 1s then P=1 else P=0**

- *All flags can be set or cleared by user*
- *Math flags-C,AC,OV and P- are also affected by ALU operations*
- *Of the three general purpose flags F0 is placed in the PSW and GF0 and GF1 are placed in PCON*

- **Stack and Stack Pointer (SP)**
  - Stack is a part of internal RAM used to store and retrieve data quickly
  - LIFO – stack operates in last in first out manner
  - Stack grows up as data is stored
  - **Stack pointer** (SP) is an 8 bit register which holds the address of top of stack
  - Default value of SP is 07
  - Value of SP can be changed to any internal RAM address
  - PUSH and POP instructions are used to store and retrieve data

Program counter and Data pointer

- 8051 has two 16-bit registers: PC and DPTR
- These registers hold the address of a byte in memory.

- **Program Counter (PC)**
  - Stores the address of the next instruction to be executed
  - Automatically incremented after each instruction fetch
  - Does not have an internal address(the only such register)
  - Default value 0000h
- **DPTR**
  - Made of two 8 bit registers DPH and DPL
  - Used for external memory access (code and data)
- **ROM**

4 kb internal ROM for program storage

- **RAM**

128 bytes **Internal RAM**

1. **Register banks-32 bytes**
   - Organized as 4 register banks
   - Each register bank contains 8 registers –R0 to R7
2. **Bit addressable memory-16 bytes**
   - 128 bits
   - May be addressed by bit address or by byte address(as 8 bits)
3. **General Purpose (scratch pad) RAM-80 bytes**
   - Byte addressable

- **Special Function Registers (SFRs)**
  - This is a set of special function registers, which can be addressed using their respective addresses which lie in the range 80H to FFH
  - SFRs control specific functions of the 8051 Microcontroller i.e, Each SFR has a very specific function.
  - Each SFR has an address (within the range 80h to FFh) and a name which reflects the purpose of the SFR.
    - For example the accumulator is a special function register having the address E0h.
  - There **are 21 Special function registers (SFR)** in 8051 micro controller and this includes I/O Port Registers (P0, P1, P2 and P3), PSW (Program Status Word), A (Accumulator), B register ,PCON (Power Control) and others like IP, IE, TMOD, TCON, SCON etc contain control and status information for interrupts, Timer / Counters and serial port.

- **I/O PORTS**
  - To communicate data with the external world the microcontroller needs ports.
  - The ports may support either parallel or serial data transfer.
  - It has 4 I/O ports namely, **Port 0, Port 1, Port 2 & Port 3**
  - 32 i/o pins arranged as four 8-bit ports
    - Port 1: is exclusively for input & output functions.
    - Port 0, 2 & 3: perform functions other than parallel data transfer.
    - All 4 ports are bidirectional.
    - The 8 port pins are connected through 8 D type port latches.

- **TIMERS**

  Many microcontroller applications require the generation of precise internal time delays between computer actions or counting of external events. These are carried out by timers/ counters.
  - The 8051 has two 16-bit timer/counters named T0 and T1.
  - These two timer/counters can be programmed independently.

- There is a bit in the TMOD SFR that specifies whether it is a timer or a counter.
    - If this bit is set, the timer/counter will work as a counter; and
    - if this bit is reset, the timer/counter will work as a timer.

➢ **SERIAL I/O**

Microcontrollers must be able to communicate with other microcontrollers or computers. One cost effective way to communicate is to send and receive data bits serially. Serial transmission and reception can be easily achieved using UART devices (Universal Asynchronous Receiver Transmitter). 8051 has a built-in UART.

- It is a Full duplex serial data receiver transmitter

➢ **INTERRUPTS**

An interrupt is an internal or external event that interrupts the microcontroller to inform it that a device needs its service. Whenever any device needs its service, the device notifies the microcontroller by sending it as interrupt signal. Upon receiving an interrupt signal, the microcontroller interrupts whatever it is doing and serves the device. The program which is associated with the interrupt is called interrupt Service Routine (ISR). The microcontroller can serve many devices based on the priority assigned to it.

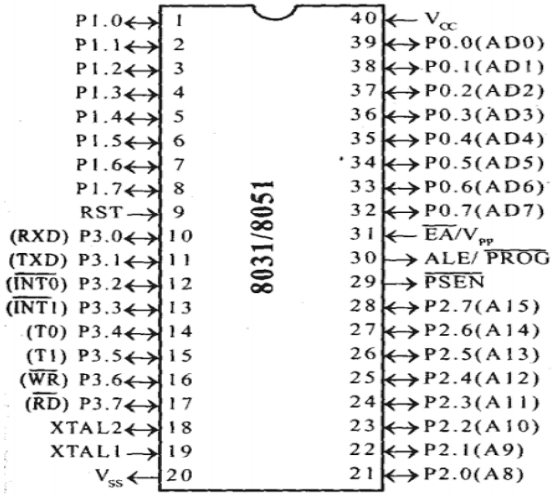
- There are 2 external and 3 internal interrupts
- External interrupts- INT0 , INT1
- Internal interrupts- timer0, timer1 and serial communication

➢ **OSCILLATOR AND CLOCK**

- All internal operations are synchronized with clock
- Quartz crystal and capacitors are used to form the resonant network that generates clock

❖ Pin Diagram

8051 microcontroller is a 40 pin Dual Inline Package (DIP). These 40 pins serve different functions like read, write, I/O operations, interrupts etc. 8051 has four I/O ports wherein each port has 8 pins which can be configured as input or output depending upon the logic state of the pins. Therefore, 32 out of these 40 pins are dedicated to I/O ports. The rest of the pins are dedicated to VCC, GND, XTAL1, XTAL2, RST, ALE, EA' and PSEN'.



Description of the Pins:

- **Pin 1 to Pin 8 (Port 1) –**

  Pin 1 to Pin 8 are assigned to Port 1 for simple I/O operations. They can be configured as input or output pins depending on the logic control i.e. if logic zero (0) is applied to the I/O port it will act as an output pin and if logic one (1) is applied the pin will act as an input pin. These pins are also referred to as P1.0 to P1.7 (where P1 indicates that it is a pin in port 1 and the number after '.' tells the pin number i.e. 0 indicates first pin of the port. So, P1.0 means first pin of port 1, P1.1 means second pin of the port 1 and so on). These pins are bidirectional pins.

- **Pin 9 (RST) –**

  Reset pin. It is an active-high, input pin. Therefore if the RST pin is high for a minimum of 2 machine cycles, the microcontroller will reset i.e. it will close and terminate all activities. It is often referred as "power-on-reset" pin because it is used to reset the microcontroller to it's initial values when power is on (high).

- **Pin 10 to Pin 17 (Port 3) –**

  Pin 10 to pin 17 are port 3 pins which are also referred to as P3.0 to P3.7. These pins are similar to port 1 and can be used as universal input or output pins. These pins are bidirectional pins.

These pins also have some additional functions which are as follows:

- **P3.0 (RXD) :**

  10th pin is RXD (serial data receive pin) which is for serial input. Through this input signal microcontroller receives data for serial communication.

- **P3.1 (TXD) :**

  11th pin is TXD (serial data transmit pin) which is serial output pin. Through this output signal microcontroller transmits data for serial communication.

- **P3.2 and P3.3 (INT0′, INT1′) :**

  12th and 13th pins are for External Hardware Interrupt 0 and Interrupt 1 respectively. When this interrupt is activated(i.e. when it is low), 8051 gets interrupted in whatever it is doing and jumps to the vector value of the interrupt (0003H for INT0 and 0013H for INT1) and starts performing Interrupt Service Routine (ISR) from that vector location.

- **P3.4 and P3.5 (T0 and T1) :**

  14th and 15th pin are for Timer 0 and Timer 1 external input. They can be connected with 16 bit timer/counter.

- **P3.6 (WR') :**

  16th pin is for external memory write i.e. writing data to the external memory.

- **P3.7 (RD') :**

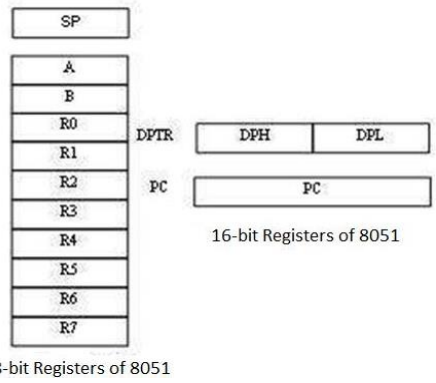
  17th pin is for external memory read i.e. reading data from external memory.

- **Pin 18 and Pin 19 (XTAL2 And XTAL1) :**

  These pins are connected to an external oscillator which is generally a quartz crystal oscillator. They are used to provide an external clock frequency of 4MHz to 30MHz.

❖ REGISTERS

Registers are used in the CPU to store information on temporarily basis which could be data to be processed, or an address pointing to the data which is to be fetched. In 8051, there is one data type is of 8-bits, from the MSB (most significant bit) D7 to the LSB (least significant bit) D0. With 8-bit data type, any data type larger than 8-bits must be broken into 8-bit chunks before it is processed.



The most widely used registers of the 8051 are A (accumulator), B, R0-R7, DPTR (data pointer), and PC (program counter). All these registers are of 8-bits, except DPTR and PC.

- **Accumulator**

The accumulator, register A, is used for all arithmetic and logic operations. If the accumulator is not present, then every result of each calculation (addition, multiplication, shift, etc.) is to be stored

into the main memory. Access to main memory is slower than access to a register like the accumulator because the technology used for the large main memory is slower (but cheaper) than that used for a register.

- **The "R" Registers**

The "R" registers are a set of eight registers, namely, R0, R1 to R7. These registers function as auxiliary or temporary storage registers in many operations. Thus "R" registers are very important auxiliary or **helper registers**. The Accumulator alone would not be very useful if it were not for these "R" registers. The "R" registers are meant for temporarily storage of values.

- **The "B" Register**

The "B" register is very similar to the Accumulator in the sense that it may hold an 8-bit (1-byte) value. The "B" register is used only by two 8051 instructions: **MUL AB** and **DIV AB**. To quickly and easily multiply or divide A by another number, you may store the other number in "B" and make use of these two instructions. Apart from using MUL and DIV instructions, the "B" register is often used as yet another temporary storage register, much like a ninth R register.

- **The Data Pointer**

The Data Pointer (DPTR) is the 8051's only user-accessible 16-bit (2-byte) register. The Accumulator, R0–R7 registers and B register are 1-byte value registers. DPTR is meant for pointing to data. It is used by the 8051 to access external memory using the address indicated by DPTR. DPTR is the only 16-bit register available and is often used to store 2-byte values.

- **The Program Counter**

The Program Counter (PC) is a 2-byte address which tells the 8051 where the next instruction to execute can be found in the memory. PC starts at 0000h when the 8051 initializes and is incremented every time after an instruction is executed. PC is not always incremented by 1. Some instructions may require 2 or 3 bytes; in such cases, the PC will be incremented by 2 or 3.

Branch, jump, and **interrupt** operations load the Program Counter with an address other than the next sequential location. Activating a power-on reset will cause all values in the register to be lost. It means the value of the PC is 0 upon reset, forcing the CPU to fetch the first opcode from the ROM location 0000. It means we must place the first byte of upcode in ROM location 0000 because that is where the CPU expects to find the first instruction.

- **The Stack Pointer (SP)**

The Stack Pointer, like all registers except DPTR and PC, may hold an 8-bit (1-byte) value. The Stack Pointer tells the location from where the next value is to be removed from the stack. When a value is pushed onto the stack, the value of SP is incremented and then the value is stored at the resulting memory location. When a value is popped off the stack, the value is returned from the memory location indicated by SP, and then the value of SP is decremented.

This order of operation is important. SP will be initialized to 07h when the 8051 is initialized. If a value is pushed onto the stack at the same time, the value will be stored in the internal RAM address 08h because the 8051 will first increment the value of SP (from 07h to 08h) and then will store the pushed value at that memory address (08h). SP is modified directly by the 8051 by six instructions: PUSH, POP, ACALL, LCALL, RET, and RETI.

- **8051 Flag Bits and PSW Register**

The program status word (PSW) register is an 8-bit register, also known as **flag register**. It is of 8-bit wide but only 6-bit of it is used. The two unused bits are **user-defined flags**. Four of the flags are called **conditional flags**, which mean that they indicate a condition which results after an instruction is executed. These four are **CY** (Carry), **AC** (auxiliary carry), **P** (parity), and **OV** (overflow). The bits RS0 and RS1 are used to change the bank registers. The following figure shows the program status word register.

The PSW Register contains that status bits that reflect the current status of the CPU.

CY	CA	F0	RS1	RS0	OV	-	P

CY	PSW.7	Carry Flag
AC	PSW.6	Auxiliary Carry Flag
F0	PSW.5	Flag 0 available to user for general purpose.
RS1	PSW.4	Register Bank selector bit 1
RS0	PSW.3	Register Bank selector bit 0
OV	PSW.2	Overflow Flag
-	PSW.1	User definable FLAG
P	PSW.0	Parity FLAG. Set/ cleared by hardware during instruction cycle to indicate even/odd number of 1 bit in accumulator.

We can select the corresponding Register Bank bit using RS0 and RS1 bits.

RS1	RS2	Register Bank
0	0	0
0	1	1
1	0	2
1	1	3

- **CY, the carry flag** − This carry flag is set (1) whenever there is a carry out from the D7 bit. It is affected after an 8-bit addition or subtraction operation. It can also be reset to 1 or 0

directly by an instruction such as "SETB C" and "CLR C" where "SETB" stands for set bit carry and "CLR" stands for clear carry.

- **AC, auxiliary carry flag** − If there is a carry from D3 and D4 during an ADD or SUB operation, the AC bit is set; otherwise, it is cleared. It is used for the instruction to perform binary coded decimal arithmetic.
- **P, the parity flag** − The parity flag represents the number of 1's in the accumulator register only. If the A register contains odd number of 1's, then P = 1; and for even number of 1's, P = 0.
- **OV, the overflow flag** − This flag is set whenever the result of a signed number operation is too large causing the high-order bit to overflow into the sign bit. It is used only to detect errors in signed arithmetic operations.

Example

Show the status of CY, AC, and P flags after the addition of 9CH and 64H in the following instruction.

MOV A, #9CH

ADD A, # 64H

```
Solution:  9C    10011100
          +64    01100100
          100    00000000
CY = 1 since there is a carry beyond D7 bit
AC = 0 since there is a carry from D3 to D4
P  = 0 because the accumulator has even number of 1's
```

Example:

1) Show the status of the CY, AC and P flag after the addition of 38H and 2FH.

38	00111000
+ 2F	00101111
67	01100111

CY = 0, since there is no carry beyond the D7 bit

AC = 1, since there is a carry from the D3 to the D4 bit

P = 1, since the accumulator has an odd number of 1s (it has five 1s)

2) Show the status of the CY, AC and P flag after the addition of 9CH and 64H.

9C	10011100
+ 64	01100100
100	00000000

CY = 1, since there is a carry beyond the D7 bit

AC = 1, since there is a carry from the D3 to the D4 bi

P = 0, since the accumulator has an even number of 1s (it has zero 1s)

3) Show the status of the CY, AC and P flag after the addition of 88H and 93H.

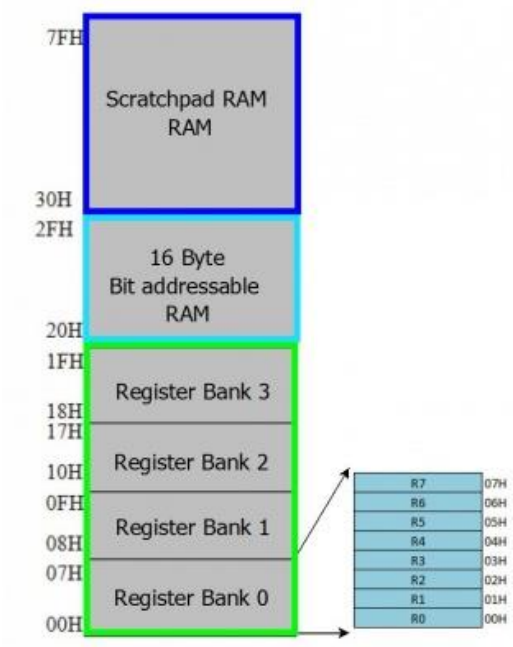
88	10001000
+ 93	10010011
11B	00011011

- **CY** = **1,** since there is a carry beyond the D7 bit
- **AC** = **0,** since there is no carry from the D3 to the D4 bi
- **P** = **0,** since the accumulator has an even number of 1s (it has four 1s)

❖ INTERNAL MEMORY

RAM Memory Space Allocation in 8051

The 8051 microcontroller has a total of 128 bytes of RAM. We will discuss about the allocation of these 128 bytes of RAM and examine their usage as stack and register.



The 128 bytes of RAM inside the 8051 are assigned the address 00 to 7FH. They can be accessed directly as memory locations and are divided into three different groups as follows −

- 32 bytes from 00H to 1FH locations are set aside for register banks and the stack.
- 16 bytes from 20H to 2FH locations are set aside for bit-addressable read/write memory.
- 80 bytes from 30H to 7FH locations are used for read and write storage; it is called as **scratch pad**. These 80 locations RAM are widely used for the purpose of storing data and parameters by 8051 programmers.

- **Register Banks in 8051**

A total of 32 bytes of RAM are set aside for the register banks and the stack. These 32 bytes are divided into four register banks in which each bank has 8 registers, R0–R7. RAM locations from 0 to 7 are set aside for bank 0 of R0–R7 where R0 is RAM location 0, R1 is RAM location 1, R2 is location 2, and so on, until the memory location 7, which belongs to R7 of bank 0.

The second bank of registers R0–R7 starts at RAM location 08 and goes to locations OFH. The third bank of R0–R7 starts at memory location 10H and goes to location to 17H. Finally, RAM locations 18H to 1FH are set aside for the fourth bank of R0–R7.

➢ **Default Register Bank**

If RAM locations 00–1F are set aside for the four registers banks, which register bank of R0–R7 do we have access to when the 8051 is powered up? The answer is register bank 0; that is, RAM locations from 0 to 7 are accessed with the names R0 to R7 when programming the 8051. Because it is much easier to refer these RAM locations by names such as R0 to R7, rather than by their memory locations.

➢ **How to Switch Register Banks**

Register bank 0 is the default when the 8051 is powered up. We can switch to the other banks using PSW register. D4 and D3 bits of the PSW are used to select the desired register bank, since they can be accessed by the bit addressable instructions SETB and CLR. For example, "SETB PSW.3" will set PSW.3 = 1 and select the bank register 1.

RS1	RS2	Register Bank
0	0	0
0	1	1
1	0	2
1	1	3

- **Stack and its Operations**
  - **Stack in the 8051**

The stack is a section of a RAM used by the CPU to store information such as data or memory address on temporary basis. The CPU needs this storage area considering limited number of registers.

How Stacks are Accessed

As the stack is a section of a RAM, there are registers inside the CPU to point to it. The register used to access the stack is known as the stack pointer register. The stack pointer in the 8051 is 8-bits wide, and it can take a value of 00 to FFH. When the 8051 is initialized, the SP register contains the value 07H. This means that the RAM location 08 is the first location used for the stack. The storing operation of a CPU register in the stack is known as a **PUSH**, and getting the contents from the stack back into a CPU register is called a **POP**.

- **Pushing into the Stack**

In the 8051, the stack pointer (SP) points to the last used location of the stack. When data is pushed onto the stack, the stack pointer (SP) is incremented by 1. When PUSH is executed, the contents of the register are saved on the stack and SP is incremented by 1. To push the registers onto the stack, we must use their RAM addresses. For example, the instruction "PUSH 1" pushes register R1 onto the stack.

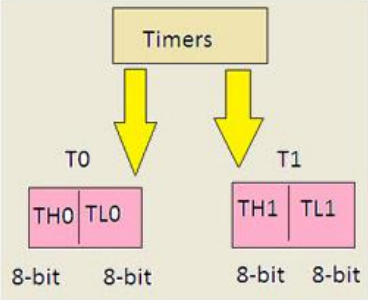
- **Popping from the Stack**

Popping the contents of the stack back into a given register is the opposite to the process of pushing. With every pop operation, the top byte of the stack is copied to the register specified by the instruction and the stack pointer is decremented once.

❖ TIMERS

Many of the microcontroller applications require counting of external events such as frequency of the pulse trains and generation of precise internal time delays between computer actions. Both these tasks can be implemented by software techniques, but software loops for counting, and timing will not give the exact result rather more important functions are not done. To avoid these problems, timers and counters in the micro-controllers are better options for simple and low-cost applications. These timers and counters are used as interrupts in 8051 microcontroller.

The 8051 has two timers, Timer 0 and Timer 1. They can be used as timers or as event counters. Both Timer 0 and Timer 1 are 16-bit wide. Since the 8051 follows an 8-bit architecture, each 16 bit is accessed as two separate registers of low-byte and high-byte. Timer can be used as a counter as well as for timing operation that depends on the source of clock pulses to counters.



Timers and counters

Counters and Timers in 8051 microcontroller contain two special function registers: TMOD (Timer Mode Register) and TCON (Timer Control Register), which are used for activating and configuring timers and counters.

Different Modes of Timers

There are four modes to operate the timers.

Mode 0: This is a 13-bit mode that means timer operation completes with “8192” pulses.

Mode 1: This is a16-bit mode, which means the timer operation completes with maximum clock pulses that “65535”.

Mode 2: This mode is an 8-bit auto reload mode, which means the timer operation completes with only “256” clock pulses.

Mode 3: Timer mode "3" is known as **split-timer mode**. When Timer 0 is placed in mode 3, it becomes two separate 8-bit timers.

❖ Port Structures

8051 has four input/ output ports to communicate with external world. These are named port 0, port 1, port 2 and port 3. Other than performing input / output operations, all ports except port 1 are allotted with additional functions.

- All ports are bidirectional.
- They are constructed with a D type output latch.
  - They have output drivers and input buffers.
  - The 8 port pins are connected through 8 D type port latches.
- We can modify their functions using software and hardware that they connect to.
- All the ports are configured as input ports on Reset.

- To configure ports as an input port ,1 must be written to that port

- To configure it as an output port 0, must be written to it.

Each port has a D-type output latch for each pin. The SFR for each port is made up of these eight latches; which can be, addressed at the SFR address for that port. For instance, the eight latches for port 0, can be addressed at location 80h; these are also bit addressable. The port latches should not be confused with the port pins; the data on the latches need not be the same as that on the pins. Two data paths are shown in Figure; upper buffer is enabled when latch data is read, and lower buffer is enabled when pin is read.

➢ Port 0

Port 0 pins may serve as inputs, outputs or, when used together, as a bi-directional low-order address and data bus for external memory.

Features:

- It is one of the SFR register with address 80H. It has dual functions.
- Port -0 has 8 pins (P0.0-P0.7).
- Port-0 can be configured 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 normal bi-directional I/O port.
- Port-0 latch is written to with 1's when used for external memory access.

Construction:

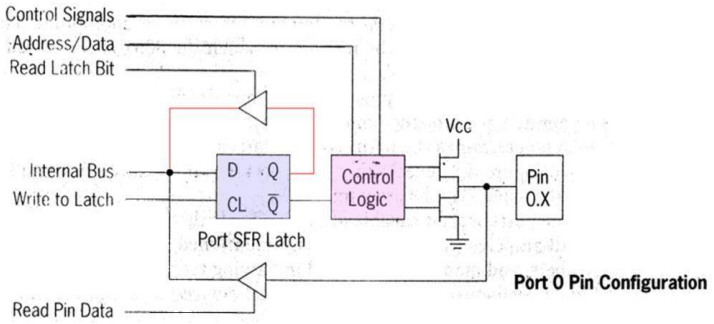
- Port 0 has a D-type latch, unidirectional buffer, and 2 FETs at each pin.

- It does not have an internal pull-up resistor.
- An external pull-up resistor is needed when Port 0 is defined as an output port.

Operation:

a) Port 0 as input port

When a pin is to be used as an input, a 1 must be written to the corresponding port 0 latch by the program, thus turning both of the output transistors off, which in turn causes the pin to "float" in a high-impedance state. The pin is connected to the input buffer.



Port 0 Pin Configuration

b) Port 0 as output port

When used as an output, the pin latches that are programmed to a 0 will turn on the lower FET grounding the pin. All latches that are programmed to a 1 still float; thus, external pullup resistors will be needed to supply a logic high when using port 0 as an output.

c) Port 0 to access external memory

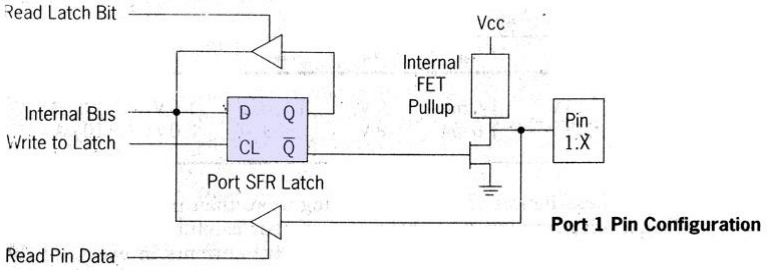
When port 0 is used as address bus to access external memory, internal control signals switch the address lines to the gates of the Field effect Transistors(FETs).

- A logic 1 on an address bit will turn the upper FET on and the lower FET off to provide a logic high at the pin.
- When the address bit is a zero, the lower FET is on and the upper FET off to provide a logic '0' at the pin.

After the address has been formed and latched into external circuits by the Address Latch Enable (ALE) pulse, the bus is turned around to become a data bus. Port 0 now reads data from the external memory and must be configured as input, so logic 1 is automatically written by internal control logic to all port 0 latches.

➢ Port 1

Port 1 pins have no dual functions .



Port 1 Pin Configuration

Features:

- Address is 90H.
- Port-1 has 8 pins (P1.1-P1.7)
- Port-1 does not have any alternate function i.e. it is dedicated solely for I/O interfacing.

Construction:

- Port 1 has one D latch, two unidirectional buffers, 1 FET, and one internal pull-up resistor at each pin.
- It has only one function – to act as an Input-Output port.

Operation:

- When Port 1 is functioning in the capacity of an input port, a digital '1' (FFH) is written to the latch. At 90H. This turns off the transistor, and the pin floats in a high impedance state. Consequently, it connects to the input buffer.
- When Port 1 is functioning in the capacity of an output port, the latch is given a 'LOW' signal (00H). This turns the FET (Field Effect Transistor) on. The pull-up resistor is OFF, and the port is used as an output port.

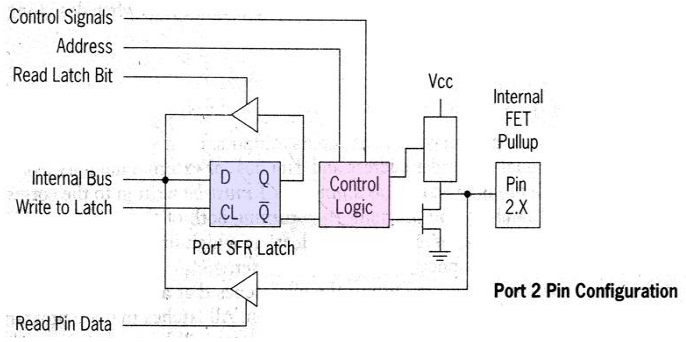
Therefore, the output latch is connected directly to the gate of the lower FET. This port has an FET circuit labeled Internal FET Pullup as an active pullup load.

➢ Port 2

Port 2 may be used as an input/output port similar in operation to port 1.The alternate use of port2 is to supply high-order address byte to address external memory.

Features:

- Address is 10H
- Port -2 has 8 pins (P2.0-P2.7).
- It has dual functions.
- Port-2 can be configured as a normal bidirectional I/O port or it can be used for Address interfacing for accessing external memory.



Port 2 Pin Configuration

Construction:

- Port 2 has a D type latch, 1 FET, an internal pull-up resistor, two unidirectional buffers, and a Control Logic block.
- Its main functions are kind of similar to those of Port 0. It can be used as an input-output port. And can access external memory in conjunction with Port 0.

Operation:

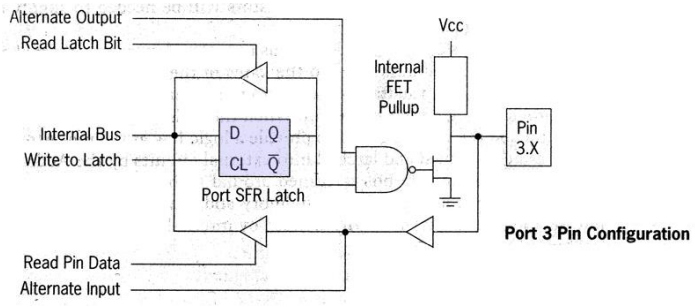
- **I/O port:**

- Quite similar to Port 0. The only difference here is that in Port 2, we use one FET with an internal pull-up resistor instead of the two FETs we saw in Port 0.

- **Memory Access:**
  - Port 2 is used in conjunction with Port 0 to generate the upper address of the external memory location that needs to be accessed. However, one key difference is that it doesn't need to turnaround and get a 1 in the latch immediately for input as in Port 0. It can remain stable.
- Port2 latches remain stable when external memory is addressed, as they do not have to be turned around for data input as is the case for port 0.

➢ Port 3

Port 3 is an input/output port similar to port 1. The input and output functions can be programmed under the control of the p3 latches or under the control of various other special function registers.



Port 3 Pin Configuration

Features:

- Address is B0H
- Port -3 has 8 pins (P3.0-P3.7).
- It has dual functions.
- Port-3 can be configured as a normal bidirectional I/O port or it can be used for alternate functions.

Construction:

The third Port of 8051 has a D-type latch. In addition to that, it has three unidirectional buffers. A FET with an internal pull-up resistor. Additionally, it also has a NAND gate connected to the FET.

Operation:

- Port 3 performs two main functions: input /output port or alternate SFR function.

a) **I/O port**

- Just like Port 2, Port 3 can function as an input-output port.

b) **Alternate SFR function**

- The input to SFR 1, we get the output of latch as 1, which turns on the NAND gate, and depending on the value of 'Alternate Output Pin,' FET will be wither ON/OFF.

The port 3 alternate uses are shown in the following table:

Pin	Alternate Use	SFR
P3.0–RXD	Serial data input	SBUF
P3.1–TXD	Serial data output	SBUF
P3.2–INT0	External interrupt 0	TCON.1
P3.3–INT1	External interrupt 1	TCON.3
P3.4–T0	External timer 0 input	TMOD
P3.5–T1	External timer 1 input	TMOD
P3.6–WR	External memory write pulse	—
P3.7–RD	External memory read pulse	—

❖ INTRRRUPTS

The most powerful and important features are interrupts in 8051 microcontroller. In most of the real-time processes, to handle certain conditions properly, the actual task must be halt for some time – it takes required action – and then must return to the main task. For executing such type of programs, interrupts are necessary. It entirely differs from the polling method wherein the processor must check sequentially each device and ask whether the service is required or not while consuming more processor time.

Interrupts are the events that temporarily suspend the main program, pass the control to the external sources and execute their task. It then passes the control to the main program where it had left off.

Interrupts are of different types like software and hardware, maskable and non-maskable, fixed and vector interrupts, and so on. Interrupt Service Routine (ISR) comes into the picture when interrupt occurs, and then tells the processor to take appropriate action for the interrupt, and after ISR execution, the controller jumps into the main program.

Types of Interrupts in 8051

The 8051 microcontroller can recognize five different events that cause the main program to interrupt from the normal execution. These five sources of interrupts in 8051are:

1. Timer 0 overflow interrupt- TF0
2. Timer 1 overflow interrupt- TF1
3. External hardware interrupt- INT0
4. External hardware interrupt- INT1
5. Serial communication interrupt- RI / 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.

Interrupt Enable (IE) Register:

This register is responsible for enabling and disabling the interrupt. It is a bit addressable register in which EA must be set to one for enabling interrupts. The corresponding bit in this register enables particular interrupt like timer, external and serial inputs. In the below IE register, bit corresponding to 1 activates the interrupt and 0 disables the interrupt.