Module V

Microcontrollers - Types of Microcontrollers – Criteria for selecting a microcontroller - Example Applications. Characteristics and Resources of a microcontroller. Organization and design of these resources in a typical microcontroller - 8051. 8051 Architecture, Register Organization, Memory and I/O addressing, Interrupts and Stack.

Microcontrollers

Microcontroller (MCU) is a chip consisting of

- Microprocessor (MPU)
- Memory
- I/O (Input/Output) ports

Microcontroller is a small computer that is capable of performing specific task(s) – e.g car alarm, washing machine, handphone, PDA etc

- Supporting Devices of microcontroller
  - Timers
  - A/D converter
  - Serial I/O
- Common communication lines
  - System Bus

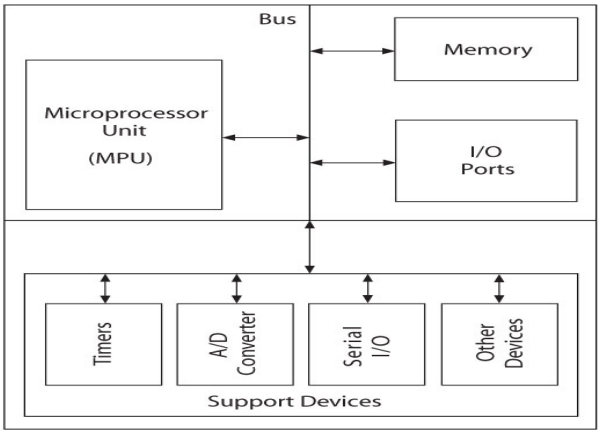
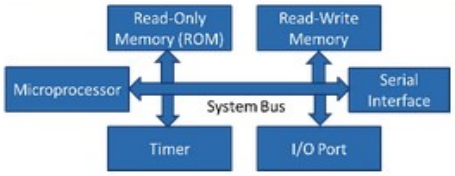
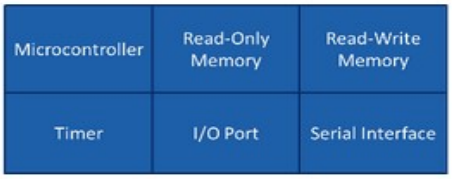


Fig: basic diagram of micro controller

Microcontrollers are designed for embedded applications or other general purpose applications

Difference between Micro processor and Microcontroller

Microprocessor	Micro Controller
	
Microprocessor is heart of Computer system.	Micro Controller is a heart of embedded system.
It is just a processor. Memory and I/O components have to be connected externally	Micro controller has external processor along with internal memory and i/O components
Since memory and I/O has to be connected externally, the circuit becomes large.	Since memory and I/O are present internally, the circuit is small.
Cannot be used in compact systems and hence inefficient	Can be used in compact systems and hence it is an efficient technique
Cost of the entire system increases	Cost of the entire system is low
Due to external components, the entire power consumption is high. Hence it is not suitable to used with devices running on stored power like batteries.	Since external components are low, total power consumption is less and can be used with devices running on stored power like batteries.
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.
Since memory and I/O components are all external, each instruction will need external operation, hence it is relatively slower.	Since components are internal, most of the operations are internal instruction, hence speed is fast.
Microprocessor have less number of registers, hence more operations are memory based.	Micro controller have more number of registers, hence the programs are easier to write.
Microprocessors are based on von Neumann model/architecture where program and data are stored in same memory module	Micro controllers are based on Harvard architecture where program memory and Data memory are separate
Mainly used in personal computers	Used mainly in washing machine, MP3 players

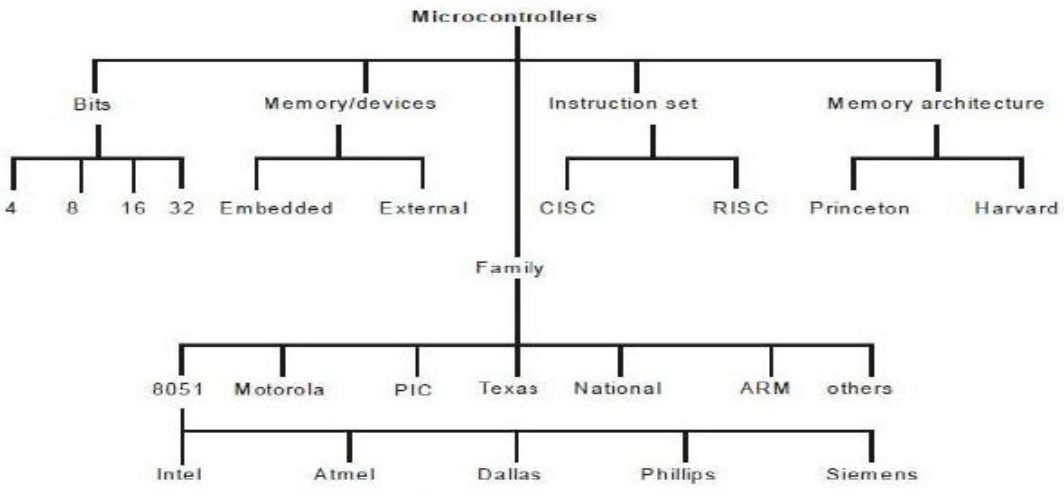
Types of microcontroller

Microcontrollers are divided into categories according to their memory, architecture, bits and instruction sets.

Based on Bits:

- **8 bits microcontroller** executes logic & arithmetic operations. Examples of 8 bits micro controller is Intel 8031/8051.
- **16 bits microcontroller** executes with greater accuracy and performance in contrast to 8-bit. Example of 16 bit microcontroller is Intel 8096.
- **32 bits microcontroller** is employed mainly in automatically controlled appliances such as office machines, implantable medical appliances, etc. It requires 32-bit instructions to carry out any logical or arithmetic function.

Types of Microcontrollers



Based on Memory:

- **External Memory Microcontroller** –Some times for a large system, the built in program memory and data memory are insufficient. To overcome this problem some microcontrollers allow the connection of external memory
- **Embedded Memory Microcontroller** – When a complete hardware required to run a particular application is provided on the microcontroller chip, it is referred to as embedded microcontroller

Based on Instruction Set:

- **CISC**- CISC means complex instruction set computer, it allows the user to apply 1 instruction as an alternative to many simple instructions.
- **RISC-** RISC means Reduced Instruction Set Computers. RISC reduces the operation time by shortening the clock cycle per instruction.

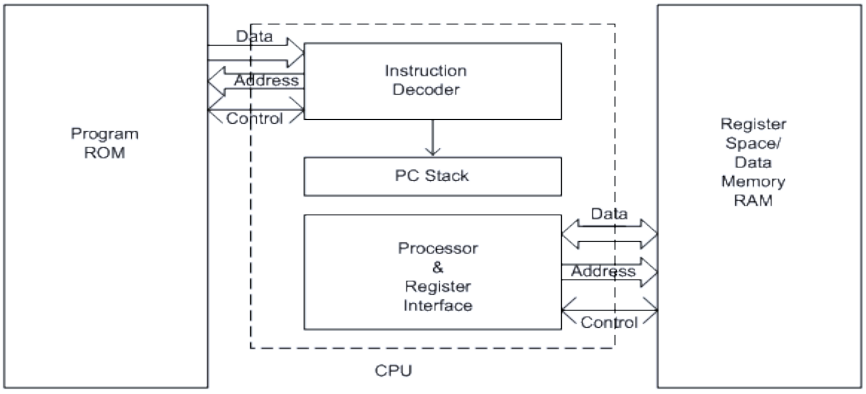
RISC	CISC
Reduced instruction set computer	Complex instruction set computer.
The microprocessor is designed using hardwired control.	The microprocessor is designed using code control
It executes at least one instruction in a cycle	Several cycles may be required to execute one instruction
The instructions have simple fixed formats with few addressing modes.	The instructions have variable formats with several complex addressing modes.
It has several general purpose resisters and large cache memory.	It has a small number of general purpose registers
The instruction set of RISC micro- processors typically includes only register to register load and store.	The instruction set of CISC microprocessor includes several instructions to access memory and CPU registers.

Based on Memory Architecture:

Harvard Memory Architecture Microcontroller

Separate memories for data and instructions.

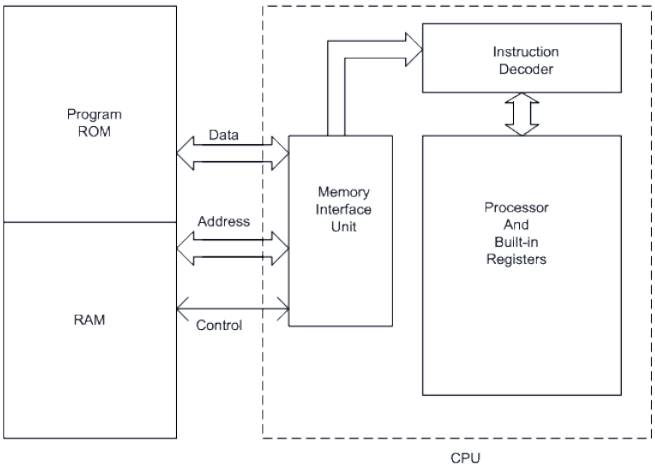
Two sets of address/data buses between CPU and memory



Princeton Memory Architecture Microcontroller

Same memory holds data, instructions.

A single set of address/data buses between CPU and memory



Microcontroller Families

8051:

8051 manufactured in 1985. This is an 8-bit microcontroller. The width of the register represents the bit number of microcontroller. For example 89C51 has 8-bit register, so 89C51 is 8-bit microcontroller. Coming to the instruction set 8051 has 250 instructions which take 1 to 4 machine cycles to execute. The speed of the 8051 microcontroller is 1 million instructions per second. 8051 has powerful instruction set; it has commands which perform more complex calculations. The ALU of the 8051 makes computations simple. 8051 microcontroller has 32 I/O pins, timers/counters, interrupts etc

AVR:

AVR is an 8-bit RISC architecture microcontroller. This is available from 1996 onwards only. 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. The speed of AVR microcontroller is 12 million instructions per second.

PIC:

PIC (Programmable interface controller) microcontrollers are available in 3 different architectures. Those are 8-bit, 16-bit and 32-bit microcontrollers. PIC has nearly 40 instructions which all are take 4 clock cycles to execute. The speed of the PIC controller is 3 million instructions per second. The programming part of the PIC microcontroller is very hard. So those who entering into embedded world freshly this is not preferable for them.

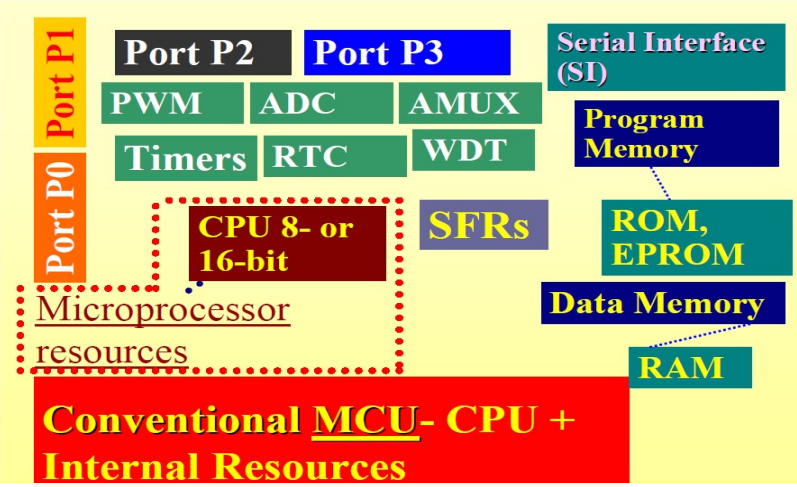
ARM:

At the time of manufacturing ARM was named as Acorn RISC Machine. Later ARM limited was established in1990. From then onwards ARM renamed as Advanced RISC machine. This is the advanced RISC controller. Most of the industries get license from ARM limited. ARM has the features like load-store architecture, fixed-length instruction set and 3-address instruction format. It has 32-bit ARM instruction set and 16-bit Thumb compressed instruction set.

Resources of a microcontroller

Or

Organization and design of these resources in a typical microcontroller – 8051



(Here just mention the names. Explanations are in the next page)

4 Ports: port0, port 1, port2 and port 3

Timers: 2 Timers in 8051(timer 0 and timer1)

Program Memory- Program functions and routines in an MCU are mostly in a non volatile read only memory(ROM)

- ROM
- EPROM

Data memory – Data variables and stacks in an MCU are mostly in a volatile read and write memory (RAM)

SFR- Special Function Register

WDT- Watchdog Timer

PWM (Pulse Width Modulator) - used to obtain the digital to analog conversion operation.

ADC (Analog to Digital Converter)

The **real time clock (RTC)** is widely used device that provides accurate time and date for many applications. A real-time clock (RTC) is a computer clock (most often in the form of an integrated circuit) that keeps track of the current time. Although the term often refers to the devices in personal computers, servers and embedded systems, RTCs are present in almost any electronic device which needs to keep accurate time.

AMUX(Analog Multiplexer)

Serial Interface

The 8051 contains UART – universal asynchronies receiver transmitter
• The serial port is full duplex

• It can transmit and receive simultaneously

Two Port 3 pins are used to provide the serial interface

P 3.0 is the receiver pin (RXD) P 3.1 is the transmitter pin (TXD)

Both synchronous and asynchronous transmission supported

Bus Width

Internal

- Internal bus width of 8051 is 8 bit. Same 8 bit bus is used for internally carrying all th data, addresses, and instruction codes.
- Registers and internal RAM s need only 8 bit addresses.

External

- External data bus width is 16 in 8051.
- Therefore external address space is 2^{16} = 64 KB

8051 Architecture

Features

- 8 bit cpu with registers A and B
- 16 bit PC and DPTR(data pointer).
- 8 bit program status word(PSW)
- 8 bit Stack Pointer
- 4K Internal ROM
- 128bytes Internal RAM
  - →4 register banks each having 8 registers
  - →16 bytes,which may be addressed at the bit level.
  - →80 bytes of general purpose data memory
- 32 i/o pins arranged as 4 8 bit ports:P0 to P3
- Two 16 bit timer/counters:T0 and T1
- Full duplex serial data receiver/transmitter
- Control registers:TCON,TMOD,SCON,PCON,IP and IE
- Two external and Three internal interrupt sources.

Oscillator and Clock Circuits.

Registers in 8051

DATA POINTER (DTPR):- DTPR is a 16 bit register.

It consists of higher byte (DPH) and a lower byte (DPL).

DPTR is used to hold the external data memory address of data being fetched.

PC (Program Counter)

- 16 bit register
- Hold address of instruction being currently fetched.

- Increments continuously to point to the next instruction.

Special Function registers

ACCUMULATOR (A):-8 bit register

it is used for data transfer and arithmetic operations. After any operation result is stored in A

B REGISTER:- it is used to store the upper 8 bit result of multiplication and divisions. It is used as temporary register

PROGRAM STATUS WORD (PSW):- This special function registers and consists of different status bits that reflect the current state of microcontroller.

It contains carry (CY), the auxiliary carry(AC), the two registers bank select bits(RS1 and RS0), the overflow flag(OV), a parity bit(P), and two user defined status flags.

CY	AC	F0	RS1	RS0	OV	-	P

STACK POINTER (SP):- This is an 8 bit register. SP is incremented before the data is stored onto the stack using PUSH/CALL instructions execution. During PUSH, first SP is incremented and then copy the data. In the POP operation, initially copy the data and then decrement the SP.

PCON (Power Control): The Power Control SFR is used to control the 8051's power control modes. Certain operation modes of the 8051 allow going "sleep" mode which requires much less power. These modes of operation are controlled through PCON.

TCON (Timer Control, Bit-Addressable): The Timer Control SFR is used to configure and modify the way in which the 8051's two timers operate. This SFR controls whether each of the two timers is running or stopped and contains a flag to indicate that each timer has overflowed.

TMOD (Timer Mode): The Timer Mode SFR is used to configure the mode of operation of each of the two timers. each timer to be a 16-bit timer, an 8-bit auto reload timer, a 13-bit timer, or two separate timers.

TL0/TH0 (Timer 0 Low/High): These two SFRs, taken together, represent timer 0.

TL1/TH1 (Timer 1 Low/High): These two SFRs, taken together, represent timer 1.

SCON (Serial Control): The Serial Control SFR is used to configure the behavior of the 8051's on-board serial port. This SFR controls the baud rate of the serial port, whether the serial port is activated to receive data, and also contains flags that are set when a byte is successfully sent or received.

SBUF (Serial Control): The Serial Buffer SFR is used to send and receive data via the on-board serial port. Any value written to SBUF will be sent out the serial port's TXD pin. Likewise, any value which the 8051 receives via the serial port's RXD pin will be delivered to the user program via SBUF.

IE (Interrupt Enable): The Interrupt Enable SFR is used to enable and disable specific interrupts.

IP (Interrupt Priority): The Interrupt Priority SFR is used to specify the relative priority of each interrupt. On the 8051, an interrupt may either be of low (0) priority or high (1) priority.

PORT0, PORT1, PORT2, PORT3 LATCHES AND DRIVERS:- Each latch and corresponding drivers of port 0-3 is allotted to the corresponding on chip I/O port.

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 1

Port-1 dedicated only for I/O interfacing.

PORT 2:

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.

PORT 3:

It works as an IO port same like Port 2 as well as it can do lots of alternate work Following are the alternate functions of port 3:

P3.0—RXD

P3.1— TXD

P3.2— INT0 BAR

P3.3— INT1 BAR

P3.4— T0

P3.5— T1

P3.6— WR BAR

P3.7— RD BAR

ALU

A unit to perform an arithmetic or logic operation at an instance

SI (Serial Interface)

- The 8051 contains UART – universal asynchronies receiver transmitter
- The serial port is full duplex
- It can transmit and receive simultaneously
- Two Port 3 pins are used to provide the serial interface
- P 3.0 is the receiver pin (RXD) P 3.1 is the transmitter pin (TXD)
- Both synchronous and asynchronous transmission supported

Register SCON controls serial data communication

Power Mode control Register

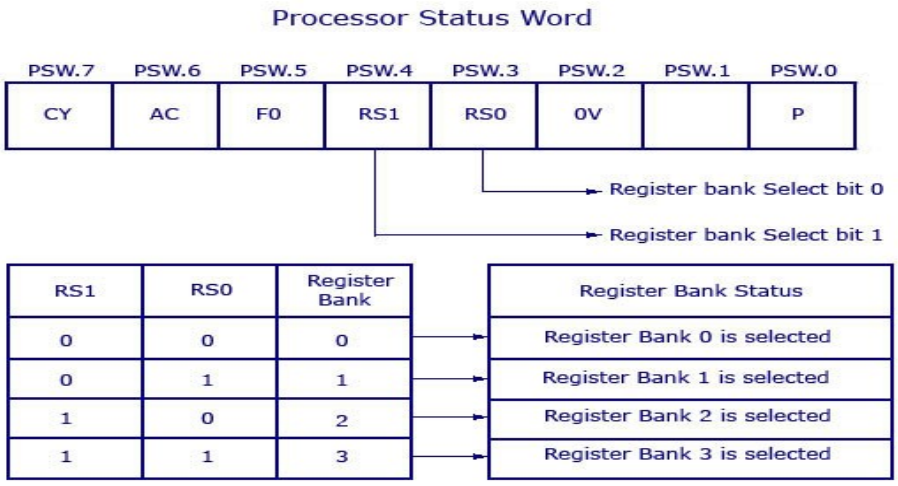
Register PCON controls processor power down, sleep modes and serial data band rate.

Timers

There are two 16-bit timers and counters in 8051 microcontroller: **timer 0 and timer 1**. Both timers consist of 16-bit register in which the lower byte is stored in TL and the higher byte is stored in TH. Timer can be used as a counter as well as for timing operation that depends on the source of clock pulses to counters.

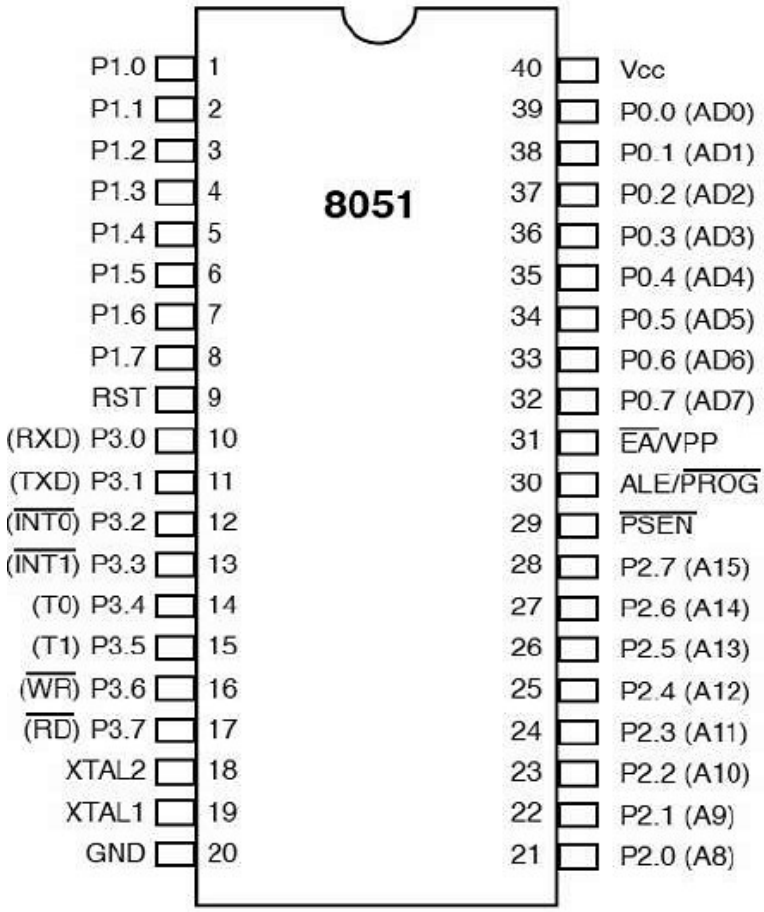
Counters and Timers in 8051 microcontroller contain two special function registers: **TMOD (Timer Mode Register) and TCON (Timer Control Register)**, which are used for activating and configuring timers and counters.

PSW



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

Pin Diagram - 8051



- Vcc（pin 40）：

  Vcc provides supply voltage to the chip.

  The voltage source is +5V.

- GND（pin 20）：ground
- XTAL1 and XTAL2（pins 19,18）：

  These 2 pins provide external clock.

- RST（pin 9）：reset

  It is an input pin and is active high（normally low）.

  It is a power-on reset.

  Upon applying a high pulse to RST, the microcontroller will reset and all values in registers will be lost.

- /EA（pin 31）：external access

  The /EA pin is connected to GND to indicate the code is stored externally.

  For 8051, /EA pin is connected to Vcc.

  "/" means active low.

- /PSEN（pin 29）：program store enable

  This is an output pin and is connected to the OE pin of the ROM.

- ALE（pin 30）：address latch enable

  It is an output pin and is active high.

  8051 port 0 provides both address and data.

  The ALE pin is used for de-multiplexing the address and data by connecting to the G pin of the 74LS373 latch.

- I/O port pins

  The four ports P0, P1, P2, and P3.

  Each port uses 8 pins.

  All I/O pins are bi-directional.

Memory organization in 8051

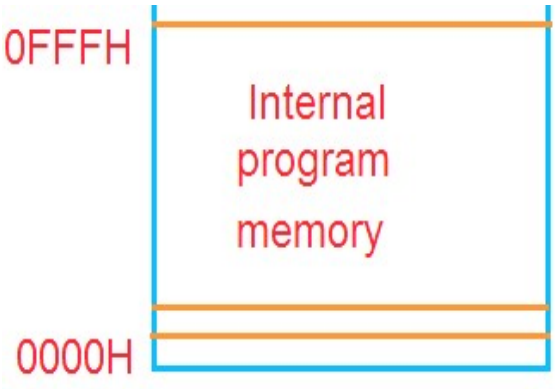
The 8051 microcontroller's memory is divided into Program Memory and Data Memory. Program Memory (ROM) is used for permanent saving program being executed, while Data Memory (RAM) is used for temporarily storing and keeping intermediate results and variables.

Program Memory (ROM)

Program Memory (ROM) is used for permanent saving program (CODE) being executed. The memory is read only. Depending on the settings made in compiler, program memory may also used to store a constant variables. The 8051 executes programs stored in program memory only.

Internal Program memory

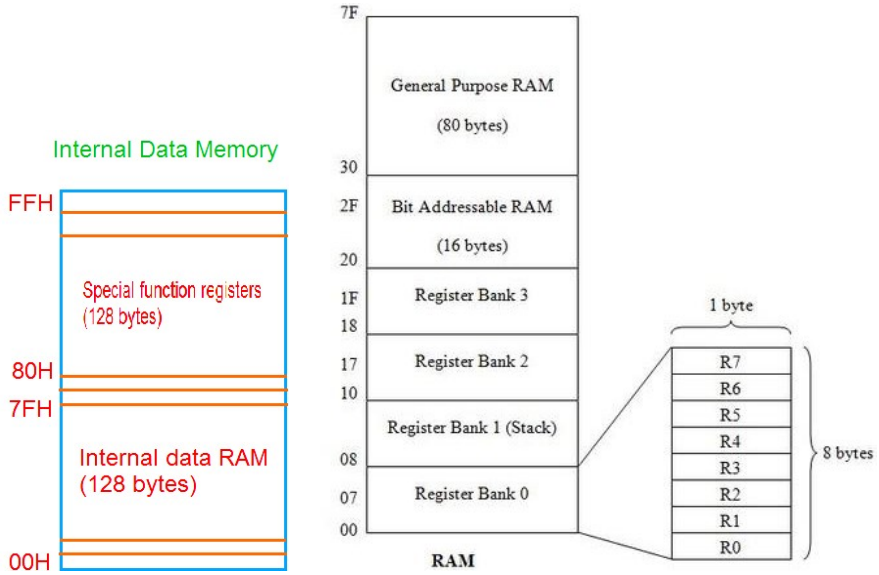
Program memory accessed through EA pin. In program memory two categories takes place:



a)If EA is high, internal program memory is accessed to 0FFFH memory location and external program memory accessed from 1000H to FFFFH memory locations.

b)If EA is low, only external program memory accessed from 0000H to FFFFH memory locations.

Internal Data Memory



The internal data memory consists of 256 bytes, these are divided into two parts:

00H-FFH for internal data RAM (128 bytes)

80H-FFH for special function registers (128 bytes)

Total 128 Byte memory is divided into three parts.

- 32 byte
- 16 byte
- 80 byte

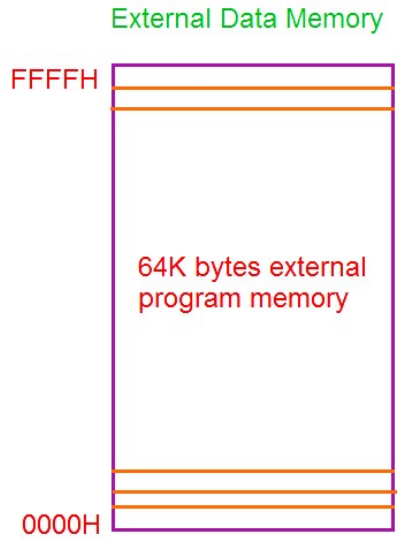
Here 16 byte classification is bit addressable. In micro-controller registers were data is stored, if one could manipulate its content bit by bit it's called bit addressable.

External Program Memory

a)If EA is high, internal program memory is accessed to 0FFFH memory location and external program memory accessed from 1000H to FFFFH memory locations.

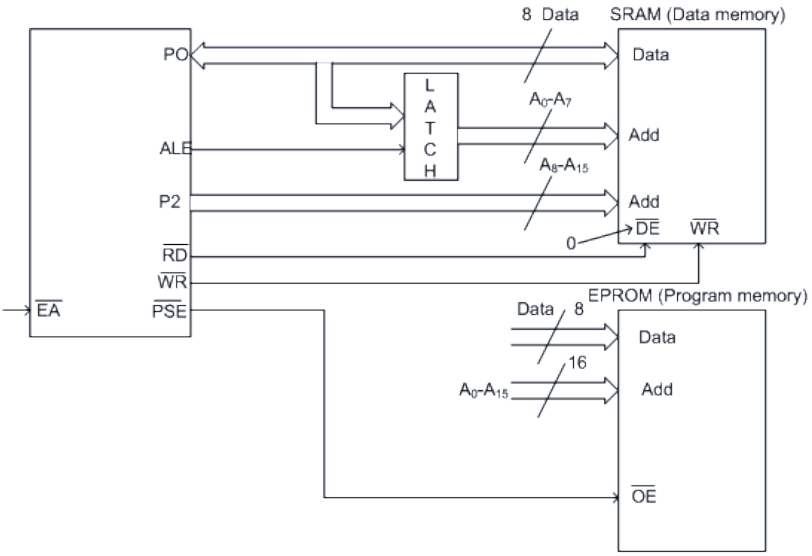
b)If EA is low, only external program memory accessed from 0000H to FFFFH memory locations.

External Data Memory



Interfacing of external memory with 8051

- External memories are interfaced using four control signals. (PSEN, ALE, RD and WR)
- ALE signal separate the A0- A7 bus for the program and X – data memories.
- PSEN signal when 0, uses the program memory code bank 2 to 31 for the code reading operation. Uses bank 0 and bank 1 also when EA is 0 during the RESET of MCU
- The RD when 0 , uses data memory for the Xdata reading operation(X Data means external data)
- The WR when 0 , uses data memory for the data write operation(X Data means external data)



I/O addressing in 8051

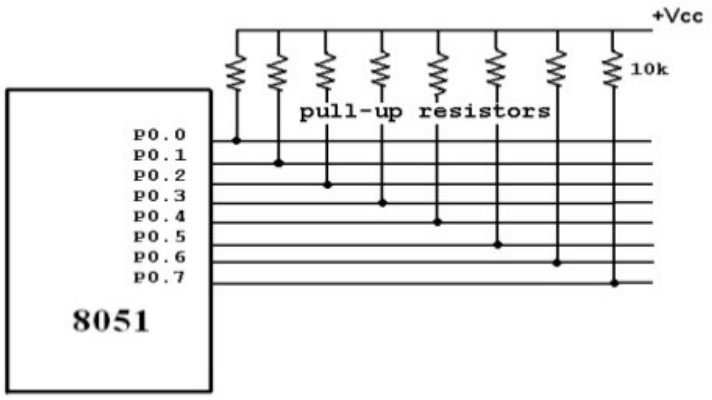
- The 8051 microcontroller has four parallel I/O ports , each of 8-bits .So, it provides the user 32 I/O lines for connecting the microcontroller to the peripherals.
- The four ports are P0 (Port0), P1(Port1) ,P2(Port 2) and P3 (Port3).
- Upon reset all the ports are output ports. In order to make them input, all the ports must be set i.e a high bit must be sent to all the port pins.
- This is normally done by the instruction

Ex: MOV A,#0FFH ; A = FF MOV P0,A ; make P0 an input port

PORT 0:

Port 0 is an 8-bit I/O port with dual purpose. If external memory is used, these port pins are used for the lower address byte address/data (AD0-AD7), otherwise all bits of the port are either input or output.

Unlike other ports, Port 0 is not provided with pull-up resistors internally, so for PORT0 pull-up resistors of nearly 10k are to be connected externally as shown in the fig



(A pull-up resistor weakly "pulls" the voltage of the wire it is connected to towards its voltage source level when the other components on the line are inactive.)

Dual role of port 0:

Port 0 can also be used as address/data bus(AD0-AD7), allowing it to be used for both address and data.
When connecting the 8051 to an external memory, port 0 provides both address and data. The 8051 multiplexes address and data through port 0 to save the pins. ALE indicates whether P0 has address or data. When ALE = 0, it provides data D0-D7, and when ALE =1 it provides address and data with the help of a 74LS373 latch.

Port 1: Port 1 occupies a total of 8 pins (pins 1 through 8). It has no dual application and acts only as input or output port. In contrast to port 0, this port does not need any pull-up resistors. Since pull-up resistors connected internally.
Upon reset, Port 1 is configured as an output port.
To configure it as an input port , port bits must be set i.e a high bit must be sent to all the port pins.
For Ex :
MOV A, #0FFH; A=FF
MOV P1,A ; make P1 an input port by writing 1's to all of its pins

Port 2 : Port 2 is also an eight bit parallel port. (Pins 21- 28). It can be used as input or output port. As this port is provided with internal pull-up resistors it does not need any external pull-upresistors.
Upon reset, Port 2 is configured as an output port. If the port is to be used as input port, all the port bits must be made high by sending FF to the port.
For ex,MOV A, #0FFH ; A=FF
MOV P2, A ; make P2 an input port by writing all 1's to it
Dual role of port 2: Port2 lines are also associated with the higher order address lines A8-A15.In systems based on the 8751, 8951, and DS5000, Port2 is used as simple I/O port
PORT 3 : Port3 is also an 8-bit parallel port with dual function.(pins 10 to 17).
The port pins can be used for I/O operations as well as for control operations. The details of these additional operations are given below in the table. Port 3 also do not need any external pull-up resistors as they are provided internally similar to the case of Port2 & Port 1. Upon reset port 3is configured as an output port. If the port is to be used as input port, all the port bits must be made high by sending FF to the port.
For ex,MOV A, #0FFH ; A= FF
MOV P3, A ; make P3 an input port by writing all 1's to it
Alternate Functions of Port 3 :

S.No	Port 3 bit	Pin No	Function
1	P3.0	10	RxD
2	P3.1	11	TxD

3	P3.2	12	$\overline{\text{INT0}}$
4	P3.3	13	$\overline{\text{INT1}}$
5	P3.4	14	T0
6	P3.5	15	T1
7	P3.6	16	$\overline{\text{WR}}$
8	P3.7	17	$\overline{\text{RD}}$

Two modes of operation

1. Single chip mode
2. Expanded Multiplexed mode

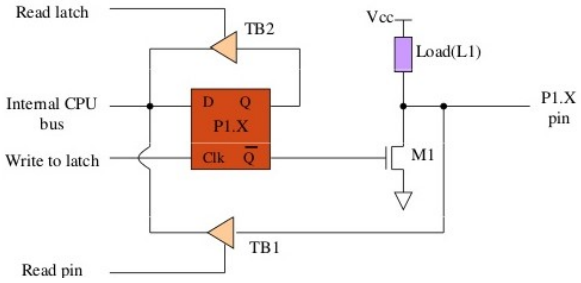
Single chip mode

- In which all the software and data internally embedded into the MCU and external memory chips are not used.

Expanded Multiplexed mode

- In this mode all software embedded either internally or externally.

Hardware Structure of I/O Pin



- Each pin of I/O ports
  - Internally connected to CPU bus
  - A D latch store the value of this pin
    - Write to latch = 1 ： write data into the D latch
  - 2 Tri-state buffer ：
    - TB1: controlled by "Read pin"
      - Read pin = 1 ： really read the data present at the pin
    - TB2: controlled by "Read latch"
      - Read latch = 1 ： read value from internal latch
  - A transistor M1 gate
    - Gate=0: open
    - Gate=1: close

Interrupts in 8051 microcontroller

- An interrupt is an external or internal event that interrupts the microcontroller to inform it that a device needs its service.
- A set of program instructions written to service an interrupt is called the Interrupt Service Routine

• 8051 has six different sources of interrupts

External: Power-up reset, INT0, INT1

Internal: Timer0, Timer1, Serial Port

Interrupt Service Routine

• When microcontroller receives an interrupt signal from any of the six interrupt sources it executes a call to interrupt service routine

• For every interrupt, there must be an interrupt service routine

• The interrupt service routine for every interrupt must be located at a fixed location in program memory, called interrupt vector.

How 8051 services an interrupt request

• 8051 finishes the instruction it is currently executing, and saves the contents of Program Counter on the stack (address of next instruction)

• It jumps to the interrupt vector location corresponding to the interrupt source

• Executes the interrupt service routine, until it encounters RETI instruction

• Returns back to the place where it was interrupted, by popping the contents of stack on PC, and starts execution at that address

Special Function Register to handling Interrupt

IE(Interrupt Enable Register)

IP(Interrupt priority Register)

Sources of Interrupt

Interrupt Enable register (IE):

EA	–	–	ES	ET1	EX1	ET0	EX0

EA	IE.7	Disables all interrupts. If EA = 0, no interrupt will be acknowledged. If EA = 1, interrupt source is individually enable or disabled by setting or clearing its enable bit.
-	IE.6	Not implemented, reserved for future use*.
-	IE.5	Not implemented, reserved for future use*.
ES	IE.4	Enable or disable the Serial port interrupt.
ET1	IE.3	Enable or disable the Timer 1 overflow interrupt.
EX1	IE.2	Enable or disable External interrupt 1.
ET0	IE.1	Enable or disable the Timer 0 overflow interrupt.
EX0	IE.0	Enable or disable External Interrupt 0.

IP(Interrupt priority Register)

-	-	PT2	PS	PT1	PX1	PT0	PX0
bit7	bit6	bit5	bit4	bit3	bit2	bit1	

-	IP.6	Reserved for future use.
-	IP.5	Reserved for future use.
PS	IP.4	It defines the serial port interrupt priority level.
PT1	IP.3	It defines the timer interrupt of 1 priority.
PX1	IP.2	It defines the external interrupt priority level.
PT0	IP.1	It defines the timer0 interrupt priority level.
PX0	IP.0	It defines the external interrupt of 0 priority level.

Interrupt Vector Table

Interrupt Number	Interrupt Description	Address
0	EXTERNAL INT 0	0003h
1	TIMER/COUNTER 0	000Bh
2	EXTERNAL INT 1	0013h
3	TIMER/COUNTER 1	001Bh
4	SERIAL PORT	0023h

Stack in 8086

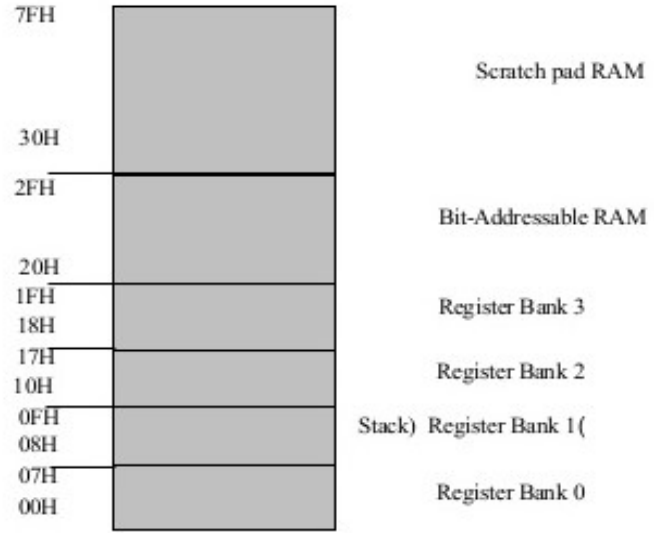
The stack is an area of random access memory (RAM) to hold the variables temporarily.

The stack is also responsible for storing address a function is called so that it can be returned correctly.

Whenever the function is called, the parameters and local variables associated with it are added to the stack (PUSH).

When the function returns, the parameters and the variables are removed ("POP") from the stack. This is why a program's stack size changes continuously while the program is running.

The register used to access the stack is called **stack pointer register**. The stack pointer is a small register used to point at the stack. When we push something into the stack memory, the stack pointer increases.



- The stack pointer in the 8051 is only 8 bits wide, which means that it can take values of 00 to FFH.
- When the 8051 is powered up, the SP register contains value 07.
- This means that RAM location 08 is the first location used for the stack by the 8051.
- The storing of a CPU register in the stack is called a PUSH, and pulling the contents off the stack back into a CPU register is called a POP.
- In other words, a register is pushed onto the stack to save it and popped off the stack to retrieve it.

Pushing onto the stack

In the 8051 the stack pointer (SP) points to the last used location of the stack. As we push data onto the stack, the stack pointer (SP) is incremented by one.

Notice that this is different from many microprocessors, notably 8086 processors in which the SP is decremented when data is pushed onto the stack.

we see that as each PUSH is executed, the contents of the register are saved on the stack and SP is incremented by 1.

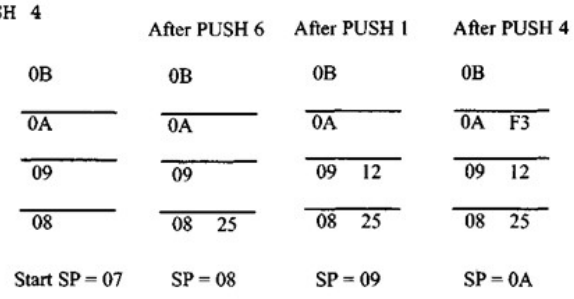
For example, the instruction "PUSH 1″ pushes register Rl onto the stack.

Example 2-8

Show the stack and stack pointer for the following. Assume the default stack area and register 0 is selected.

```
MOV  R6,#25H
MOV  R1,#12H
MOV  R4,#0F3H
PUSH 6
PUSH 1
PUSH 4
```

Solution:



Popping from the stack

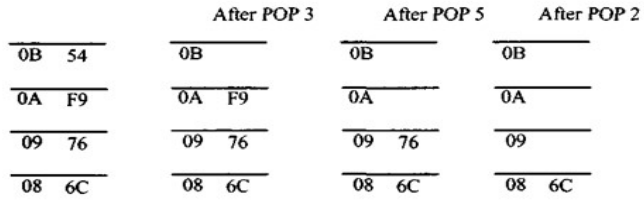
Popping the contents of the stack back into a given register is the opposite process of pushing. With every pop, the top byte of the stack is copied to the register specified by the instruction and the stack pointer is decremented once. Example demonstrates the POP instruction.

Example 2-9

Examining the stack, show the contents of the registers and SP after execution of the following instructions. All values are in hex.

```
POP  3    ;POP stack into R3
POP  5    ;POP stack into R5
POP  2    ;POP stack into R2
```

Solution:



Extra Topics

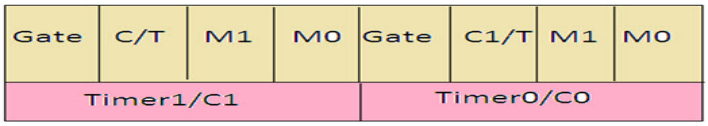
Timers and counters

Many of the microcontroller applications require counting of external events such as frequency of the pulse trains and generation of 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.

There are two 16-bit timers and counters in 8051 microcontroller: **timer 0 and timer 1**. Both timers consist of 16-bit register in which the lower byte is stored in TL and the higher byte is stored in TH. Timer can be used as a counter as well as for timing operation that depends on the source of clock pulses to 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.

Timer Mode Control (TMOD): TMOD is an 8-bit register used for selecting timer or counter and mode of timers. Lower 4-bits are used for control operation of timer 0 or counter0, and remaining 4-bits are used for control operation of timer1 or counter1.



Timer Mode Control (TMOD)

Gate: If the gate bit is set to '0', then we can start and stop the "software" timer in the same way. If the gate is set to '1', then we can perform hardware timer.

C/T: If the C/T bit is '1', then it is acting as a counter mode, and similarly when set C/T bit is '0'; it is acting as a timer mode.

M0	M1	Mode	Timer Pulses
0	0	M0	13-bit-2^13-8192
0	1	M1	16-bit-2^16-65535 pulses
1	0	M2	8-bit-autoreload mode-2^8= 256 pulses
1	1	M3	Split mode(load the values in T0 automatically start the T1

Mode select bits: The M1 and M0 are mode select bits, which are used to select the timer operations. There are four modes to operate the timers.

Mode 0: This is a 13-bit mode that means the 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: This mode is a split-timer mode, which means the loading values in T0 and automatically starts the T1.

Timer Control Register (TCON): TCON is another register used to control operations of counter and timers in microcontrollers. It is an 8-bit register wherein four upper bits are responsible for timers and counters and lower bits are responsible for interrupts.

TF1	TR1	TFO	TRO	IE1	ITO	IEO	ITO

Timer Control Register (TCON)

TF1: The TF1 stands for 'timer1' flag bit. Whenever calculating the time-delay in timer1, the TH1 and TL1 reaches to the maximum value that is "FFFF" automatically.

Whenever the TF1=1, then clear the flag bit and stop the timer.

TR1: The TR1 stands for timer1 start or stop bit. This timer starting can be through software instruction or through hardware method.

TR1=1; (Start timer)

TF0: The TF0 stands for 'timer0' flag-bit. Whenever calculating the time delay in timer1, the TH0 and TL0 reaches to a maximum value that is 'FFFF', automatically.

Whenever the TF0=1, then clear the flag bit and stop the timer.

TR0: The TR0 stands for 'timer0' start or stop bit; this timer starting can be through software instruction or through hardware method.

TR0=1; (Start timer)

Serial Interface

The serial port of 8051 is full duplex, i.e., it can transmit and receive simultaneously.

The register SBUF is used to hold the data. The special function register SBUF is physically two registers. One is, write-only and is used to hold data to be transmitted out of the 8051 via TXD. The other is, read-only and holds the received data from external sources via RXD.

Serial Port Control Register (SCON)

Register SCON controls serial data communication

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

Mode select bits

SM0	SM1	Mode
0	0	Mode 0
0	1	Mode 1
1	0	Mode 2
1	1	Mode 3

SM2: multi processor communication bit

REN: Receive enable bit

TB8: Transmitted bit 8 (Normally we have 0-7 bits transmitted/received)

RB8: Received bit 8

TI: Transmit interrupt flag

RI: Receive interrupt flag

Power Mode control Register

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

b7							b0
SMOD	—	—	—	GF1	GF0	PD	IDL

SMOD: Serial baud rate modify bit

GF1: General purpose user flag bit 1

GF0: General purpose user flag bit 0

PD: Power down bit

IDL: Idle mode bit