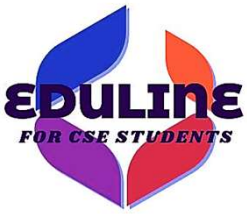
MODULE 5

MICROCONTROLLERS

CO – Students will be able to Outline features of microcontrollers and develop low level programs.



Prepared By Mr. EBIN PM, Chandigarh University, Punjab

MICROCONTROLLERS

- Micro controller is a single chip computer or a CPU with all the peripherals like RAM, ROM, I/O ports, Timers etc. on the same chip
- It is a small computer on a single IC
- It can be called as a heart of embedded systems
- Microcontrollers are designed to execute a single specific task to control a single system.
- Microcontrollers are generally used in projects and applications that required direct control of user

➢ **Eg:** 8051, Motorola's 6811, Z8, PIC 16x etc.

Prepared By Mr. EBIN PM, Chandigarh University, Punjab EDULINE 2

MICRO PROCESSOR	MICROCONTROLLER
General purpose device which is called a CPU	Dedicated chip which is also called single chip computer
Do not contain on-chip I/O port, timers, memories etc.	It include RAM, ROM, serial and parallel interface, timers etc.
Instructions are mainly nibble or byte addressable	Instructions are both bit addressable as well as byte addressable
Design is complex and expensive	Design is rather simple and cost effective
Large number of instructions and is complex	Less number of instructions and is simple
Microprocessor has Zero status flag	Microcontroller has no zero flag
Heart of computer system	Heart of embedded system

Prepared By Mr. EBIN PM, Chandigarh University, Punjab EDULINE 3

MICRO PROCESSOR	MICROCONTROLLER
Since memory and I/O has to be connected externally the circuit become large	Since memory and I/O are present internally, the circuit is small
Less number of registers, hence more operations are memory based	Have more number of registers, hence the programs are easier to write
Mainly used in personal computers	Used in washing machine, MP3 players etc.
These are based on Von Neumann model	Based on Harvard architecture

Prepared By Mr. EBIN PM, Chandigarh University, Punjab EDULINE 4

- **Microcontrollers are classified in terms of**
  - **Internal bus width**
  - **Embedded microcontroller**
  - **Instruction set**
  - **Memory architecture**
  - **IC chip or VLSI core file and family**

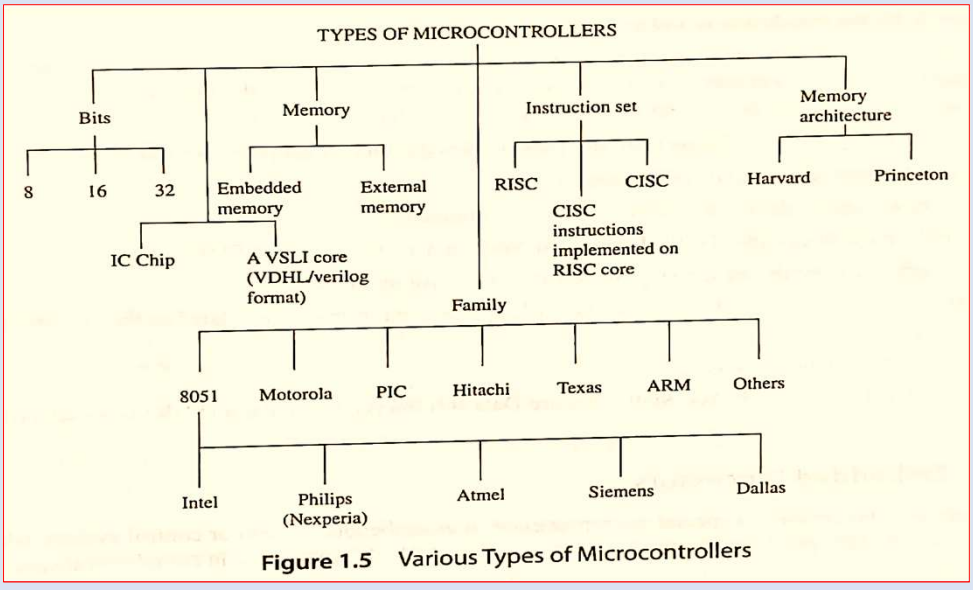


Figure 1.5 Various Types of Microcontrollers

❖Applications of Microcontrollers

➢**In biomedical instruments**

- ECG LCD display cum recorder
- Blood cell recorder cum analyzer
- Patient monitor system

➢**In communication systems**

- Numeric pagers
- Cellular phones
- Cable TV terminals
- Video games

Prepared By Mr. EBIN PM, Chandigarh University, Punjab EDULINE 7

➢**Peripheral controllers of a computer**

- Keyboard controller
- Printer controller
- LAN controller
- Disk drive controller

➢**Instruments**

- Industrial process controller
- Electronic smart weight display system

➢Automatic signal tracker; Robotics system; CNC machine controller; CRT display controller

Prepared By Mr. EBIN PM, Chandigarh University, Punjab EDULINE 8

8051 MICROCONTROLLER - ARCHITECTURE

- 8051 has 4KB internal program memory.
- The 8051 has
  - Processor
  - ROM & RAM
  - Interrupt control circuit
  - Internal timing devices (timers T0 & T1)
  - Serial Interface (SI)
  - Special Function Registers (SFR)
  - Four Ports (P0, P1, P2 and P3)

- **Features in 8051 are given below:**

1. 8051 can be used as a single- chip micro computer with embedded program in ROM or Flash. The ROM is 4KB in 8051
2. 8051 has 128Byte RAM. There are a number of Special Function Registers
3. Each program needs a stack. 8051 has a stack . SP(Stack pointer) is 8 bit register
4. There is a program counter. The lower bytes of program counter is sent at the bus A_0-A_7. The A_0-A_7 bus pins are also common to the data bus D_0-D_7
5. There is a Program Counter higher byte at bus A_8-A_{15}. The A_8-A_{15} bus pins are used in the expanded chip mode.

6. A microcontroller can be connected to I/O devices, using ports
7. There are two external pins for interrupts INT0 & INT1. There is interrupt control circuit for the 8051 interrupt service mechanism. Two special function registers IP (Interrupt Priority) and IE (Interrupt Enable) used for priorities and mask.
8. There are two programmable timers/event counters , T_0 & T_1 to do real time control of event and task.

The internal architecture of 8051 contains the following blocks:

1. **Accumulator (A) :** 8 bit register, saves the operand for operations by ALU. Its important function is to accumulate the result after an ALU operation.
2. **B register (B) :** 8 bit register B saves a second operand for the ALU. It accumulates the part of the result of multiplication or division.
3. **PSW (Processor Status Word):** 8 bit register to save the status information
4. **Stack Pointer (SP):** 8 bit register is incremented before the data is stored on to the stack using PUSH instruction.

5. **Data Pointer (DPTR) :** 16 bit register , holds the external data memory address of the data being currently fetched or to be fetched. Data is fetched using indirect addressing mode. The DPTR consist of 2 bytes DPH(Higher) DPL (Lower)
6. **Port P0:** 8 bit port **P0** is for I/O in a single chip mode. In the expanded mode P_0 is for the data bus- cum-lower order address signals AD_0-AD_7
7. **Port P2:** 8 bit port P_2 is for the I/O in a single chip mode. In the expanded mode P_2 is for higher order address signals A_8-A_{15}
8. **Port P1:** 8 bit port P_1 is for the I/O in a single chip mode . In the expanded mode it is used for few interfacing signals.

Port P3: 8 bit port P_3 is for the I/O in a single chip mode.

- Also used for serial interface signals
- P3 pin also used for timer T0 and T1 inputs, and interrupts $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ inputs.
- P3 pin is also used for sending $\overline{\text{RD}}$ and $\overline{\text{WR}}$ signals for the memory read , write in the expanded mode.

10. **Serial Data Buffer (SBUF):** It internally contains two independent registers

- Transmit buffer ➔ Parallel-in Serial – out register
- Receive buffer ➔ Serial – in Parallel-out register

11.Timer Register: These are two 16 bit registers ,can be accessed as their lower and upper bytes

- TL0 ➔ Lower byte of the timing register 0
- TH0 ➔ Higher byte of the timing register 0
- TL1 ➔ Lower byte of the timing register 1
- TH1 ➔ Higher byte of the timing register 1

12.Control Registers : The special function registers IP, IE, TMOD, TCON, SCON and PCON contain control and status information for interrupts, timers/counters and serial port.

- IP – Interrupt Priority register
- IE –Interrupt Enable register
- TMOD – Timer mode register

- TCON – Timer Control register
- SCON – Serial Control register
- PCON – Power Control register

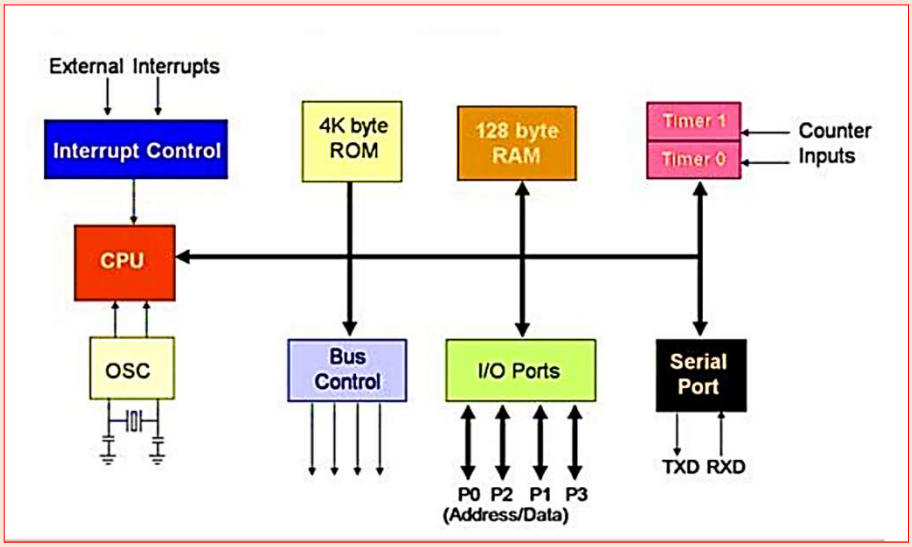
13.Timing & Control unit : This unit derives all the necessary timing and control signals required for the internal operation of the circuit. It also derives control signals required for controlling the external system bus.

14.Oscillator : This circuit generate the basic timing clock signals for the operation of the circuit using crystal oscillator

15.Instruction Register: This register decodes the opcode of an instruction to be executed , and gives information to the timing and control unit to generate necessary signals for the execution of the instruction.

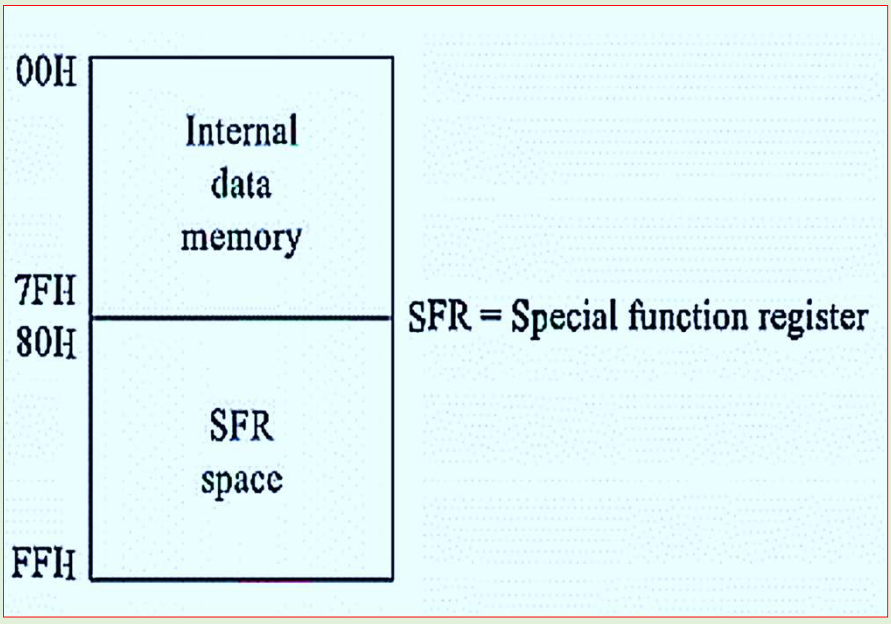
16. **RAM :** RAM is 128 byte memory for the read and write and is indirectly and directly addressable. A RAM address is between 0x00 and 0x7F
17. **ROM:** EPROM or flash EEPROM of 4KB or 8KB or 16KB
18. **ALU:** Perform 8 bit arithmetic and Logical operations over the operands held by the temporary registers TMP1 and TMP2.
19. **SFR Register Bank:** 4 register banks each of 8 registers and these are also part of the internal RAM.

❖Block Diagram of 8051 Microcontroller



REGISTER ORGANIZATION OF 8051

- The internal data memory of 8051 is divided into 2 groups
  - Set of eight registers (R0 to R7)
  - Scratch pad memory (High speed internal memory)
- The address range 00H to 07H is used to access the registers, and the rest are scratch pad memory
- 8051 provides 4 register banks, but only one register bank can be used at any point of time.
- To select the register bank, two bits of PSW are used.



- The following addressing can be used to select register bank

Address Range	Register Bank
00H to 07H	Register Bank 0
08H to 0FH	Register Bank 1
10H to 17H	Register Bank 2
18H to 1FH	Register Bank 3

- The concept of 4 register bank is very useful. For servicing the interrupts, this feature is very good.
- The interrupt program can use one bank and the interrupt service subroutine can access another bank for better performance.

❖Special Function Registers (SFRs)

- Register A & B is used to store operands
- PSW – Processor Status Word
- IP- Interrupt Priority Register
- IE- Interrupt Enable Register
- SBUF - Serial Buffer
- SCON - Serial Control Register
- TH_1 - Timer1 higher 8 bits
- TH_0 - Timer 0 higher 8 bits
- TL_1 - Timer 1 lower 8 bits
- TL_0 - Timer0 lower 8 bits

- TMOD – Timer mode Register
- TCON – Timer Control Register
- PCON – Power Control Register
- SP - Stack Pointer
- P0 – Port 0
- P1 – Port1
- P2 – Port2
- P3 – Port3

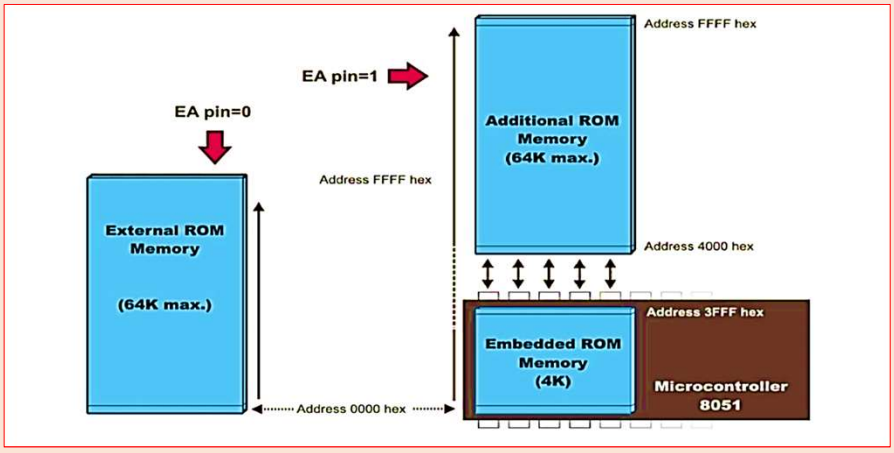
MEMORY ORGANIZATION OF 8051

➢In 8051 microcontroller , the memory is divided in to program memory and data memory.

❖Program Memory

- ROM is used for permanent saving program (CODE) being executed. The memory is read only.
- Program memory may also used to store constant variables. The 8051 executes programs stored in program memory only
- 8051 memory organization allows external program memory to be added. Microcontroller handle external memory depends on the pin EA (External Access) logical state.

- External ROM is accessed whenever the EA pin is connected to ground (=0) or when the PC contains an address higher than the last address in the internal 4KB ROM (0FFFH). 8051 designs can use internal and external ROM automatically.



❖Internal Data Memory (RAM)

- The 128 byte internal RAM is organized into three distinct areas.

A) 32 bytes from address 00H to 1FH, that create 32 working registers organized as 4 banks of 8 registers each.

- The 4 register banks are numbered 0 to 3 and are made up of 8 registers named R_0 to R_7.
- Each register can be accessed by name or by its RAM address.

Eg: If bank 3 is currently selected, then R_0 of bank 3 is R_0 or address 18H

- Bits RS_0 and RS_1 in the PSW determine which bank of registers is currently in use at any time when the program is running.

➢Bank 0 is selected on reset.

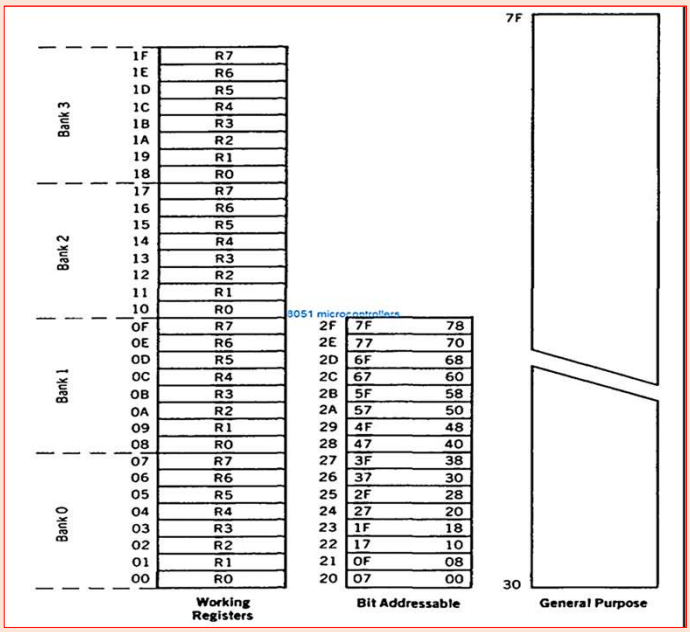
B) A bit-addressable area of 16 bytes occupies RAM byte addresses 20H to 2FH; forming a total of 128 addressable bits.

- An addressable bit maybe specified by its bit address of 00H to 7FH.

Eg: bit address 4FH is also bit 7 of byte address 29H

- Addressable bits are useful when the program need only remember a binary event.(switch on, light off, etc.)

C) A general purpose RAM area above the bit area from 30H to 7FH, addressable as bytes.



❖External Data Memory

- Access to external memory is slower than access to internal data memory.
- There maybe up to 64K Bytes of external data memory. Several 8051 devices provide on-chip XRAM space.
- It can be accessed through the indirect addressing mode.

❖I/O addressing:

- 8051 microcontroller have 4 I/O ports each of 8-bit, which can be configured as input or output.
- Total 32 I/O pins allows the microcontroller to be connected with the peripheral devices.

INTERRUPTS IN 8051

- Interrupt is a signal send by an external device to the processor so as to request the processor to perform a particular task or work.
- When a processor recognizes an interrupt, it saves the processor status in stack. Then it call and execute interrupt service routine(ISR). At the end of ISR, it restores the processor status and the program control is transferred to main program.
- Types of interrupts are:

  1) **Software & Hardware interrupt**

  2) Vectored & non-vectored interrupt

  3) Maskable & non-maskable interrupt

❖Types of Interrupt in 8051

➢ The 5 sources of interrupt in 8051 are:

1) Timer 0 overflow interrupt – TF0
2) Timer 1 overflow interrupt – TF1
3) External hardware interrupt – INT0
4) Inernal hardware interrupt – INT1
5) Serial communication interrupt – RI/TI

There are two SFR's for interrupt handling

IE -> Interrupt Enable Register

IP -> Interrupt Priority Register

❖Interrupt Enable (IE) Register

- Responsible for enabling and disabling the interrupt.
- It is a bit addressable register.

7	6	5	4	3	2	1	0
EA	X	X	ES	ET1	EX1	ET0	EX0

- EX_0 -> Enable/Disable – External Interrupt 0
- ET_0 -> Enable/Disable – Timer 0 overflow interrupt
- EX_1 -> Enable/Disable – External Interrupt 1
- ET_1 -> Enable/Disable – Timer 1 overflow interrupt
- ES -> Enable/Disable – Serial port interrupt
- EA -> Enable/Disable – All interrupts

❖Interrupt Priority (IP) register

- It is possible to change the priority levels of the interrupts by setting or clearing the corresponding bit in the IP register.
- If the interrupt priorities are not programmed, the microcontroller executes in predefined manner in the order INT0, TF0, INT1, TF1, SI

❖TCON Register

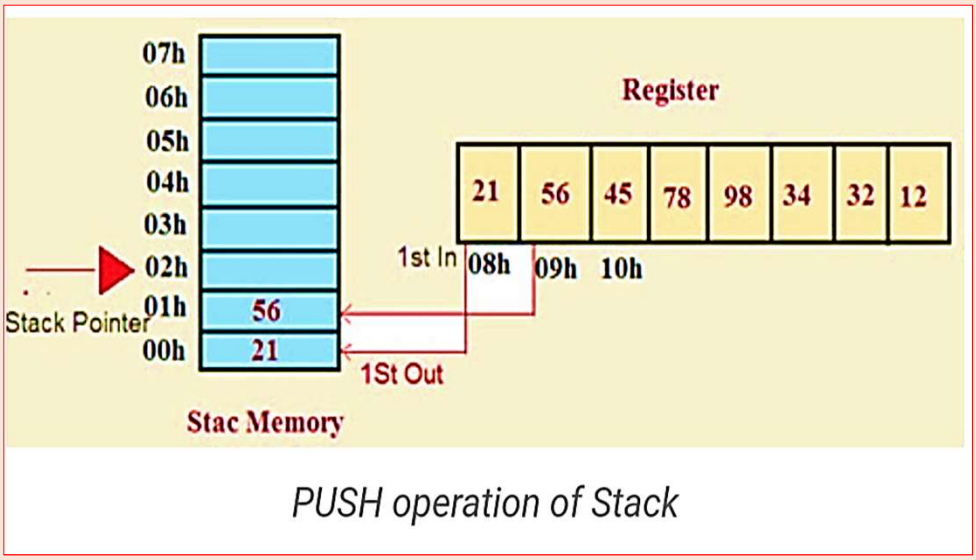
- TCON register specifies the type of external interrupt to the 8051 microcontroller.

INTERRUPT	ADDRESS
IE0	0003
TF0	000B
IE1	0013
TF1	001B
SERIAL	0023

STACK IN 8051

- Stack is an area of RAM allocated to hold temporarily all the parameters of the variable
- Whenever the function is called , the parameters and local variables are added to the stack (PUSH)
- When the function returns, the parameter and the variables are removed(POP) from the stack.
- The register used to access the stack is called Stack Pointer register. The Stack Pointer (SP) is a small register used to point at the stack.
- When we push something in to the stack memory, the stack pointer increases.

❖PUSH Operation

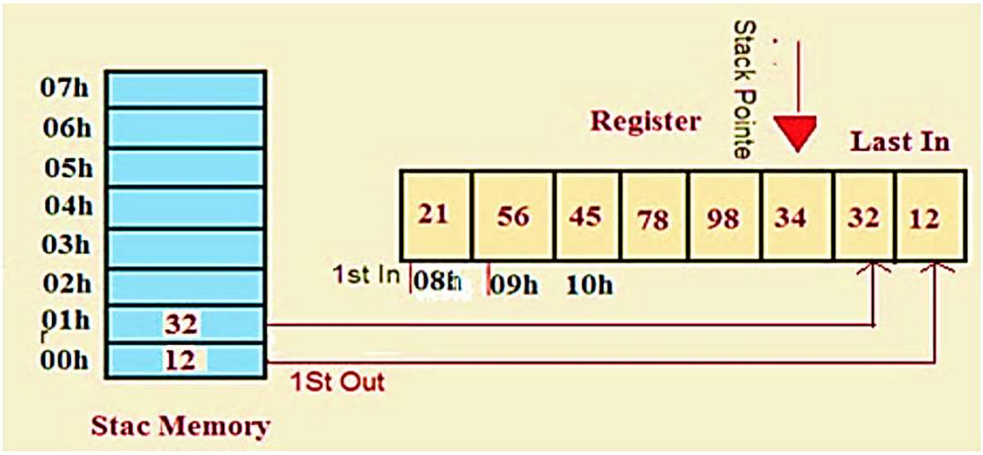


PUSH operation of Stack

- The PUSH is used for taking the values from any register and storing in the starting address of the stack pointer. ie; 00H by using PUSH operation
- For the next PUSH, it increments +1 , and stores the value in the next address of the stack pointer. ie, 01H
- Eg: 0000H

```
MOV 08H, #21H
MOV 09H, #56H
PUSH 00H
PUSH 01H
END
```

❖POP Operation



POP Operation in Stack

- It is used for placing the values from the stack pointer's maximum address to any other register's address
- If we use this POP again , then it decrements by 1 and the value stored in any register is given as POP
- Eg: 0000H

```
MOV 00H, #12H
MOV 01H, #32H
POP 1FH
POP 0EH
END
```

8051 ADDRESSING MODES

- It is the way in which the instruction is specified or different ways that are used by the processor to access or store data.
- 8051 provides 5 addressing modes.

1. Immediate
2. Direct
3. Register
4. Register Indirect
5. Indexed

1. Immediate Addressing Mode

- Here the source operand is constant.
- Immediate data is preceded by the sign ' # '
- This addressing mode can be used to load information in to any of the registers.
- Eg: MOV A, #25H; (25H is load to register A)
  MOV R4, #64H;
  MOV DPTR, #40H

2. Direct Addressing Mode

- In this mode the direct address of memory location is provided in instruction to fetch the operand.
- Only the internal RAM and SFR's address can be used in this type of instruction.
- Eg: MOV A, 30H; (content of RAM address 30H is copied in to Accumulator)

  MOV 56H, A; (save the content of A into RAM location 56H)

  MOV R6, R2; (it is invalid)

3. Register Addressing Mode

- Here the operand is contained in the specific register of microcontroller
- The user must provide the name of register from where the operand/Data need to be fetched.
- The permitted registers are A, R7-R0 of each register bank.
- Eg: MOV A, R0 ; (content of R0 register is copied in to Accumulator)

  MOV DPTR, A (Invalid)

4. Register Indirect Addressing Mode

- Here the address of memory location is indirectly provided by a register
- The " @" sign indicates that the register hold the address of memory location , ie. Fetch the content of the memory location whose address is provided in register.
- Eg: MOV A, @R0 ; (copy the content of memory location whose address is given in R0 register)

5. Indexed Addressing Mode

- This addressing mode is basically used for accessing data from Look- up table located in program code ROM of 8051
- Here the address of memory is indexed, ie, added to form the actual address of memory.
- Eg: MOVC A, @A+ DPTR (Here C means code. The content of A register is added with content of DPTR and the result is copied to A register)
- Because the data elements are stored in the program space ROM of the 8051, it uses the instruction MOVC instead of MOV

8051 INSTRUCTION SETS

DATA TRANSFER	ARITHMETIC	LOGICAL	BOOLEAN	PROGRAM BRANCHING
MOV	ADD	ANL	CLR	LJMP
MOVC	ADDC	ORL	SETB	AJMP
MOVX	SUBB	XRL	MOV	SJMP
PUSH	INC	CLR	JC	JZ
POP	DEC	CPL	JNC	JNZ
XCH	MUL	RL	JB	CJNE
XCHD	DIV	RLC	JNB	DJNZ
	DA A	RR	JBC	NOP
		RRC	ANL	LCALL
		SWAP	ORL	ACALL
			CPL	RET

- An 8051 Instruction consists of an Opcode followed by Operand(s) of size Zero Byte, One Byte or Two Bytes.
- The Op-Code part of the instruction contains the Mnemonic, which specifies the type of operation to be performed. All Mnemonics or the Opcode part of the instruction are of One Byte size.
- Coming to the Operand part of the instruction, it defines the data being processed by the instructions. The operand can be any of the following:
  - Data value
  - I/O Port
  - Memory Location
  - CPU register

- There can multiple operands and the format of instruction is as follows:

  MNEMONIC DESTINATION OPERAND, SOURCE OPERAND

- Based on the operation they perform, all the instructions in the 8051 Microcontroller Instruction Set are divided into five groups. They are:

1. **Data Transfer Instructions**
2. **Arithmetic Instructions**
3. **Logical Instructions**
4. **Boolean or Bit Manipulation Instructions**
5. **Program Branching Instructions**

1. Data Transfer Instructions

- Data transfer instructions are associated with transfer of data between registers or external program memory or external data memory

Sl. No	Instruction Format	Function Performed
1.	MOV dest, src	Copy the Content of Source to Destination
2.	MOVX dest, src	Used to move to/from external data memory only
3.	MOVC dest, src	Used to move from program memory (ROM) Only
4.	PUSH src	Copies one byte from source to stack top
5.	POP dest	Copies one byte from stack top to destination
6.	XCH	Exchanges data between two sources
7.	SWAP A	Exchanges upper and lower nibbles of A

2. Arithmetic Instructions

- It performs several basic operations such as addition, subtraction, division, multiplication etc. After execution , the result is stored in the first operand.

Sl. No	Instruction Format	Function Performed
1.	ADD A ,Rn	Adds the register to the accumulator
2.	ADDC A ,Rn	Adds the register to the accumulator with a carry flag
3.	SUBB A ,Rn	Subtracts the register from the accumulator with a borrow
4.	INC A	Increments the accumulator by 1
5.	DEC A	Decrements the accumulator by 1
6.	MUL AB	Multiplies A and B
7.	DIV AB	Divides A by B
8.	DA A	Decimal adjustment of the accumulator according to BCD code
9.	CLR A	Clear the value in A(a=0)
10.	CJNE dest,src,target	compare the Source and Destination, and jump to target if they are not equal

3. Logical Instructions

- Logical Instructions perform logical operations

Sl. No	Instruction Format	Function Performed
1.	ANL dest,src	Logically AND the source and the Destination
2.	ORL dest,src	Logically OR the source and the Destination
3.	CPL dest	Complement- Logically NOT the destination
4.	XRL dest,src	Logically XOR the source and the Destination

4. Boolean or Bit Manipulation Instructions

- Boolean or Bit Manipulation Instructions will deal with bit variables.
- There is a special bit-addressable area in the RAM and some of the Special Function Registers (SFRs) are also bit addressable.

Boolean or Bit manipulation instructions are:

a) Data manipulation instructions
b) Port instructions
c) Complement instructions
d) Rotate instructions

a) Data Manipulation Instructions

a. Data Manipulation Instructions

Sl. No	Instruction Format	Function Performed
1.	SETB Bit	Set the indicated Bit
2.	CLR Bit	Clear the indicated Bit
3.	CPL Bit	Complement the Indicated Bit
4.	MOV C, Bit	Move the indicated bit to C(Carry Flag)
5.	MOV Bit, C	Move the C(Carry Flag) to indicated bit
6.	ANL C, Bit	Move to C(Carry Flag) the logical AND of C and the indicated bit
7.	ANL Bit, C	Move to the bit , the logical AND of C and the indicated bit
8.	ORL C, Bit	Move to C(Carry Flag) the logical OR of C and the indicated bit
9.	ORL Bit, C	Move to the bit , the logical OR of C and the indicated bit

b) Port Instructions

- Ports can only take data in (from outside) or put data out.
- There are no special instructions for ports in 8051, but it uses MOV instructions for this.

Ex: **MOV A, P1** (copy the content of port1 to accumulator)

MOV P2, R1 (copy the content of Register R1 to port2)

C) Complement Instructions

- Here only A register and direct RAM address can be used as the destination.

Ex: **CPL A** (Complement the contents of A)

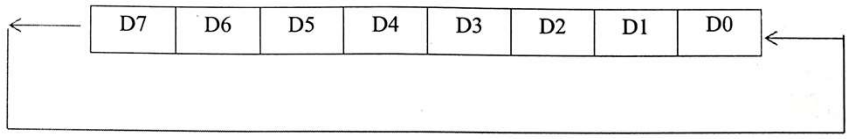
CPL 43H (Complement the contents of RAM address 43H)

d) Rotate Instructions

- There are 4 rotate and 1 swap instructions for the chip.
- Only the A register can be used as the destination

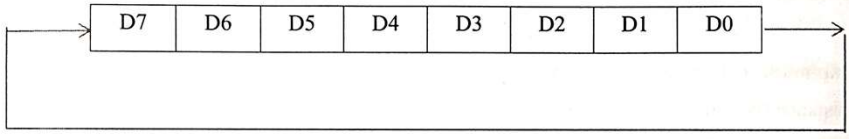
i) **RL A ; Rotate left the content of A**

The A register is rotated left and D7 appears in the D0 position after rotation.



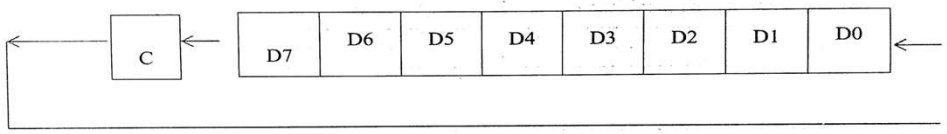
ii) **RR A ; Rotate right the content of A**

The A register is rotated right and the bit D0 is moved to D7



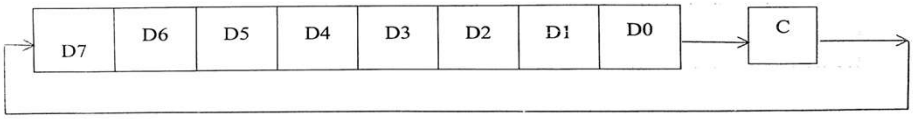
iii) RLC A ; Rotate left through carry

The A register is shifted left and bit D7 is moved to carry, while the carry bit is moved to D0 of A

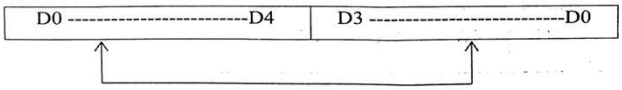


iv) RRC A ; Rotate RIGHT through carry

The A register is shifted right and bit is moved to D7, while the D0 moves into the carry.



v) SWAP A ; This is a special instruction where the lower and upper nibbles of A are swapped.



Prepared By Mr. EBIN PM, Chandigarh University, Punjab EDULINE 55

5. Program Branching Instructions (Flow control instructions)

- **Flow control instructions** are used to divert the flow of the program
- These instructions are used to implement loops and subroutine calls
- Branch instructions change the flow of a program by modifying the program counter (PC)

❖ Unconditional Branch Instructions:

1) SJMP Target

- SJMP stands for Short Jump and it is a relative jump
- Here the destination of jumping is expressed as a relative number and it will be short.ie, 8 bits only
- This 8 bit number represents distance between current PC value and target address

Prepared By Mr. EBIN PM, Chandigarh University, Punjab EDULINE 56

- If this number is negative it is backward and the maximum range of which is **-128**
- If it is a forward jumping the number is positive with maximum range of **+127**

2) LJMP Target

- This is a Long Jump instruction and is not relative
- It is a three byte instruction
- When this instruction is executed, the current PC value is simply replaced by the 16 bit number in the instruction, which may vary from 0000 to FFFF.

3) AJMP Target

- This is an Absolute Jump instruction and also a relative jump.
- Here the range of jump is 2K and 11 bits specify the destination range.

❖ **Conditional Branch Instructions :**

- These are the instructions which make the programs really useful.
- Computers are used for repetitive and conditional tasks and conditional branching is the method for it.
- **All conditional jumps are short jumps**

Sl. No	Instruction Format	Function Performed
1.	JC target	Jump if CY=1
2.	JNC target	Jump if CY=0
3.	JZ target	Jump if the register A=0
4.	JNZ target	Jump if the register A is not zero
5.	JB bit, target	Jump if the bit=1
6.	JNB bit, target	Jump if the bit=0
7.	JBC bit, target	Jump if the bit=1. Then clear bit
8.	DJNZ byte, target	Decrement the byte, and jump if the byte is zero