JUNAID . GIRKAR

14/12/21 POA 60004190057

TT2 TE COMPS A4

Q1

ANS The term addressing modes refers to the way in which the operand of an instruction is specified. The addressing mode specifies a rule for interpreting or modifying the address field of the field before the operand is actually executed.

According to their ~~typ~~ purpose, they are divided into two types :-

- Addressing modes for data
- Addressing modes for branch

According to different ways of specifying an operand by 8086 microprocessor, different addressing modes are used in 8086 programming. They are :-

- IMMEDIATE ADDRESS MODE : In this mode, the data operand is a part of the instruction itself.

E.g : MOV CX, 4929 H, ADD AX, 2387 H, MOV AL, FFH.

- REGISTER ADDRESSING MODE : It means that the register is the source of an operand for an instruction.

E.g : MOV CX, AX ; copies contents of 16 bit AX register into 16 bit CX
ADD BX, AX
XOR AX, DX

60004190057

JAGirkar

- DIRECT ADDRESSING MODE: the addressing mode in which the effective address of the memory location is written directly in the instruction

E.g: MOV AX, [1592H], MOV AL, [0300H]

- REGISTER INDIRECT ADDRESSING MODE: This addressing mode allows data to be addressed at any memory location through an offset address held in any of the following registers: BP, BX, DI & SI

E.g: MOV AX, [BX] ; suppose BX contains 48H, then contents of 48H
; are moved to AX

ADD CX, {BX}

- BASED ADDRESSING MODE: In this addressing mode, the offset address of the operand is given by the sum of contents of the BX / BP registers and the 8-bit / 16-bit displacement

E.g: MOV DX, [BX+04], ADD CL, [BX+08]

- INDEXED ADDRESSING MODE: In this addressing mode, the operands offset address is found by adding the contents of SI or DI register and 8/16-bit displacements.

E.g: MOV BX, [SI+16]; ADD AL, [DI+1C]

60004190057
JAGirkar

- **BASED-INDEX ADDRESSING MODE :** In this, the offset adress of the operand is computed by summing the base register to the contents of an index register.

E.g: ADD CX, [AX+SI], MOV AX, [AX+DI]

- **BASED INDEXED WITH DISPLACEMENT MODE :** In this, the operands offset is computed by adding the base register contents. An index registers contents and 8 or 16-bit displacement

E.g: MOV AX, [BX+DI+08], ADD CX, [BX+SI+16]

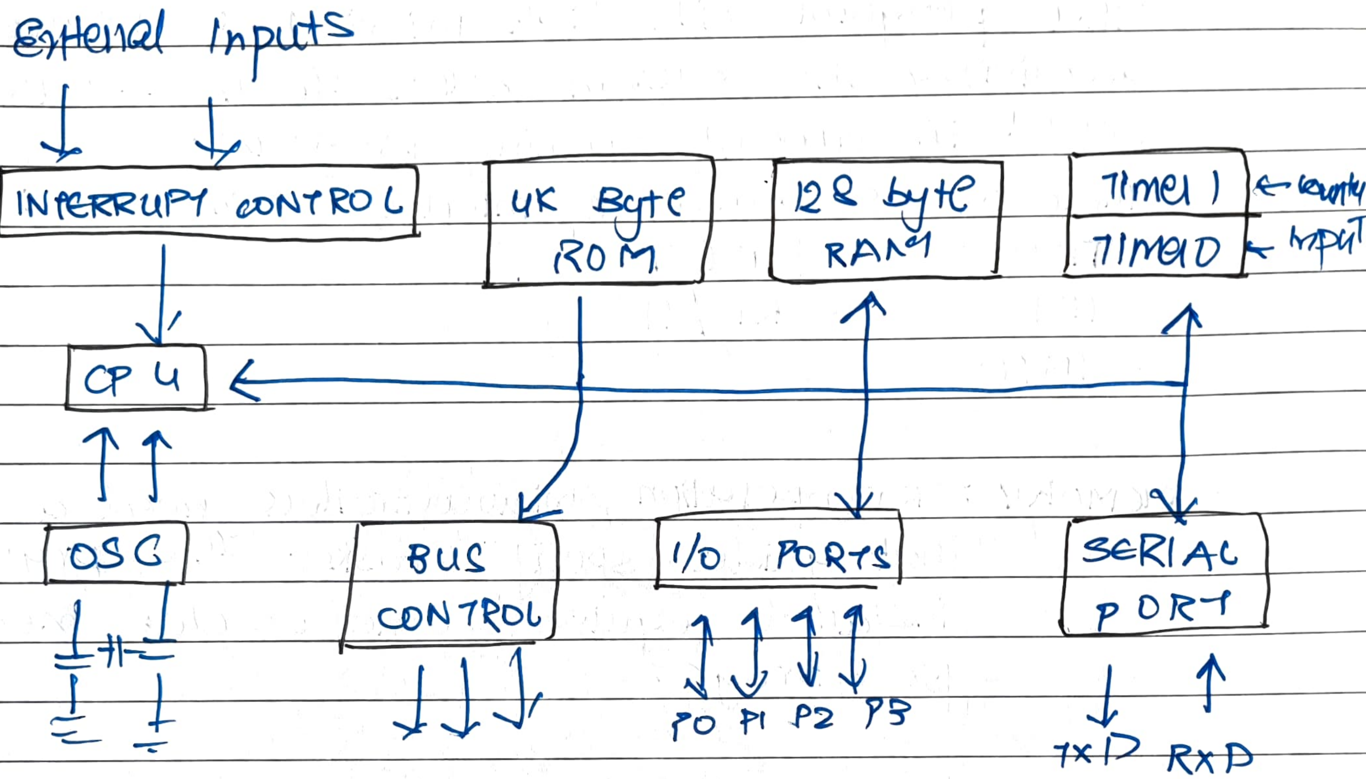
Q2
ANS

```
Data Segment
String1  DB   08h, 14h, 05h, 0fh, 09h
res  db  ?
data  ends

code  segment
 Assume  cs : code ,  ds : data
 Start  :
    mov  ax, data
    mov  ds , ax
    mov  cx, 04h
    mov  bl, 00h
    LEA  SI , String1
   up:
       mov   al , [SI]
       cmp  al , bl
       jl nxt
   nxt :
          inc  si
          dec  cx
          jnz  up

    mov  res , bl
    int  3
    code  ends
    end  start
```

Q3



8051 microcontroller
Architecture diagram.

Basic components present in 8051 microcontroller architecture:

- CPU : It acts as a mind of any processing machine. It synchronizes and manages all processes that are carried out in microcontroller. User has no control over CPU functioning. It interprets the program stored in ROM and performs the projected duty. CPU manages the different types of registers in 8051 microcontroller.

- INTERRUPTS : It is a sub-routine call that given by the microcontroller when some other program with high priority is request for acquiring the system buses then interrupts occur in current running program.

There are 5 sources of interrupts in 8051 :

- TF 0
- TF1
- INT0
- INT 1
- RI / TI

- MEMORY : For operation, microcontrollers need a program that guides specific tasks. This prgram installed requires some on chip memory for storage.

- BUS : Bus is a group of wires which uses as a communication canal or acts as a means of data transfer. A bus can bear 8 bits, 16 bits. all together. 2 types of buses in 8051 :-
  - Address bus
  - Data bus

- OSCILLATOR : As it is a digital circuit, it needs a timer that is externally connected or on-chip. Therefore 8051 uses two 16 bit counters and timers. For the operation of these timers & counters, an oscillator is used