8085

8-bit.

8086.

16-bit.

Data bus → 16-bit.

Address → 20bit

2^{20} : 1MB.

Microprocessor 8086

$2^{20} = 1M$.

- It is a 16-bit Microprocessor having 20 bit address lines and 16 bit data lines that provides up to 1MB storage.
- The Architecture of 8086 consists of a 16-bit ALU, a set of 16-bit register and provides segmented memory addressing capability, a rich instruction set, powerful interrupt structure, fetched instruction queue for overlapped fetching and execution etc.
- It supports two modes of operation, i.e. Maximum mode and Minimum mode. Maximum mode is suitable for system having multiple processors and Minimum mode is suitable for system having a single processor.
- The complete architecture of 8086 can be divided into two parts

Bus Interface Unit (BIU)

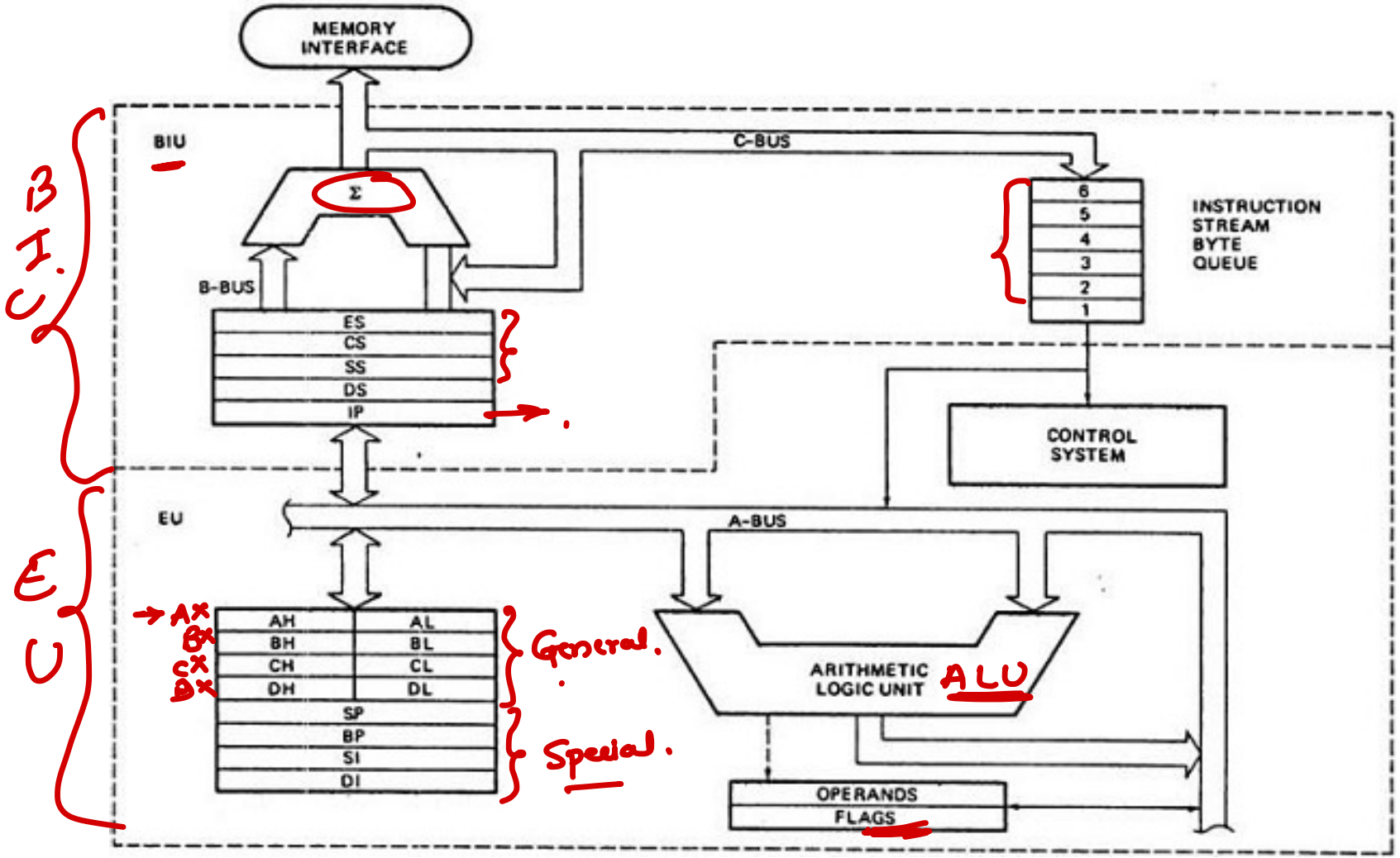
- BIU contains Instruction queue, Segment registers, Instruction pointer, Address adder.

Execution Unit (EU)

- EU contains Control circuitry, Instruction decoder, ALU, Registers, Flag register

Microprocessor 8086

X86



16-bit → AX, BX, CX, DX → General.

AH (8-bit), AL, BH, BL, CH, CL, DH, DL.

Acc ← Result

BX → offset address to store data.

CX → count.

DX → division.

Special Purpose :

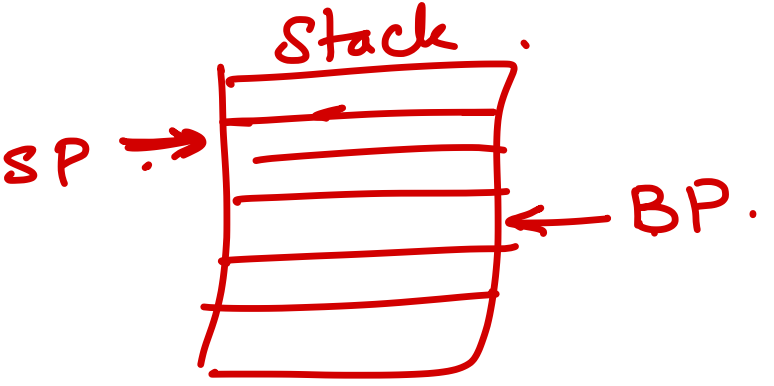
SI → string. } Offset addr of data.
DI → string. }

SP → Offset addr of Stack.

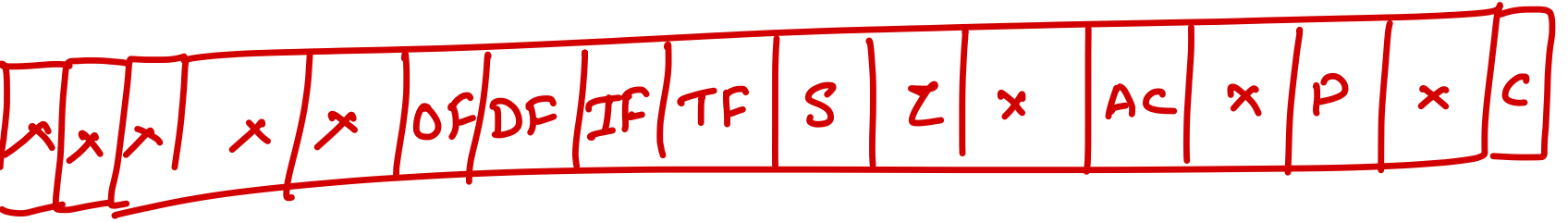
BP → stack → offset addr of stack.

IP → Offset addr of code.

PC.



Flag ⟶ 9 achive.



Microprocessor 8086

EU (Execution Unit)

- Execution unit gives instructions to BIU stating from where to fetch the data and then decode and execute those instructions. Its function is to control operations on data using the instruction decoder & ALU.

ALU

- It handles all arithmetic and logical operations, like +, –, ×, /, OR, AND, NOT operations.

Instruction pointer

- It is a 16-bit register used to hold the address of the next instruction to be executed.

Base Pointer

- BP can hold offset address of any location in the stack segment. It is used to access random locations of the stack.

Source Index

- It holds offset address in Data Segment during string operations.

Destination Index

- It holds offset address in Extra Segment during string operations.

Microprocessor 8086

General purpose register

- There are 8 general purpose registers, i.e., AH, AL, BH, BL, CH, CL, DH, and DL. These registers can be used individually to store 8-bit data and can be used in pairs to store 16 bit data. The valid register pairs are AH and AL, BH and BL, CH and CL, and DH and DL. It is referred to the AX, BX, CX, and DX respectively.
- **AX register** – It is also known as accumulator register. It is used to store operands for arithmetic operations.
- **BX register** – It is used as a base register. It is used to store the starting base address of the memory area within the data segment.
- **CX register** – It is referred to as counter. It is used in loop instruction to store the loop counter.
- **DX register** – This register is used to hold I/O port address for I/O instruction.

Stack pointer register

- It is a 16-bit register, which holds the address from the start of the segment to the memory location, where a word was most recently stored on the stack.

Flag Register

It is a 16-bit register that behaves like a flip-flop, i.e. it changes its status according to the result stored in the accumulator. It has 9 flags and they are divided into 2 groups − Conditional Flags and Control Flags.

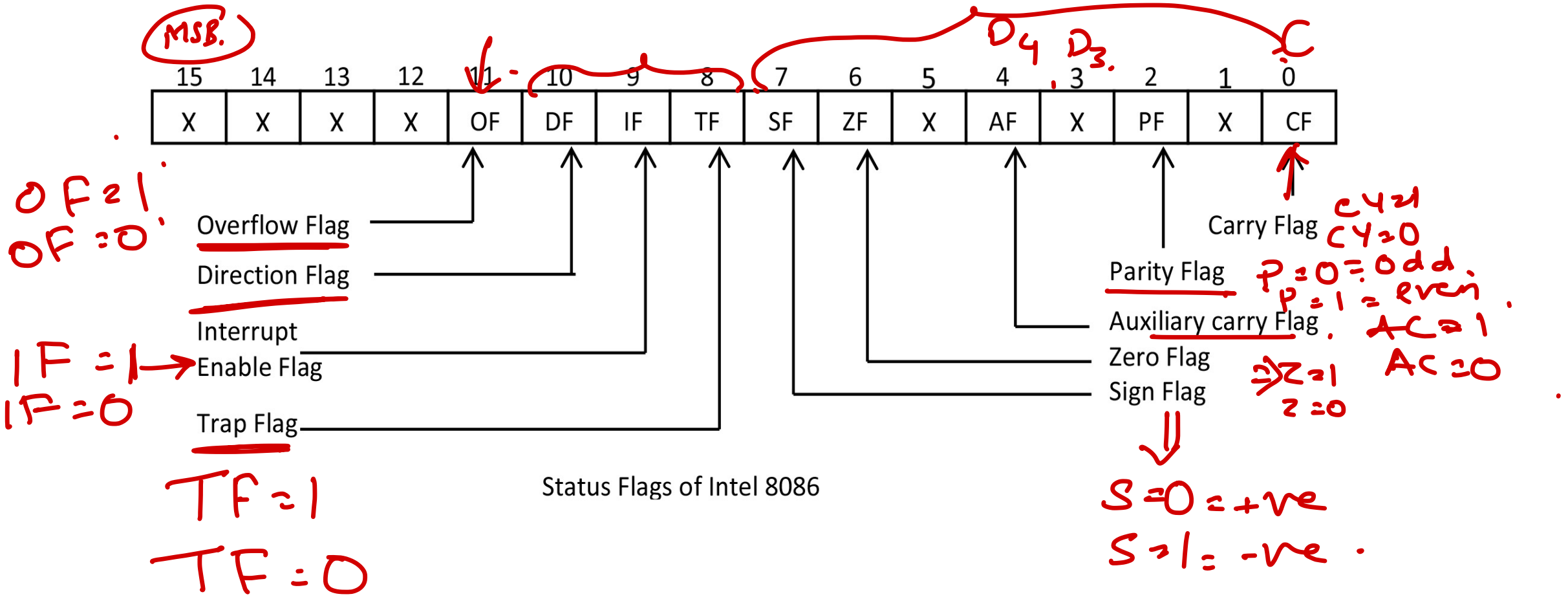
Conditional Flags

It represents the result of the last arithmetic or logical instruction executed. List of conditional flags are

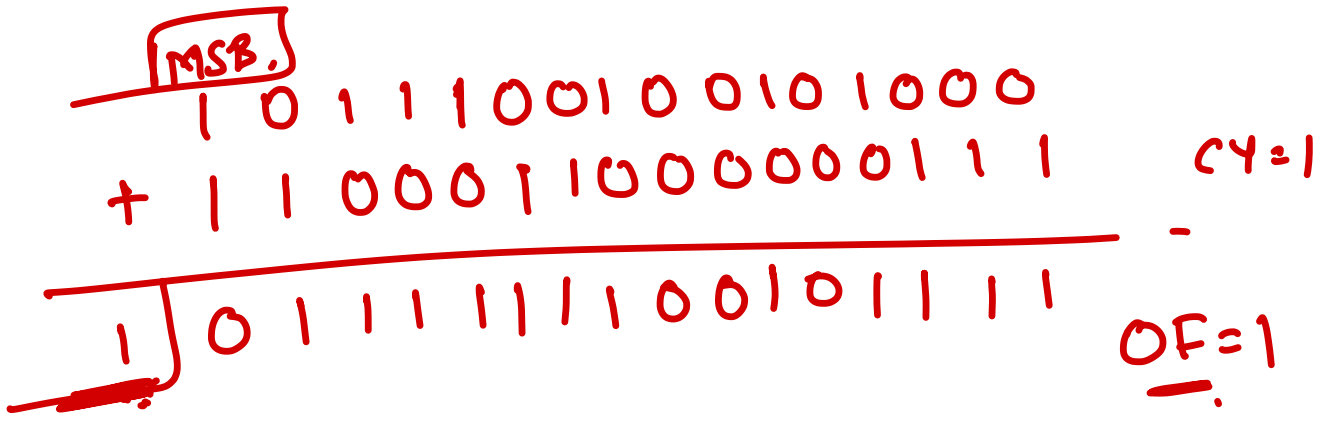
- **Carry flag** − This flag indicates an overflow condition for arithmetic operations.
- **Auxiliary flag** − When an operation is performed at ALU, it results in a carry/barrow from lower nibble (i.e. D0 – D3) to upper nibble (i.e. D4 – D7), then this flag is set, i.e. carry given by D3 bit to D4 is AF flag. The processor uses this flag to perform binary to BCD conversion.
- **Parity flag** − This flag is used to indicate the parity of the result, i.e. when the lower order 8-bits of the result contains even number of 1's, then the Parity Flag is set. For odd number of 1's, the Parity Flag is reset.
- **Zero flag** − This flag is set to 1 when the result of arithmetic or logical operation is zero else it is set to 0.
- **Sign flag** − This flag holds the sign of the result, i.e. when the result of the operation is negative, then the sign flag is set to 1 else set to 0.
- **Overflow flag** − This flag represents the result when the system capacity is exceeded.

Flag → Conditional, Control.

Flag Register



Status Flags of Intel 8086



Df =>

Sunbeam .

DF=0
autoincre.

DF=1
autodecre

Sunbeam

Control Flags

- Control flags controls the operations of the execution unit. List of control flags are
- **Trap flag** – It is used for single step control and allows the user to execute one instruction at a time for debugging. If it is set, then the program can be run in a single step mode.
- **Interrupt flag** – It is an interrupt enable/disable flag, i.e. used to allow/prohibit the interruption of a program. It is set to 1 for interrupt enabled condition and set to 0 for interrupt disabled condition.
- **Direction flag** – It is used in string operation. As the name suggests when it is set then string bytes are accessed from the higher memory address to the lower memory address and vice-a-versa.

BIU (Bus Interface Unit)

- BIU takes care of all data and addresses transfers on the buses for the EU like sending addresses, fetching instructions from the memory, reading data from the ports and the memory as well as writing data to the ports and the memory. EU has no direction connection with System Buses so this is possible with the BIU. EU and BIU are connected with the Internal Bus.

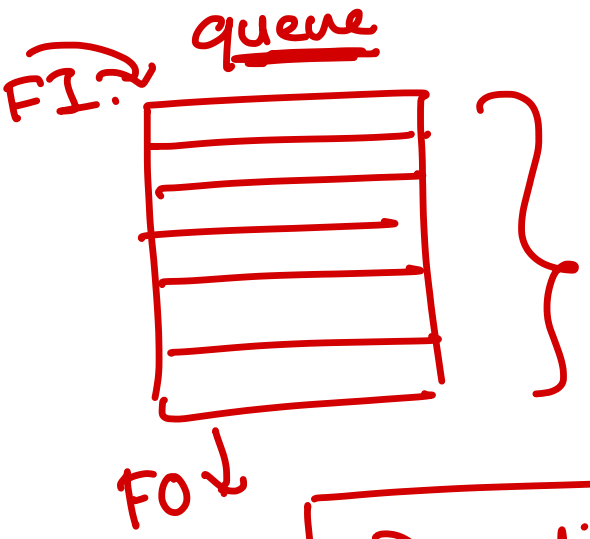
Instruction queue

- BIU contains the instruction queue. BIU gets upto 6 bytes of next instructions and stores them in the instruction queue. When EU executes instructions and is ready for its next instruction, then it simply reads the instruction from this instruction queue resulting in increased execution speed.
- Fetching the next instruction while the current instruction executes is called **pipelining**.

Segment register

- BIU has 4 segment buses, i.e. CS, DS, SS& ES. It holds the addresses of instructions and data in memory, which are used by the processor to access memory locations.

MVI A,20H
MVI B,50H
ADD B
OUT
HLT



pipeline.

Code Segment register: CS .

- CS is 16-bit register holds the base address for the Code Segment. All programs are stored in the Code Segment and accessed via the IP.

Data Segment register: DS .

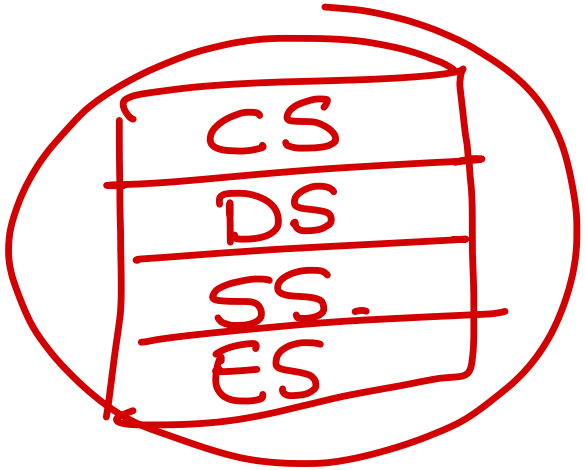
- DS is 16-bit register holds the base address for the Data Segment.

Stack Segment register: SS

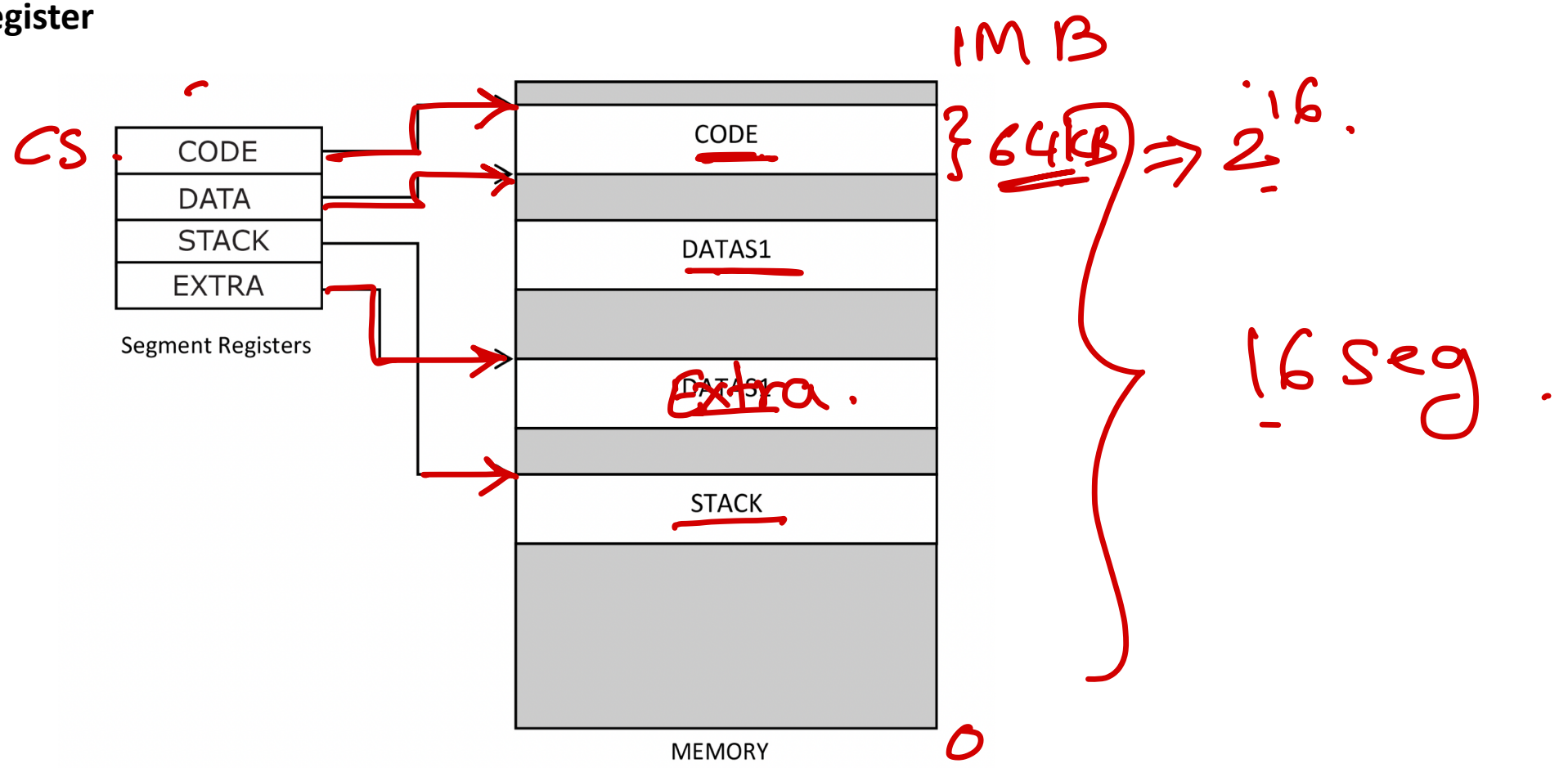
- SS is 16-bit register holds the base address for the Stack Segment.

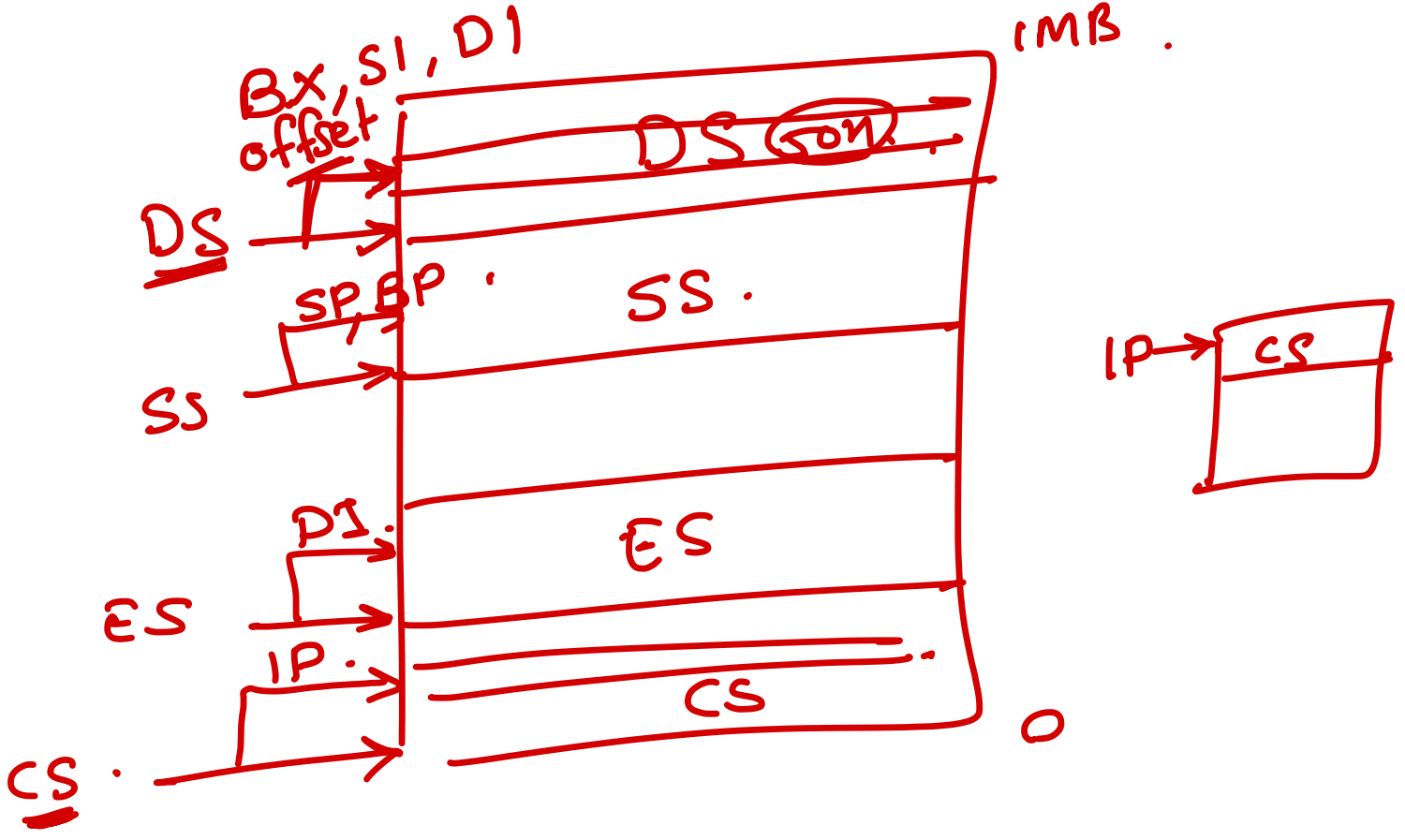
Extra Segment register: ES .

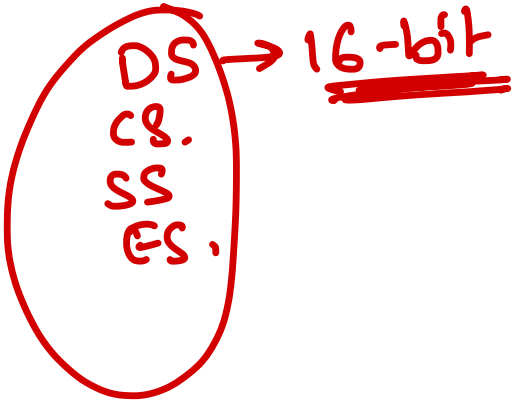
- ES is 16-bit register holds the base address for the Extra Segment.

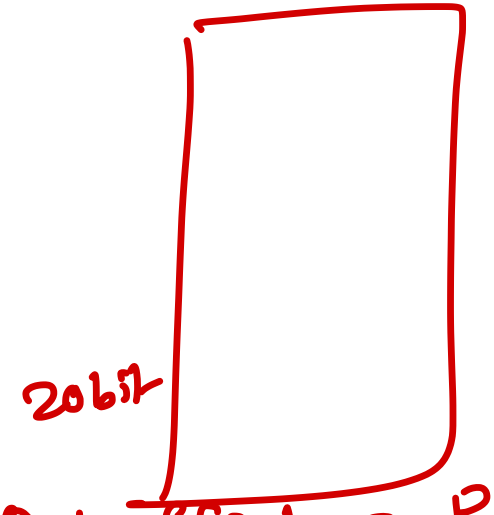


Segment Register




BX, SI, DI
offset
DS
DS (20H)
SP, BP
SS
SS
DI
ES
ES
IP
CS
CS
1MB
0
IP
CS





segment $\times 10$ + offset = Physical address.

DS $\times 10$ + BX.

CS = 3000H. → IP = 1234H.

PA = CS $\times 10$ + IP.

= 3000H $\times 10$ + 1234.

= 31234H //.

DS. = 2432H.

SHL →

$D_{15}D_{14}D_{13}D_{12}D_{11}D_{10}D_9D_8D_7D_6D_5D_4D_3D_2D_1D_0$.

16bit 0 0 1 0 0 1 0 0 0 0 1 1 0 0 1 0.

0 0 1 0 0 1 0 0 0 0 1 1 0010 0000.

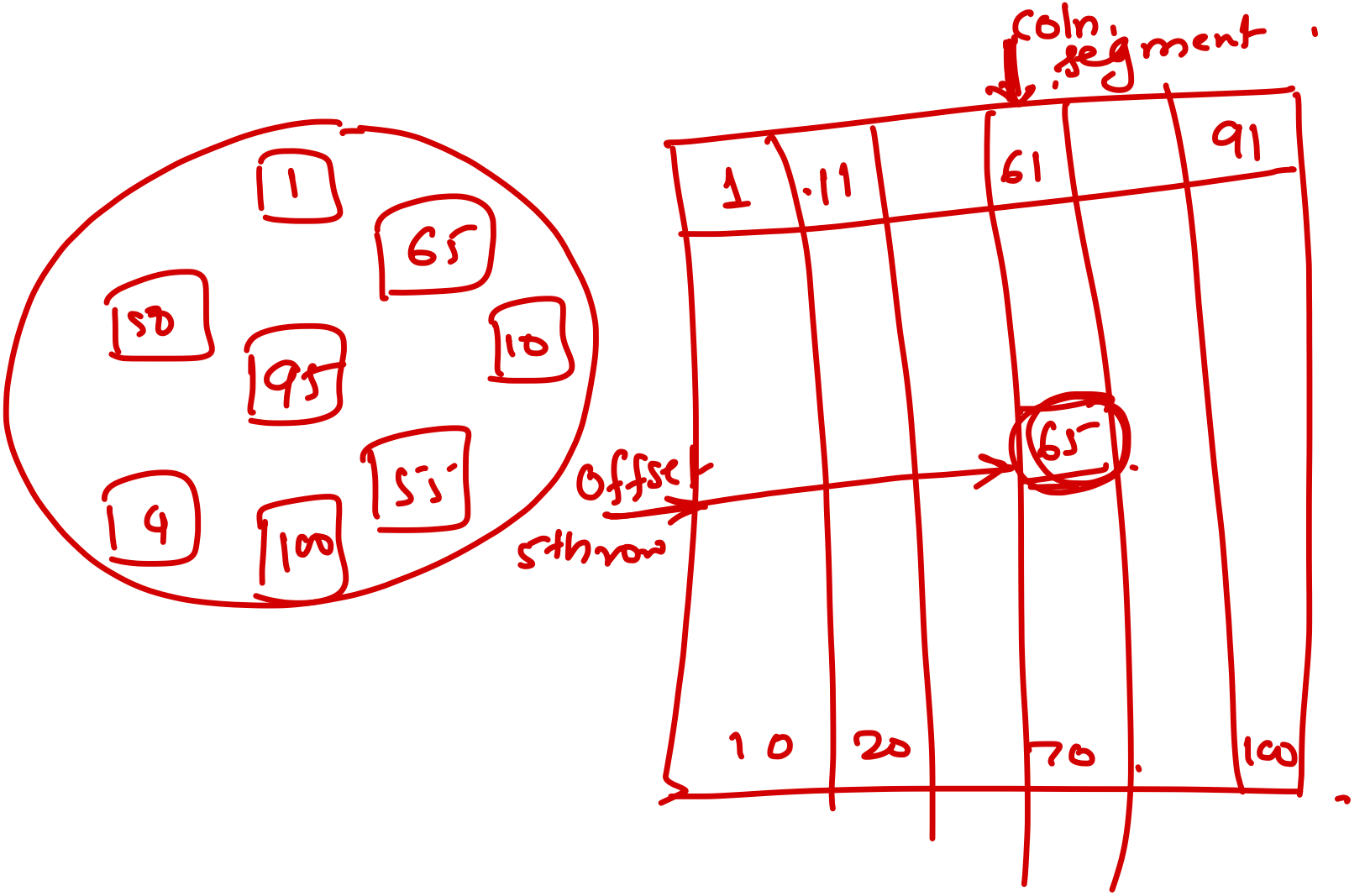
DS. ×10 + BX.

SI

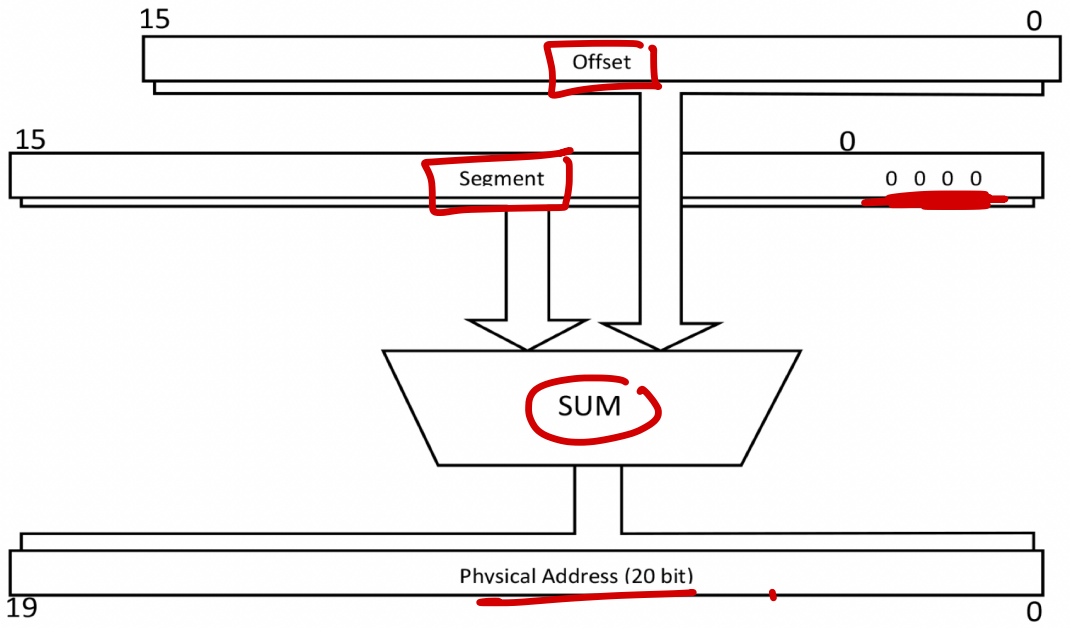
DI

8-bit dis.

16-bit disp.


1
65
50
10
95
55
4
100
Offset
5th row
Coln segment
1
11
61
91
65
10
20
70
100

Physical Address Calculation

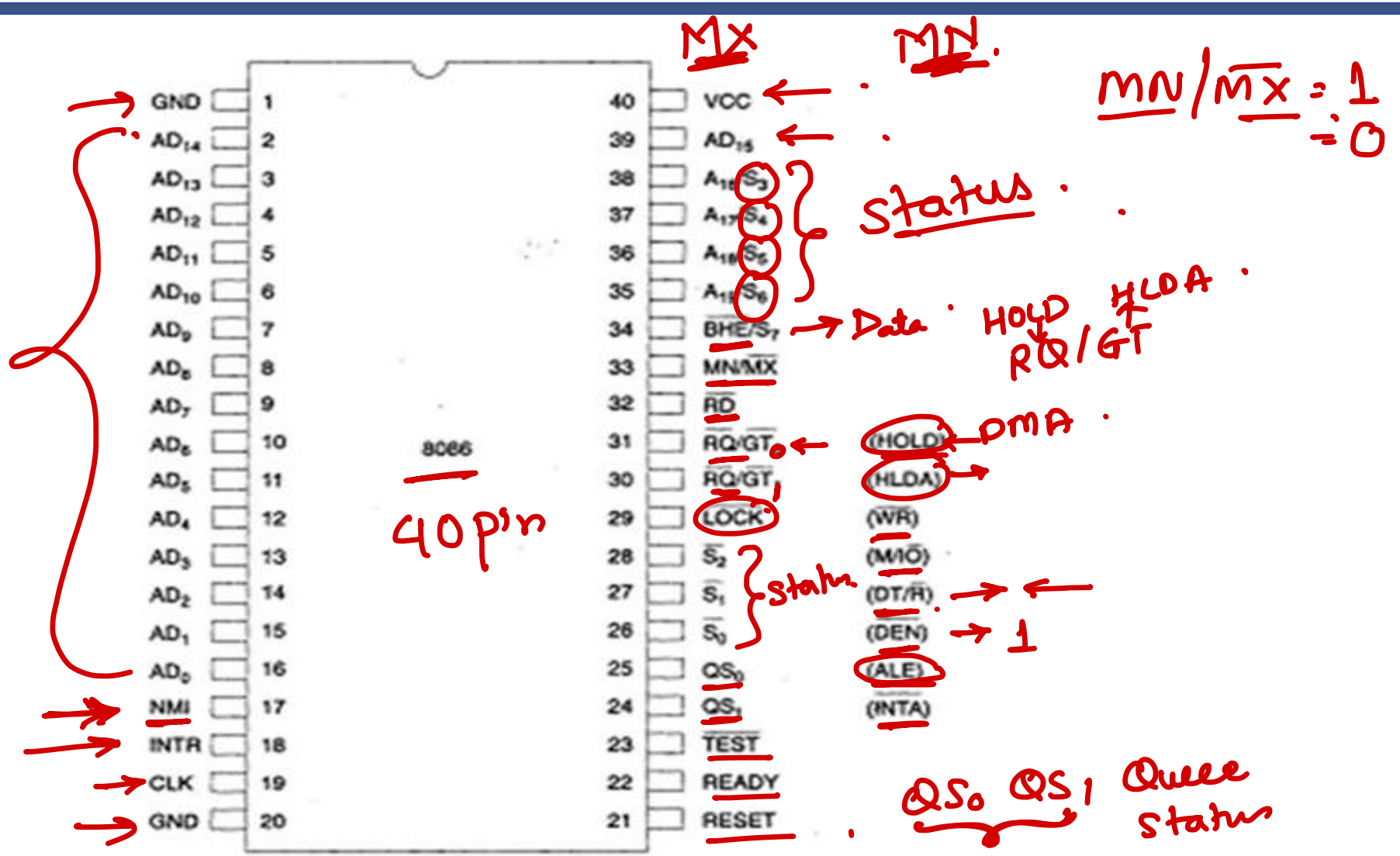


The BIU has a Physical Address Generation Circuit. It generates the 20 bit physical address using Segment and Offset addresses using the formula:

Physical Address = Segment Address x 10H + Offset Address

Microprocessor 8086

- Pin 8086



Microprocessor 8086

The description of the pins of 8086 is as follows:

AD0-AD15 : Bidirectional address/data lines. These are low order address bus. They are multiplexed with data.

A16/S3 - A19/S7: High order address lines. These are multiplexed with status signals.

BHE/S7 : Bus High Enable/Status. During T1, it is low. It enables the data onto the most significant half of data bus, D8-D15. 8-bit device connected to upper half of the data bus use BHE signal. It is multiplexed with status signal S7.

RD : For read operation. It is an output signal. It is active LOW signal .

Ready : The addressed memory or I/O sends acknowledgment through this pin. When HIGH, it denotes that the peripheral is ready to transfer data.

RESET : System reset. The signal is active HIGH.

Microprocessor 8086

- **CLK :**Clock 5MHz.
- **INTR:** Interrupt Request.
- **NMI :** Non-maskable interrupt request.
- **TEST :** Wait for test control. When LOW the microprocessor continues execution otherwise waits.
- **VCC :** Power supply +5V dc.
- **GND :** Ground.
- **LOCK :** It is an active LOW signal. When this signal is LOW, all interrupts are masked and no HOLD request is granted. In a multiprocessor system all other processors are informed through this signal that they should not ask the CPU for relinquishing the bus control.
- **RQ / GTO , RQ / GTI** : Request/Grant : Same like HOLD and HLDA signal
- **SQ0 , SQ1 : Queue Status .**Queue Status is valid during the clock cycle after which the queue operation is performed.

Microprocessor 8086

- **INTA :** Interrupts acknowledgement. On receiving interrupt signal, the processor issues an interrupt acknowledgment signal. It is active LOW.
- **ALE :** Address latch enable. ALE = 1, 8086 sends this signal to latch indicates address is available and when ALE = 0 , data is available .
- **DEN :** Data Enable. When Intel 8287/8286 octal bus transceiver is used this signal. It is active LOW.
- **DT/R :** Data Transmit/Receives. This signal controls the direction of data flow through the transceiver. When it is HIGH, data is sent out. When it is LOW, data is received.
- **M/IO :** Memory or I/O access. When this signal is HIGH, the CPU wants to access memory. When this signal is LOW, the CPU wants to access I/O device.
- **WR :**When this signal is LOW, the CPU performs memory or I/O write operation.
- **HLDA :** Hold Acknowledgment. It is sent by the processor when it receives HOLD signal. It is active HIGH signal.
- **HOLD :** When another device in microcomputer system wants to use the address and data bus, it sends HOLD request to CPU through this pin.

MOV ~~MVI~~ MVI A,20H AX → AL, AH

Addressing modes

Immediate addressing Mode - the data is provided in the instruction directly.

- MOV CX,4374H

MOV AX,2486H.

Direct Addressing Modes – the instruction operand specifies the memory address where data is located.

- MOV BL, [4374H]

MOV AX,[2486H] → add

Register Addressing Mode – copy the data from one to another register.

- MOV CX, AX

MOV AX,BX → reg.

Register Indirect Addressing Modes - Instruction specifies a register containing an address, where data is located. This addressing mode work with SI, DI, BX and BP register.

- MOV AL, [BX] or MOV AL, CS: [BX]

MOV AX,[BX] → offset

Register Relative Addressing Modes- Effective address is formed by adding an 8-bit or 16-bit displacement with contents of any one of register BX, BP, SI and DI

- MOV AX,50H[BX][SI]

↳ 8-bit
16-bit

MOV AX,50H[BX]

PH = DS x 10 + BX + 50H.

data

Indexed Addressing Modes - Offset is stored in one of the index register DS is default for register SI & DI

- MOV AX, [SI]

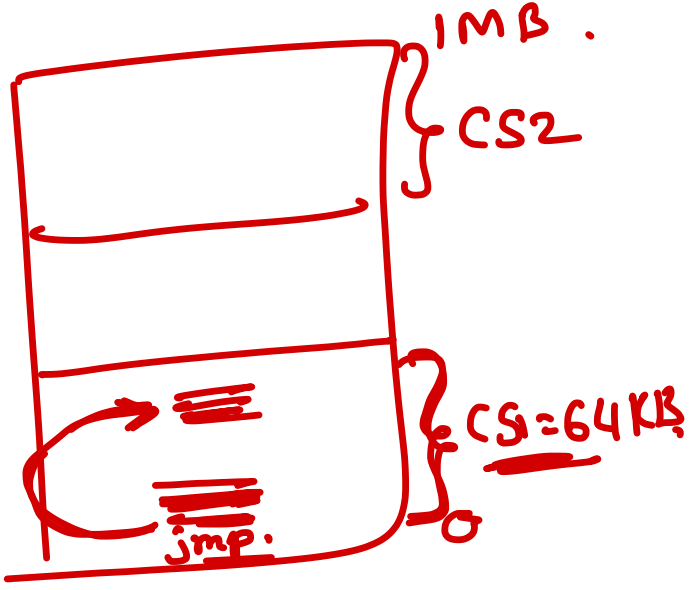
Based Indexed Addressing Modes - Effective address of data is formed by adding contents of base register (BX or BP)

- MOV AX, [BX][SI]

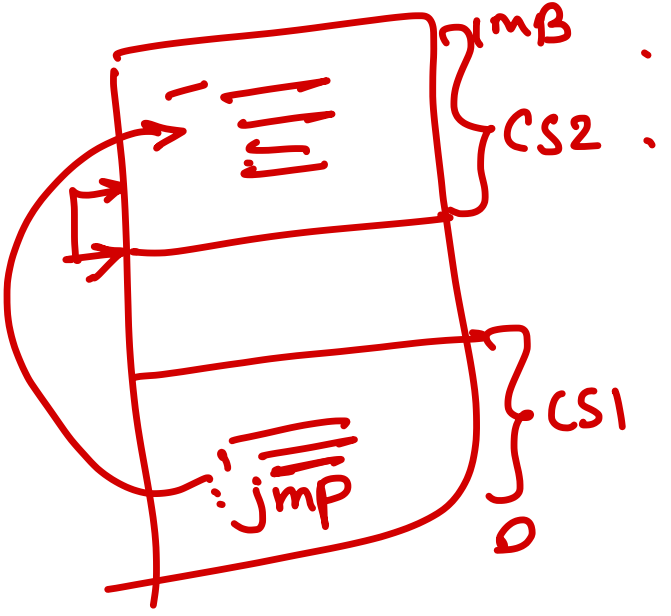
Relative Based Indexed Addressing Modes - Effective address is formed by adding an 8-bit or 16-bit displacement

- MOV AX,50H[BX][SI]

Intra



Inter.



Intrasegment Addressing Modes - Address to which the control is to be transferred lies in the same segment.

- (8 -bit) short jump - 128 to +127
- (16 -bit) long jump - 32768 to +32767

Intrasegment Indirect Addressing Modes –

- JMP [BX]

Intersegment Addressing Mode-Transfer to different segment CS & IP Destination are specified directly

- JMP 5000H:2000H

seg off

jmp CS:IP

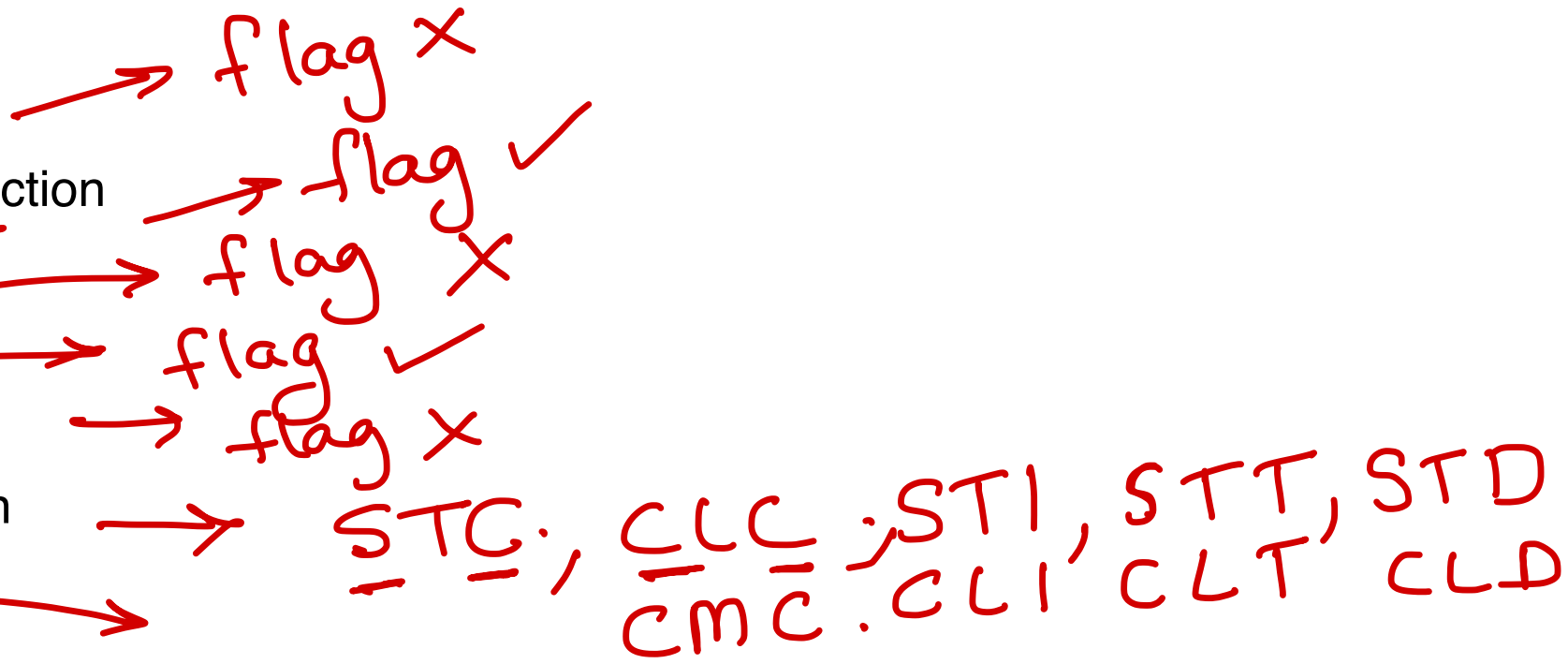
Intersegment Indirect Addressing Mode - Jump to an address in other segment specified at effective address 2000H in DS/CS (far jump)

CS

- JMP [2000H]

Instruction Set of 8086

- Data transfer instruction
- Arithmetic and Logical instruction
- Branching Instruction
- Loop Instruction
- Machine control Instruction
- Flag manipulation Instruction
- Shift & rotate Instruction
- String Instruction



Questions

Question 1

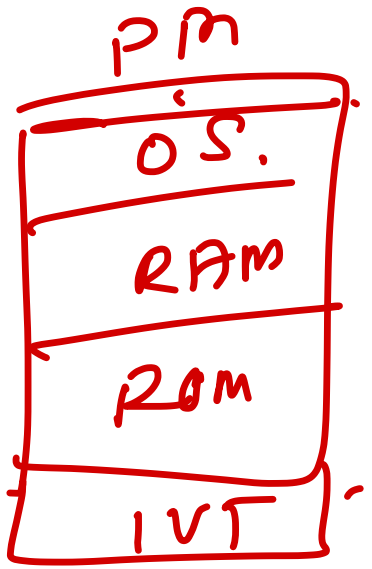
1) Virtual memory consists of

a) Static RAM
b) Dynamic RAM
c) Magnetic memory
d) None of these

Question 2

2) Cache memory acts between

a) CPU and RAM
b) RAM and ROM
c) CPU and Hard Disk
d) None of these



Question 3

3) Which of the following memory is non-volatile?

a) SRAM
b) DRAM
c) ROM
d) All of the above

Question 4

4) What does MAR stand for?

a) Main Address Register
b) Memory Access Register
c) Main Accessible Register
d) Memory Address Register

Question 5

5) In 8085, 16-bit address bus, which can address upto?

$2^{16} = 64KB$.

a) 16KB
b) 32KB
c) 64KB
d) 128KB

Question 6

6) It is also a 16-bit register works like stack, which is always incremented/decremented by 2 during push & pop operations.

a) Stack pointer
b) Temporary register
c) Flag register
d) Program counter

Question 7

7) This signal indicates that another master is requesting the use of the address and data buses.

a) READY
b) HOLD
c) HLDA
d) INTA

Question 8

8) MVI K, 20H is an example of?

a) Immediate addressing mode
b) Register addressing mode
c) Direct addressing mode
d) Indirect addressing mode

Question 9

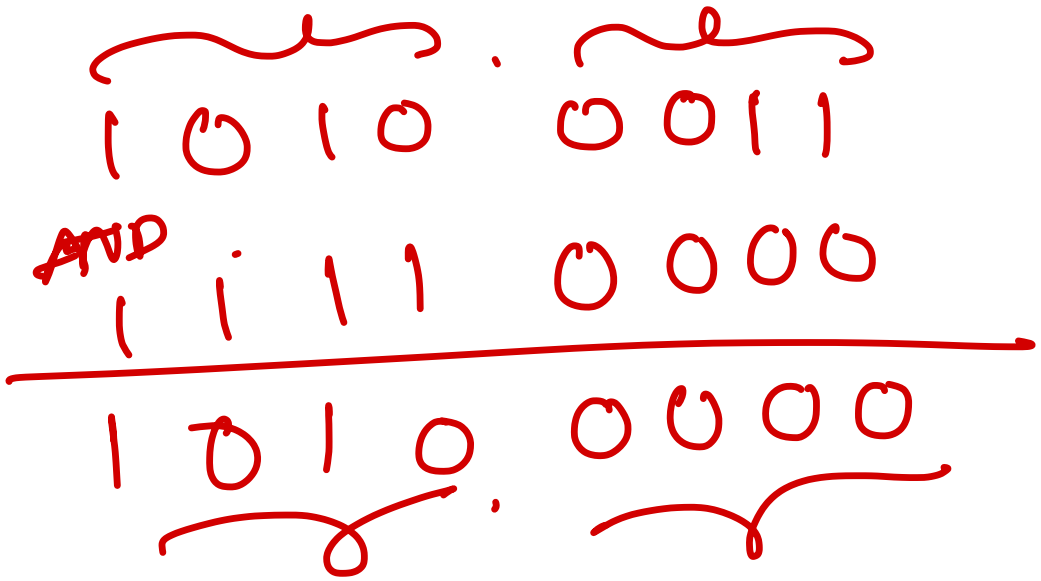
9) 8086 has ___ address bus.

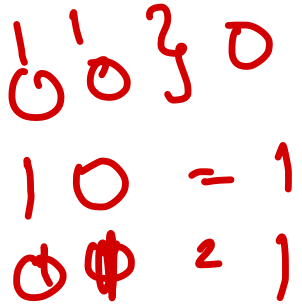
a) 16-bit
b) 18-bit
c) 20-bit
d) 24-bit

Question 10

10) An instruction which can be used to clear lower 4 bits of addressing 8085 is

a) XOR FFH
b) OR F0H
c) ANI F0H
d) None of Above





Question 11

11) Which of the below will not affect the status flags

a) JMP
b) ANI
c) OR
d) None of Above

Question 12

12) If data and instruction are stored in different memories the architecture is termed as _____

a) Von Neumann architecture
b) Harvard architecture
c) CICS Architectures
d) None of Above

Question 13

13) Which of the following is not true in case of pipelined processor

a) Single cycle execution
b) Multi Cycle execution
c) None

Question 14

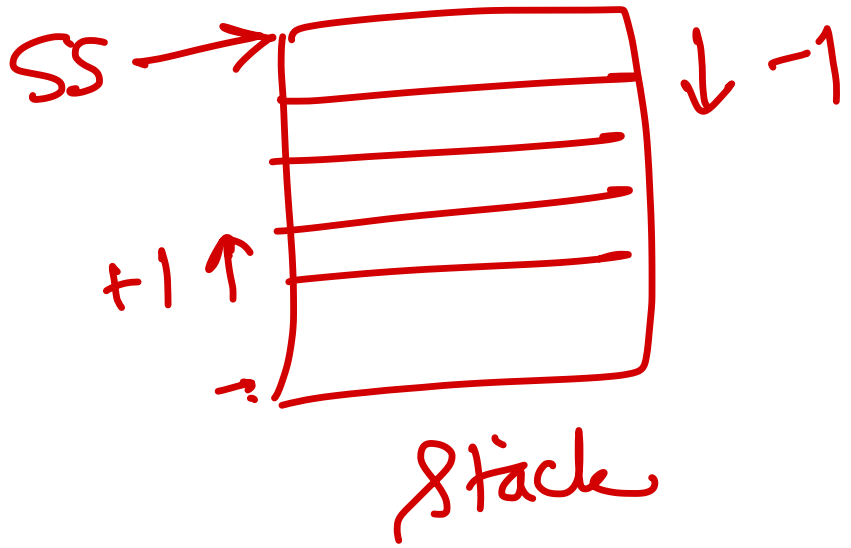
14) The kind of computer architecture that has common block of memory for both code as well as data and simultaneously cannot access both is called as _____

a) Von Neumann architecture
b) Harvard architecture
c) RISC architecture
d) Modified Harvard architecture

Question 15

15) The stack of X-86 based processor is

a) Upward growing stack
b) Downward growing stack
c) Configurable by the the programmer as upward growing or downward growing
d) None of the above



Question 16

16) The number of data lines of a processor basically indicates

a) The no of bits the processor can access
b) The total memory capacity(size) accessible by the processor
c) The processor has memory mapped I/O
d) None of above

Question 17

17) Which of the following are the typical characteristics of RISC machine

a) Instruction taking multiple cycle
b) Highly pipelined
c) Instruction interpreted by micro programs
d) Multiple register sets

Question 18

18) Memory access in RISC architecture is limited to instruction

a) CALL & RET
b) PUSH & POP
c) STA & LDA
d) MOV & JMP

Question 19

19) Single-bit indicators that may be set or cleared to show the result of logical or arithmetic operation are

a) Flags
b) Registers
c) Monitors
d) Decisions

Question 20

20) The register in the 8085 is used to keep track of the memory address of next op-code to be run in the program is

a) Stack pointer

b) Program Counter

c) Instruction pointer

d) Accumulator

Question 21

21) A register in the microprocessor that keeps track of the answer or result of any arithmetic or logic operation is

a) Stack Pointer
b) Program Counter
c) Instruction Pointer
d) Accumulator

Question 22

22) Which one of the following is not a vectored interrupt?

a) TRAP.
b) INTR
c) RST 7.5 .
d) RST 3

Question 23

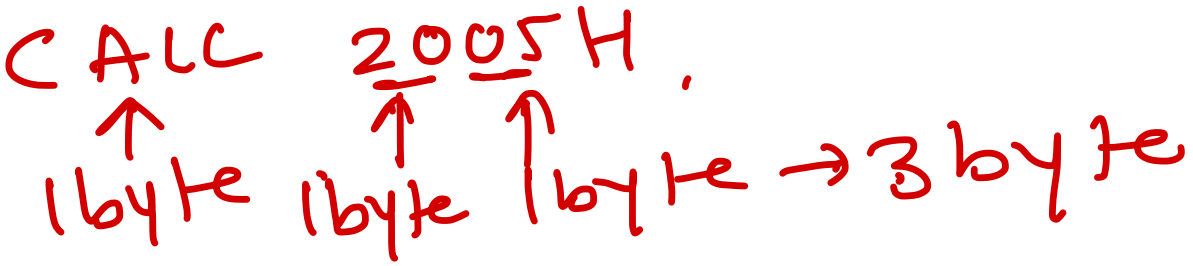
23) The processor status word of 8085 microprocessor has five flags namely

a) S, Z, AC, P, CY
b) S, OV, AC, P, CY
c) S, Z, OV, P, CY
d) S, Z, AC, P, OV

Question 24

24) CALL instruction is a _____ instruction.

a) 4 bytes
b) 2 bytes
c) 1 bytes
d) 3 bytes



Question 25

25) RST0 - RST7 are the _____ in 8085.

a) hardware interrupts
b) logical interrupts
c) software interrupts
d) conditional interrupts

Question 26

26) SIM used to implement the hardware interrupts (RST 7.5, RST 6.5, RST 5.5) by setting various bits to form masks or generate output data via the Serial Output Data (SOD) line.

a) RIM
b) SIM
c) EI
d) DI

Question 27

27) After the execution of CMP A instruction A = 20H & B = 50H. what is the status ?

a) ZF is set and CY is reset

b) ZF is set and CY is unchanged

c) ZF is reset and CY is set

d) ZF is reset and CY is unchanged

A < B .

Question 28

28) S_0 and S_1 pins are used for

a) serial communication
b) indicating the processor's status
c) acknowledging the interrupt
d) none of the above

Question 28

28) S_0 and S_1 pins are used for

a) serial communication
b) indicating the processor's status
c) acknowledging the interrupt
d) none of the above

① Determine the no of FF needed to construct a register capable of storing.

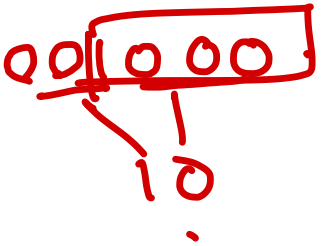
(a) 6-bit binary no → 6FF.

(b) Hex. no upto F → 4-bit → 4FF.

F → 1111

(c) Octal upto 10 → 4bit → 4FF.

10 → 001 000



Q) How many FF are required to implement each in Johnson. count.

① Mod 10:

Mod = 2N. → FF.

2 × 5 → 5FF.

② Mod 16.

2 × 8 = 16.

↓

8FF.

Q) Ring counter. → N Counter.

Mod 10.
↳ 10FF.

Mod 6.
↳ 6FF.