* What happens when the PC is powered up?

Ans: When the PC is powered up, the processor is put in a reset state, the CS register is set to FFFFh and IP is set to 0000h. So, the first instruction it executes is located at FFFF0h. This memory location is in ROM and it contains the instruction that transfer control to the starting point of the BIOS routines.

The BIOS routines first check for system and memory errors and then initialize the interrupt vectors and BIOS data area. Finally, BIOS loads the operating system from the system disk, which is done in two steps. First, the BIOS load a small program called boot program and then the boot program loads the actual operating system routines.

* Define multicore and manycore processor?

Ans:

A multicore processor is an integrated circuit with two or more separate processing units, called cores. Each core reads and executes program instructions, as if the computer had several processors.

Manycore processor is a specialist multi-core processor designed for a high degree of parallel processing, containing numerous simpler, independent processor cores. Manycore processors are used extensively in embedded computers and high-performance computing.

* Explain the difference between the following instructions:
MOV AX, 2437H and MOV AX, [2437H]

Ans: MOV AX, 2437H means -
The AX register will contain the value of 2437H as its content.
And MOV AX, [2437H] means -
AX will contain the contents ~~of DS:2437H.~~ that is located in DS:2437H.

* What are the advantages of 80286 microprocessor over 8086 microprocessor?

Ans:

(1) 80286 offers more than twice as much performance per clock cycle as the 8086.

(2) 80286 has twice as many functional units as 8086.

(3) 80286 also operates in protected virtual address mode which enables multitasking and memory protection.

(4) 80286 establishes more addressabl memory.

* Compare RISC processor with CISC processor?

Ans:

RISC	CISC
(1) Single-cycle instructions	(1) Multiple-cycle instructions
(2) Highly pipelined	(2) Less pipelined
(3) Few addressing modes	(3) Many addressing modes
(4) Heavy use of RAM	(4) Efficient use of RAM than RISC
(5) Simple, standardized instructions	(5) Complex and variable length instructions
(6) Uses more registers	(6) Uses less registers
(7) RISC architecture gives more importance to software.	(7) CISC architecture gives more impotance to hardware.

* Write two differences between BIOS routine and DOS routine?

Ans:

DOS routine	BIOS routine
(1) DOS routines operate over entire PC family.	(1) BIOS routines are machine specific.
(2) DOS routine is associated with INT 21H	(2) BIOS routine is associated with INT 16H

* Disadvantage of DMA:

DMA operation is carried out by DMA controller so the extension of resources make DMA operation more expensive.

* Write an assembly program to perform the following:
Put the sum 1+3+5+7+9....+25 in Bx

Sol:

```
.STACK
.MODEL SMALL
.STACK 100H
.DATA
.CODE
XOR BX,BX
MOV AX,@DATA
MOV DS,AX
XOR BX,BX
MOV AX,1

TESTVAL: CMP AX,25
         JG ENDPRO
         JLE ADDVAL

ADDVAL: ADD BX,AX
        #ADD AX,2
        JMP TESTVAL

ENDPRO: MOV AH,4CH
        INT 21H
```

* What happens ~~it~~ after executing a CALL and RET instruction?

Ans:

The execution of RET instruction causes the stack to be popped into IP and control returns to the calling program.

The call instruction invokes a procedure. For a ~~near~~ NEAR procedure, execution of CALL causes the offset address of the near instruction in line after the CALL to be saved on the stack and the IP gets the offset of the first instruction in the procedure.

* Have any problem in the following code If so, explain it and solve this problem.

Ans
```
MOV CX,0
MOV AH,2
MOV DL,'*'
TOP: INT 21H
LOOP TOP
```

Ans: The problem with the given code is that the initial value of CX is 0.

CX is used as a loop counter. For every cycle of a loop operation, the value of CX decreases by 1 and the loop operation goes on until CX is 0. But here the initial value of CX is 0 and so the loop operation will not start at all.

Place a value greater than 0 in CX depending on the number of loop cycles you want.

* What are the parts of a machine instruction? Write down the steps that CPU goes through to execute a machine instruction?

Ans:

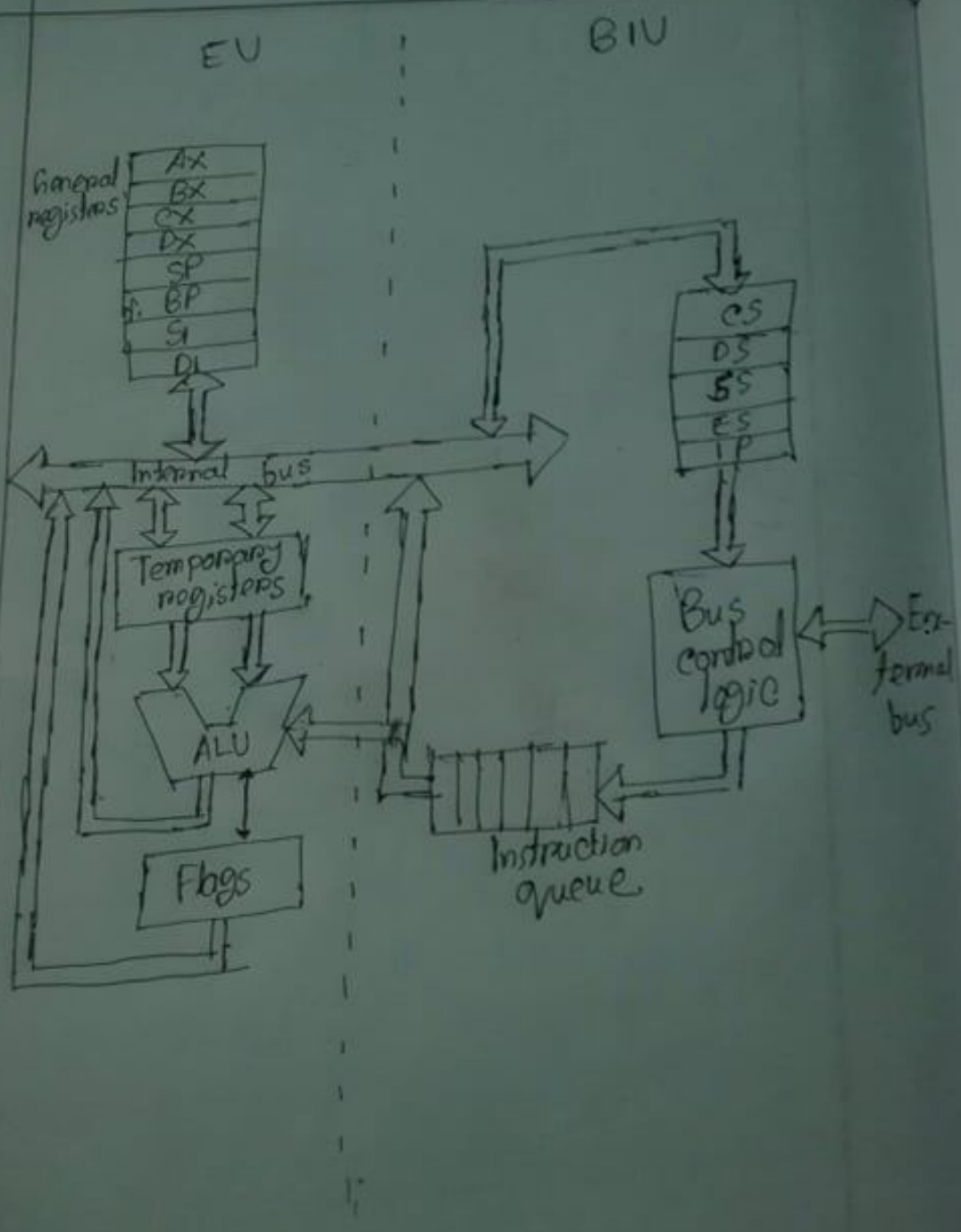
A machine instruction has two parts: an opcode and operands. The opcode specifies the type of an information and the operation. operands are often given as memory addresses of data to be operated on.

The CPU goes through the following steps to execute a machine instruction.

(a) Fetch: (1) Fetch an instruction from memory
(2) Decode the instruction to determine the operation
(3) Fetch data from memory if necessary

(b) Execute: (4) Perform the operation on the data.
(5) Store the result in memory if needed.

* Internal architecture of 8086



* What is the function of instruction queue?

Ans: Instruction queue is used to prefetch the next instructions in a separate buffer while the processor is executing the current instruction.

* What is the function of instruction pointer (IP)?

Ans: The instruction pointer contains the offset address of the next instruction to be executed by the execution unit.

These upcoming executable instructions are prefetched by the ~~execu~~ instruction queue.

80 → 10000000 → 128
B → 00001011 → 11
01111010 → 75

* A memory location has physically address 80FD2H. In what segment does it have offset BFD2H?

Ans:

We know that,

Physical address = 10× segment address + offset address

⇒ 10× segment address

= Physical address − offset address

= 80FD2H − BFD2H

= 75000h

⇒ segment address = 7500h

* Features of 8086 microprocessor:

(1) 16 bit data bus
(2) 16 bit ALU
(3) 2 functional units
(4) 1 MB RAM
(5) 20 bit address bus
(6) 40 pin dual in line package
(7) Designed to operate in 2 modes. (a) Maximum mode
(b) Minimum mode
(8) 14 ~~16bit~~ 16-bit registers
(9) Can support up to 64K I/O ports

* What is the function of a co-processor?

Ans: A co-processor is used to supplement the functions of a primary processor.

Microprocessors require the help of co-processors in floating point arithmetic.

* What is macro?

Ans: A macro is a set of instructions grouped under a single unit.

It is another method for implementing modular programming in the 8086 microprocessors.

In other words, a macro is a programmable pattern which translates a certain sequence of input into a preset sequence of output.

* What is PSP?

Ans: When a program is loaded in memory, DOS prefaces it with a 256 byte program segment prefix. The PSP contains information about the program.

* What do you mean by the following?

(i)
```
MOV AX, @DATA
MOV DS, AX
MOV ES, AX
```

Ans: Here, @DATA is the name of the data segment defined by .DATA. The assembler translates the name @DATA into a segment number. DOS places its segment number in both DS and ES before executing the program. The result is that DS does not contain the segment number of the data segment. The given lines correct this issue.

* What is the meaning of the following?

```
LEA DX, MSG
MOV AH, 9
INT 21H
```

Ans: (1) LEA DX, MSG: puts of a copy of the offset address of MSG in DX, MSG remains unchanged.

(2) MOV AH, 9: enables a request for function 9 of the INT 21H dos routine to be executed.

(3) INT 21H: Executes the 9th function of the DOS routine.

* translate the following high level language assignment statements to assembly language?

(i) A = B - 2C - A + 1

Ans:

```
MOV AX, 2
IMUL C
SUB B, AX
MOV
SUB
```

```
MOV SUB
MOV AX, A
SUB B, AX
INC B
MOV AX, B
MOV A, AX
```

* Basic differences between 8086 and 8088 microprocessor:

8086	8088
(1) 8086 has a 16-bit data bus	(1) 8088 has a 8-bit data bus
(2) 8086 has a faster clock rate	(2) 8088 has a slow clock rate

* Similarity between 8086 and 8088:
Both have the same instruction set

*

80386	80386SX
32 bit data bus	16 bit data bus

* Uses of I/O ports?

Ans: I/O devices are connected to the computer through I/O circuits. Each of these circuits contain several registers called I/O ports.

I/O ports are used for data and control commands. I/O ports function as transfer points between the CPU and I/O devices.

* What do you mean by a 20bit address bus?

Ans: A 20 bit address bus can enable the accessing of upto 2^{20} memory locations. 2^{20} bytes = 1MB.

* Write an assembly program to display a 5x5 grid of "*"

Sol:

```
.MODEL SMALL
.STACK 100H
.DATA
VAR1    DW  0AH,0DH,'*****$'
.CODE
MAIN PROC
MOV AX,@DATA
MOV DS,AX
LE MOV AH,9
  LEA DX,VAR1
  INT 21H
  INT 21H
  INT 21H
  INT 21H
  INT 21H
MOV AH,4CH
INT 21H
MAIN ENDP
END MAIN
```

* Write code to reverse the bit pattern in AL. Put the reversed bit pattern back to AL.

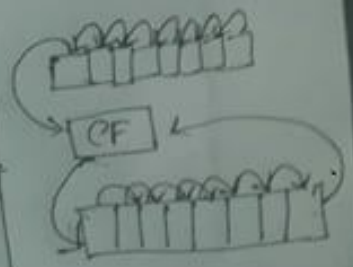
Ans:

```
.MODEL SMALL
.STACK 100H
.DATA
.CODE
MAIN PROC
MOV AX,@DATA
MOV DS,AX

MOV CX,8

REVERSE:
    SHL AL,1
    RCR BL,1
    LOOP REVERSE
    MOV AL,BL
```

the first shifted bit of AL becomes the rightmost bit of BL



Explanation:

The bit shifted into CF is the one that BL receives from RCR operation. The bits shifted in from RCR operation shifts rightwards but bits shifted from right to left.

* What do you mean by (i). CRF file (ii). EXE file (iii). LST file (iv) Object file

Ans: (ii). EXE file: .EXE file is an executable file which is a main entry point for execution of a program.

(i). CRF file: .CRF or cross reference file is a listing of names that appear in the program and the line numbers on which they occur. It is useful for locating variables and labels in a large program.

(iv) Object file: The machine language file created by the assembler from the source program file.

(iii). LST file: The source listing file is a line-numbered text file that displays assembly language code and the corresponding machine code side by side and gives other information about the program.

*What will be the output after executing the following code:

```
XOR AX,AX
MOV CX,16
TOP: ROL DX,1
JNC NEXT
INC AX
NEXT: LOOP TOP
```

Ans: The AX will contain the number of 1 bits present in DX.

After ROL executes, if rotated bit is 1, then CF becomes 1 and JNC is skipped and AX is incremented.

If rotated bit is 0, the loop gets to the next cycle.

And by the end of 16 loop cycles, we get the number of 1 bits of DX stored in AX.

* What are the advantages of using a stack?

Ans:

(1) Saves the contents of registers for the calling program while a procedure executes

(2) Stack holds data or addresses that will be acted upon by a procedure.

(3) Section of memory is set aside for storing return addresses

* Write the differences between 16bit microprocessor and 32 bit microprocessor?

16bit microprocessor	32 bit microprocessor
(1) Internal bus is 16bits	(1) Internal bus is 32 bits
(2) ALU performs operation on 16 bits	(2) ALU performs operation on 32 bits

* Write a macro to place the largest of two ords in AX?

Ans:

```
GET_BIG  MACRO   WORD1, WORD2
         LOCAL   EXIT
         MOV     AX, WORD1
         CMP     AX, WORD2
         JG      EXIT
         MOV     AX, WORD2
EXIT:    ENDM
```

* Write macro to put the largest of three macros in AX?

Ans:

```
GET_BIG     FIRST, SECOND
        MOV     AX, FIRST
        CMP     AX, SECOND
        JG      FUNC1
        MOV     AX, SECOND

FUNC1:
        GET_BIG     SECOND, THIRD
        MOV     AX, SECOND
        CMP     AX, THIRD
        JG      FUNC2
        MOV     AX, THIRD

FUNC2:
        ENDM
```

* Maximum mode pin functions of 8086:

(a) READY: When it is high it indicates that the device is ready to transfer data. When it is low, it indicates wait state.

(b) RESET: RESET pin requires the microprocessor to terminate its present activity immediately.

(c) BHE: BHE is used in selection of the proper byte or bytes of memory or I/O word to be read and write.

(d) S_3 and S_4: S_3 and S_4 form together a 2 bit binary code that identifies which of the 8086 segment registers are used to generate the physical address that was output on the address bus during the current bus cycle.

S_4	S_3	Segment register
0	0	Extra
0	1	Stack
1	0	Code/none
1	1	Data

(e) QS_1 and QS_0: These are queue status signals. They provide the status of the instruction queue.

QS_0	QS_1	Status
0	0	No operation
0	1	First byte of opcode from the queue
1	0	Empty the queue
1	1	Subsequent byte from the queue

(f) LOCK: When LOCK signal is active, it indicates the processors not to ask the CPU to leave the system bus

(3) S_0, S_1, S_2 :

These are status signals that provide the status of the operation, which is used by the Bus Controller 8288 to generate memory and I/O control signals.

S_2	S_1	S_0	status
0	0	0	Interrupt acknowledgement
0	0	1	I/O Read
0	1	0	I/O Write
0	1	1	Halt
1	0	0	Opcode fetch
1	0	1	Memory read
1	1	0	Memory write
1	1	1	Passive

* Describe the minimum mode pin functions of 8086?

Ans:

(1) ~~HOLDA~~ HLDA: It stands for Hold Acknowledgement signal. This signal acknowledges the HOLD signal.

(2) HOLD: This signal indicates to the processor that external devices are requesting to access the address/data buses.

(3) ALE: ALE stands for "address enable latch". This signal indicates the availability of a valid address on the address/data lines.

(4) M/$\overline{IO}$: It stands for minimum/maximum. It indicates what mode the processor is to operate in. When it is high, it works in minimum mode and vice versa.

(5) M/IO: This signal is used to distinguish between memory and I/O operations. When it is high, it indicates I/O operation otherwise memory operation.