Name :- Krunal Rank

Roll No :- U18CO081

Class :- B Tech 3rd Year,

Computer Engineering.

U18CO081

MIT
Tutorial 8

Ans 1/2 The flag register is one of the special purpose registers.
8086 has a 16 bit flag register and there are 9 valid flags.

D_{15}	D_{14}	D_{13}	D_{12}	D_{11}	D_{10}	D_9	D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
				O	D	I	T	S	Z		AC		P		CY

Status flags:-
S :- At any operation if the MSB is 1, then it indicates number is negative and it is set.
CY:- set if arithmetic operation results into carry.
Ac: Sets if arithmetic operation results into carry from lower half to upper half.
P:- Sets if parity of Accumulator value is odd.
O :- Overflow flag is set to 1 if overflow occurs.
Z :- Sets if arithmetic operation results into 0.

Control flag :-
D:- Directional flag is set to 1 if a string needs to be accessed from higher memory to lower, 0, otherwise.
I :- If I=1, microprocessor will recognise interrupts.
T:- Trap flag is used for on chip debugging.

Ans 2:- The physical address of memory location is located with the help of Base ~~Register~~ Address and an offset.

In order to access memory location, we can't pass a 20 bit address directly due to address line limitation. The 16 bit address with respect to the segment is called an offset.

Hence, using the base address, we calculate the physical address of the memory.
This Base Address for Data segment is stored in DS and Code segment is stored in CS.

Hence, Physical Address = Base Address + Offset.

Ans 3:-

CMP :- CMP [Reg/Memory] [Reg./Memory/Immediate]
Operand1 - Operand2

CMP AL, BL → AL - BL → Sets zero if equal
→ No carry if AL > BL
→ Carry if AL < BL

LAHF :- Loads AH from 8 low bits of flag register.
AH = Flag register.

XCHG :- Exchanges values of two operands.
XCHG AX, BX
BX ↔ AX

LEA:- Load Effective Address
REG = Address of memory.

```
Mov BX, 35H
Mov DI, 12H
Mov SI, [BX+DI]  ; SI = 47H
```

PUSH:- Stores 16 bit value in the stack.

```
SP = SP-2
SS:[SP] = Operand
Mov AX, 1234H
PUSH DX
POP DX      ; DX = 1234H
RET
```

LDS:- Load memory double word into word register and DS.
REG = first word
DS = Second word.

```
LDS AX, m
RET
m DW 1234H
  DW 5678H.
; AX = 1234H  DS = 5678H.
```

Ans 4: An assembler directive is a statement to give direction to the assembler to perform task of the assembly process.

ASSUME :- Tells the assembler what names have been chosen for Code, Data Extra and Stack segments.

DW :- Used to assign 16 bit data word to a variable.
WORD DW 2346H

DQ : Define 4 word length word to a variable.
BIG DQ 2435948740192A92BH

PROC :- Used to identify the start of the procedure. The term FAR or NEAR is used to define the type of procedure.
NEAR : Resides in same code segment
FAR : Resides in different code segment.

ENDP :- Used to end a procedure.

Ans 5:
1) MOV CL,34H → Immediate Addressing
2) MOV BX,[4172H] → Direct Addressing
3) MOV DS, AS → Register Addressing
4) MOV AX, [SI+BX+04] → Base Index with Displacement addre

Ans 6:- Given,

AX = 85H

BX = 64H

SP = 2000H

i) PUSH AX

SP = 1FFDH

AX = 85H

BX = 64H

ii) Pop BX.

SP = 2000H

AX = 85H

BX = 85H

Ans 7:- AX = ACH = 1010 1100

CY = 1.

i) ROL AX, 2 ; AX = 1011 0010

ii) RCL AX, 2 ; AX = 1011 0010 H.

Ans 8:-

DIV :- Unsigned division

When operand is a byte:-

AL = AX / operand

AH = Remainder

When operand is a word:-

AX = (DX AX) / operand

DX = Remainder

IDIV :- Signed Division

When operand is a byte :-

AL = AX / Operand

AH = Remainder (modulus)

When operand is a word :-

AX = (DXAX) / Operand

DX = Remainder (modulus)