Name: Ayush Jain
SAP ID: 60004200132
Div: B2
Computer Engineering

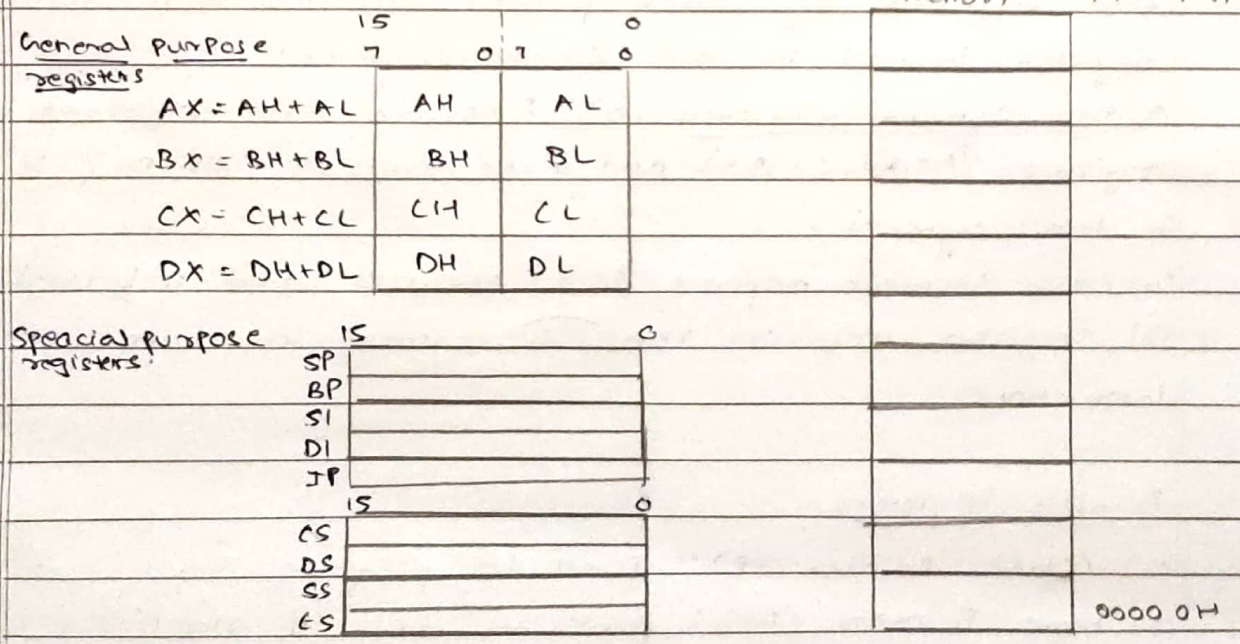
POA - Assignment 2

Q. 1) 8086 programmer's model:

→ Programming model for a microprocessor shows the various internal registers that are accessible to the programmer. The following figure is a model for 8086. In general each register has speacial function.

In a programming model there are:
4 general purpose registers
4 segment registers
2 Pointer registers
2 Index registers
1 instruction pointer register
1 flag registers.

Programmers model of 8086



Flag registers

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	OF	DF	IF	TF	SF	ZF	-	AF	-	PF	-	CF

General Purpose Registers :

(1) AX : Accumulator performs arithmetic, logical and data transfer instruction.

(2) BX : Base register pointer used to point based, based indexed or register indirect addressing.

(3) CX : Counter register uses as a long counter also as a counter in a string manipulation.

(4) DX : Data register, used as a port number in I/O operation, also in multiplication and division.

Segment Register :

(1) Code segment register (CS) : Used to access to instructions referred by instruction pointer (IP) register.

(2) Stack segment register (SS) : The processor assume all the data reffered by stack pointer (SP) and base pointer (BP) register located in the stack segment.

(3) Data segment register (DS) : Assume data referred by general register (AX, BX, CX, DX) and index register (SI, DI) is located in data segment.

(4) Extra Segment register (ES) : Assumes data referred by DI register referres the ES segment in string manipulation instructions.

Pointer Registers :

(1) Stack Pointer (SP) : Points to program stack.

(2) Base Pointer (BP) : points to data in stack.

Index Registers:

(1) Source Index (SI): used to point memory locations in the data segment address by DS.

(2) Destination Index (DI): access the memory locations address by ES with string operations.

Instruction Pointer (IP)

Address of the next instructions to be executed in pointer. IP and CS are used to compute the memory address of the instructions code to be fetched.

Flag Register:

Flag register indicates the status of a processor. 9 are active out of 16.

Q. 2) 8086 addressing modes with examples:

→ Addressing modes are different ways by which CPU can access data on operands. They determine how to access a specific memory address.

Types of addressing modes:

1) Register Addressing mode:

This involves the use of registers.

eg: MOV AX, CX

(2) Immediate addressing mode :

2 operands, 1 is register and other is a constant value.

eg: mov AX, B0 H

(3) Direct Addressing modes :

It loads or stores data from memory to register and vice-versa.

```
eg: mov AX, 2162H
    mov DS, AX
    mov CX, 2H
    mov [481], CX
    mov BX, [48]
```

(4) Register Indirect Addressing :

It uses the offset address which resides in one of these registers ie. BX, SI, DI

```
eg: mov AX, 0705h
    mov DS, AX
    mov CX, 42A h
    mov, SI, CX
    mov [SI], CX
```

(5) Based Relative Addressing Mode.

This uses base register either BX or BP

```
eg: mov AX, 0708 h
    mov CX, 0154 h
```

(6) Index Relative Addressing Mode:

This uses index register, ie. DI and SI

eg: mov Ax, 6708h

mov [DI+5], DX

(7) Based Index Addressing Modes.

Continues the use of based and index addressing.

eg: mov AX, 0708h

mov DX, 0154h

mov DI, 213h

mov [BP+DI+20], DX

Q. 3) Difference between macros and Procedure.

Macro	Procedure
1) It contains set of instructions to support modular programming	1) It contains set of instructions which can be called seperatively.
2) It is used for small set (<10)	2) It is used for large set (>10)
3) Memory required is high.	3) Memory required is less.
4) CALL/RET are not required	4) CALL/RET are required.
5) Faster than procedure.	5) Slower than procedure.
6) Passed as a part of statement when call micro.	6) Generated only once when the procedure is defined.
7) No overhead times.	7) Overhead times take place.
8) In a macro parameters is passed as part of statement that calls micro.	8) In a procedure parameters are passed in registers and memory locations of stack.

Q.4) Example of programms in 8086.

→ Factorial using Macro:

```
fact macro f
  up:
   mul f
   dec f
   jnz up
end M

data segment
    num dw 05h
    result dw ?
ends

Stack segment
    dw 08 dup(0)
ends

Code segment :
  Start:
   mov AX, data
   MOV DX, AX
   mov CX, Num
   mov AX, 0001h
   fact num
   mov result, AX
ends
```