Course Code : CSE-331L / EEE-332L / ETE-332L
Course Name : Microprocessor Interfacing & Embedded System
Instructor : Azmary Zannat Aurin
Semester : Fall 2022

Marks Distribution:

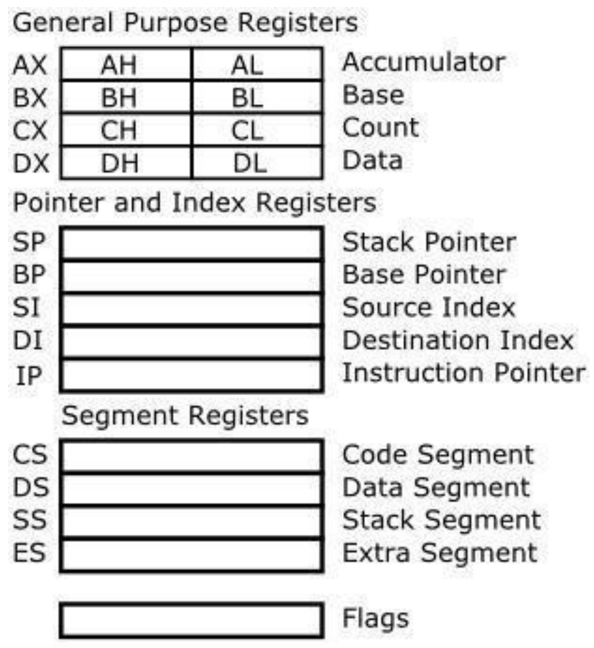
Class performance and attendance	10%
Quiz (best 2 out of 3)	10%
Assignment	10%
Mid	20%
Final	30%
Project	20%

Introduction:
In this session, you will be introduced to assembly language programming and to the emu8086 emulator software. emu8086 will be used as both an editor and as an assembler for all your assembly language programming.

Microcontroller vs. Microprocessor:
A microprocessor is a CPU on a single chip. If a microprocessor, its associated support circuitry, memory, and peripheral I/O components are implemented on a single chip, it is a microcontroller. Cost and space of microprocessor is higher than microcontroller.
We will be learning more about Microprocessor 8086 in this course.

Registers of MPU 8086:

- ➢ Total number of registers: 14
- ➢ Each register size: 16 bits



General Purpose Registers (4 registers):

- ➢ Each GPR has two separate parts: Higher order byte and Lower order byte (each with 8 bits size). Data on each part can be separately manipulated
- ➢ Can perform 16 bits and 8 bits data read/write operations

	AH	AL
AX: 0011 0000 0011 1001 b	0011 0000 b	0011 1001 b
AX: 1111 0100 1010 0001 b	1111 0100 b	1010 0001 b
AX: F4A1 h	F4 h	A1 h
AX: 4 h	?	?

AX (Accumulator Register): Used in arithmetic, logic and data transfer operations

BX (Base Register): used as an address register

CX (Count Register): used for program loop count

DX (Data Register): used in arithmetic and I/O operations

Segment Registers (4 registers):

Program code, data and stack are loaded into different memory segments.

- ➢ Stack segment: used for temporary storage of addresses and data

- ➢ Code segment: program instructions are loaded in this segment.
- ➢ Data segment: variables are declared in this segment
- ➢ Extra segment: another data segment in the memory

Pointer and Index Registers (5 registers):

- ➢ Points to memory locations
- ➢ Unlike segment registers, they can be used for general arithmetic operations
- ➢ IP register: contains the offset of the next instruction in the code segment

Flag Register (1 register):

- ➢ Indicates the status of the microprocessor

What is Assembly Language:

- ➢ A low-level programming language
- ➢ Converted to machine code using an **assembler**
- ➢ Important to low-level embedded system designs
- ➢ Designed for specific processor

Steps required to run an assembly program:

- ➢ Write the necessary assembly source code
- ➢ Save the assembly source code
- ➢ Compile/Assemble source code to create machine code
- ➢ Emulate/Run the machine code

Structure of Assembly Language Programming for MPU 8086:

```
Label: OperationToPerform operand1 operand2 ;
Label: OperationToPerform Destination Source ;
```

Label: - symbolic name for memory location

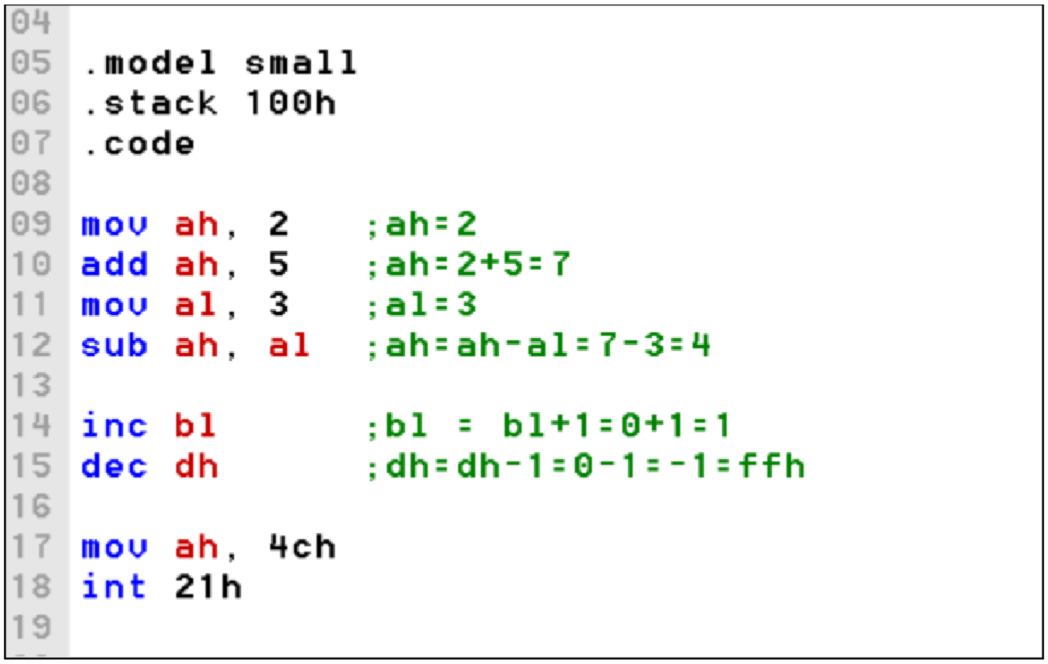
OperationToPerform - instruction name

; - comments

Operands: REG, MEMORY, Immediate

- ➢ **REG:** Any valid register
- ➢ **Memory:** Referring to a memory location in RAM
- ➢ **Immediate:** Using direct values (can never be a destination)

Instruction	Algorithm (= is assignment)
MOV	MOV Destination, Source **Algorithm:** destination = source
ADD	ADD Destination, Source **Algorithm:** destination = destination + source
SUB	SUB Destination, Source **Algorithm:** destination = destination - source
INC	INC Destination **Algorithm:** destination = destination + 1
DEC	DEC Destination **Algorithm:** destination = destination - 1
** source remains unchanged	



```
.model small
.stack 100h
.code

mov ah, 2     ;ah=2
add ah, 5     ;ah=2+5=7
mov al, 3     ;al=3
sub ah, al    ;ah=ah-al=7-3=4

inc bl        ;bl = bl+1=0+1=1
dec dh        ;dh=dh-1=0-1=-1=ffh

mov ah, 4ch
int 21h

```