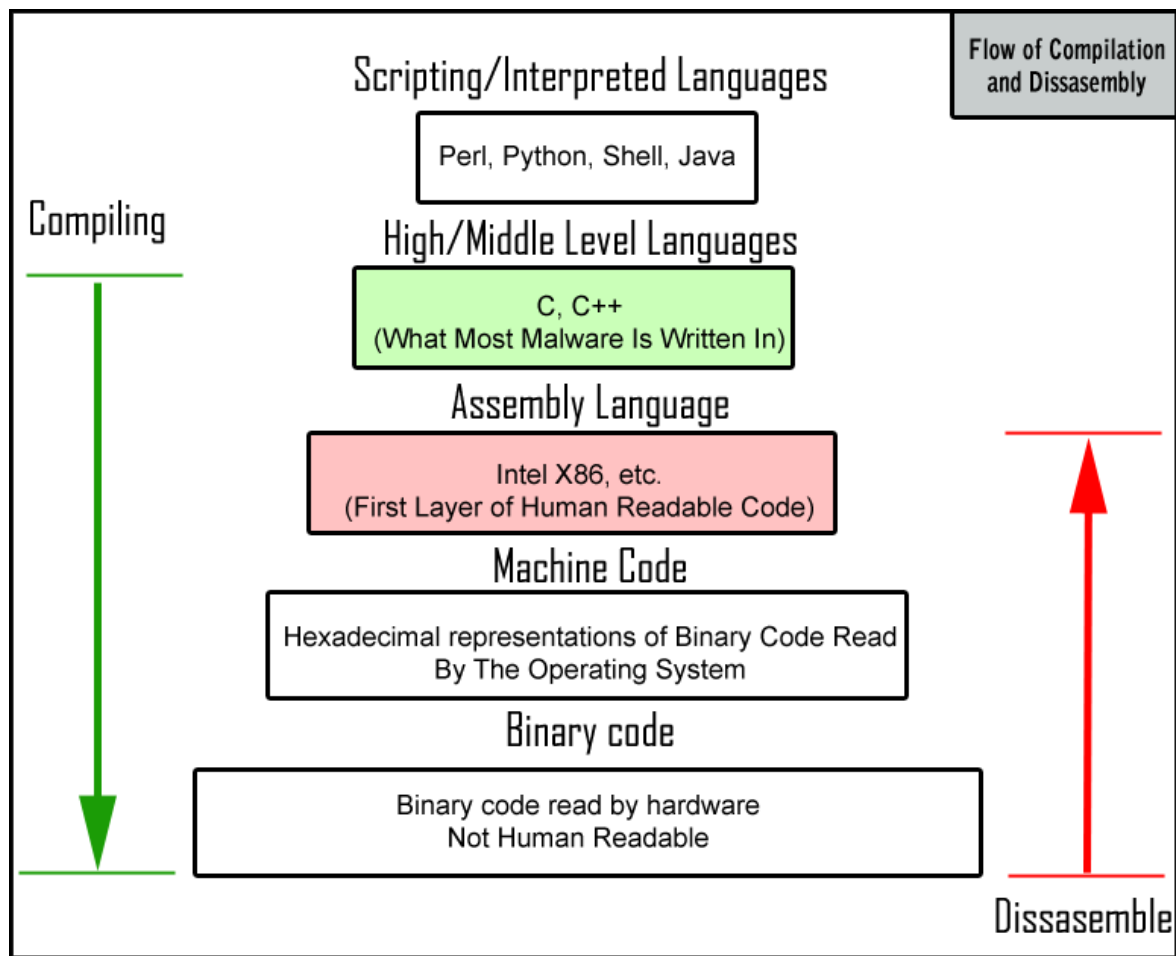


Welcome to the World of Assembly Language Programming

Prepared by
Madhusudan Basak
Lecturer
Department of CSE, BUET

The beginning !

- What is Assembly Language?



About Assembly

- Assembler is human-readable machine code
- gives you complete control over the system's resources

Why use it ?

- Direct hardware manipulation
- Access to specialized processor instructions
- Performance and efficiency
- Speed optimization
- Control the code behavior

Disadvantages ☹️

- Hard and tedious
- Bug-prone
- Non-portable(machine dependent)
- Difficult to debug

A question ?

- In what kind of situations, assembly language will be the only solution?
- an Operating System
- drivers and communication with custom hardware/electronics.
- an Compiler (assembler can be the output)
- optimizations

Definition from wiki

- An **assembly language** is a [low-level programming language](#) for a [computer](#), or other programmable device, in which there is a very strong (generally [one-to-one](#)) correspondence between the language and the [architecture's machine code instructions](#). Each assembly language is specific to a particular computer architecture, in contrast to most [high-level programming languages](#), which are generally [portable](#) across multiple architectures, but require [interpreting](#) or [compiling](#).

Chapter 1

Microcomputer Systems

The Components of a Microcomputer System

- Typical components
 - System unit
 - Keyboard
 - Display Screen
 - Disk drives
- Functionally three parts
 - CPU
 - Memory circuits
 - I/O circuits

Memory

- Bytes and Words
- Bit position
- Memory operations
- RAM and ROM
- Buses

CPU

- Two main components
 - Execution Unit(EU)
 - Bus Interface Unit(BIU)

I/O Ports

- Serial Ports
- Parallel Ports

Instruction Execution

- Fetch
- Execute

Instruction Execution

- The Process of executing an instruction

– 00000001 00000110 00000000 00000000



ADD
operation



AX
Register



Memory
Address
00000000



Memory
Address
00000000

- The process is not less than a movie !!!

I/O Devices

- Magnetic Disks
- Keyboard
- Display Monitor
- Printers

Programming Languages

- Machine Language
- Assembly Language
- High-Level Languages

Chapter 2

REPRESENTATION OF NUMBERS AND CHARACTERS

Number Systems

- Binary Number System
- Hexadecimal Number System
- Decimal Number System

Conversion Between Number Systems

- Converting Binary and Hex to Decimal
- Converting Decimal to Binary and Hex
- Conversions between Hex and Binary

Addition and Subtraction

- Hexadecimal Addition Table

How integers are represented in the Computer

- Unsigned integers
- Signed integers
 - One's complement
 - Two's complement
- Decimal Interpretation

Character Representation

- ASCII (American Standard Code for Information Interchange) code
- Normally 128 ASCII Codes
- 32-126 are printable
- 0 to 31 and 127 for control purposes
- For Extended set, there are 256 characters

Chapter 3

ORGANIZATION OF THE IBM PERSONAL COMPUTERS

The Intel 8086 Family of Microprocessors

- The 8086 and 8088 Microprocessors
- The 80186 and 80188 Microprocessors
- The 80286 Microprocessor
- The 80386 and 80386SX Microprocessors
- The 80486 and 80486SX Microprocessors

Organization of the 8086/8088 Microprocessors

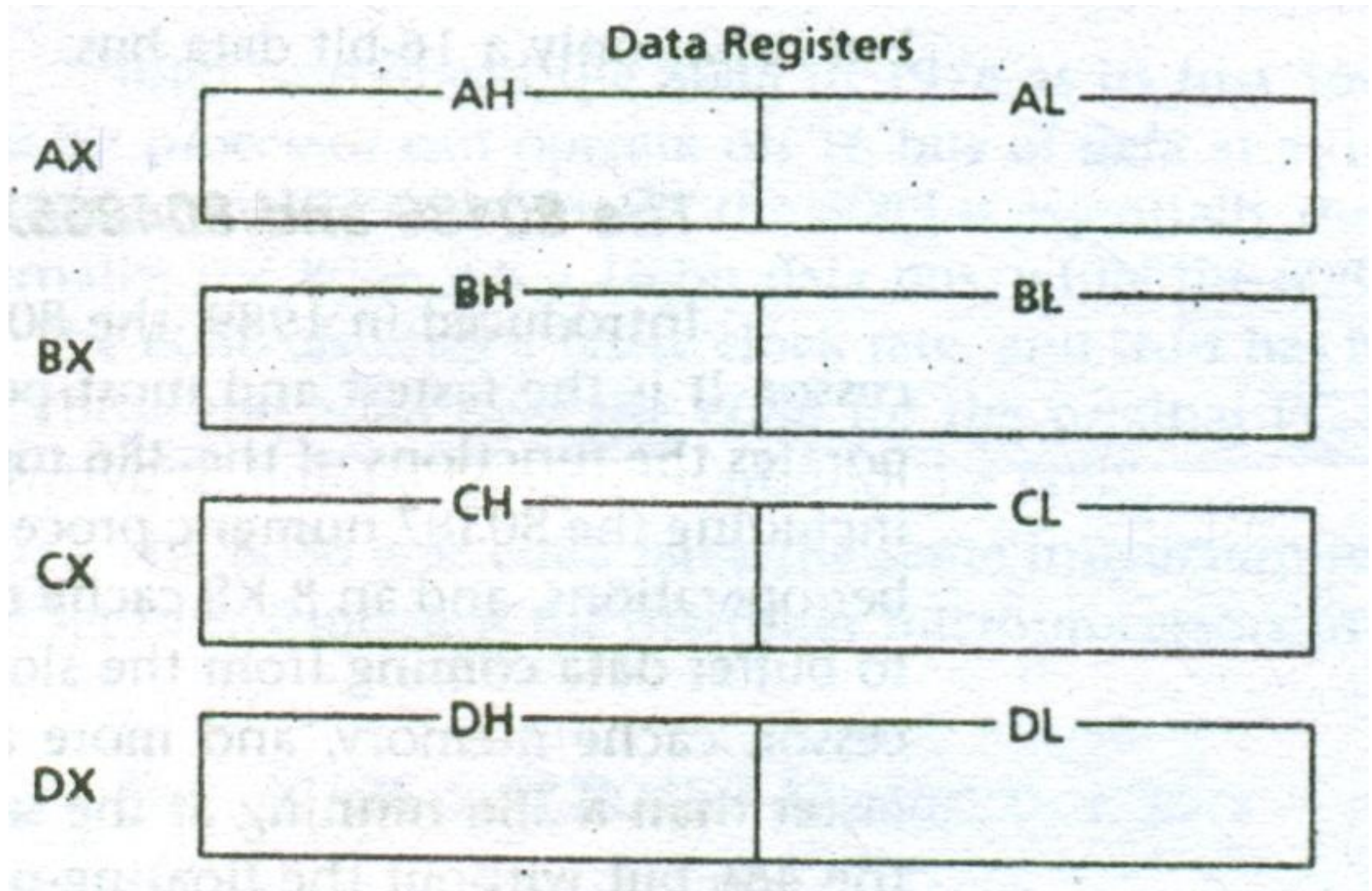
- Registers
 - 3 Types of registers
 - Data Registers
 - Address Registers
 - Status Register

Data Registers

- AX (Accumulator Register)
- BX (Base Register)
- CX (Count Register)
- DX (Data Register)

- Each one has two parts like AX(AH+AL)

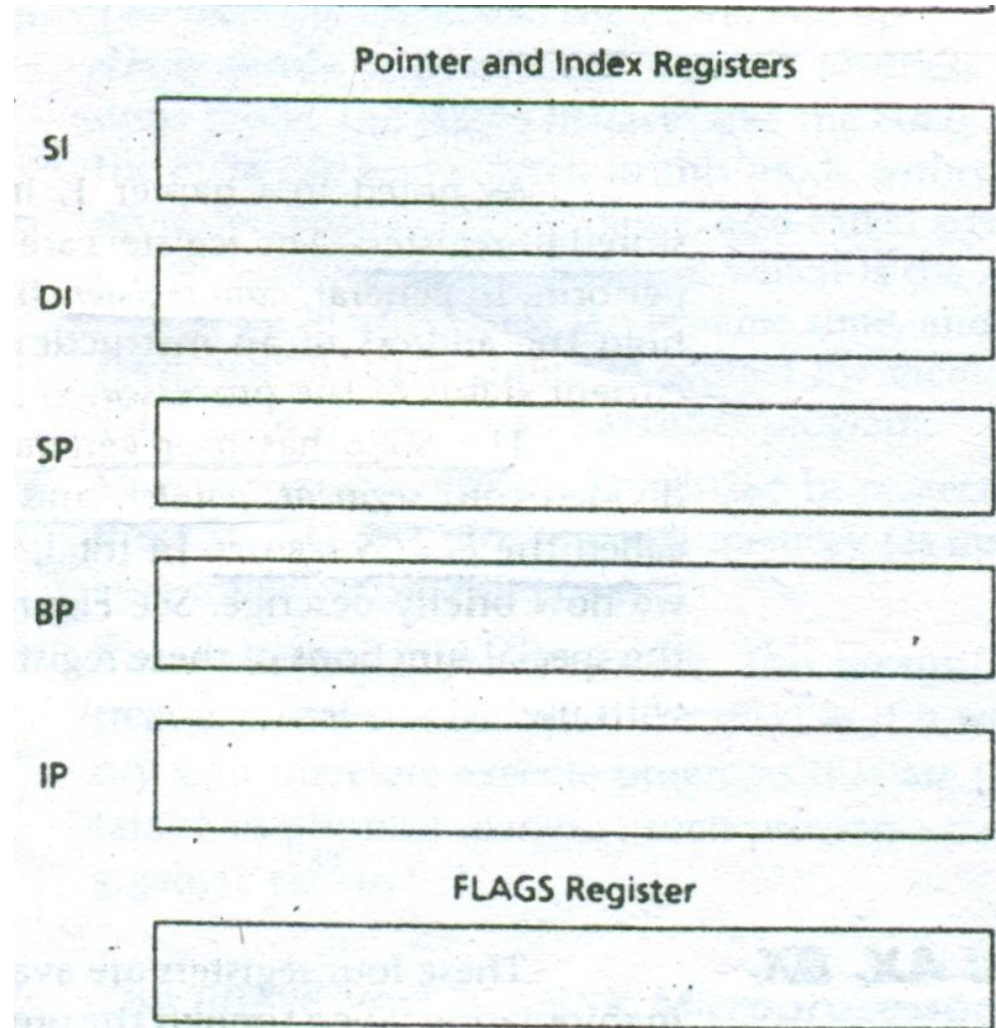
Data Registers



Segment Registers

	Segment Registers
CS	<input type="text"/>
DS	<input type="text"/>
SS	<input type="text"/>
ES	<input type="text"/>

Pointer and Index Registers, Instruction pointers and Flags Register



Segment Registers

- Memory Segment
- Segment: Offset Address
- Location of Segments

Task	Segment Registers	Pointer and Index Registers
Code	CS	IP
Data	DS	SI
	ES	EI
Stack	SS	SP
		BP

Overlapping Scenario

	10021	11010101
	10020	01001001
Segment 2 ends →	1001F	11110011
	1001E	10011100
	...	
	10010	01111001
Segment 1 ends →	1000F	11101011
	1000E	10011101
	...	
	10000	01010001
Segment 0 ends →	0FFFF	11111110
	0FFFE	10011111
	...	
	...	
	00021	01000000
Segment 2 begins →	00020	01101010
	0001F	10110101
	...	
	00011	01011001
Segment 1 begins →	00010	11111111
	0000F	10001110
	...	
	00003	10101011
	00002	00000010
	00001	10101010
Segment 0 begins →	00000	00111000

- If you have a segment A4FBh and offset 4872h. Then what is the 20 bit physical address?

A4FB0h

+4872h

A9822h (20 bit physical address)

Organization of the PC

- The Operating System
 - DOS (Disk Operating System)
 - BIOS (Basic Input/Output System)

Memory Organization of the PC

- 8086/8088 has only 1MB of memory
- Not all memory for application programs
- Interrupt Vector, Video Display Memory etc are needed
- IBM fixed all the positions and allowed all to live happily

Address		Segment
FFFFFh		F000h
F0000h		
FFFFFFh		E000h
E0000h		
20000h		1000h
OFFFh		
10000h		0000h
OFFFh		
00000h		

BIOS	Address
	F0000h
Reserved	
	E0000h
Reserved	
	D0000h
Reserved	
	C0000h
Video	
	B0000h
Video	
	A0000h
Application program area	
DOS	
BIOS and DOS data	
	00400h
Interrupt vectors	
	00000h

I/O ports addresses

- Interrupt controller (20h-21h)
- Keyboard controller (60h-63h)
- etc...

Start-up Operation

- Boot program