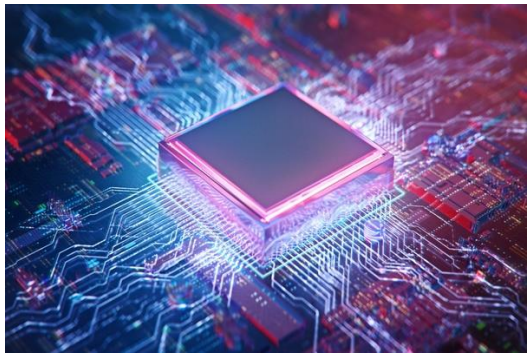


EL2003
Computer Organization and Assembly Language

Dr. Muhammad Usman Abbasi
Assistant Professor | Head of Electrical Engineering Department
National University of Computer and Emerging Sciences - FAST

1. Introduction to Assembly Language

- 1.1. Basic Computer Architecture
- 1.2. Registers
- 1.3. Instruction Groups
- 1.4. Intel iapx88 Architecture
- 1.5. History
- 1.6. Register Architecture
- 1.7. Our First Program
- 1.8. Segmented Memory Model

1.4. Intel iapx88 Architecture

- We will be using IBM PC based on Intel architecture
  - wide availability
  - free assemblers
  - free debuggers
  - wide use in a variety of domains

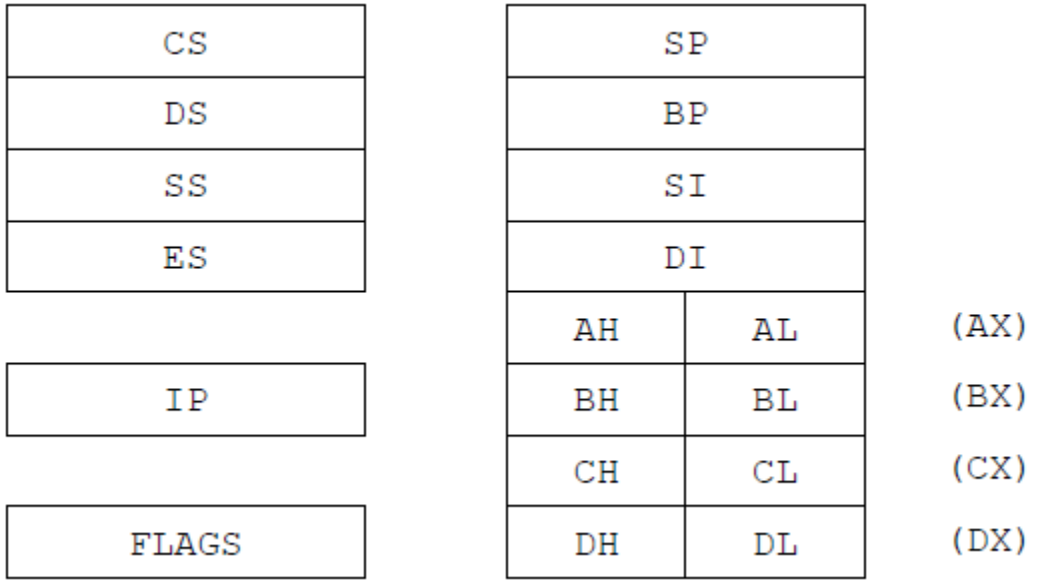
- It was a very successful processor also called 8088
- 8088 is a 16bit processor with its accumulator and all registers of 16 bits.
- iAPX386 which is very advanced and powerful processor (32-bit)

- iAPX88 stands for “Intel Advanced Processor Extensions 88

1.5. History

- Reading assignment

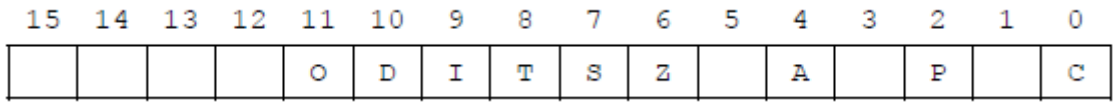
1.6. Register Architecture



1.6. Register Architecture

- **General Registers (AX, BX, CX, and DX)**
  - The registers AX, BX, CX, and DX behave as general-purpose registers in Intel architecture and do some specific functions in addition to it.
  - The A of AX stands for Accumulator.
  - The B of BX stands for Base.
  - The C of CX stands for Counter.
  - The D of DX stands for Destination (destination in I/O operations).
- **Index Registers (SI and DI)**
- **Instruction Pointer (IP)**
- **Stack Pointer (SP)**
- **Base Pointer (BP)**

- **Flags Register**



C	Carry	When two 16bit numbers are added the answer can be 17 bits long or when two 8bit numbers are added the answer can be 9 bits long. This extra bit that won't fit in the target register is placed in the carry flag where it can be used and tested.
P	Parity	Parity is the number of "one" bits in a binary number. Parity is either odd or even. This information is normally used in communications to verify the integrity of data sent from the sender to the receiver.
A	Auxiliary Carry	A number in base 16 is called a hex number and can be represented by 4 bits. The collection of 4 bits is called a nibble. During addition or subtraction if a carry goes from one nibble to the next this flag is set. Carry flag is for the carry from the whole addition while auxiliary carry is the carry from the first nibble to the second.
Z	Zero Flag	The Zero flag is set if the last mathematical or logical instruction has produced a zero in its destination.

S	Sign Flag	A signed number is represented in its two's complement form in the computer. The most significant bit (MSB) of a negative number in this representation is 1 and for a positive number it is zero. The sign bit of the last mathematical or logical operation's destination is copied into the sign flag.
T	Trap Flag	The trap flag has a special role in debugging which will be discussed later.
I	Interrupt Flag	It tells whether the processor can be interrupted from outside or not. Sometimes the programmer doesn't want a particular task to be interrupted so the Interrupt flag can be zeroed for this time. The programmer rather than the processor sets this flag since the programmer knows when interruption is okay and when it is not. Interruption can be disabled or enabled by making this bit zero or one, respectively, using special instructions.
D	Direction Flag	Specifically related to string instructions, this flag tells whether the current operation has to be done from bottom to top of the block (D=0) or from top to bottom of the block (D=1).
O	Overflow Flag	The overflow flag is set during signed arithmetic, e.g. addition or subtraction, when the sign of the destination changes unexpectedly. The actual process sets the overflow flag whenever the carry into the MSB is different from the carry out of the MSB

1.7. Our First Program

- **English Language Version**
- “Program is an ordered set of instructions for the processor.” Our first program will be instructions manipulating AX and BX in plain English.
  - move 5 to ax
  - move 10 to bx
  - add bx to ax
  - move 15 to bx
  - add bx to ax

- **Assembly Language Version**
  - operation destination, source
  - operation destination
  - operation source
  - operation

- **Assembler, Linker, and Debugger**
  - Netwide Assembler (NASM)
  - “A full-screen debugger” or AFD
  - nasm ex01.asm –o ex01.com –l ex01.lst

1.8. Segmented Memory Model

- Reading assignment