Microcomputer Architecture

A computer system has three main components: a Central Processing Unit (CPU) or processor, a Memory Unit and Input Output Units (devices). In any microcomputer system, the component which actually processes data is entirely contained on a single chip called Microprocessor (MPU). This MPU can be programmed using assembly language. Writing a program in assembly language requires a knowledge of the computer hardware (or Architecture) and the details of its instruction set.

The main internal hardware features of a computer are the processor, memory and registers (registers are special processor components for holding address and data).
The external hardware features are the computer Input/Output devices such as keyboard, monitor…

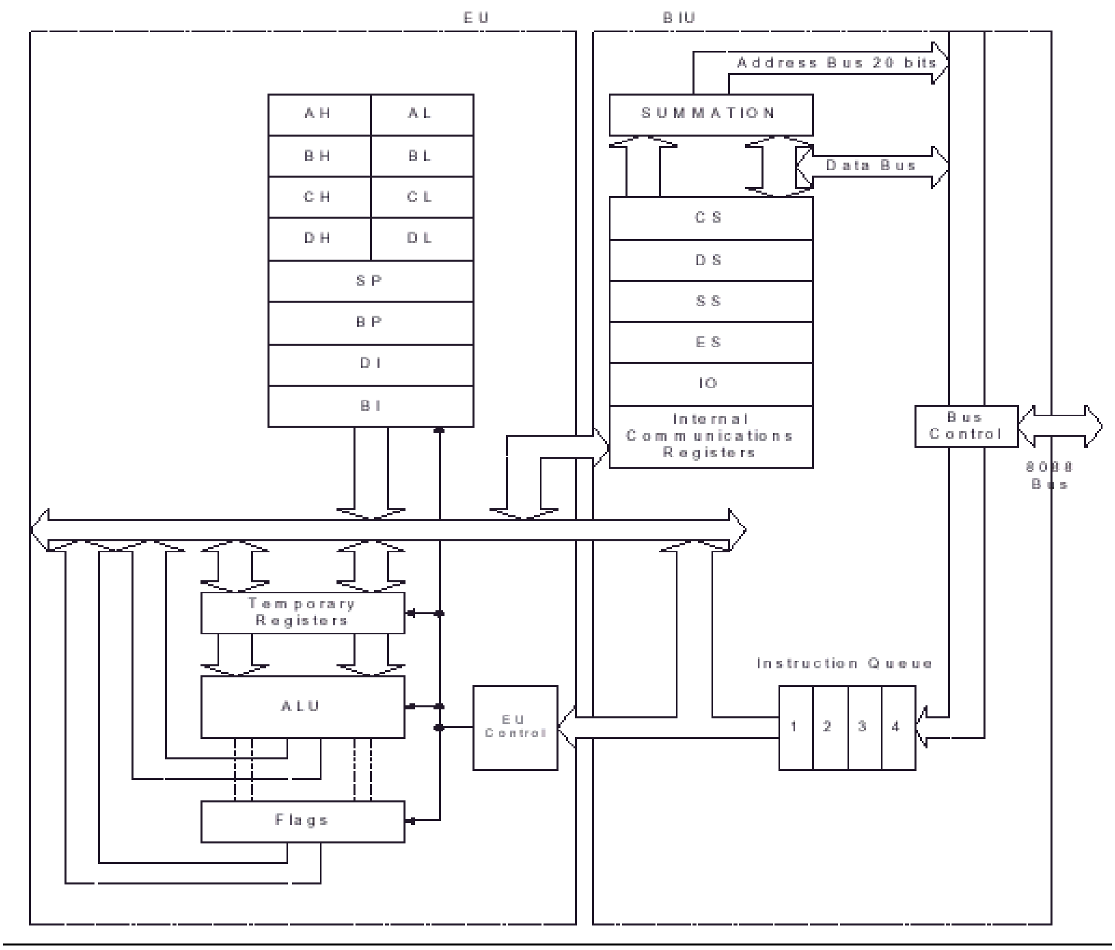
Software consists of the operating system (O.S) and various programs and data files stored on disk

A MICROPROCESSOR is a multipurpose programmable logic device that reads binary instructions from a storage device called memory accepts binary data as input and processes data according to those instructions and provides results as input.
Intel introduced its first 4-bit microprocessor 4004 in 1971 and 8-bit microprocessor 8008 in 1972. These microprocessors could not survive as general purpose microprocessors due to their design and performance limitations. Launching of the first general purpose 8-bit microprocessor 8080 in 1974 by Intel is considered to be the first major stepping stone towards the development of advanced microprocessors. The

microprocessor 8085 followed 8080, with a few more features added to its architecture, which resulted in a functionally complete microprocessor. The main limitations of the 8-bit microprocessors were their low speed of execution, low memory addressing capability, limited number of general purpose registers and a less powerful instruction set . All these limitations of the 8-bit microprocessors tempted the designers to go for more powerful processors in terms of advanced architecture, more processing capability, larger memory addressing capability and a more powerful instruction set. The 8086 was a result of such developmental design efforts.

In the family of 16-bit microprocessors, Intel's 8086 was the first one launched in 1978. The introduction of the 16-bit processor was a result of the increasing demand for more and more powerful and high speed computational resources. 8086 microprocessor has a much more powerful instruction set along with the architectural developments which imparted substantial programming flexibility and improvement in speed over the 8-bit microprocessors.

The architecture of 8086 provides a number of improvements over 8085 architecture. It supports a 16-bit ALU, a set of 16-bit registers and provides segmented memory addressing capability, a rich instruction set, powerful interrupt structure, fetched instruction queue for overlapped fetching and execution etc. The internal block diagram, shown in Figure (1), describes the overall organization of different units inside the chip.



Figure(1)

The complete architecture of 8086 can be divided into two parts (a) Bus Interface Unit (BIU) and (b) Execution Unit (EU). The bus interface unit contains the circuit for physical address calculations and a predecoding instruction byte queue (6 bytes long). The bus interface unit makes the system bus signals available for external interfacing of the devices. In other words, this unit is responsible for establishing communications with external devices and peripherals including memory via the bus. As already stated, the 8086 addresses a segmented memory. The complete physical address which is 20-bits long is generated using segment and offset registers, each 16-bits long.

Computer Compenants

1) The processor

The CPU or processor acts as the controller of all actions or services provided by the system. The operations of a CPU can be reduced to three basic steps: fetch, decode, and execute. Each step includes intermediate steps, some of which are:

1- Fetch the next instruction:

- Place it in a holding area called a queue.
- Decode the instruction.

2- Decode the instruction

- Perform address translation.
- Fetch operand from memory.

3- Execute the instruction.

- Perform the required calculation.
- Store results in memory or register.
- Set status flag attached to the CPU.

As there are a great many variations in architecture between the different kinds of CPU, we shall begin my looking at a simplified model of the structure. The model to be used can be seen on the right of this page, and is a good basis on which to build your knowledge of the workings of a microprocessor. The simplified model consists of five parts, which are:

a) Arithmetic & Logic Unit (ALU)

The part of the central processing unit that deals with operations such as addition, subtraction, and multiplication of integers and Boolean operations. It receives control signals from the control unit telling it to carry out these operations.

b) Control Unit (CU)

This controls the movement of instructions in and out of the processor, and also controls the operation of the ALU. It consists of a decoder, control logic circuits, and a clock to ensure everything happens at the correct time. It is also responsible for performing the instruction execution cycle..

c) Register Array

This is a small amount of internal memory that is used for the quick storage and retreival of data and instructions. All processors include some common registers used for specific functions, namely the program counter, instruction register, accumulator, memory address register and stack pointer.

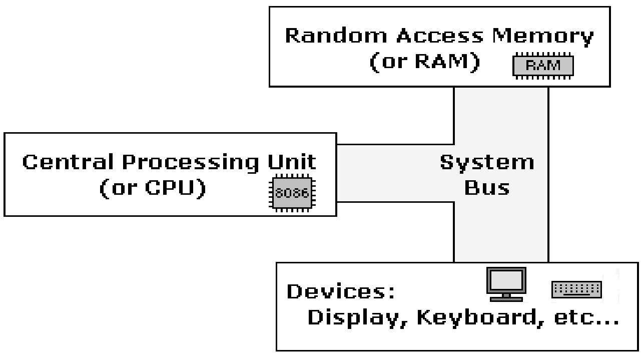


Figure -2-

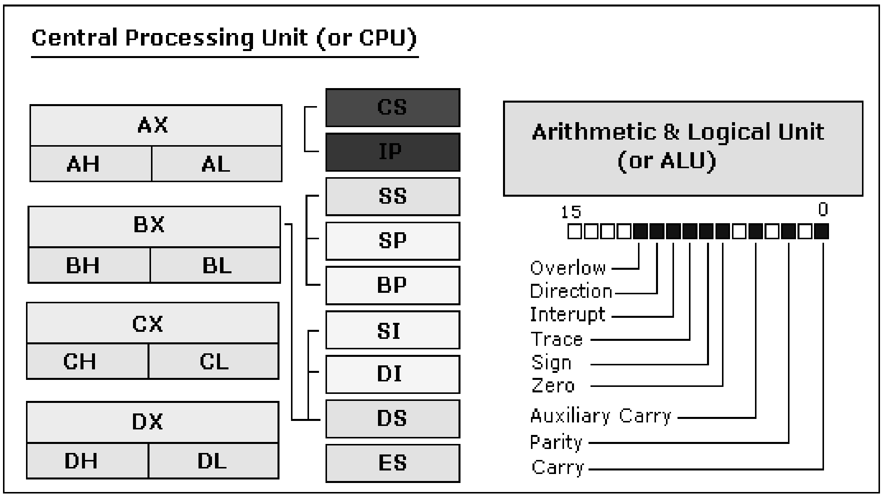
The **system bus** (shown figure 2) connects the various components of a computer. The **CPU** is the heart of the computer, most of computations occur inside the **CPU**.

RAM is a place to where the programs are loaded in order to be executed.

Inside the CPU

Registers

Registers are 8, 16, or 32-bit high speed storage locations directly inside the CPU, The CPU has an internal data bus that is generally twice as wide as its external data bus.



Registers 8086 block diagram

Figure – 3 -

General Registers

All general registers of the 8086 microprocessor can be used for arithmetic and logic operations. The general registers are:

1. **AX (Accumulator):** This is accumulator register. It gets used in arithmetic, logic and data transfer instructions. In manipulation and division, one of the numbers involved must be in AX or AL.
2. **BX (Base Register):** This is base register. BX register is an address register. It usually contain a data pointer used for based, based indexed or register indirect addressing.
3. **CX (Count register):** This is Count register. This serves as a loop counter. Program loop constructions are facilitated by it. Count register can also be used as a counter in string manipulation and shift/rotate instruction.
4. **DX (Data Register):** This is data register. Data register can be used as a port number in I/O operations. It is also used in multiplication and division.

Despite the name of a register, it's the programmer who determines the usage for each general purpose register. The main purpose of a register is to keep a number (variable). The size of the above registers is 16 bit, it's something like: **0011000000111001b** (in binary form), or **12345** in decimal (human) form.

4 general purpose registers (AX, BX, CX, DX) are made of two separate 8 bit registers, for example if AX= **0011000000111001b**, then AH=**00110000b** and AL=**00111001b**. therefore, when you modify any of the 8 bit registers 16 bit register is also updated, and vice-versa. the same is for other 3 registers, "H" is for high and "L" is for low part.

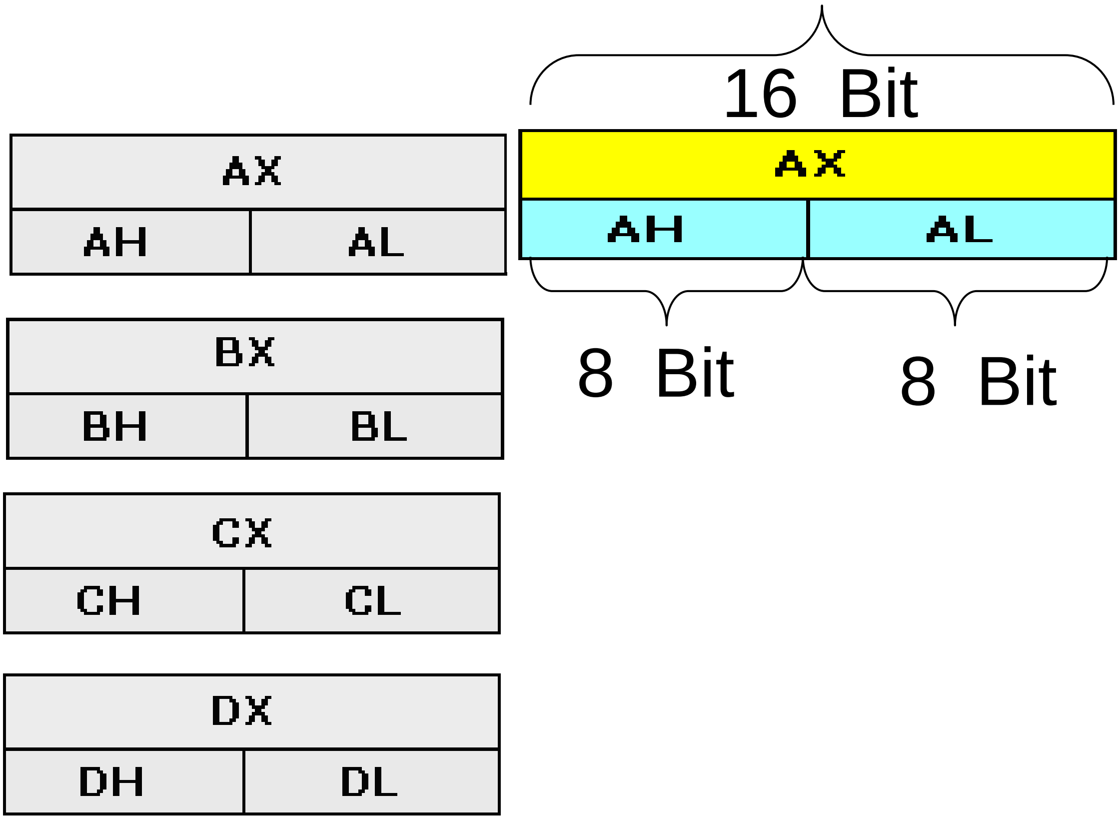


Figure -3-

Because registers are located inside the CPU, they are much faster than memory. Accessing a memory location requires the use of a system bus, so it takes much longer. Accessing data in a register usually takes no time. Therefore, you should try to keep variables in the registers. Register sets are very small and most registers have special purposes which limit their use as variables, but they are still an excellent place to store temporary data of calculations.

Assembly Language Instruction

- ## MOV instruction

MOV REG, memory

MOV memory, REG

MOV REG, REG

MOV memory, immediate

MOV REG, immediate

Where: REG: AX, BX, CX, DX, AH, AL, BL, BH, CH, CL, DH, DL, DI, SI, BP, SP.

H/w :

1. Load 12cdh into BX

2.copy the lower 8-bit from BX to AH

3. Load 10110111b to address memory contain in the BX reg.

4. Replace the data between BH and CL