UNIT - I

**Chapter - 1 Basic structure of computers**

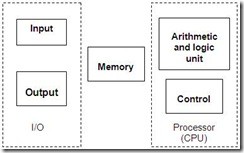
**Digital Computer/Computer** - A computer is a fast electronic calculating machine that accepts digitized input information, process it according to a list of internally stored instructions, and produces the resulting output information.

**1.1 Types of computers** - Different differ widely in size, cost, computational power, and intended use:

* Embedded computers: - are integrated into a larger device in order to automatically monitor and control a physical process. Ex:- vacuum cleaners, washing machines.
* Personal computers – desktop, notebook. A desktop computer is the most common form of a personal computer. Desktop computers have processing and storage units, visual display and audio output units, and I/O devices. Another example of a personal computer is a portable notebook computer.
* Workstations – They make use of high-resolution graphics input/output capability, more computational power and reliability. Workstations are used for engineering simulations, animations, rendering of images or any computing intensive jobs.
* Enterprise systems (mainframes) and servers – much more computing power and storage capacity, accessible via internet. Servers are used to maintain websites. They can hold databases or can host websites.
* Supercomputers – large-scale numerical calculations. Supercomputers are used to compute highly complex scientific/mathematical models.

# 1.2 Functional units

A computer consists of 5 functionally independent main parts:



* 1. Input
     + Computers accept coded information through i**/**p units which read the data.
     + Input Devices-Keyboard, Joysticks, trackballs and mouse.
  2. Memory
* Function is to store programs and data.
* 2 classes-primary storage, secondary storage.
* Primary storage-fast memory that operates at electronic speeds.
* Words, address-access to any word.
* Word length-number of bits in each word.
* RAM
* Time required to access one word at time is called memory access time.
* Cache memory: a smaller, faster RAM unit is called a cache. It is used to hold the section of a program that is currently being executed, along with any associated data. The purpose of cache is to provide high instruction execution rate.
* Secondary storage-used for large amount of data.
* Secondary storage devices-magnetic tapes, disk and optical disks.
  1. ALU
* Most computer operations are executed in ALU
* All arithmetic & logic operations.
* Registers-high speed storage elements.
* Operations are done by bringing all operands to processor.
  1. Output
* Function is to send processed results to outside world.
* O**/**P device-printer.
  1. Control unit
* This is the center that sends control signals to other units &senses their states.
* Timing signals.

The operation of a computer can be summarized as,

1. Computer accepts information &data through I**/**P unit.
2. Information is processed in ALU.
3. Processed information leaves computer through O**/**P unit
4. All activities are directed by CU.

# 1.3 Basic operational concepts

An Instruction consists of 2 parts, 1) Operation code (Opcode) and 2) Operands.

Operands

Opcode

* The data/operands are stored in memory.
* The individual instructions are brought from the memory to the processor. Then, the processor performs the specified operation.
* Consider a typical instruction ADD R2, Loc
* This instruction is an addition operation. The following are the steps to execute the instruction:

Step 1: Fetch the instruction from main-memory into the processor.

Step 2: Fetch the operand at location Loc from main-memory into the processor.

Step 3: Add the memory operand (i.e. fetched contents of LOC) to the contents of register R2.

Step 4: Store the result (sum) in R2.

The same instruction can be realized using 2 instructions as: Load R2, Loc

Add R3, R2, R4

The following are the steps to execute the instruction:

Step 1: Fetch the instruction from main-memory into the processor.

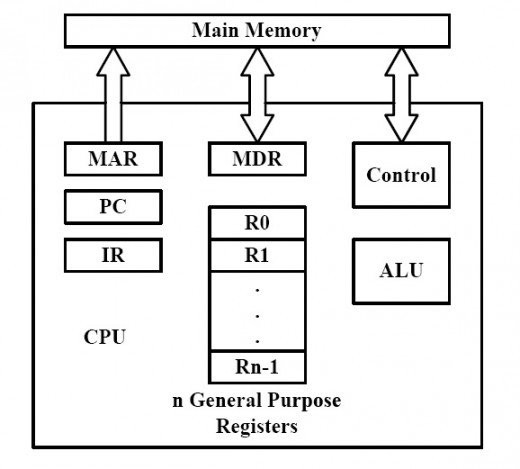
Step 2: Fetch the operand at location Loc from main-memory into the register R2.

Step 3: Add the content of Register R2 and the contents of register R4. Step 4: Store the result (sum) in R3.

Ex: store R4, Loc this instruction copies operand in R4 to memory location Loc.

MAIN PARTS OF PROCESSOR

* The processor contains ALU, control-circuitry and many registers.
* The Processor contains ’n’ general-purpose registers R0 through Rn-1.
* The Instruction Register (IR) holds the instruction that is currently being executed.
* The control-unit generates the timing-signals that determine when a given action is to take place.
* The Program Counter (PC) contains the memory-address of the next-instruction to be fetched & executed.During the execution of an instruction, the contents of PC are updated to point to next instruction.
* The Memory Address Register (MAR) holds the address of the memory-location to be accessed.
* The Memory Data Register (MDR) contains the data to be written into or read out of the addressed location. MAR and MDR facilitates the communication with memory.



STEPS TO EXECUTE AN INSTRUCTION

* 1. The address of first instruction (to be executed) gets loaded into PC.
  2. The contents of PC (i.e. address) are transferred to the MAR & control-unit issues Read signal to memory.
  3. After certain amount of elapsed time, the first instruction is read out of memory and placed into MDR.
  4. Next, the contents of MDR are transferred to IR. At this point, the instruction can be decoded & executed.
  5. To fetch an operand, it's address is placed into MAR & control-unit issues Read signal. As a result, the operand is transferred from memory into MDR, and then it is transferred from MDR to ALU.
  6. Likewise required number of operands is fetched into processor.
  7. Finally, ALU performs the desired operation.
  8. If the result of this operation is to be stored in the memory, then the result is sent to the MDR.
  9. The address of the location where the result is to be stored is sent to the MAR and a Write cycle is initiated.
  10. At some point during execution, contents of PC are incremented to point to next instruction in the program.

# 1.4 Number Representation and Arithmetic:

**Number Representation**

Most natural way to represent a number in a computer system is by a binary number. A text character can be represented by a character code. Binary number are represented by using any of the following systems:

* **Sign-and-magnitude** - Most significant bit determines sign, remaining unsigned bits represent magnitude
* **1’s complement** - Most significant bit determines sign. To change sign from unsigned to negative, invert all the bits ( -3 is obtained by complementing each bit in vector 0011 to yield 1100).
* **2’s complement** - Most significant bit determines sign. To change sign from unsigned to negative, invert all the bits and add 1. This is equivalent to subtracting the positive number from 2n. The representations are as given in the table below

# 

**Addition & Subtraction of Signed Numbers: The** 3 systems of representing signed numbers systems differ only in the way they represent negative number.

• Sign and magnitude system – simplest representation – most awkward for addition and subtraction.

• 1’s complement method is somewhat better.

• 2’s complement is the most efficient method.

**Circle representation of Integer Mod N:** This is a graphical technique to compute (a+b) mod 16. This can be also used for addition involving signed numbers. Both the cases are shown below



The operation (7+4) mod 16 yields the value 11. To perform this graphically using the above representation locate 7 on the circle and then move 4 units in the clock wise direction to arrive at the answer 11.

Next let us consider adding +7 to -3. 2’s complement representation for 7 is 0111 and -3 is 1101. Thus locate 0111 and then move 1101(13 steps) in clockwise direction to arrive at 0100 = +4.

**Overflow in integer arithmetic:** In 2’s complement arithmetic addition of opposite sign numbers will never result in overflow. If the numbers are the same sign and the result is the opposite sign, overflow has occurred. E.g. 0100 + 0111 =1011 (but 1011 is -5) Overflow occurs when carry-in to the high-order bit does not equal carry out.

**The rules governing addition and subtraction of n-bit signed numbers using the 2’scomplement representation system may be stated as follows:**

1. To add two numbers, add their n-bit representations, ignoring the carry-out bit from

the most significant bit (MSB) position. The sum will be the algebraically correct value in 2’s-complement representation if the actual result is in the range−2n−1 through+2n−1 − 1.

1. To subtract two numbers X and Y , that is, to perform X − Y , form the 2’s-complement of Y , then add it to X using the add rule. Again, the result will be the algebraically correct value in 2’s-complement representation if the actual result is in the range −2n−1 through +2n−1 − 1.

# 1.5 Character Representation

**Characters:** Apart from numbers computers must be able to handle alphanumeric text information consisting of characters. Characters can be letters of alphabets, decimal digits, punctuation marks etc. Most widely used code was ASCII and now a days unicode is being used widely.

# 1.6 Performance

The most important measure of performance of a computer is how quickly it can execute programs. The speed of a computer is affected by the design of

* 1. Instruction-set.
  2. Hardware & the technology in which the hardware is implemented.
  3. Software including the operating system.

Because programs are usually written in a HLL (High Level Language) the performance is affected by the compiler that translates HLL program to machine language. For best performance, it is necessary to design the compiler, machine instruction set and hardware in a co-ordinated way.

# PROCESSOR CACHE

At the start of program execution, all program instructions are stored in the main- memory. As execution proceeds, instructions are fetched into the processor, and a copy is placed in the cache. Later, if the same instruction is needed a second time, it is read directly from the cache. A program will be executed faster if movement of instruction/ data between the main-memory and the processor is minimized which is achieved by using the cache.

# PROCESSOR CLOCK

Processor circuits are controlled by a timing signal called a Clock.The clock defines regular time intervals called Clock Cycles.To execute a machine instruction, the

processor divides the action to be performed into a sequence of basic steps such that each step can be completed in one clock cycle.