**FUNCTIONS OF CPU**

1. **Fetch:** The first step, fetch, involves retrieving an instruction (which is represented by a number or sequence of numbers) from program memory. The instruction’s location (address) in program memory is determined by a program counter (PC), which stores a number that identifies the address of the next instruction to be fetched. After an instruction is fetched, the PC is incremented by the length of the instruction so that it will contain the address of the next instruction in the sequence.
2. **Decode:** The instruction that the CPU fetches from memory determines what the CPU will do. In the decode step, performed by the circuitry known as the instruction decoder, the instruction is converted into signals that control other parts of the CPU. The way in which the instruction is interpreted is defined by the CPU’s instruction set architecture (ISA).

In some CPU designs, the instruction decoder is implemented as a hardwired, unchangeable circuit. In others. A micro-program is used to translate instructions into sets of CPU configuration signals that are applied sequentially over multiple clock pulses. In some cases, the memory that stores the micro-program is re-writable, making it possible to change the way in which the CPU decodes instructions.

1. **Execute:** After the fetch and decode steps, the execute step is performed. Depending on the CPU architecture, this may consist of a single action or a sequence of series. During each action, various parts of the CPU are electrically connected so they can perform all or part of the desired operation and then the action is completed, typically in response to a clock pulse. Very often the results are written to an internal CPU register for quick access ny subsequent instructions. In other cases results may be written to slower,but less expensive and higher capacity main memory.
2. **Store:** The CPU program must give feedback after executing an instruction, and the output data is written to the memory.