

## Hardwired Vs. Micro-programmed Control Unit:

In a system or computer, most of the tasks are controlled with the help of a processor or CPU (Central processing unit), which is the main component of a computer. The CPU usually has two main modules: **control unit** (CU) and **arithmetic and logic unit** (ALU). The control unit (CU) is used to synchronize the tasks by sending control signals that govern the timing of the events. On the other hand, mathematical and logical operations can be handled with the help of ALU. Micro programmed control units and hardwired control units can be two categories of control units. We can execute an instruction with the help of these two types of control units.

In the **hardwired control unit**, the execution of operations is much faster, but the implementation, modification, and decoding are difficult. In contrast, implementing, modifying, decoding **micro-programmed control units** is very easy. The micro-programmed control unit is also able to handle complex instructions. With the help of control signals generated by micro-programmed and hardwired control units, we are able to fetch and execute the instructions.

## Control Signals

In order to generate the control signals, basic designing is required in both the type of control system. The functionality of a processor's hardware is realized with the help of these control signals.

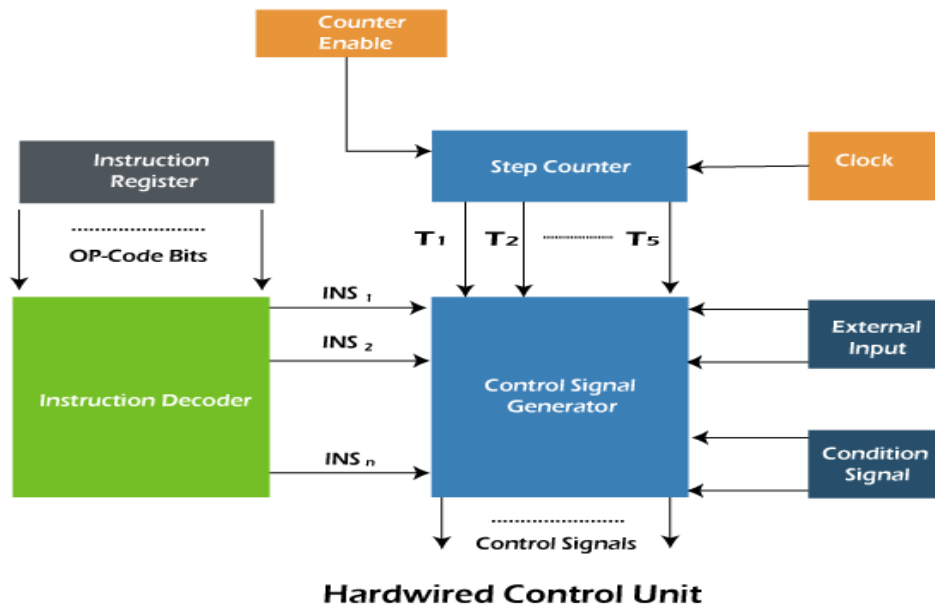
Purposes of Control signals:

- Control signals are used to know what operation is going to be performed.
- It is used to know about the sequence of operations that are performed by the processor.
- It is used to know about the timing at which an operation must be executed and many other types of things.

## Hardwired Control Unit

By generating appropriate control signals, the hardwired control unit is able to execute the instructions at a correct time and proper sequence. As compared to the micro-programmed, the hardwired CU is generally faster. In this CU, the control signals are generated with the help of PLA circuit and state counter. Here the Central processing unit requires all these control signals. With the help

of hardware, the hardwired control signals are generated, and it basically uses the circuitry approach.



- The **instruction register** is a type of processor register used to contain an instruction that is currently in execution. As we can see, the instruction register is used to generate the OP-code bits respective of the operation as well as the addressing mode of operands.
- The above generated Op-code bits are received in the field of an **instruction decoder**. The instruction decoder interprets the operation and instruction's addressing mode. Now on the basis of the addressing mode of instruction and operation which exists in the instruction register, the instruction decoder sets the corresponding Instruction signal  $INS_i$  to 1. Some steps are used to execute each instruction, i.e., **instruction fetch, decode, operand fetch, Arithmetic and logical unit, and memory store**.
- The information about the current step of instruction must be known by the control unit. Now the **Step Counter** is implemented, which is used to contain the signals from  $T_1, \dots, T_5$ . Now on the basis of the step which contains the instruction, one of the signals of a step counter will be set from  $T_1$  to  $T_5$  to 1.
- Now we have a question that how the step counter knows about the current step of instruction? So to know the current step, a **Clock** is implemented. The one-clock cycle of the clock will be completed for

each step. For example, suppose that if the step counter sets T3 to 1, then after completing one clock cycle, the step counter will set T4 to 1.

- Now we have a question, i.e., what will happen if the execution of an instruction is interrupted for some reason? Will the step counter still be triggered by the clock? The answer to this question is **No**. As long as the execution in current step is completed, the **Counter Enable** will "disable" the Step Counter so that it will stop then increment to the next step signal.
- Now we have a question, i.e., what if the execution of instruction depends on some conditions? In this case, the **Condition Signals** will be used. There are various conditions in which the signals are generated with the help of control signals that can be less than, greater than, less than equal, greater than equal, and many more.
- The **external input** is the last one. It is used to tell the Control Signal Generator about the interrupts, which will affect the execution of an instruction.

So, on the basis of the input obtained by the conditional signals, step counter, external inputs, and instruction register, the control signals will be generated with the help of Control signal Generator.

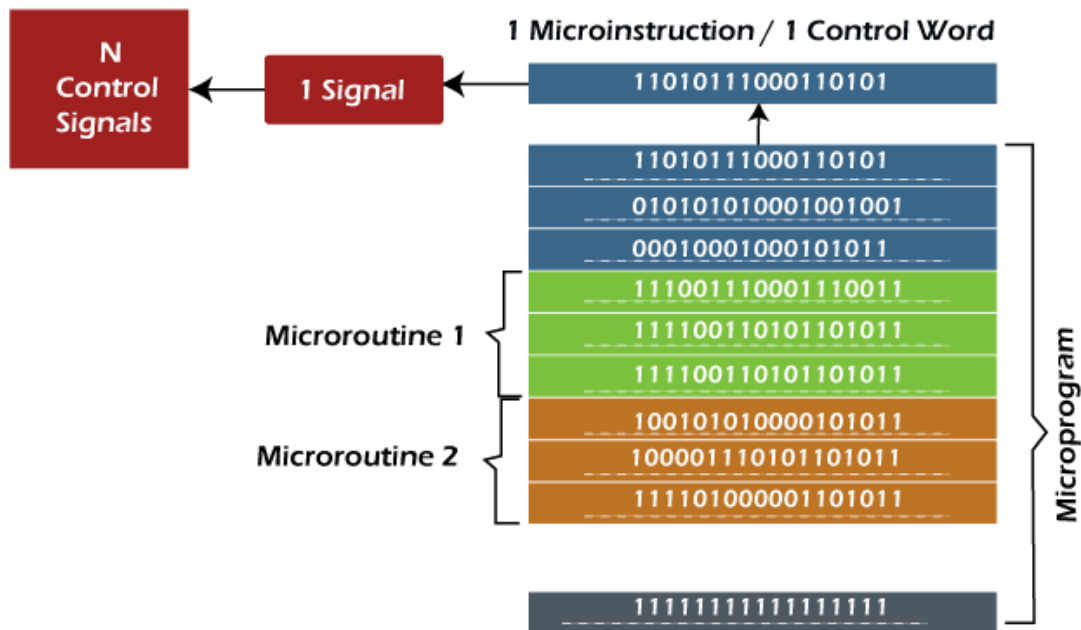
### **Micro-programmed Control Unit**

A micro-programmed control unit can be described as a simple logic circuit. We can use it in two ways, i.e., it is able to execute each instruction with the help of generating control signals, and it is also able to do sequencing through microinstructions. It will generate the control signals with the help of programs. At the time of evolution of CISC architecture in the past, this approach was very famous. The program which is used to create the control signals is known as the "Micro-program". The micro-program is placed on the processor chip, which is a type of fast memory. This memory is also known as the control store or control memory.

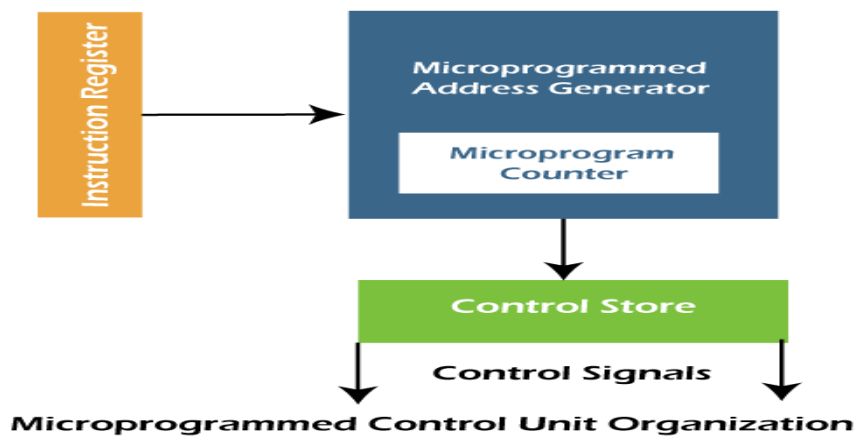
A micro-program is used to contain a set of microinstructions. Each microinstruction or control word contains different bit patterns. The n bit words are contained by each microinstruction. On the basis of the bit pattern of a control word, every control signals differ from each other.

Like the above, the instruction execution in a micro-programmed control unit is also performed in steps. So for each step, the micro-program contains a control

word/ microinstruction. If we want to execute a particular instruction, we need a sequence of microinstructions. This process is known as the micro-routine.



Now we will learn about the organization of Micro-program CU. Then we will learn about the flow of instruction execution with the help of instruction execution steps, which are described as follows:



- **Instruction** fetch is the **first step**. In this step, the instruction is fetched from the IR (Instruction Register) with the help of a Microinstruction address register.
- **Decode** is the **second step**. In this step, the instructions obtained from the instruction register are decoded with the help of a microinstruction address generator. Here we will also get the starting address of a micro-routine. With the help of this address, we can easily perform the

operation, which is specified in the instruction. It will also load the starting address into the micro-program counter.

- **Increment** is the third step. In this step, the control word, which corresponds to the starting address of a micro-program, will be read. When the execution proceeds, the value of the micro-program counter will be increased so that it can read the successive control words of a micro-routine.
- **End bit** is the fourth step. The microinstruction of a micro-routine contains a bit, which is known as the end bit. In this step, the execution of the microinstruction will be successfully completed when the end bit is set to 1.
- This is the last step, and in this step, the micro-program address generator will again go back to **Step 1** so that a new instruction is fetched and this process continues with the next program.

In the micro-programmed control unit, the micro-programs are stored with the help of Control memory or Control store. The implementation of this CU is very easy and flexible, but it is slower as compared to the Hardwired control unit.