Introduction of Control Unit and its Design

Control Unit is the part of the computer's central processing unit (CPU), which directs the operation of the processor. It was included as part of the Von Neumann Architecture by John von Neumann. It is the responsibility of the Control Unit to tell the computer's memory, arithmetic/logic unit and input and output devices how to respond to the instructions that have been sent to the processor. It fetches internal instructions of the programs from the main memory to the processor instruction register, and based on this register contents, the control unit generates a control signal that supervises the execution of these instructions.

A control unit works by receiving input information to which it converts into control signals, which are then sent to the central processor. The computer's processor then tells the attached hardware what operations to perform. The functions that a control unit performs are dependent on the type of CPU because the architecture of CPU varies from manufacturer to manufacturer. Examples of devices that require a CU are:

- Control Processing Units(CPUs)
- Graphics Processing Units(GPUs)

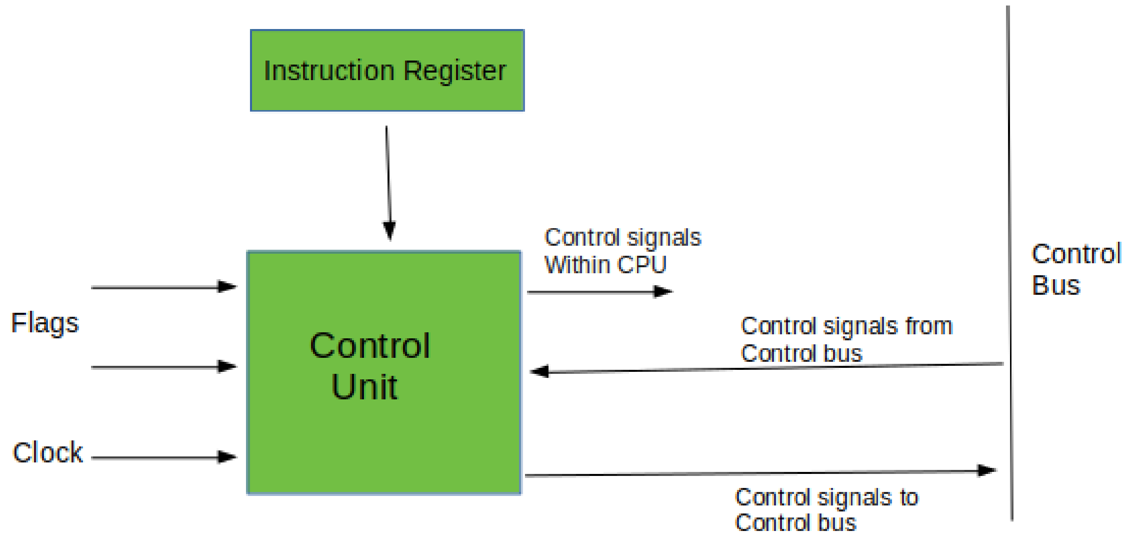

Figure 1: Block Diagram of control unit

Functions of the Control Unit –

1. It coordinates the sequence of data movements into, out of, and between a processor's many sub-units.
2. It interprets instructions.
3. It controls data flow inside the processor.
4. It receives external instructions or commands to which it converts to sequence of control signals.
5. It controls many execution units (i.e. ALU, data buffers and registers) contained within a CPU.
6. It also handles multiple tasks, such as fetching, decoding, execution handling and storing results.

Types of Control Unit –

There are two types of control units: Hardwired control unit and Micro programmable control unit.

1. **Hardwired Control Unit –**

   In the Hardwired control unit, the control signals that are important for instruction execution control are generated by specially designed hardware logical circuits, in which we cannot modify the signal generation method without physical change of the circuit structure. The operation code of an instruction contains the basic data for control signal generation. In the instruction decoder, the operation code is decoded. The instruction decoder constitutes a set of many decoders that decode different fields of the instruction opcode.

   As a result, few output lines going out from the instruction decoder obtains active signal values. These output lines are connected to the inputs of the matrix that generates control signals for executive units of the computer. This matrix implements logical combinations of the decoded signals from the instruction opcode with the outputs from the matrix that generates signals representing consecutive control unit states and with signals coming from the outside of the processor, e.g. interrupt signals. The matrices are built in a similar way as a programmable logic arrays.

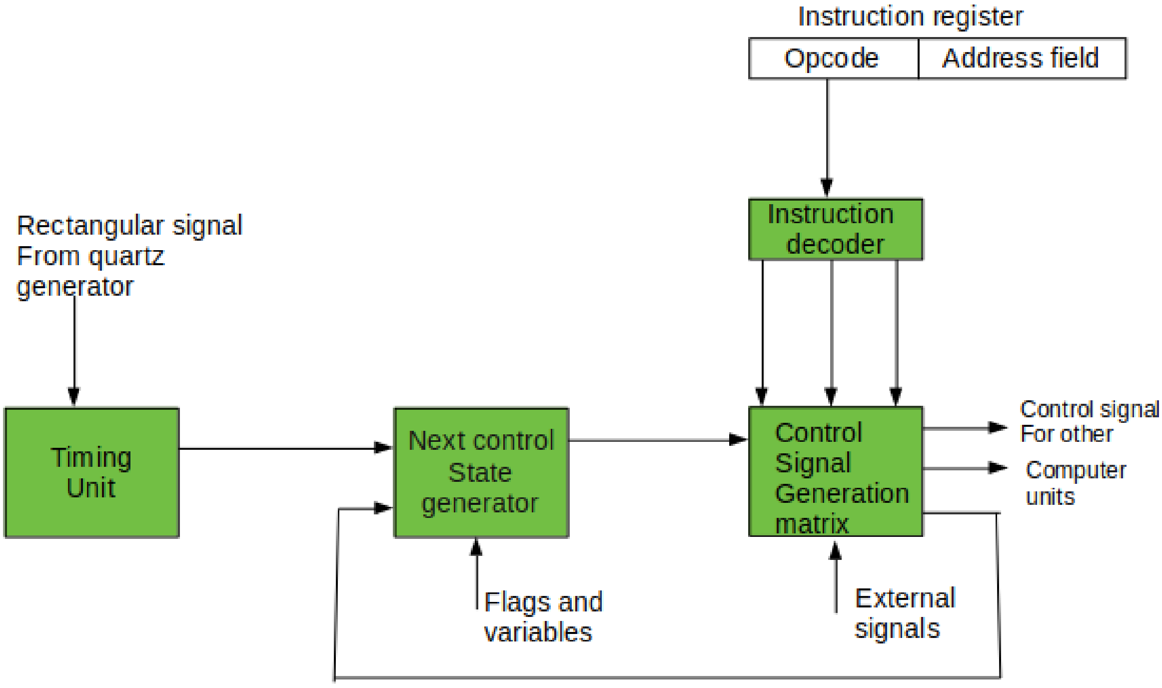

Figure 2: Hardwired Control Unit

Control signals for an instruction execution have to be generated not in a single time point but during the entire time interval that corresponds to the instruction execution cycle. Following the structure of this cycle, the suitable sequence of internal states is organized in the control unit.

A number of signals generated by the control signal generator matrix are sent back to inputs of the next control state generator matrix. This matrix combines these signals with the timing signals, which are generated by the timing unit based on the rectangular patterns usually supplied by the quartz generator. When a new instruction arrives at the control unit, the control units is in the initial state of new instruction fetching. Instruction decoding allows the control unit enters the first state relating execution of the new instruction, which lasts as long as the timing signals and other input signals as flags and state information of the computer remain unaltered. A change of any of the earlier mentioned signals stimulates the change of the control unit state.

This causes that a new respective input is generated for the control signal generator matrix. When an external signal appears, (e.g. an interrupt) the control unit takes entry into a next control state that is the state concerned with the reaction to this external signal (e.g. interrupt processing). The values of flags and state variables of the computer are used to select suitable states for the instruction execution cycle.

The last states in the cycle are control states that commence fetching the next instruction of the program: sending the program counter content to the main memory address buffer register and next, reading the instruction word to the instruction register of computer. When the ongoing instruction is the stop instruction that ends program execution, the control unit enters an operating system state, in which it waits for a next user directive.

2. **Micro programmable control unit –**

The fundamental difference between these unit structures and the structure of the hardwired control unit is the existence of the control store that is used for storing words containing encoded control signals mandatory for instruction execution.

In microprogrammed control units, subsequent instruction words are fetched into the instruction register in a normal way. However, the operation code of each instruction is not directly decoded to enable immediate control signal generation but it comprises the initial address of a microprogram contained in the control store.

- **With a single-level control store:**

  In this, the instruction opcode from the instruction register is sent to the control store address register. Based on this address, the first microinstruction of a microprogram that interprets execution of this instruction is read to the microinstruction register. This microinstruction contains in its operation part encoded control signals, normally as few bit fields. In a set microinstruction field decoders, the fields are decoded. The microinstruction also contains the address of the next microinstruction of the given instruction microprogram and a control field used to control activities of the microinstruction address generator.

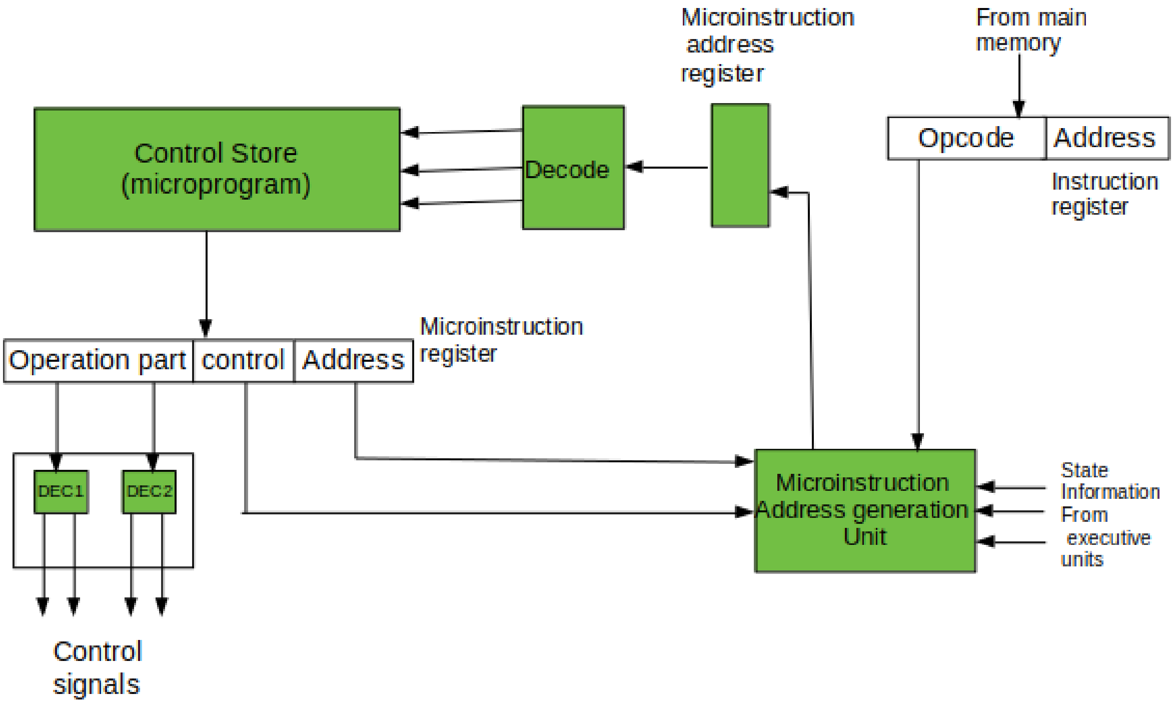

Figure 3: Micro programmable control unit with a single-level control store

The last mentioned field decides the addressing mode (addressing operation) to be applied to the address embedded in the ongoing microinstruction. In microinstructions along with conditional addressing mode, this address is refined by using the processor condition flags that represent the status of computations in the current program. The last microinstruction in the instruction of the given microprogram is the microinstruction that fetches the next instruction from the main memory to the instruction register.

- **With a two-level control store :**

  In this, in a control unit with a two-level control store, besides the control memory for microinstructions, a Nano-instruction memory is included. In such a control unit, microinstructions do not contain encoded control signals. The operation part of microinstructions contains the address of the word in the Nano-instruction memory, which contains encoded control signals. The Nano-instruction memory contains all combinations of control signals that appear in microprograms that interpret the complete instruction set of a given computer, written once in the form of Nano-instructions.

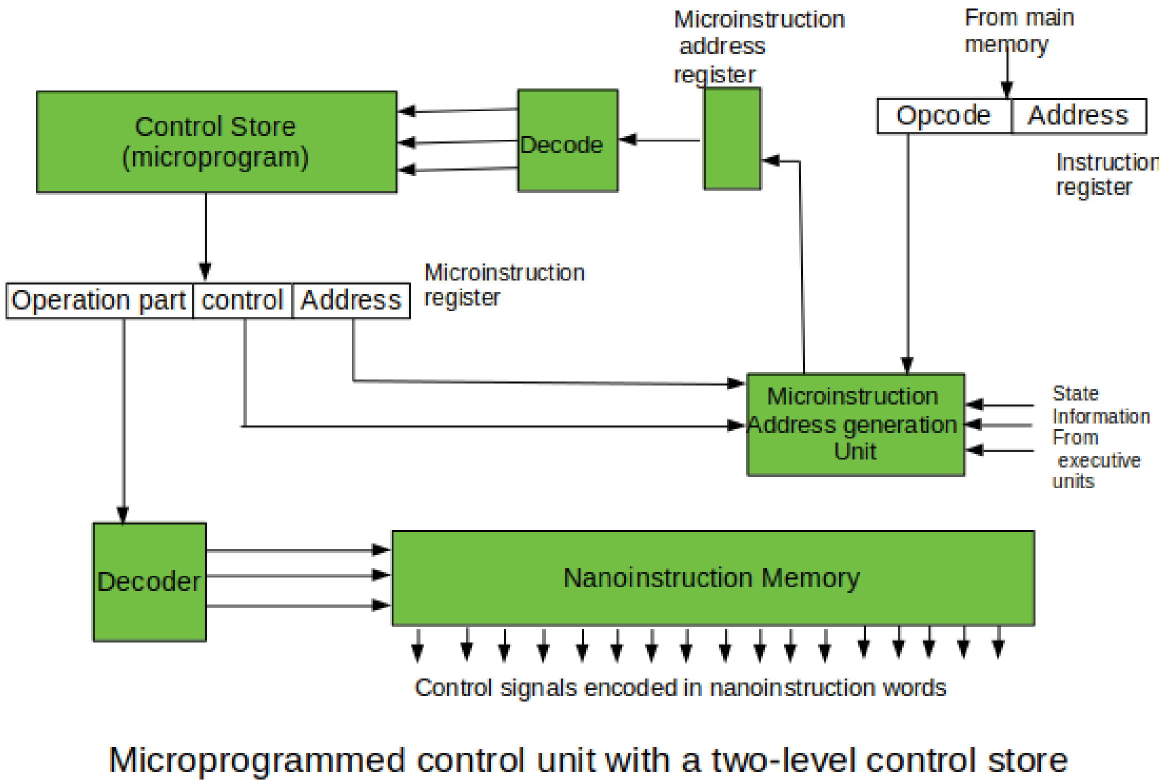

Figure 4: Micro programmable control unit with a two-level control store

In this way, unnecessary storing of the same operation parts of microinstructions is avoided. In this case, microinstruction word can be much shorter than with the single level control store. It gives a much smaller size in bits of the microinstruction memory and, as a result, a much smaller size of the entire control memory. The microinstruction memory contains the control for selection of consecutive microinstructions, while those control signals are generated at the basis of Nano-instructions. In Nano-instructions, control signals are frequently encoded using 1 bit/ 1 signal method that eliminates decoding.

Difference between Hardwired and Microprogrammed Control Unit:

ATTRIBUTES	HARDWIRED CONTROL UNIT	MICROPROGRAMMED CONTROL UNIT
1. Speed	Speed is fast	Speed is slow
2. Cost of Implementation	More costly.	Cheaper.
3. Flexibility	Not flexible to accommodate new system specification or new instruction redesign is required.	More flexible to accommodate new system specification or new instruction sets.
4. Ability to Handle Complex Instructions	Difficult to handle complex instruction sets.	Easier to handle complex instruction sets.
5. Decoding	Complex decoding and sequencing logic.	Easier decoding and sequencing logic.
6. Applications	RISC Microprocessor	CISC Microprocessor
7. Instruction set of Size	Small	Large
8. Control Memory	Absent	Present
9. Chip Area Required	Less	More
10. Occurrence	Occurrence of error is more	Occurrence of error is less

Difference between Horizontal and Vertical micro-programmed Control Unit

Basically, **control unit (CU)** is the engine that runs the entire functions of a computer with the help of control signals in the proper sequence. In the **micro-programmed** control unit approach, the control signals that are associated with the operations are stored in special memory units. It is convenient to think of sets of control signals that cause specific micro-operations to occur as being "microinstructions". The sequences of microinstructions could be stored in an internal "*control*" memory.

Micro-programmed control unit can be classified into two types based on the type of Control Word stored in the Control Memory, viz., Horizontal micro-programmed control unit and Vertical micro-programmed control unit.

- In *Horizontal micro-programmed* control unit, the control signals are represented in the decoded binary format, i.e., 1 bit/CS. Here 'n' control signals require n bit encoding. On the other hand.
- In *Vertical micro-programmed* control unit, the control signals are represented in the encoded binary format. Here 'n' control signals require $\log_2 n$ bit encoding.

Comparison between Horizontal micro-programmed control unit and Vertical micro-programmed control unit:

HORIZONTAL M-PROGRAMMED CU	VERTICAL M-PROGRAMMED CU
It supports longer control word.	It supports shorter control word.
It allows higher degree of parallelism. If degree is n, then n Control Signals are enabled at a time.	It allows low degree of parallelism i.e., degree of parallelism is either 0 or 1.
No additional hardware is required.	Additional hardware in the form of decoders are required to generate control signals.
It is faster than Vertical micro-programmed control unit.	it is slower than Horizontal micro-programmed control unit.

HORIZONTAL M-PROGRAMMED CU	VERTICAL M-PROGRAMMED CU
It is less flexible than Vertical micro-programmed control unit.	It is more flexible than Horizontal micro-programmed control unit.
Horizontal micro-programmed control unit uses horizontal microinstruction, where every bit in the control field attaches to a control line.	Vertical micro-programmed control unit uses vertical microinstruction, where a code is used for each action to be performed and the decoder translates this code into individual control signals.
Horizontal micro-programmed control unit makes less use of ROM encoding than vertical micro-programmed control unit.	Vertical micro-programmed control unit makes more use of ROM encoding to reduce the length of the control word.

Example: Consider a hypothetical Control Unit which supports 4 k words. The Hardware contains 64 control signals and 16 Flags. What is the size of control word used in bits and control memory in byte using:

a) Horizontal Programming

b) Vertical Programming?

Solution: