Micro Programmed Control Unit

The function of the control unit in a digital computer is to initiate sequences of microoperations. The number of different types of microoperations that are available in a given system is finite. To start with microprogrammed control unit initiates a series of sequential steps of microoperations. During any given time, certain microoperations are to be initiated, while others remain idle. Various essential features of microprogrammed control unit include:

Control Word: The control variables at any given time can be represented by a string of 1's and 0' a called a control word. As such, control words can be programmed to perform various operations on the components are stored in memory is called a micro programmed control unit.

Microinstruction: Each word in control memory contains within it a microinstruction. The microinstruction specifies one or more micro operations for the system.

Microprogram: A sequence of microinstructions constitutes a microprogram.

Control Memory: Since alterations of the microprogram are not needed once the control unit is in operation, the control memory can be a read-only memory (ROM). The content of the words in ROM are fixed and cannot be altered by simple programming since no writing capability is available in the ROM. ROM words are made permanent during the hardware production of the unit.

Functioning: The use of a microprogram involves placing all control variables in words of ROM for use by the control unit through successive read operations. The content of the word in ROM at a given address specifies a microinstruction.

A more advanced development known as **dynamic microprogramming** permits a microprogram to be loaded initially from an auxiliary memory such as a magnetic disk. Control units that use dynamic microprogramming employ a **writable control memory**.

Organization of Microprogrammed Control Unit: A computer that employs a micro programmed control unit will have two separate memories a main memory and a control memory.

1. The **main memory** is available to the user for storing the program. The contents of main memory may alter when the data are manipulated and every time that the program is changed. The user's program in main memory consists of machine instructions and data.
2. In contrast, the **control memory** holds a fixed microprogram that cannot be altered by the occasional user. The microprogram consists of microinstructions that specify various internal control signals for execution of register microoperations. Each machine instruction initials a series of microinstructions in control memory. These microinstructions generate the rations to fetch the instruction from main memory; to evaluate the effective address, to execute the operation specified by the instruction, and to return control to the fetch phase in order to repeat the cycle for the next instruction.

The general configuration of a microprogrammed control unit is demonstrated in the block diagram of following figure:

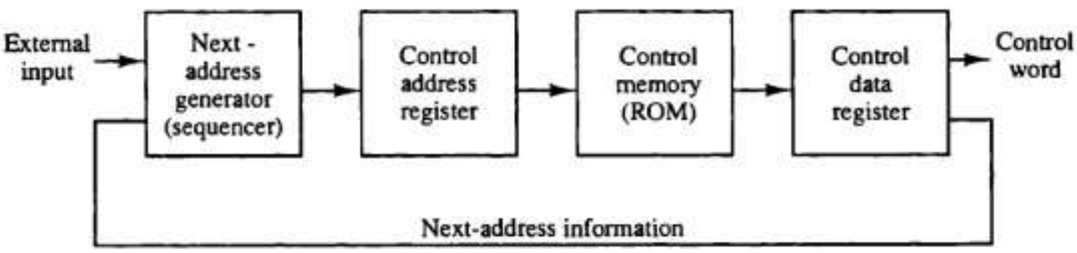

Figure 1: Microprogrammed Control Unit Organisation

Compiled by: Dr. Sandeep Maan, Assoc Prof in Computer Sc, GCG Sec-14, Gurugram.
Reference: M Morris Mano, Computer System Architecture, Third Edition, Pearson Education

Various components of this organization include:

1. **Control Memory:** The control memory is assumed to be a ROM, within which all control information is permanently stored.
2. **Control Address Register:** The control memory address register specifies the address of the microinstruction.
3. **Control Data Register:** The control data register holds the microinstruction read from memory.
4. **Microprogram Sequencer:** The next address generator is sometimes called a microprogram sequencer, as it determines the address sequence that is read from control memory. The address of the next microinstruction can be specified in several ways, depending on the sequencer inputs. Typical functions of a microprogram sequencer are

    1. Incrementing the control address register by one,
    2. Loading into the control address register an address from control memory
    3. Transferring an external address
    4. or loading an initial address to start the control operations.

5. **Pipeline Register:** The control data register holds the present microinstruction while the next address is computed and read from memory; the data register is sometimes called a pipeline register.

Working: The microinstruction contains a control word that specifies one or more microoperations for the data processor. Once these operations are executed, the control must determine the nest address. The location of the next microinstruction may be the one next in sequence, or it may be located somewhere else in the control memory. For this reason, it is necessary to use some bits of the present microinstruction to control the generation of the address of the next microinstruction. The next address may also be a function of external input conditions. While the microoperations are being executed, the next address is computer in the next address generator circuit and then transferred into the control address register to read the next microinstruction. Thus a microinstruction contains bits for initiating microoperations in the data processor part and bits that determine the address sequence for the control memory.

Microprogrammed Control Unit Vs Hardwired Control Unit: The main advantage of the micro programmed control is the fact that once the hardware configuration is established; there should be no need for further hardware or wiring changes. If we want to establish a different control sequence for the system, all we need to do is specify a different set of microinstructions for control memory. The hardware configuration should not be changed for different operations; the only thing that must be changed is the microprogram residing in control memory.
Most computers based on the reduced instruction set computer (RISC) architecture concept, use hardwired control rather than a control memory with a microprogram.

Microinstruction Sequencing (Address Sequencing in Microprogrammed Control Unit)
An initial address is loaded into the control address register when power is turned on in the computer. This address is usually the address of the first microinstruction that activates the instruction fetch routine. The fetch routine may be sequenced by incrementing the control address register through the rest of its microinstructions. At the end of the fetch routine, the instruction is in the instruction register of the computer.
The control memory next must go through the routine that determines the effective address of the operand. A machine instruction may have bits that specify various addressing modes, such as indirect address and index registers. The effective address computation routine in control memory can be

reached through a branch microinstruction, which is conditioned on the status of the mode bits of the instruction. When the effective address computation routine is completed, the address of the operand is available in the memory address register.
The next step is to generate the microoperations that execute the instruction fetched from memory. The microoperation steps to be generated in processor register depend on the operation code part of the instruction. Each instruction has its own microprogram routine stored in a given location of control memory. The transformation from the instruction code bits to an address in control memory where the routine is located is referred to as a mapping process. A mapping procedure is a rule that transforms the instruction code into a control memory address. Once the required routine is reached, the microinstructions that execute the instruction may be sequenced by incrementing the control address register, but sometimes the sequence of microoperations will depend on values of certain status bits in processor registers. Micro programs that employ subroutines will require an external register for storing the return address. Return addresses cannot be stored in ROM because the unit has no writing capability.
When the execution of the instruction is completed, control must return to the fetch routine. This is accomplished by executing an unconditional branch microinstruction to the first address of the fetch routine. In summary, the address sequencing capabilities required in control memory are:

1. Incrementing of the control address register.
2. Unconditional branch or conditional branch, depending on statues bit conditions.
3. A mapping process from the bits of the instruction to an address for control memory.
4. A facility for subroutine call and return.

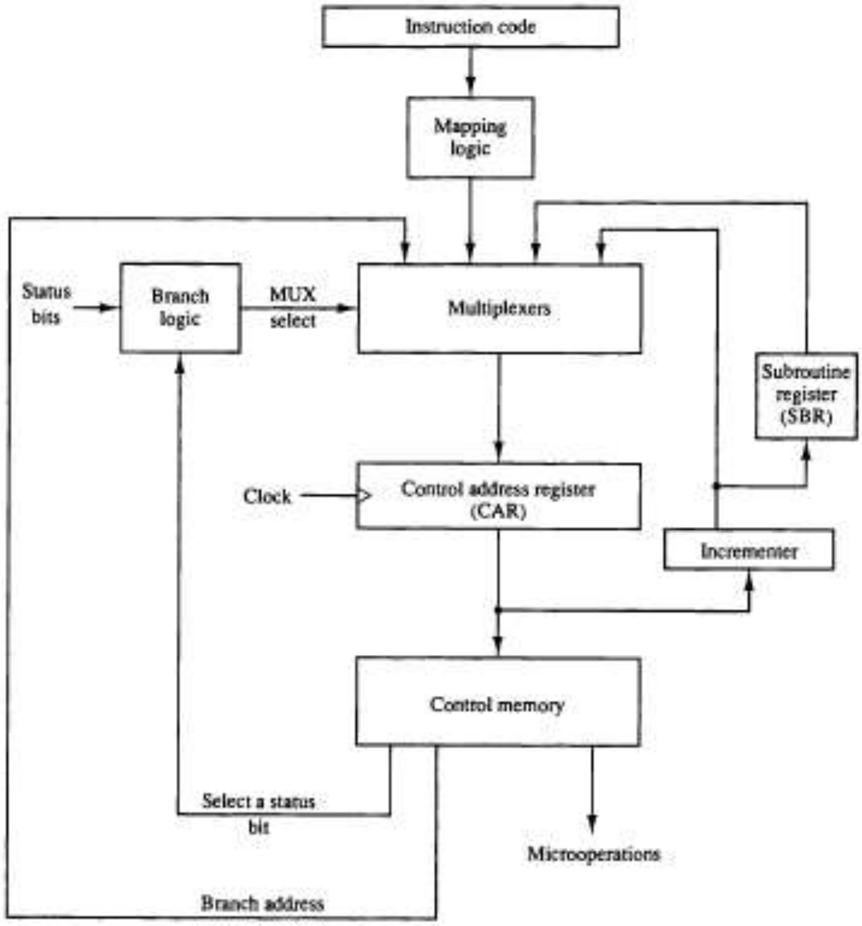

Figure 2:Address Sequencing in Microprogrammed Control Unit