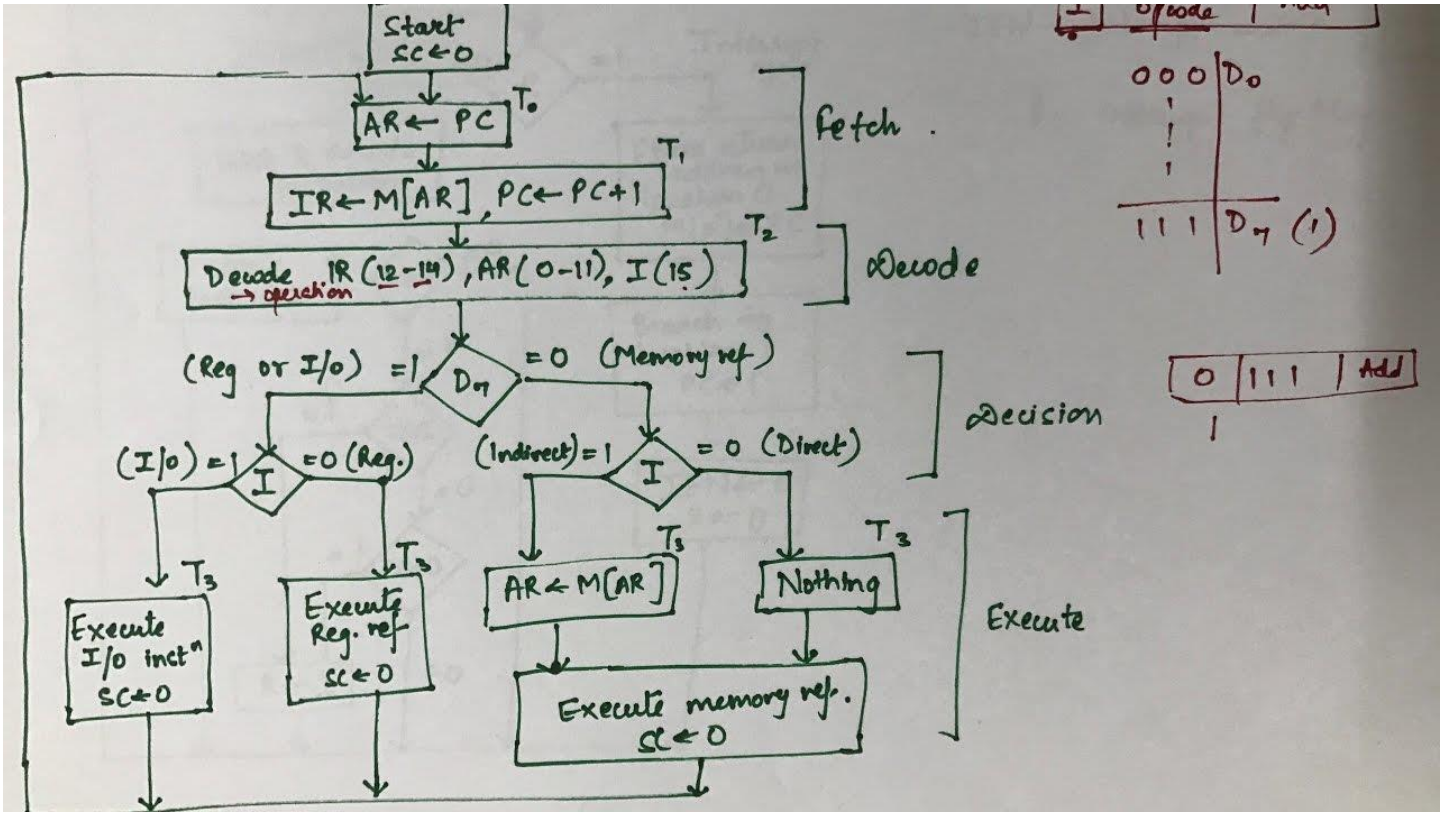
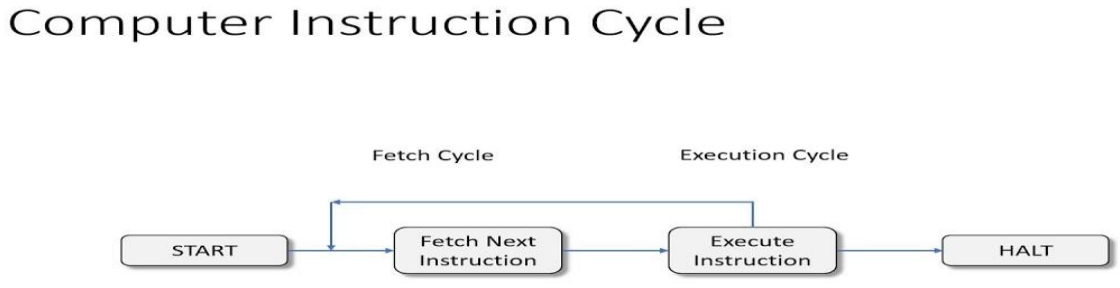
Instruction cycle



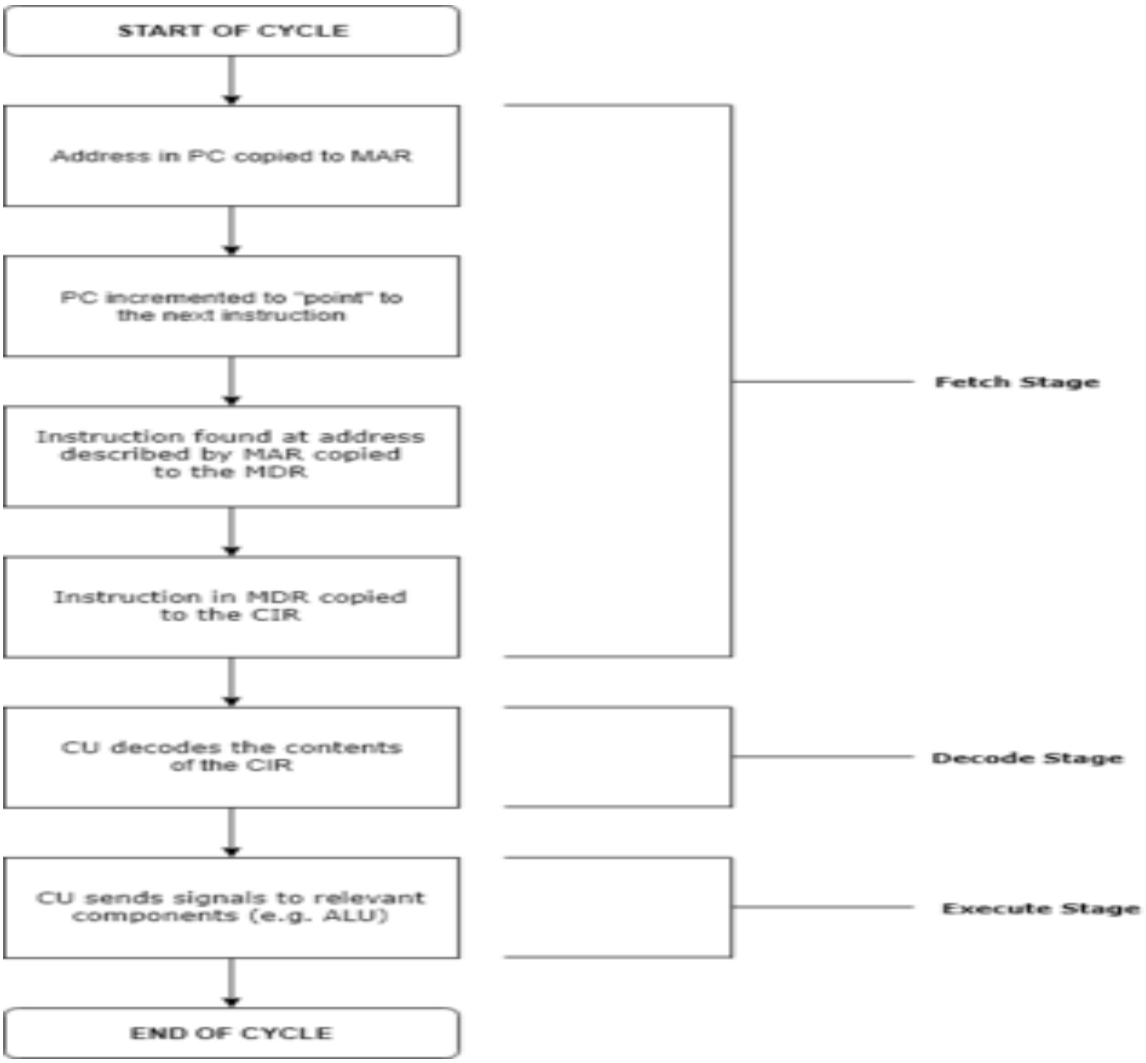


The instruction cycle (also known as the fetch–decode–execute cycle or simply the fetch-execute cycle) is the cycle which the central processing unit (CPU) follows from boot-up until the computer has shut down in order to process instructions. It is composed of three main stages: the fetch stage, the decode stage, and the execute stage.

This is a simple diagram illustrating the individual stages of the fetch-decode-execute cycle.

In simpler CPUs, the instruction cycle is executed sequentially, each instruction being processed before the next one is started. In most modern CPUs, the instruction cycles are instead executed concurrently, and often in parallel, through an instruction pipeline: the next instruction starts being processed before the previous instruction has finished, which is possible because the cycle is broken up into separate.

Execution of a complete instruction



Microinstruction Sequencing:

A micro-program control unit can be viewed as consisting of two parts:

1. The control memory that stores the microinstructions.
2. Sequencing circuit that controls the generation of the next address.

A micro-program sequencer attached to a control memory inputs certain bits of the microinstruction, from which it determines the next address for control memory. A typical sequencer provides the following address-sequencing capabilities:

1. Increment the present address for control memory.

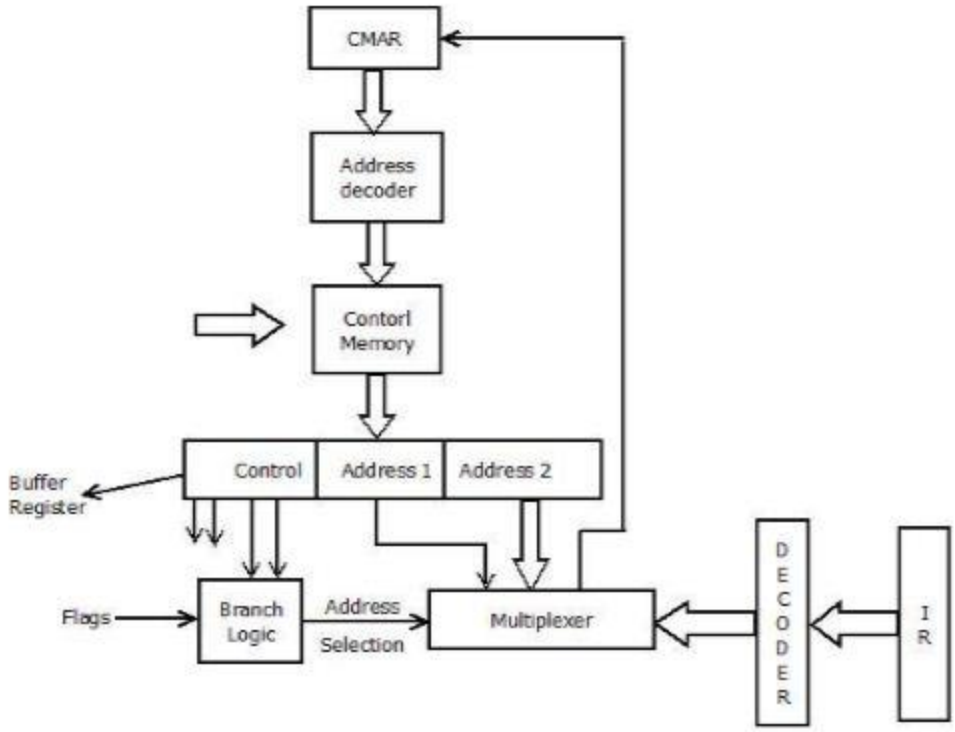
2. Branches to an address as specified by the address field of the micro instruction.
3. Branches to a given address if a specified status bit is equal to 1.
4. Transfer control to a new address as specified by an external source (Instruction Register).
5. Has a facility for subroutine calls and returns.

Depending on the current microinstruction condition flags, and the contents of the instruction register, a control memory address must be generated for the next micro instruction.

There are three general techniques based on the format of the address information in the microinstruction:

1. Two Address Field.
2. Single Address Field.
3. Variable Format

Two Address Field:



The simplest approach is to provide two address field in each microinstruction and multiplexer is provided to select:

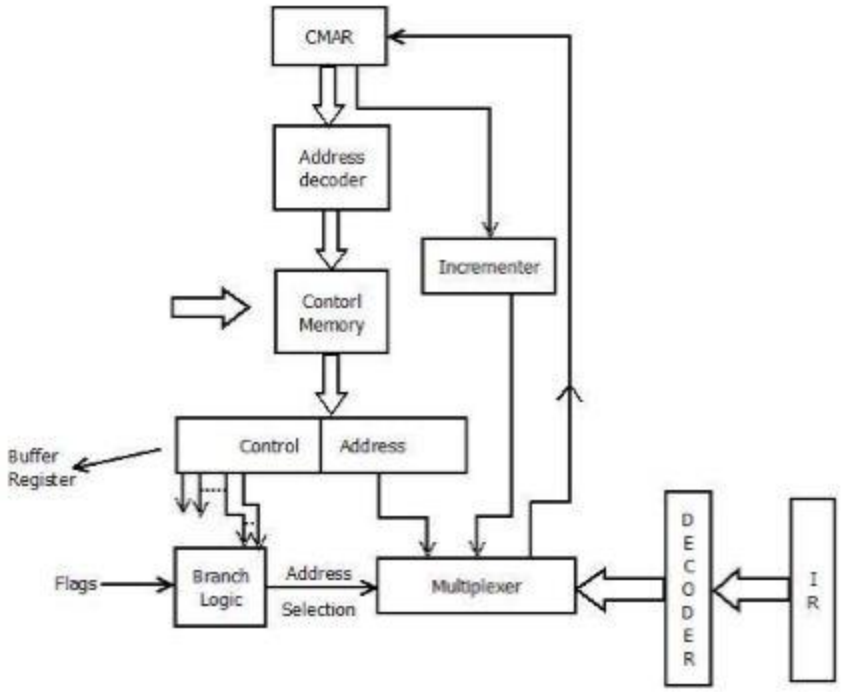
- Address from the second address field.
- Starting address based on the OPcode field in the current instruction.

The address selection signals are provided by a branch logic module whose input consists of control unit flags plus bits from the control partition of the micro instruction.

Single Address Field:

Two-address approach is simple but it requires more bits in the microinstruction. With a simpler approach, we can have a single address field in the micro instruction with the following options for the next address.

- Address Field.
- Based on OPcode in instruction register.
- Next Sequential Address.



The address selection signals determine which option is selected. This approach reduces the number of address field to one. In most cases (in case of sequential execution) the address field will not be used. Thus the microinstruction encoding does not efficiently utilize the entire microinstruction.

Variable Format:

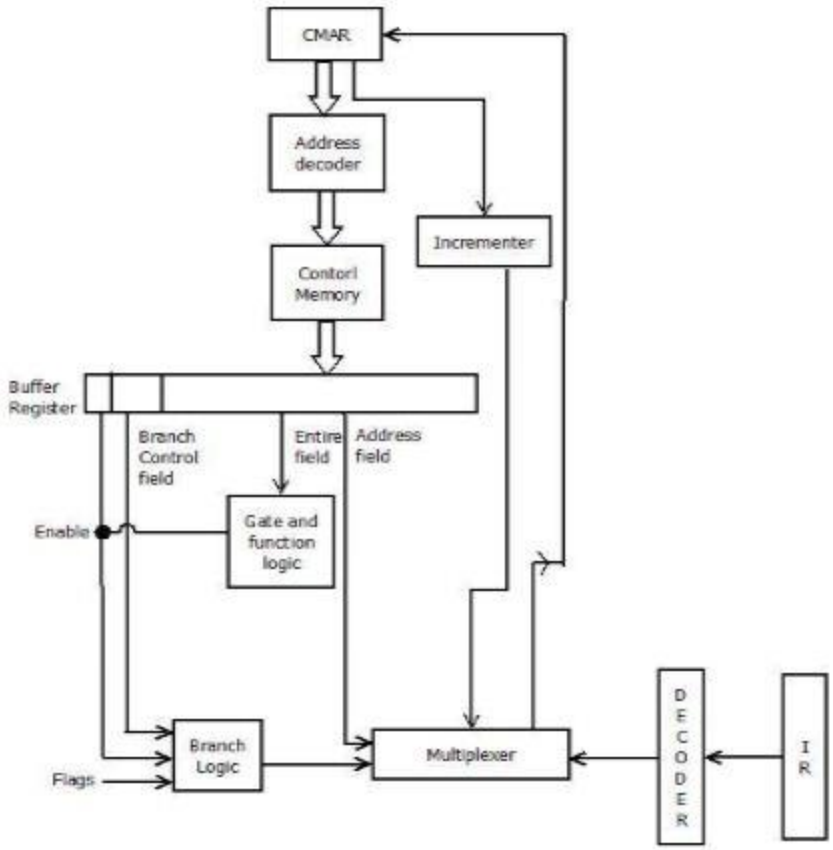


Fig. 3.41: Branch control logic, variable format

In this approach, there are two entirely different microinstruction formats. One bit designates which format is being used. In this first format, the remaining bits are used to activate control signals. In the second format, some bits drive the branch logic module, and the remaining bits provide the address. With the first format, the next address is either the next sequential address or an address derived from the instruction register. With the second format, either a conditional or unconditional branch is specified.