Microinstruction Format

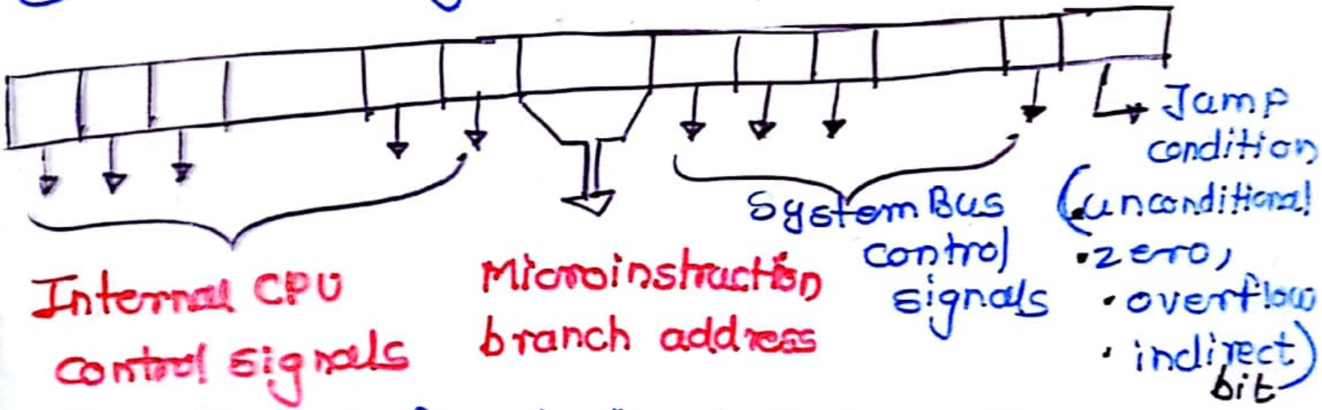
Microinstructions are commonly divided into two types

1. Horizontal microinstruction
2. Vertical microinstruction

Horizontal microinstruction:

In the horizontal microinstruction each bit of the microinstruction represents a control signal. A horizontal microinstruction has the following general attributes:

1. Long format
2. high degree of parallelism
3. little encoding of control information.


Jump condition (unconditional, zero, overflow, indirect bit)
Internal CPU control signals
Microinstruction branch address
System Bus control signals

The format of a horizontal microinstruction is as follows:

- There is one bit for each internal control line.
- There is one bit for each system bus control line.
- There is condition field for each condition for conditional branching.
- Address field that stores the address of the microinstruction to be exeecuted neset when a branch is taken

Execution of a horizontal microinstruction

⇓

Turn on all the control lines with bit value equal to 1

⇓

If the condition bit is false — If the condition bit is true

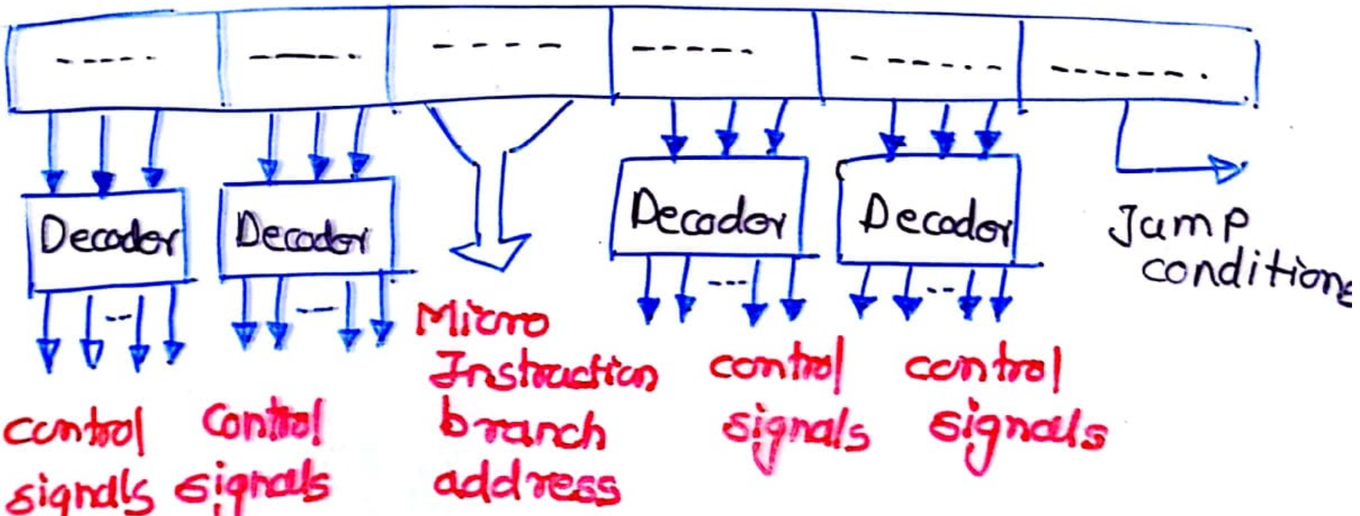
Execute the next microinstruction in sequence.

Next microinstruction to be executed is indicated in the address field.

Vertical Microinstructions

A vertical microinstruction allows encoding of control information. Vertical microinstructions are characterized by

- Short formats
- Limited ability to express parallel micro-operations.
- considerable encoding of control information.



Vertical microinstruction using decoders

In vertical microinstructions many similar control signals can be decoded into few microinstruction bits. For example for 32 ALU operations, only 5 encoded bits will be required in vertical microinstruction. However these encoded bits need to be passed from respective decoders to get the individual control signals.

In general, a horizontal control unit is faster yet requires wide instruction words whereas, vertical control unit although requires decoders however, are shorter in length.

Grouping of Control Signals

It is possible to perform many operations in parallel at machine level. A typical microinstruction format has to take advantage of the inherent parallelism at the operation level.

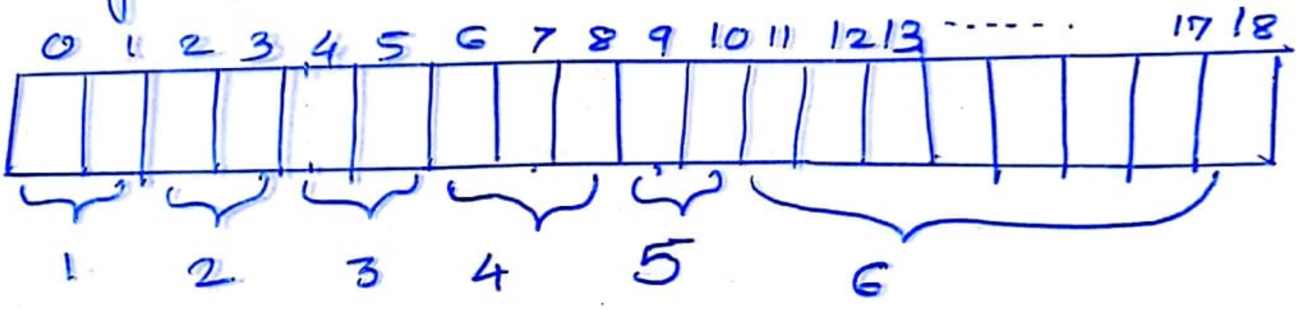
A microinstruction is divided into k disjoint control fields. Each control field handles a limited set of microoperations, any one of which can be performed simultaneously with the microoperations specified by the remaining control fields. A control field often specifies control-line values for a single device such as an adder, a register, or a bus.

The design of an encoded microinstruction format with distinct groups can be stated as.

- Organize the format into independent fields. Each field depicts a set of actions (control signals) control signals from different fields can be activated simultaneously.

- Actions from fields are manually exclusive. only one action specified for a given field could occur at a time.

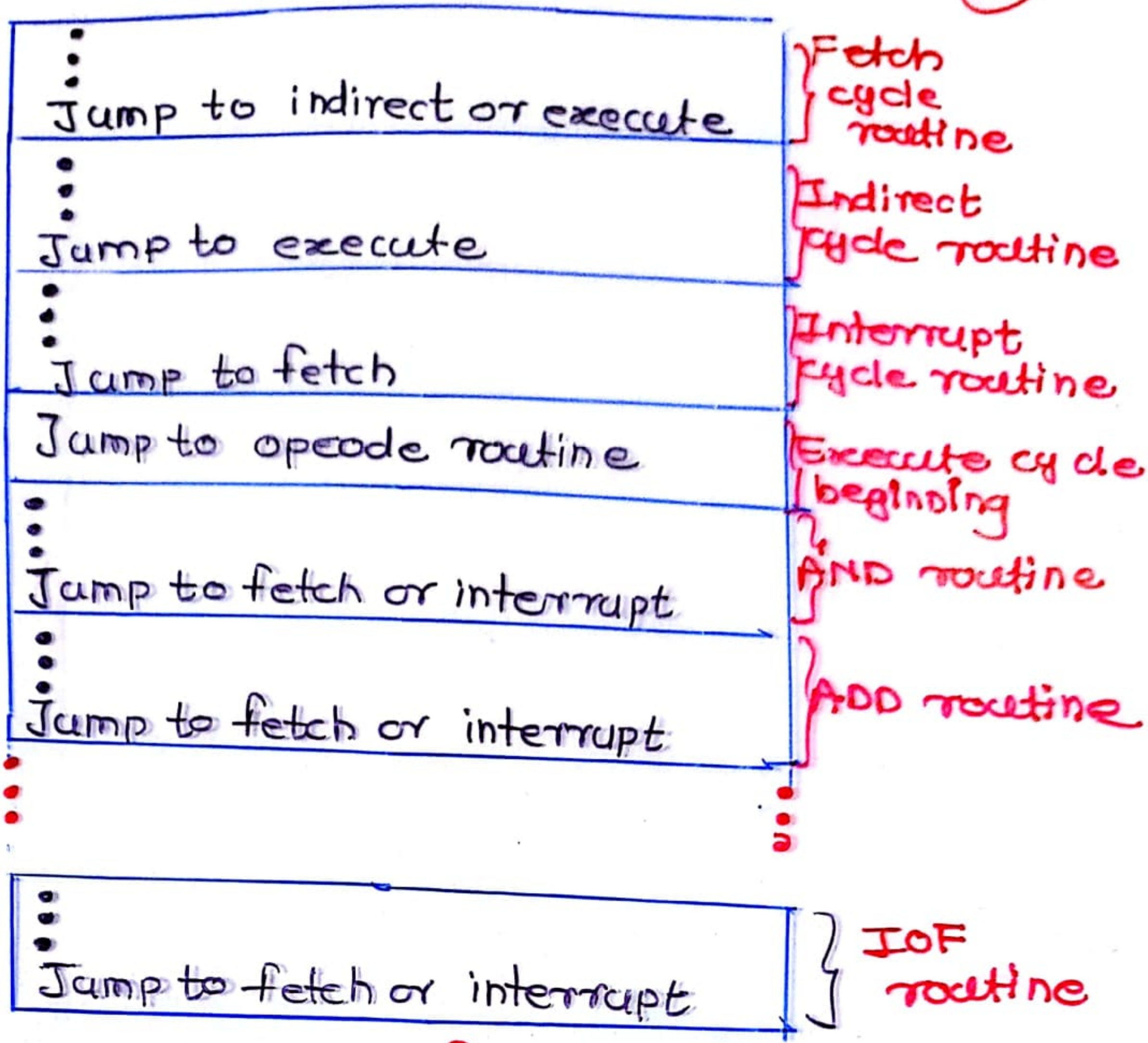
For a hypothetical machine, control signals can be grouped as shown below.



Field definition:

① Register transfer ② Memory operation
③ Sequencing operation ④ ALU operation
⑤ Register selection ⑥ Constant

Next figure shows how these control words or microinstructions could be arranged in a control memory. The microinstructions in each routine are to be executed sequentially. Each routine ends with a branch or jump instruction indicating where to go next. There is a special execute cycle routine whose only purpose is to signify that one of the machine instruction routines (AND, ADD and so on) is to be executed next, depending on the current opcode.



Organization of control memory

Microinstruction Sequencing

A microprogram control unit can be viewed as consisting of two parts.

① The control memory that stores the microinstructio

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 microinstruction.

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)

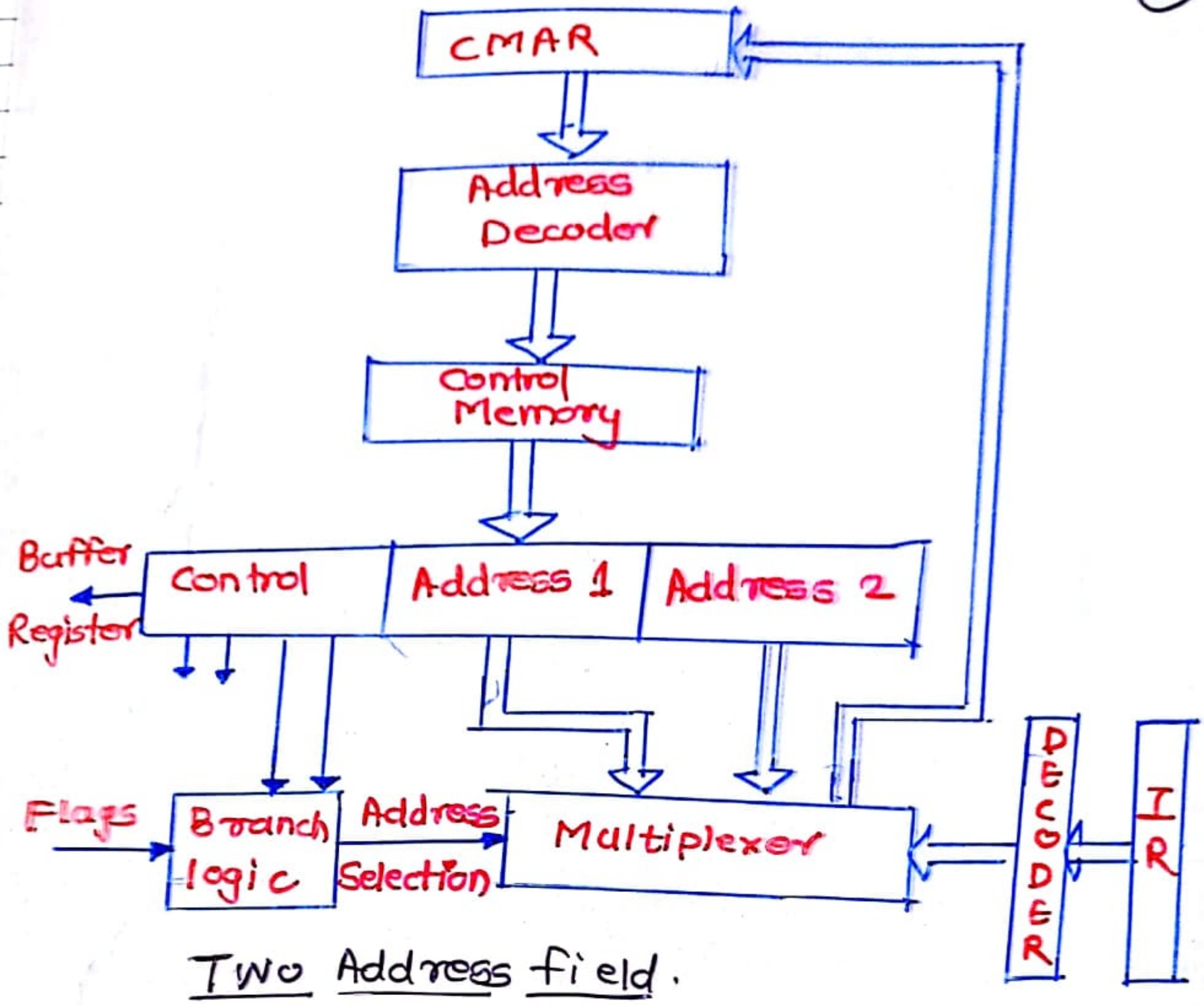
5) Has a facility for subroutine calls and returns.

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.

- Two address field
- Single address fied
- Variable format.

① Two address field :



Two Address field.

The simplest approach is to provide two address field in each microinstruction. A multiplexer is provided to select

① Address from the first address field.

② Address from the second address field.

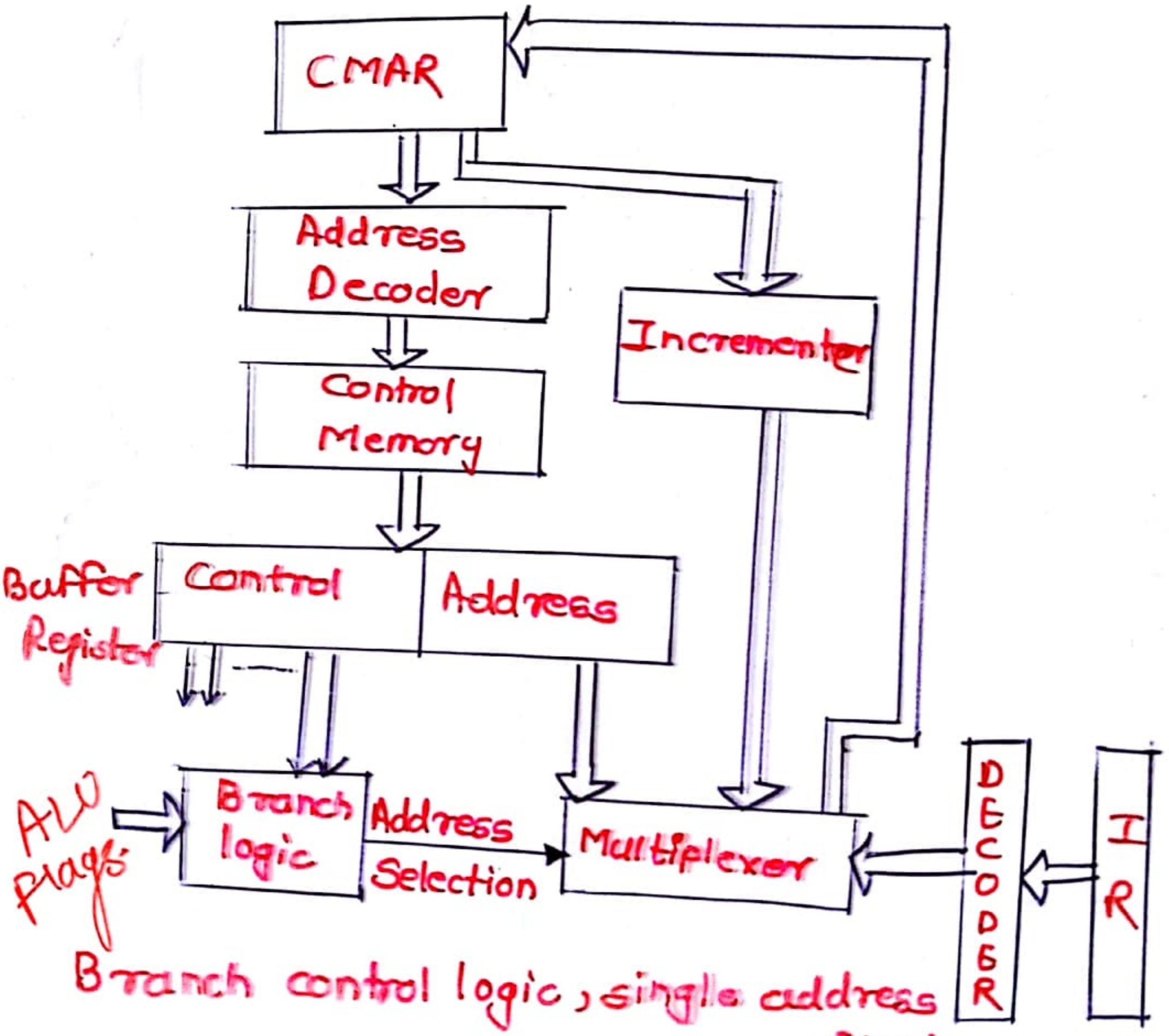
③ starting address based on the OPcode field of the current instruction.

The address selection signals are provided by a branch logic module whose input consists of control unit flags plus bits from the ~~second~~ control portion of the microinstruction.

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 microinstruction with the following option. for the next address.

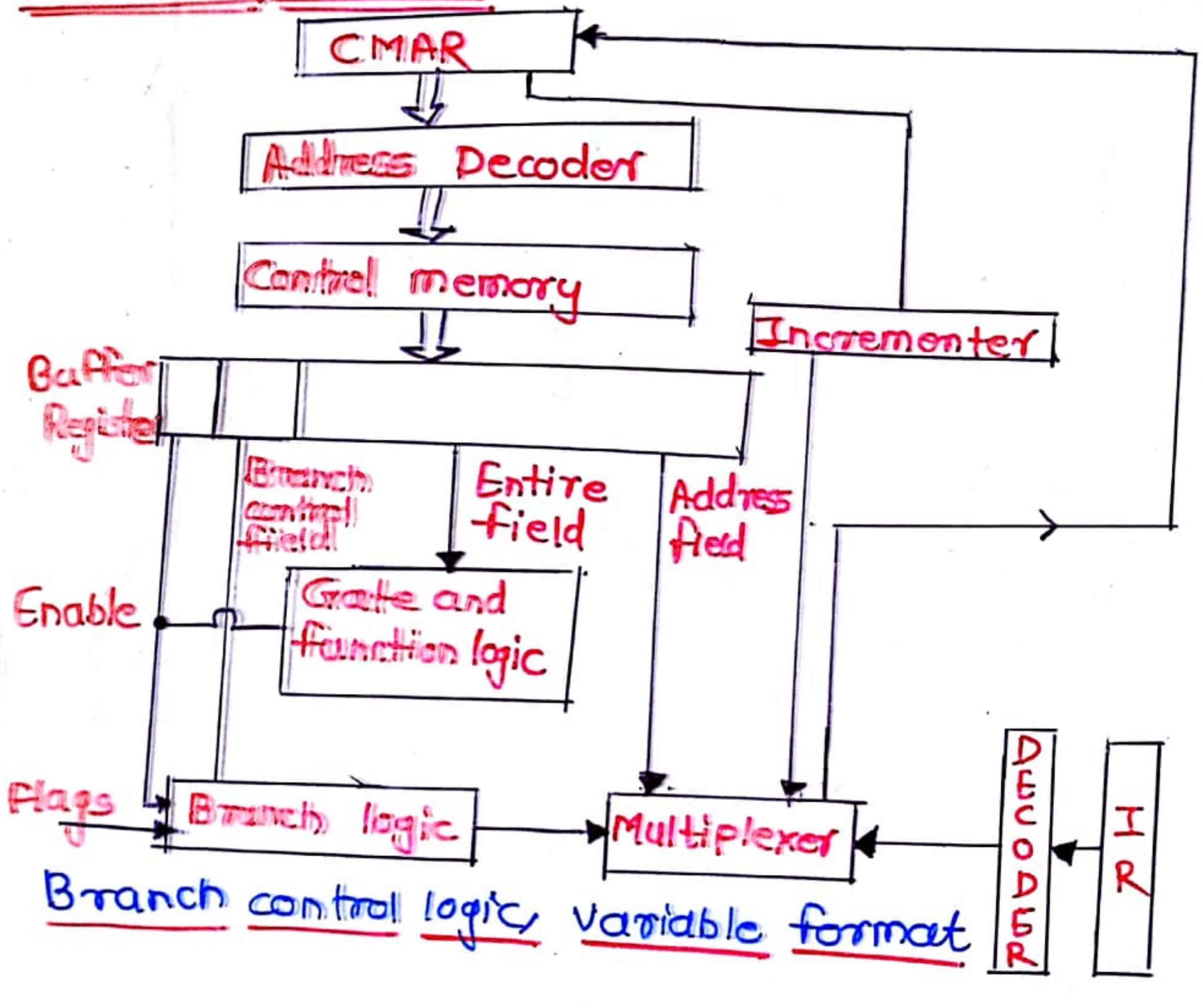
- Address field
- Based on OPcode in the instruction register.
- Next sequential address.



Branch control logic, single address field

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.



Branch control logic, variable format

In this approach, there are two entirely different microinstruction formats. one bit designates which format is being used. In the 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.