

COA

Addressing Modes

The way in which the operand of an instruction is specified.

Sequential control flow addressing modes:-

⇒ Register based :-

i) Register direct :- operand is placed in one of the 8 or 16 bit GPRs.

⇒ One register reference required to access the data.

Example: MOV AX, CX



⇒ Memory Based :

- i) Implied / Implicit : Data is available in opcode itself, so no EA.
example: CMA (complement Accumulator)
& all **zero** address instructions.
- ii) Immediate : Data is present in the address field of instruction.



Example: MOV AL, 35H

limitation in this mode is: range of constants are limited by size of address field.

* this mode is used to access constants or to initialize registers to constant value.

* Designed like one address instruction.

- iii) Direct / Absolute : Effective address of data is part of the instruction.

⇒ One memory reference is needed to access the data.

* Used to access static variables.

Example: - MOV R1, [1000]

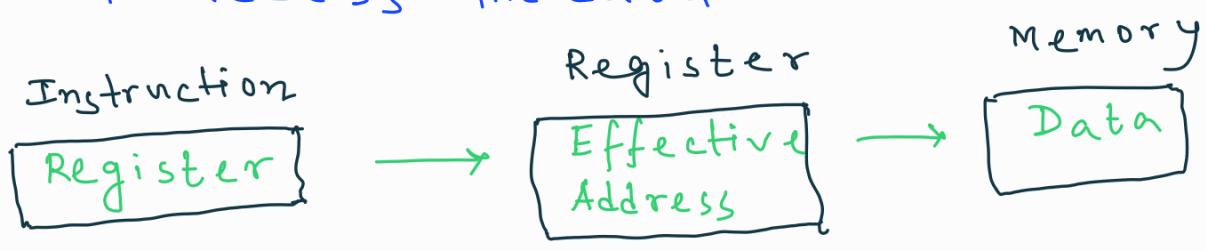
Instruction



Memory



- Array as parameter
- iv) Indirect : Address field of instruction containing the address of effective address.
 - ⇒ two references required to access data.
 (1st ref. to access effective address
 + 2nd ref. to access data)
 - ⇒ Register indirect : Operand's offset is placed in any of the registers BX, BP, SI, DI as specified in the instruction. effective address of the data is in base register or an index register that is specified by the instruction.
 - ⇒ Two register reference required to access the data.



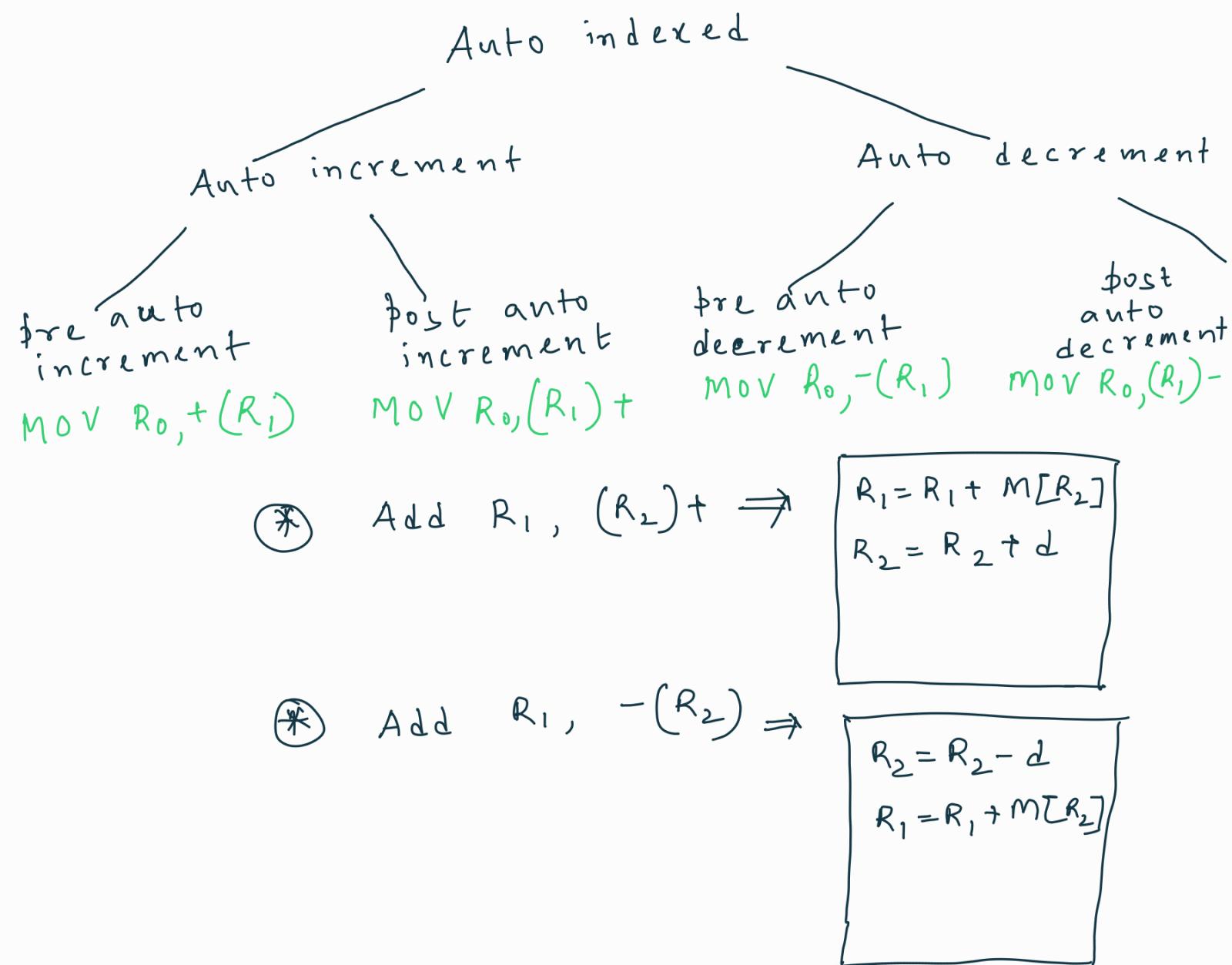
example: $MOV AX, [BX]$

- ⇒ Memory indirect : Effective address stored in memory.
- ⇒ Two memory references needed to access the data.

v) Auto indexed : Effective address of the operand is the content of register specified in the instruction. After accessing the operand the contents of this register is automatically incremented (or decremented) to point to the next consecutive memory location (R_1)⁺.

\Rightarrow One register ref. + one memory ref.
+ one ALU ref. required to access the data.

* Useful for stepping through arrays in loop.



Auto increment or decrement modes are useful for implementing LIFO data structure.
⇒ base address is required which is stored in base register.

$$EA = \text{Base Register} + \text{Step Size}$$

vi) Based Indexed: Operand's offset is the sum of the content of a base reg. BX or BP and index reg. SI or DI.

Example: ADD AX, [BX + SI]

Register that holds the index value is
Register indexed addressing mode

* indexed : MOV AX, [SI + 05]

Transfer of control flow Addressing mode:

⇒ Relative / PC relative: — Used to access instruction within the segment (intra segment) so that only offset address is required.
↳ present in the address field of instruction.

$$EA = \text{Base Address} + \text{Offset Address}$$

$$PC = PC + IR[\text{add. field}]$$

- * position dependent
- * Relocation at runtime.

⇒ Base Register Addressing Mode:

Used to access instruction segments (inter segment) so that base address & offset is required.

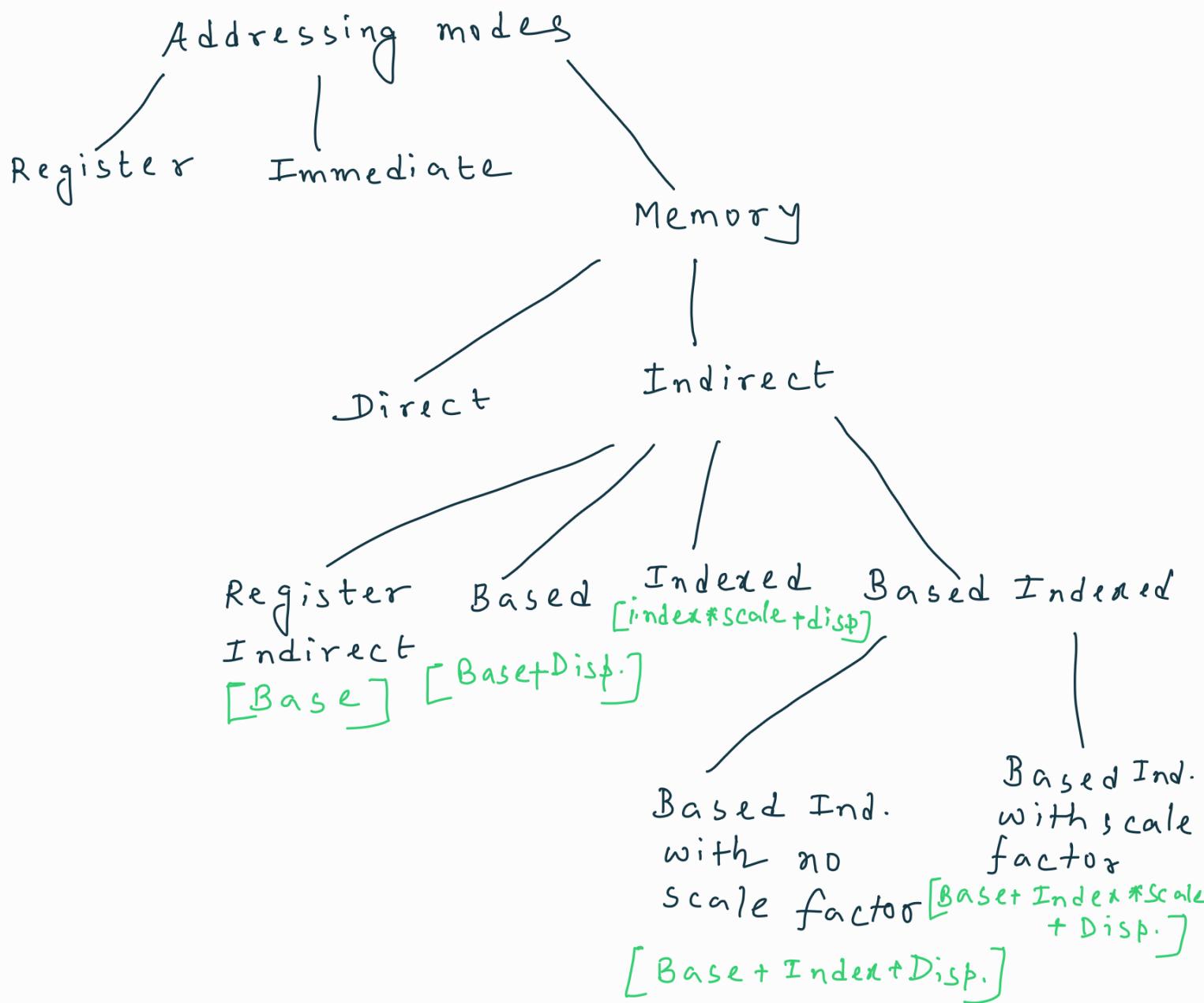
implemented using
mnemonics: →
JUMP, BRANCH, SKIP

Transfer of control
instruction

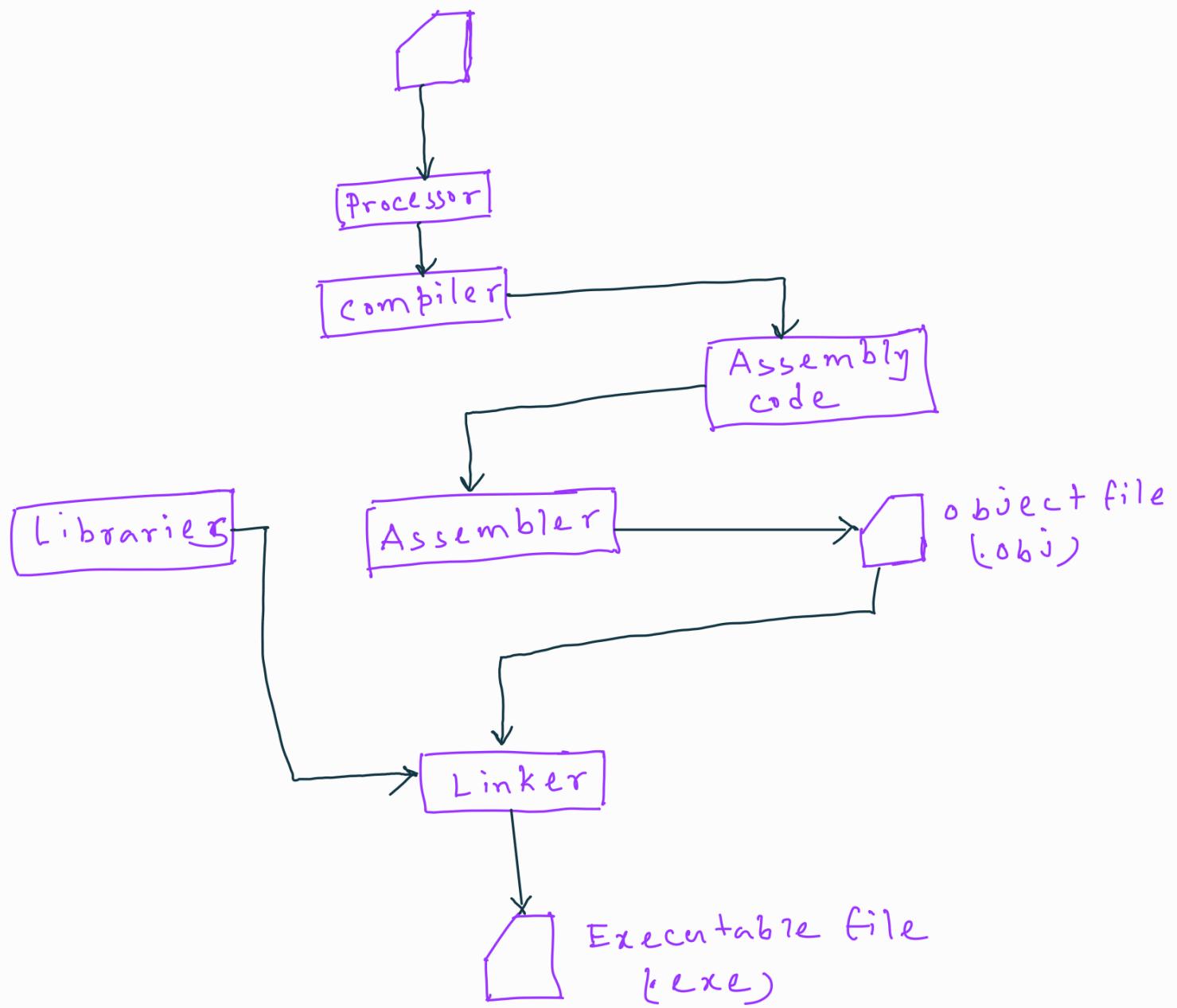
Conditional
(IF, SWITCH, FOR...)

Unconditional
(HALT)

Pentium Addressing modes (32 bit Add.)



Steps for generating executable file



Control unit

CU is the part of Computer's CPU which directs the operation of the processor.

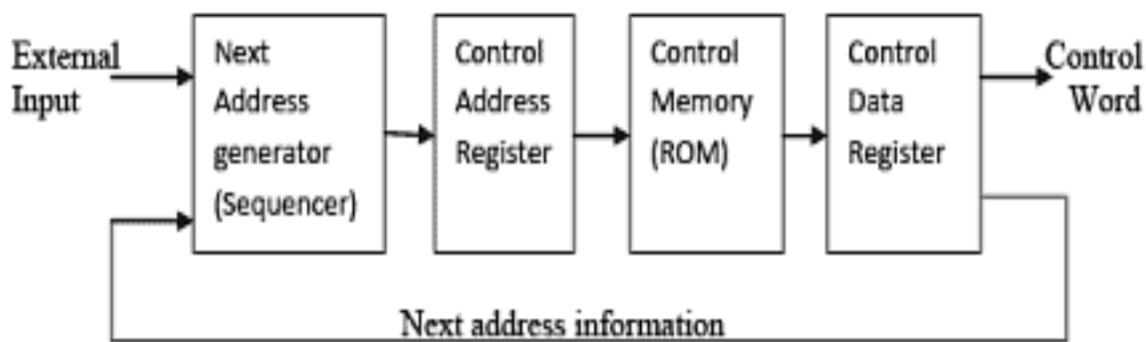
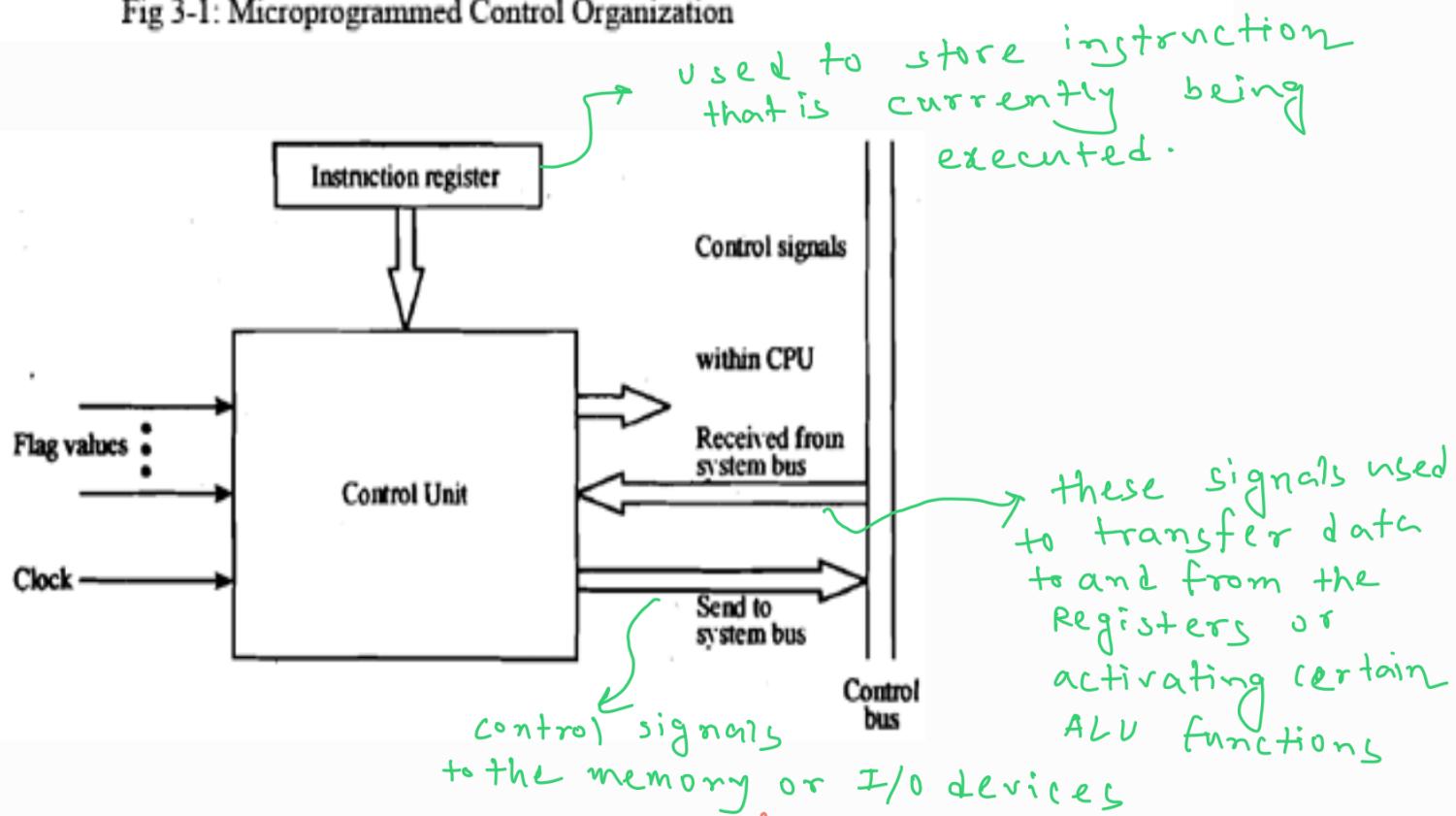
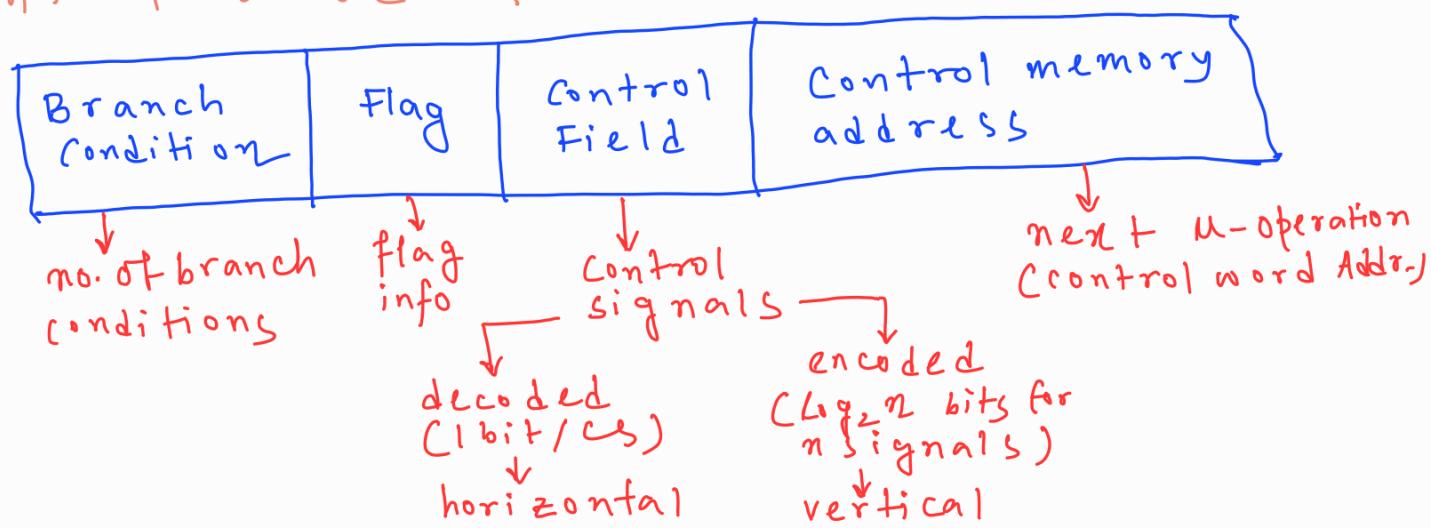
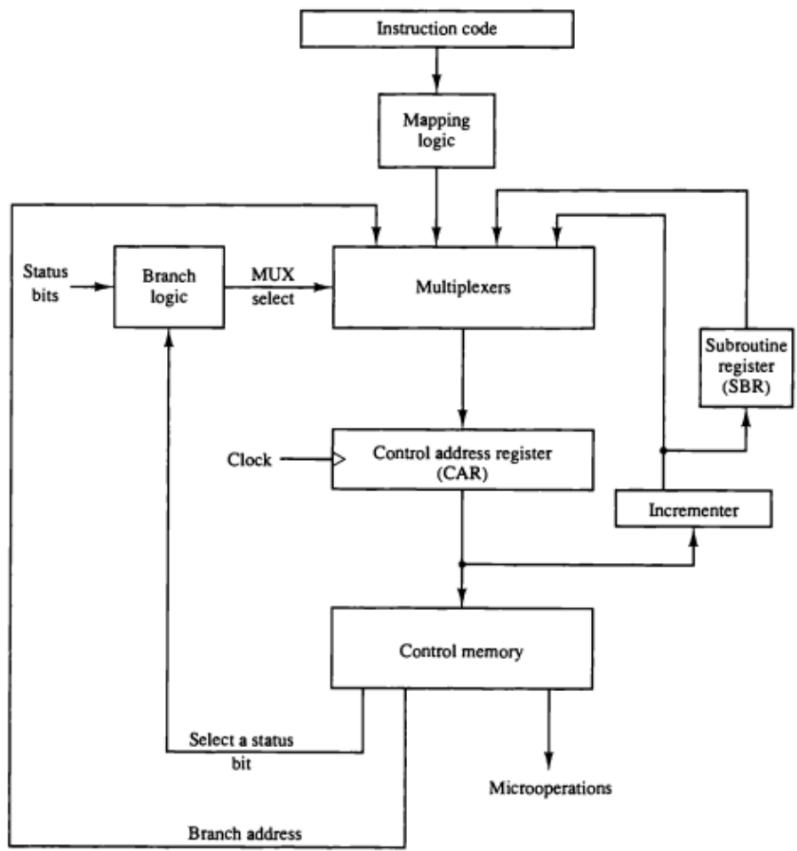


Fig 3-1: Microprogrammed Control Organization



Control word format :-





* A bus which is used to provide communication between the major components of a computer is called system bus.

Micro instruction Format

3	3	3	2	2	7
F1	F2	F3	CD	BR	AD

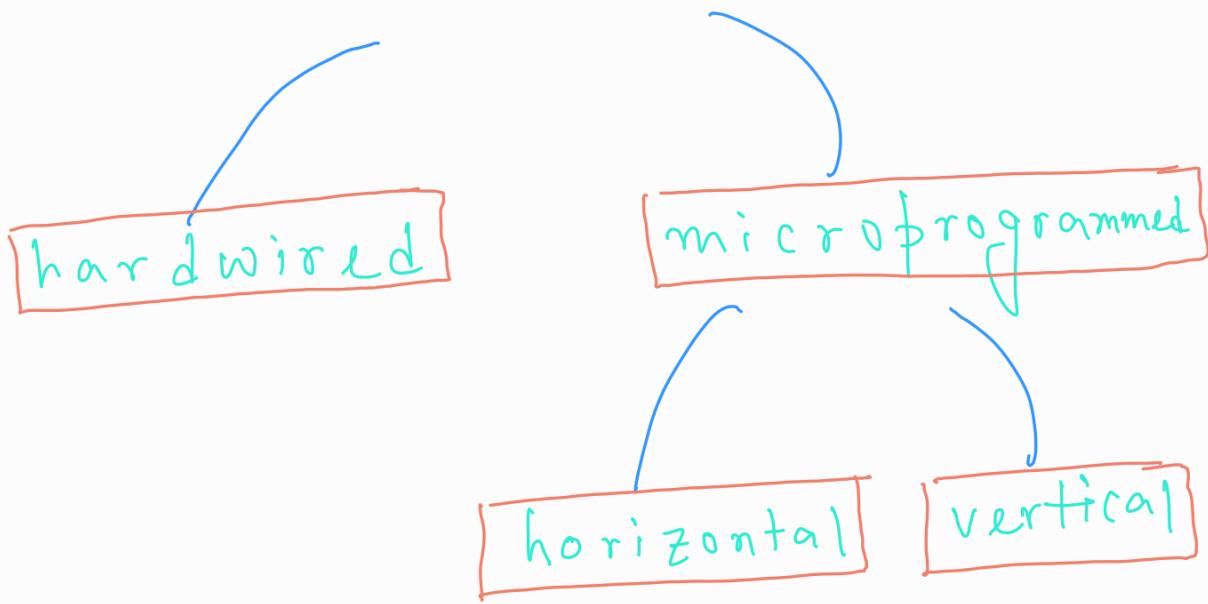
F1, F2, F3: Microoperation fields

CD: Condition for branching

BR: Branch field

AD: Address field

Types of control unit



Speed comparison :—

hardwired > horizontal > vertical

flexibility comparison :—

vertical > horizontal > hardwired

Control unit

Horizontal Microprogrammed control unit :-

The control signals are represented in the decoded binary format i.e. 1 bit/CS.

- ⇒ to represent n control signal n bits needed
- ⇒ Longer control word
- ⇒ used in parallel processing application.
- ⇒ higher degree of parallelism (if degree is n then n control signals can be enabled at a time)
- ⇒ no additional hardware reqd. (decoder), means faster than vertical microprogrammed CU.
- ⇒ Flexible than hardwired.

Vertical microprogrammed control unit

- ⇒ Supports shorter control word
- ⇒ low degree of parallelism
(either 0 or 1)
- ⇒ additional hardware (decoder) required
- ⇒ slower than horizontal microprogrammed control unit.
- ⇒ makes more use of ROM encoding to reduce length of control word.

