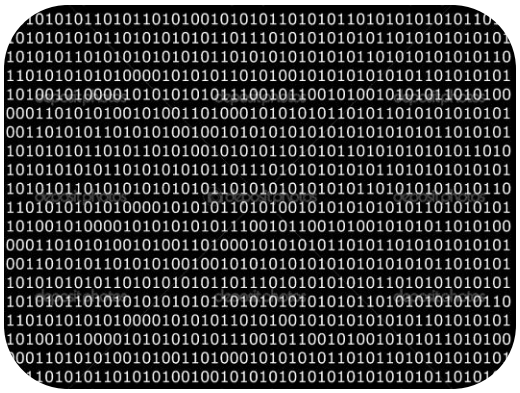
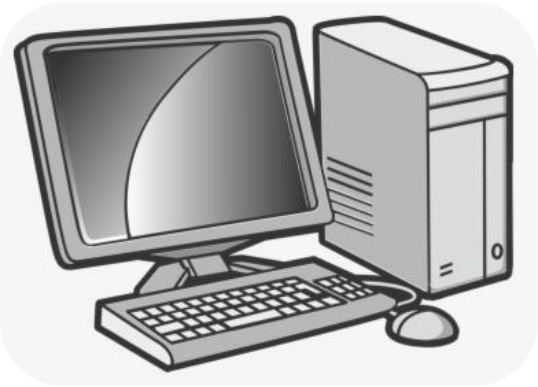
AMIRAJ

COLLEGE OF ENGINEERING & TECHNOLOGY

# Unit-4

# Micro programmed Control Organization:





| | | |
|---|---|---|
| **Subject:- COA**<br>**Code:-3140707** | **Prepared by:**<br>**Asst. Prof. S Y JOSHI**<br>**(CSE Department,ACET)** | AMIRAJ<br>COLLEGE OF ENGINEERING & TECHNOLOGY |

# Topics to be covered

- Control Memory,
- Address sequencing,
- Micro program example,
- Design of Control Unit

# Control Memory

- The function of the control unit in a digital computer is to initiate sequences of micro operations.
- The control unit initiates a series of sequential steps of micro operations.
- The control variables at any given time can be represented by a string of 1's and 0's called a *control word*.
- As such, control words can be programmed to perform various operations on the components of the system.
- A control unit whose binary control variables are stored in memory is called **micro programmed control unit**.
- A memory that is a part of control unit is called a control memory.

# Address sequencing

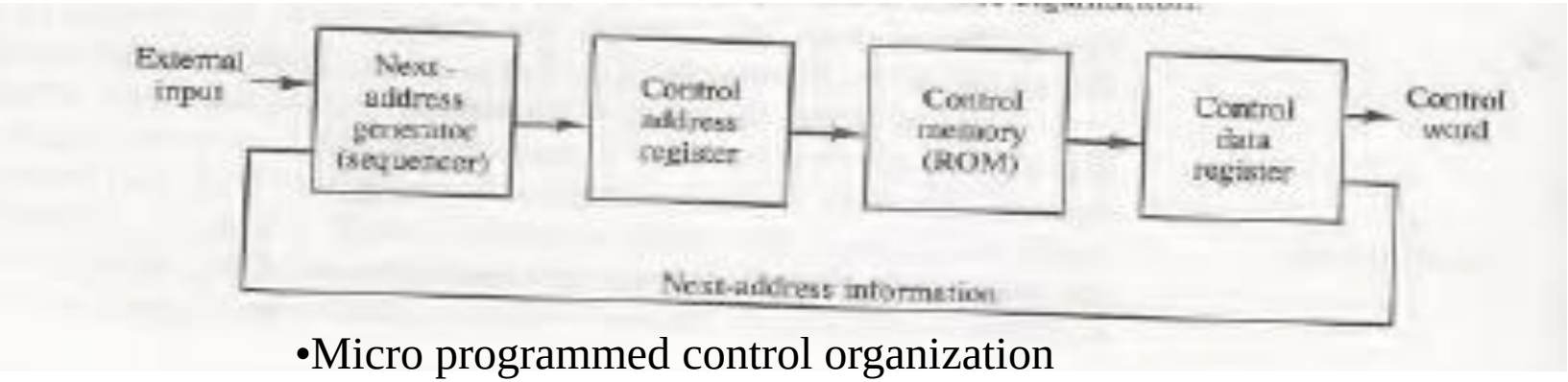
•Control memory address register specifies the address of the microinstruction, and the control data register holds the microinstruction read from memory.

•The microinstruction contains a control word that specifies one or more micro operations for the data processor.

•The next address generator is sometimes called a micro program sequencer, as it determines the address sequence that is read from control memory.

•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.

•It allows the execution of the micro operations specified by the control word simultaneously with the generation of the next microinstruction.

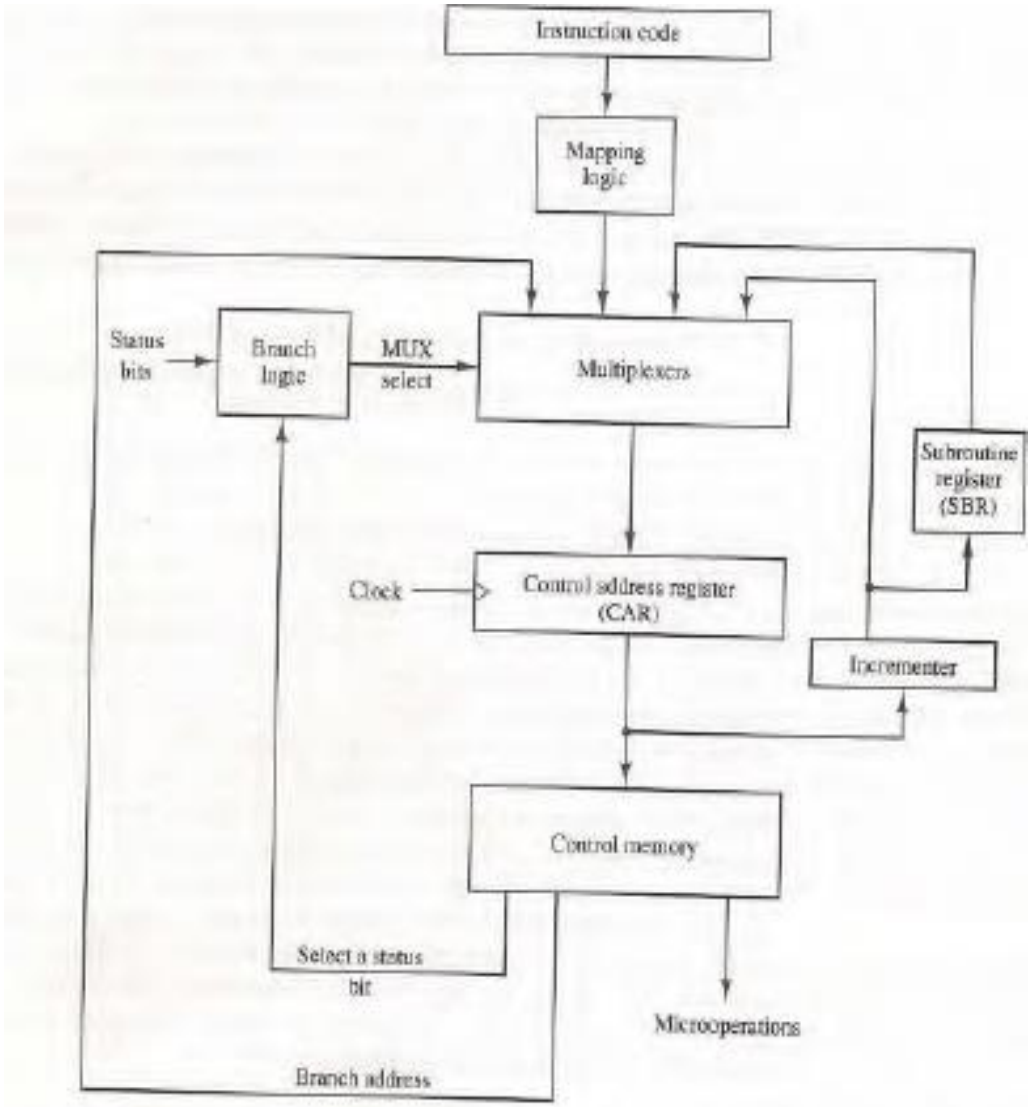



•Micro programmed control organization

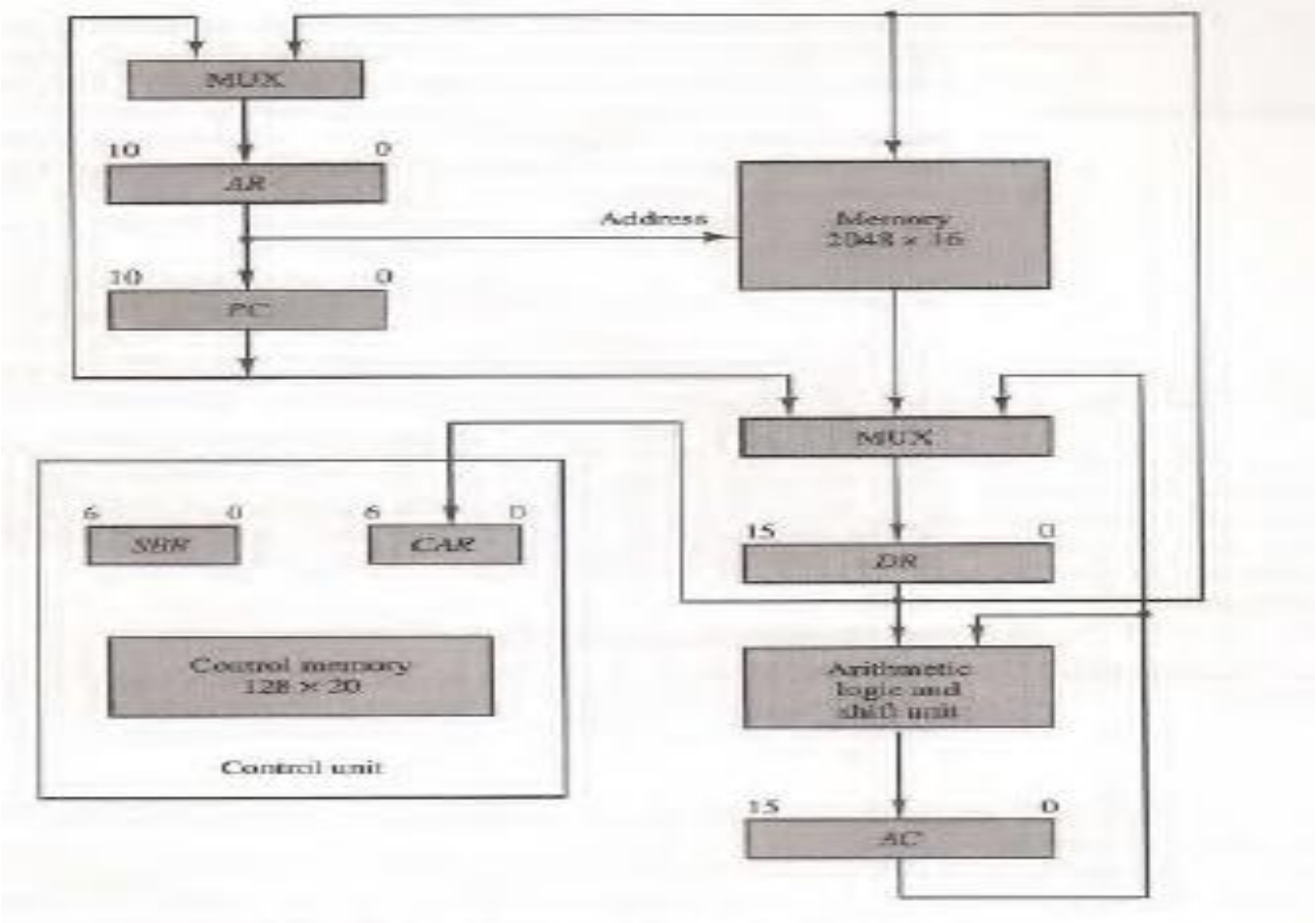
# Address sequencing

The address sequencing capabilities required in a control memory are:

1. Incrementing of the control address register.
2. Unconditional branch or conditional branch, depending on status 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.



# Micro program example

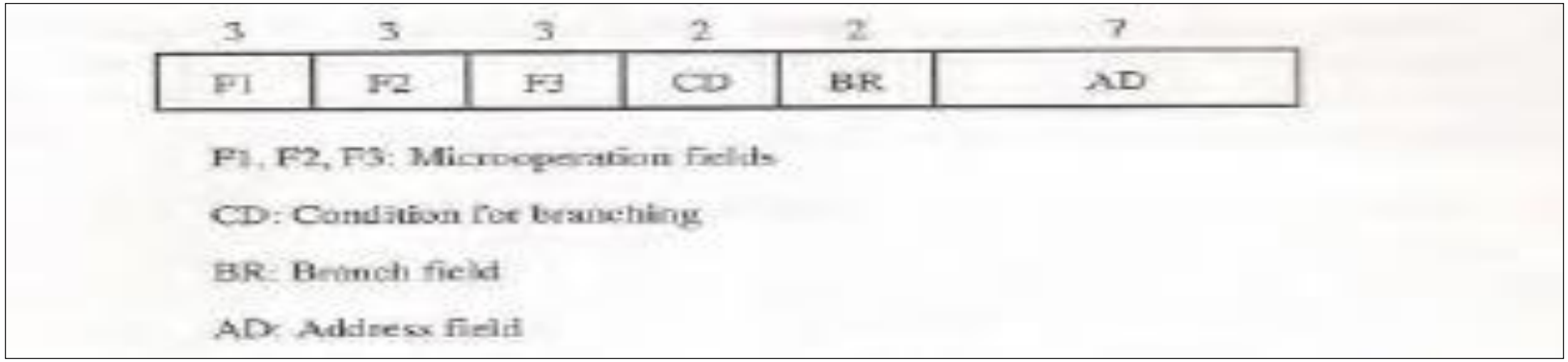



Computer hardware configuration.



AMIRAJ
COLLEGE OF ENGINEERING & TECHNOLOGY

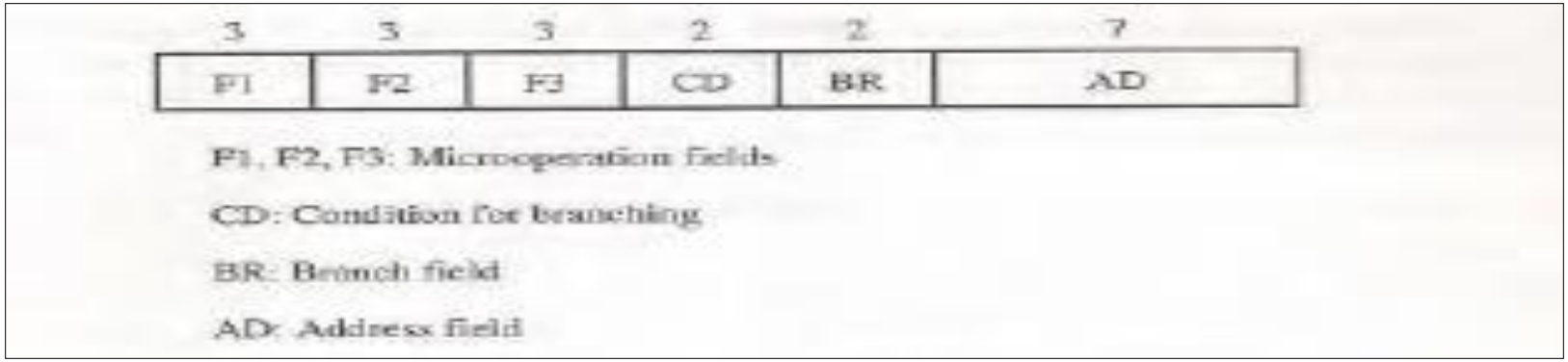
# Micro program example




•Microinstruction code format (20 bits).

- The micro operations are subdivided into three fields of three bits each.
- The three bits in each field are encoded to specify seven distinct micro operations
- The CD (condition) field consists of two bits which are encoded to specify four status bit conditions
- The BR (branch) field consists of two bits. It is used, in conjunction with the address field AD

# Micro program example




•Microinstruction code format (20 bits).

- The micro operations are subdivided into three fields of three bits each.
- The three bits in each field are encoded to specify seven distinct micro operations
- The CD (condition) field consists of two bits which are encoded to specify four status bit conditions
- The BR (branch) field consists of two bits. It is used, in conjunction with the address field AD

# Micro program example

The fields specify the following information.

1. The label field may be empty or it may specify a symbolic address. A label is terminated with a colon (:).

2. The microoperations field consists of one, two, or three symbols, separated by commas, . There may be no more than one symbol from each F field. The NOP symbol is used when the microinstruction has no micro operations. This will be translated by the assembler to nine zeros.

3. The CD field has one of the letters U, I, S, or Z.

4. The BR field contains one of the four symbols defined in previous Table

5. The AD field specifies a value for the address field of the microinstruction in one of three possible ways:
a. With a symbolic address, which must also appear as a label
b. With the symbol NEXT to designate the next address in sequence
c. When the BR field contains a RET or MAP symbol, the AD field is left empty and is converted to seven zeros by the assembler.

# Micro program example

| F1 | Microoperation | Symbol |
|---|---|---|
| 000 | None | NOP |
| 001 | $AC \leftarrow AC + DR$ | ADD |
| 010 | $AC \leftarrow 0$ | CLRAC |
| 011 | $AC \leftarrow AC + 1$ | INCAC |
| 100 | $AC \leftarrow DR$ | DRTAC |
| 101 | $AR \leftarrow DR(0\text{–}10)$ | DRTAR |
| 110 | $AR \leftarrow PC$ | PCTAR |
| 111 | $M[AR] \leftarrow DR$ | WRITE |

| F2 | Microoperation | Symbol |
|---|---|---|
| 000 | None | NOP |
| 001 | $AC \leftarrow AC - DR$ | SUB |
| 010 | $AC \leftarrow AC \vee DR$ | OR |
| 011 | $AC \leftarrow AC \wedge DR$ | AND |
| 100 | $DR \leftarrow M[AR]$ | READ |
| 101 | $DR \leftarrow AC$ | ACTDR |
| 110 | $DR \leftarrow DR + 1$ | INCDR |
| 111 | $DR(0\text{-}10) \leftarrow PC$ | PCTDR |

| F3 | Microoperation | Symbol |
|---|---|---|
| 000 | None | NOP |
| 001 | $AC \leftarrow AC \oplus DR$ | XOR |
| 010 | $AC \leftarrow \overline{AC}$ | COM |
| 011 | $AC \leftarrow$ shl $AC$ | SHL |
| 100 | $AC \leftarrow$ shr $AC$ | SHR |
| 101 | $PC \leftarrow PC + 1$ | INCPC |
| 110 | $PC \leftarrow AR$ | ARTPC |
| 111 | Reserved | |

| CD | Condition | Symbol | Comments |
|---|---|---|---|
| 00 | Always = 1 | U | Unconditional branch |
| 01 | $DR(15)$ | I | Indirect address bit |
| 10 | $AC(15)$ | S | Sign bit of $AC$ |
| 11 | $AC = 0$ | Z | Zero value in $AC$ |

| BR | Symbol | Function |
|---|---|---|
| 00 | JMP | $CAR \leftarrow AD$ if condition = 1<br>$CAR \leftarrow CAR + 1$ if condition = 0 |
| 01 | CALL | $CAR \leftarrow AD$, $SBR \leftarrow CAR + 1$ if condition = 1<br>$CAR \leftarrow CAR + 1$ if condition = 0 |
| 10 | RET | $CAR \leftarrow SBR$ (Return from subroutine) |
| 11 | MAP | $CAR(2\text{–}5) \leftarrow DR(11\text{–}14)$, $CAR(0,1,6) \leftarrow 0$ |

# Design of Control Unit

- The bits of the microinstruction are usually divided into fields, with each field defining a distinct, separate function.
- The various fields encountered in instruction formats provide control bits to initiate microoperations in the system, special bits to specify the way that the next address is to be evaluated, and an address field for branching.
- The number of control bits that initiate microoperations can be reduced by grouping mutually exclusive variables into fields and encoding the k bits in each field to provide 2^k micro operations

# Design of Control Unit

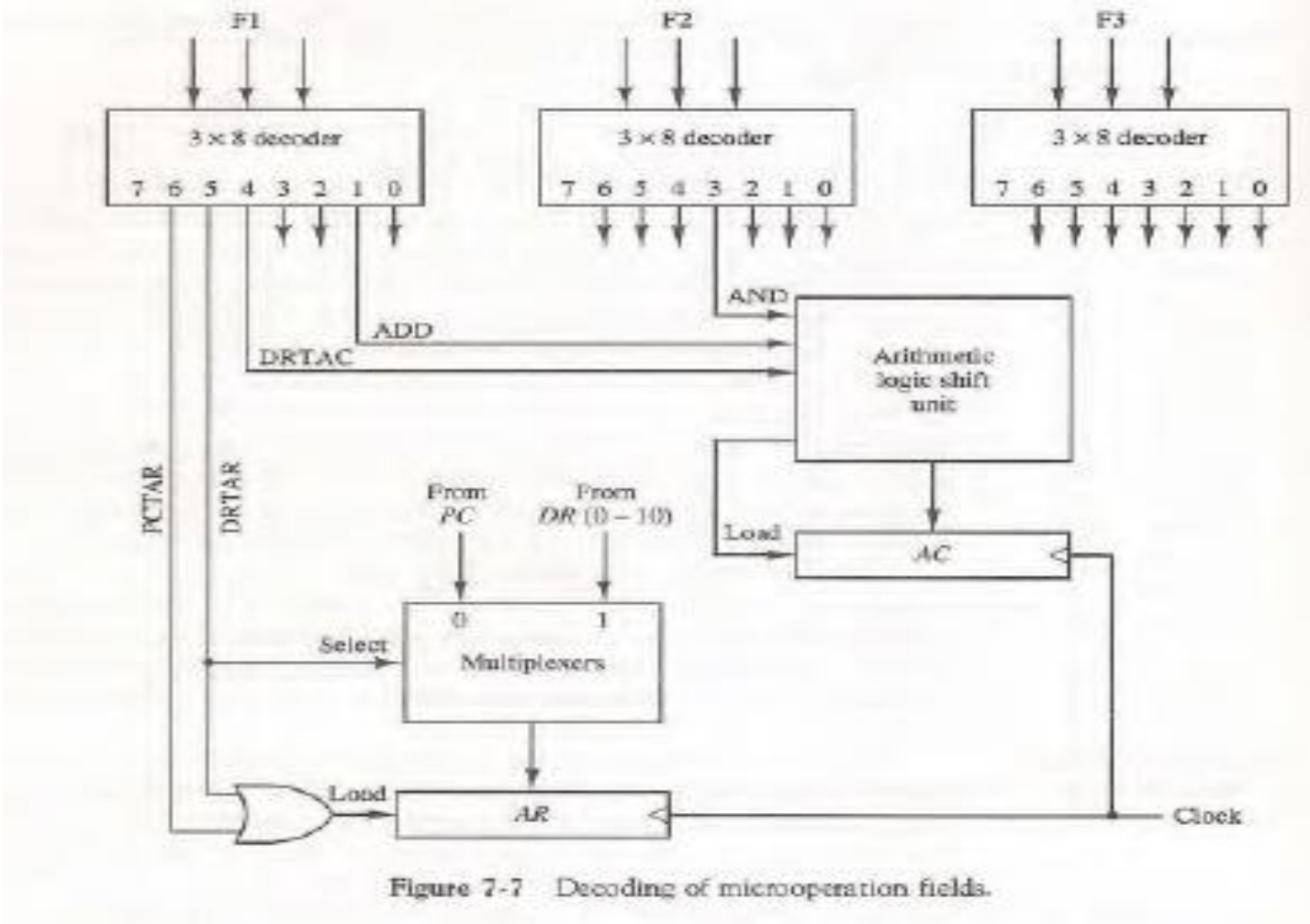



Figure 7-7 Decoding of microoperation fields.



AMIRAJ COLLEGE OF ENGINEERING & TECHNOLOGY

# Design of Control Unit

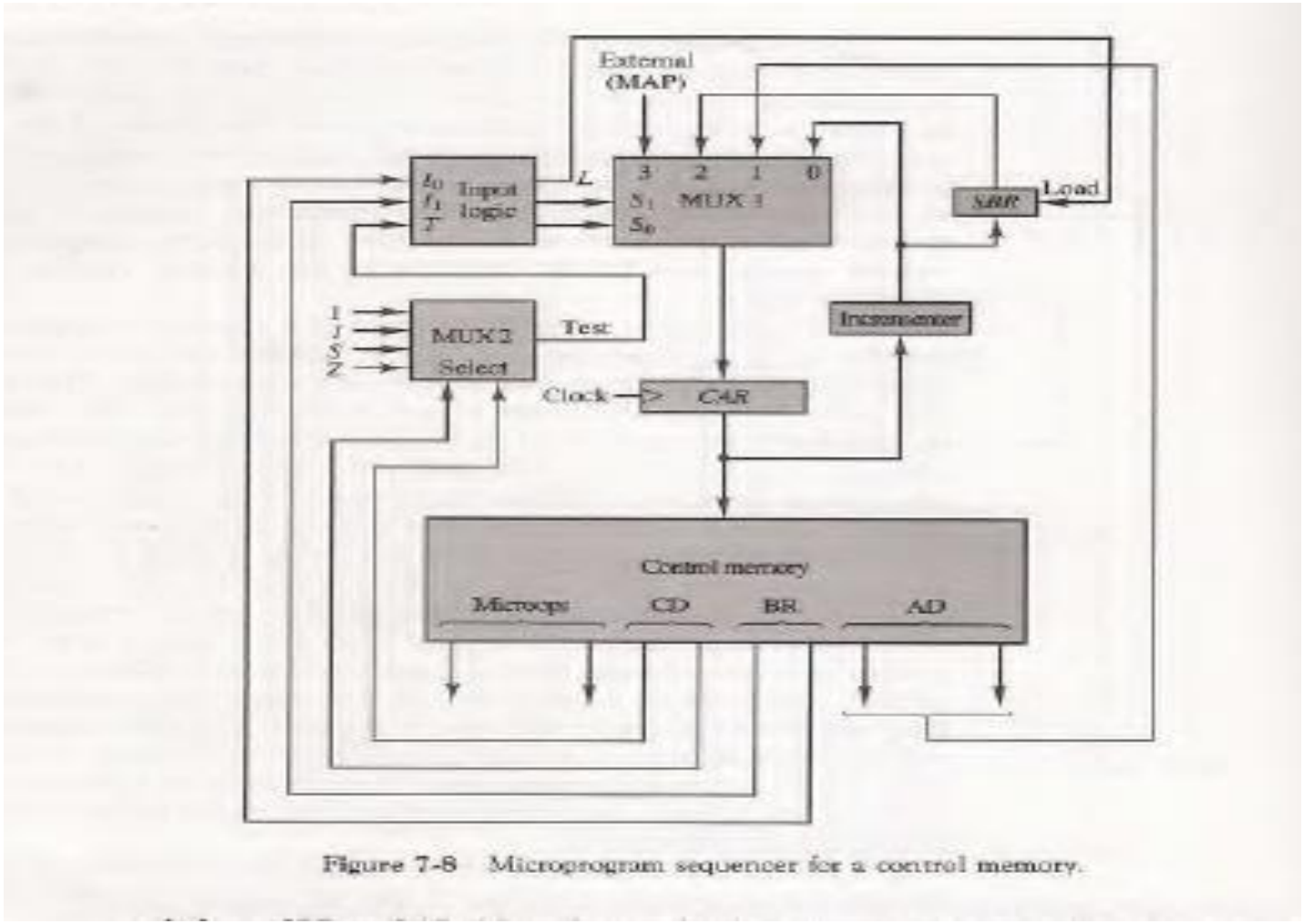



Figure 7-8 Microprogram sequencer for a control memory.



AMIRAJ
COLLEGE OF ENGINEERING & TECHNOLOGY

# References

- **Images , descriptive Tables , from Computer System Architecture, Morris Mano, 3rd edition Prentice Hall**

- **Note: These pdf/ppt notes are for purpose of teaching aids to classroom/online sessions study, and in no case imply for GTU syllabus or GTU exam. For GTU syllabus or exam related preparation, one may, however will need to attend college/online lectures and refer books given by GTU in their official syllabus.**