Unit-3

Control Unit Organization

Control Unit

- All the control unit does is generate a set of control signals
- Each control signal is on or off
- Represent each control signal by a bit
- Have a control word for each micro-operation
- While doing instruction fetch, the instruction consisted of state information and in each state the signals that are necessary to setup the data path that information is available from the micro word

RUNGTA

- Have a sequence of control words for each machine code instruction
- Add an address to specify the next micro-instruction, depending on conditions.
- A control unit undertakes the following responsibilities:
  - Instruction Interpretation
  - Instruction Sequencing

Instruction Interpretation

- In the interpretation phase, the control unit reads instructions from the memory, using the PC as a pointer.
- Then it identifies the instruction type, obtains the necessary operands, and routes them to the appropriate functional units of the execution unit.
- Then the necessary signals are issued to the execution unit to perform the desired operation, and the results are routed to the specified destination.

Instruction Sequencing

- In the sequencing phase, the control unit decides the address of the next instruction to be executed and loads it into the PC.

Functional Block Diagram of Control Unit

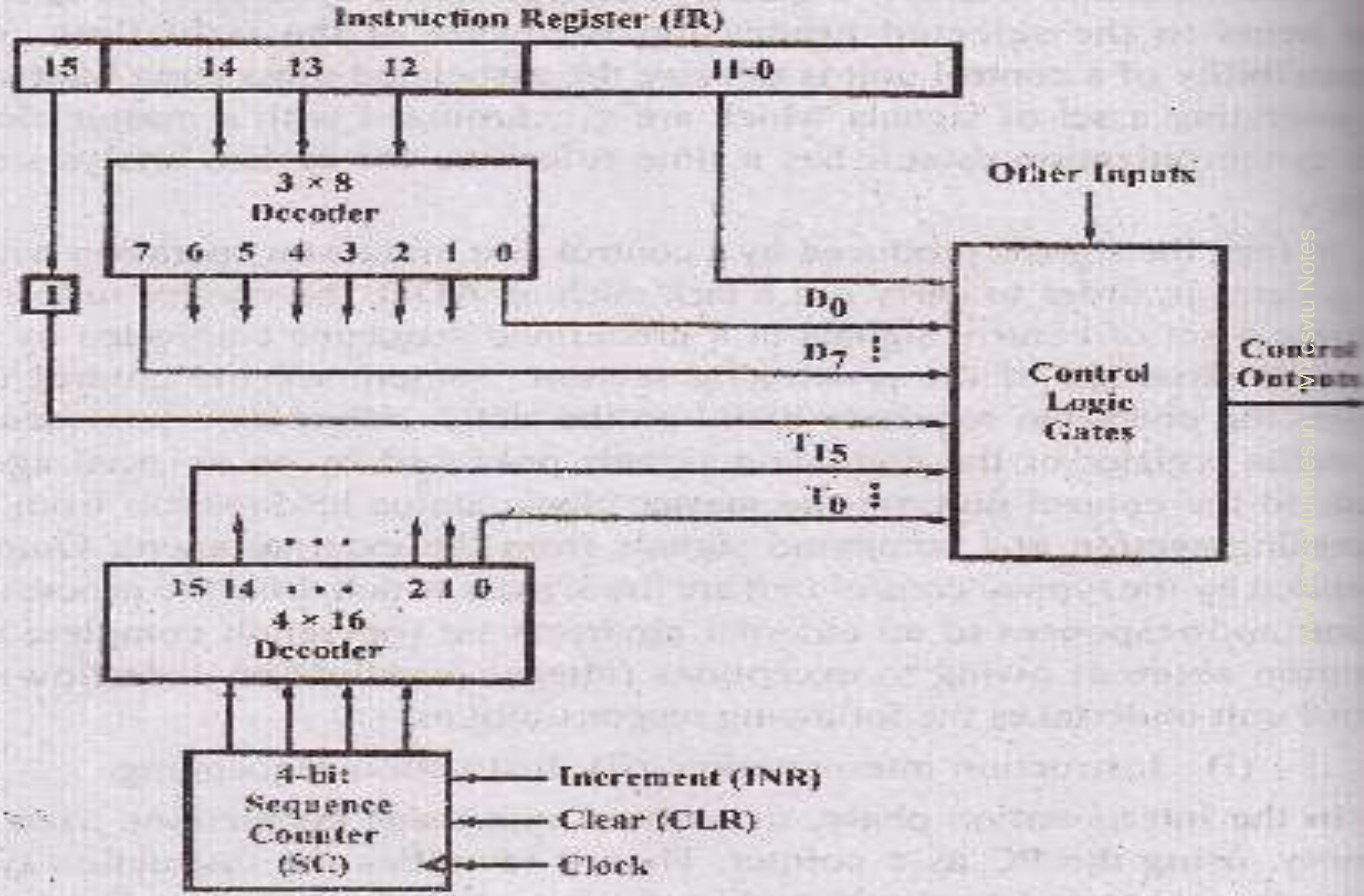
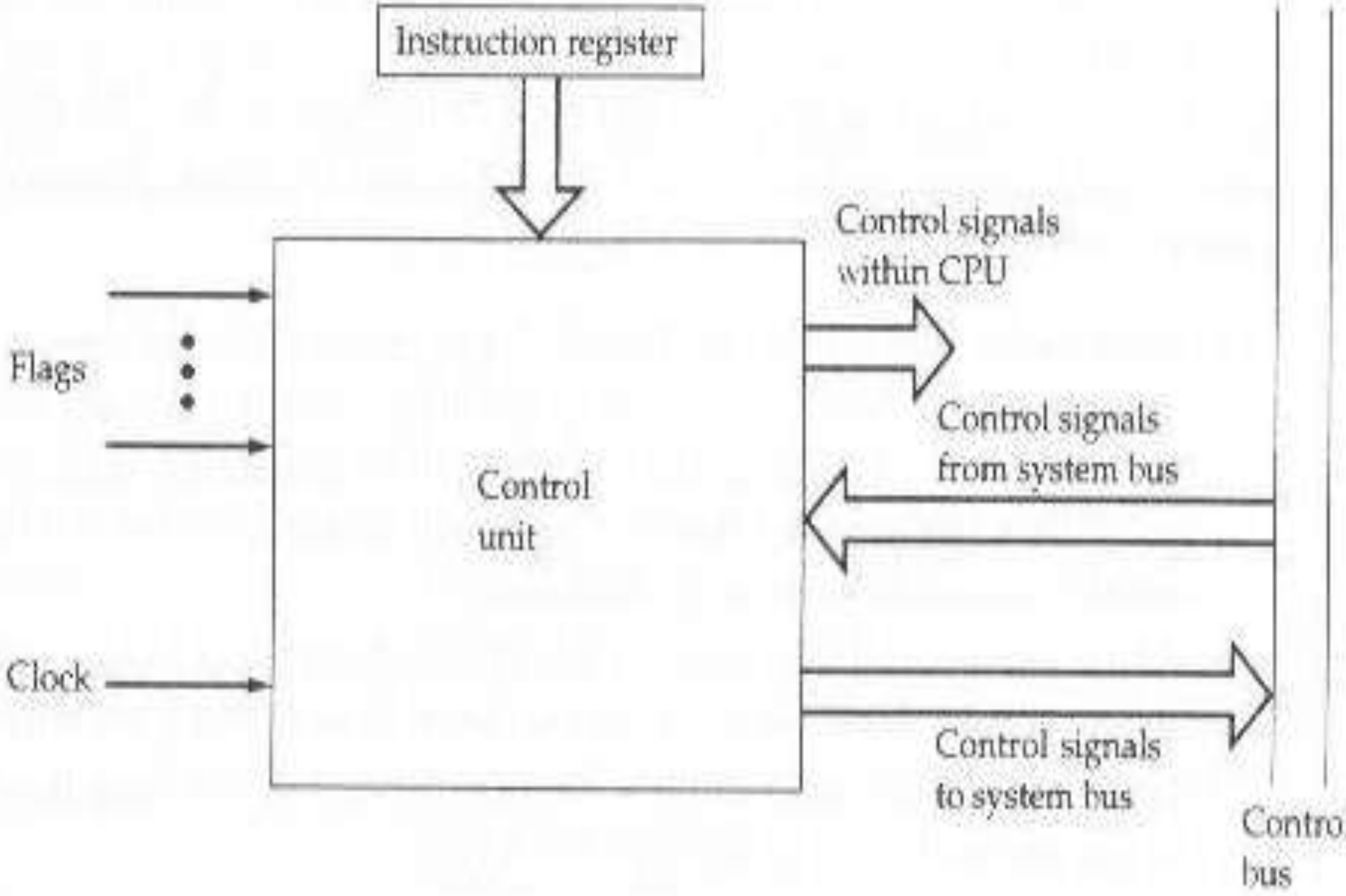


Fig. 2.1 Functional Block Diagram of Control Unit

- Figure above shows the functional block diagram of the control unit.
- It consists of two decoders-a sequence counter and a number of control logic gates.
- An instruction read from the memory is kept in the instruction register (IR).
- The instruction register as shown in the figure is divided into 3 parts.
  - 1 bit
  - The operation code
  - And bits through 0 through 11

- The 8 outputs of the decoder are represented by the symbols D_0 through D_7
- The subscripted decimal number is equivalent to the binary value of the corresponding operation code.
- Bit 15 of the instruction is shifted to a flip flop denoted by a symbol I.
- Bits 0 through 11 are applied to the control logic gates.
- The 4-bit sequence counter can count in binary from 0 through 15
- The outputs of the counter are decoded into 16 timing signals T_0 through T_{15}

- Sequence counter (SC) can be incremented or cleared synchronously.
- Most of the time, the counter is incremented to provide the sequence of timing signals out of 4 x 16 decoder.
- Once in a while, the counter is cleared to 0, causing the next active timing signal to be T_0



Model of a Control Unit

Deepak Bhalla (Asst. Professor (MCA & IT Dept.))

- Figure above is a general model of the control unit, showing all of its inputs and outputs.
- **The inputs are as follows :**
- **Clock** : This is how the control unit "keep time". The control unit causes one micro operation to be performed for each clock pulse. This is sometimes referred to as the processor cycle time or the clock cycle time.
- **Instruction Register :** The opcode of the current instruction is used to determine which micro operations to perform during the execution cycle.

RUNGTA

- **Flags :** These are needed by the control unit to determine the status of the processor and the outcome of the previous ALU operations.
- For example , for the increment -and –skip- if zero (ISZ) instruction , the control unit sill increment the PC if zero flag is set.
- **Control Signals From Control Bus :** The control bus portion of the system bus provides signal to the control unit, such as interrupt signals and acknowledgement.

- **Control signals from control bus:** The control bus portion of the system bus provides signals to the control unit, such as interrupt signals and acknowledgments.

The outputs are as follows:

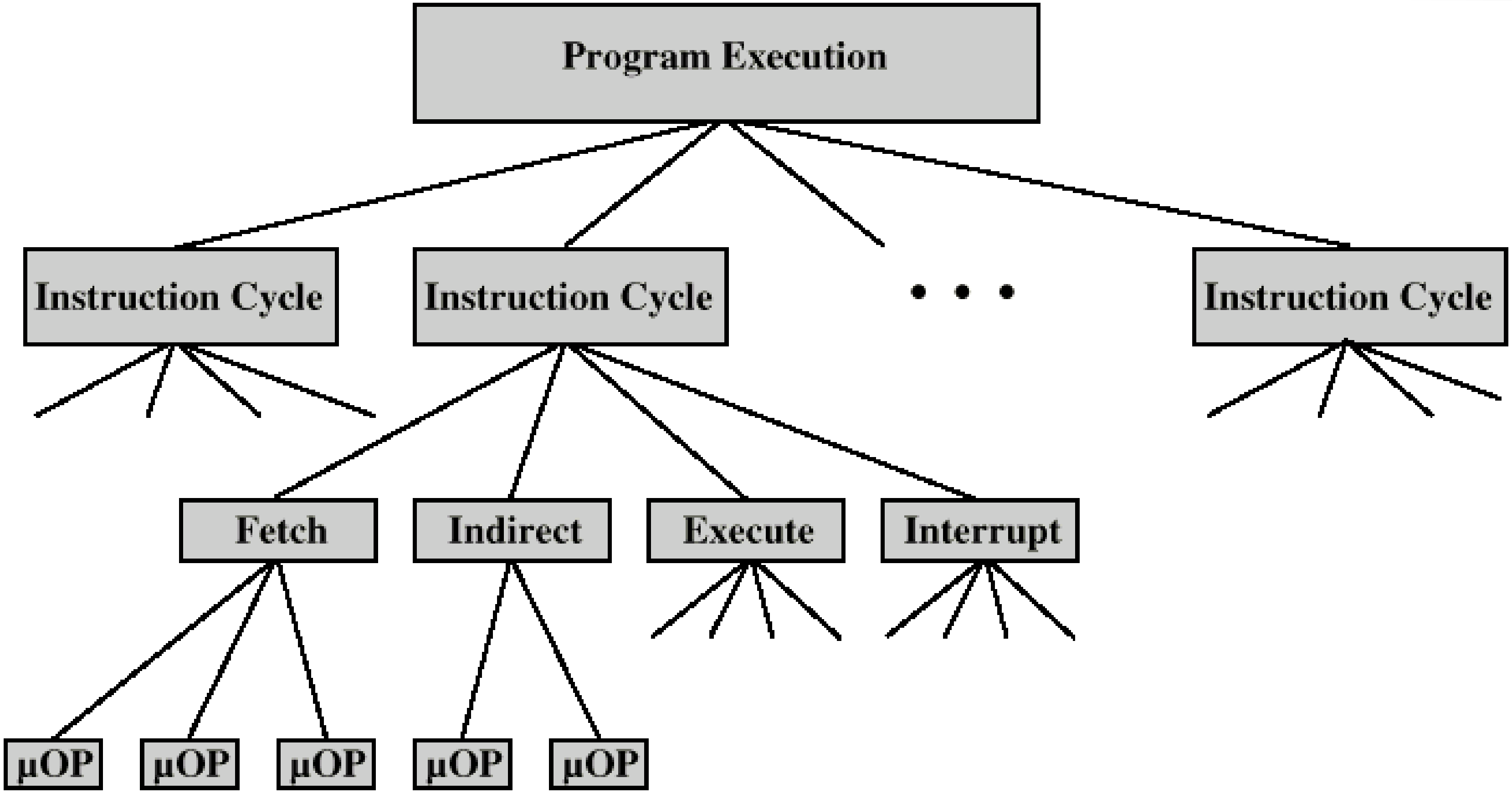
- **Control signals within the processor:** These are two types: those that cause data to be moved from one register to another, and those that activate specific ALU functions.
- **Control signals to control bus:** These are also of two types: control signals to memory, and control signals to the I/O modules.

The new element that has been introduced in this figure is the control signal. Three types of control signals are used: those that activate an ALU function, those that activate a data path, and those that are signals on the external system bus or other external interface. All of these signals are ultimately applied directly as binary inputs to individual logic gates.

Micro-Operations

- A computer executes a program
- Fetch/execute cycle
- Each cycle has a number of steps
- Called micro-operations
- Each step does very little
- Atomic operation of CPU

Constituent Elements of Program Execution



Fetch - 4 Registers

- Memory Address Register (MAR)
  - Connected to address bus
  - Specifies address for read or write op
- Memory Buffer Register (MBR)
  - Connected to data bus
  - Holds data to write or last data read
- Program Counter (PC)
  - Holds address of next instruction to be fetched
- Instruction Register (IR)
  - Holds last instruction fetched

Fetch Sequence

- Address of next instruction is in PC
- Address (MAR) is placed on address bus
- Control unit issues READ command
- Result (data from memory) appears on data bus
- Data from data bus copied into MBR
- PC incremented by 1 (in parallel with data fetch from memory)
- Data (instruction) moved from MBR to IR/IBR
- MBR is now free for further data fetches

Fetch Sequence (symbolic)

- t1: MAR <- (PC)
- t2: MBR <- (memory)
- PC <- (PC) +1
- t3: IR <- (MBR)
- (tx = time unit/clock cycle)
- or
- t1: MAR <- (PC)
- t2: MBR <- (memory)
- t3: PC <- (PC) +1
- IR <- (MBR)

Indirect Cycle

- MAR <- ($IR_{address}$) - address field of IR
- MBR <- (memory)
- $IR_{address}$ <- ($MBR_{address}$)

- MBR contains an address
- IR is now in same state as if direct addressing had been used
- (What does this say about IR size?)

Interrupt Cycle

- t1: MBR <-(PC)
- t2: MAR <- save-address
- PC <- routine-address
- t3: memory <- (MBR)
- This is a minimum
  - May be additional micro-ops to get addresses
  - N.B. saving context is done by interrupt handler routine, not micro-ops

Execute Cycle (ADD)

- Different for each instruction
- e.g. ADD R1,X - add the contents of location X to Register 1 , result in R1
- t1: MAR <- ($IR_{address}$)
- t2: MBR <- (memory)
- t3: R1 <- R1 + (MBR)
- Note no overlap of micro-operations

Execute Cycle (ISZ)

- ISZ X - increment and skip if zero
  - t1: MAR <- ($IR_{address}$)
  - t2: MBR <- (memory)
  - t3: MBR <- (MBR) + 1
  - t4: memory <- (MBR)
  - if (MBR) == 0 then PC <- (PC) + 1
- Notes:
  - if is a single micro-operation
  - Micro-operations done during t4

Functional Requirements

- Define basic elements of processor
- Describe micro-operations processor performs
- Determine functions control unit must perform

Types of Micro-operation

- Transfer data between registers
- Transfer data from register to external
- Transfer data from external to register
- Perform arithmetic or logical ops

Functions of Control Unit

- Sequencing
  - Causing the CPU to step through a series of micro-operations in the proper sequence , based on the program being executed.
- Execution
  - Control unit causes each micro operation to be performed.
- This is done using Control Signals

Control unit implementations

- **To execute instructions, a computer's processor must generate the control signals used to perform the processor's actions in the proper sequence.**
- **This sequence of actions can either be executed by another processor's software (for example in software emulation or simulation of a processor) or in hardware.**
- **Hardware methods fall into two categories: the processor's hardware signals are generated either by hardwired control, in which the instruction bits directly generate the signals, or by microprogrammed control in which a dedicated microcontroller executes a microprogram to generate the signals.**

Hardware Control Unit

- When a control signals are generated by hardware using conventional logic design techniques, the control unit is said to be Hardwired.
- A diagram of hardwired control unit is shown in the figure below:

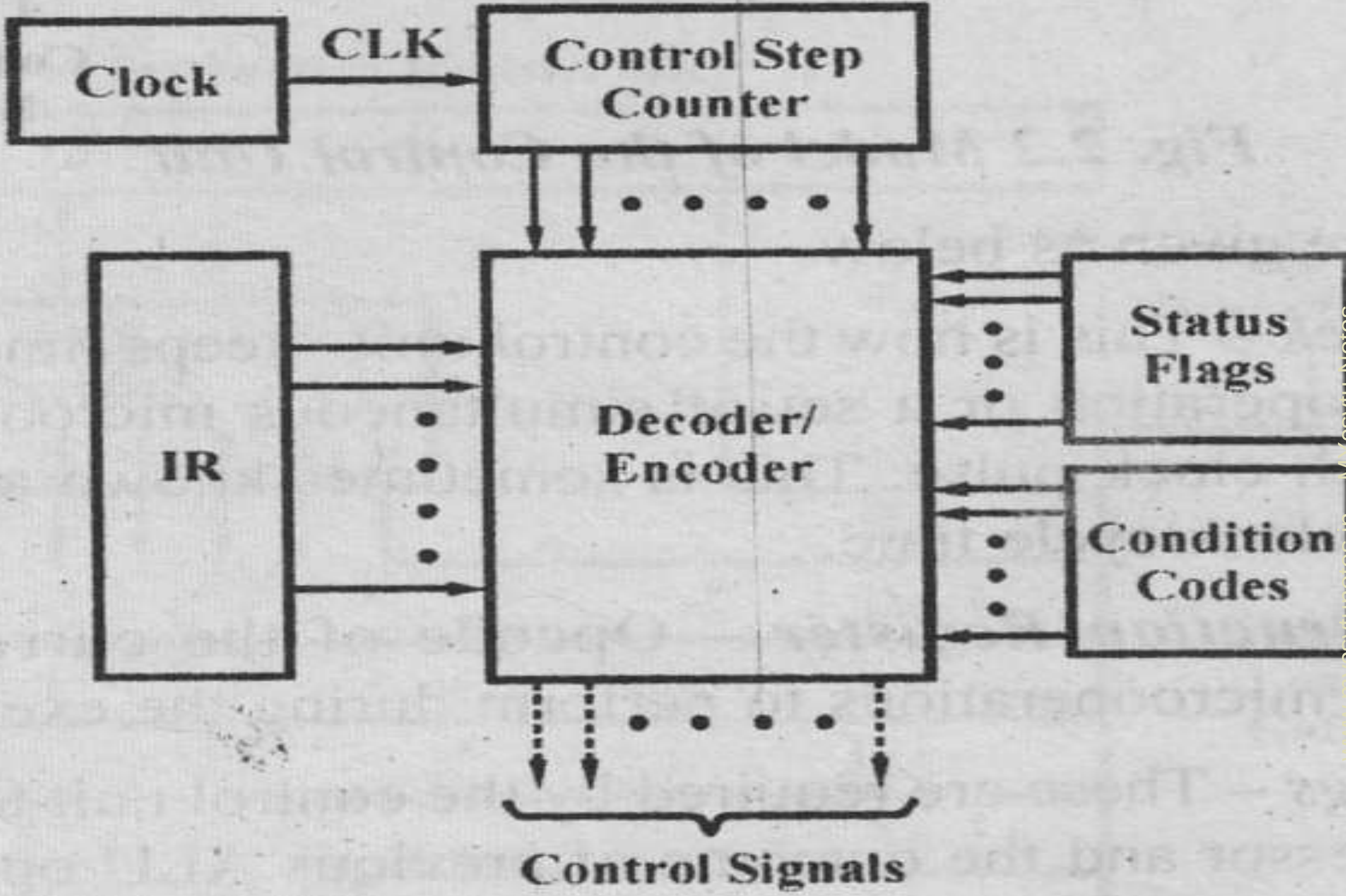


Fig. 2.3 Control Unit Organization

- In this case, the control unit is a combinatorial circuit; it gets a set of inputs (from IR, flags, clock, system bus) and transforms them into a set of control signals.
- This control unit may be based on the counter driven by a clock signal,CLK.
- The required control signals are uniquely determined by the following information :

i) **Contents of the control counter.**

ii) **Contents of the instruction register**

iii) **Contents of the condition code and other status flags.**

By status flags, we mean the signals representing the states of various sections of the CPU and various control lines connected to it.

- **In order to gain some insight into the structure of the control unit, we will start by giving a simplified view of the hardware involved.**
- **The decoder encoder block in fig 2.3 is a combinational circuit which produces the required control outputs, depending on the state of all its input.**
- **By separating the decoding and encoding functions, we obtain the more detailed block diagram in figure below**

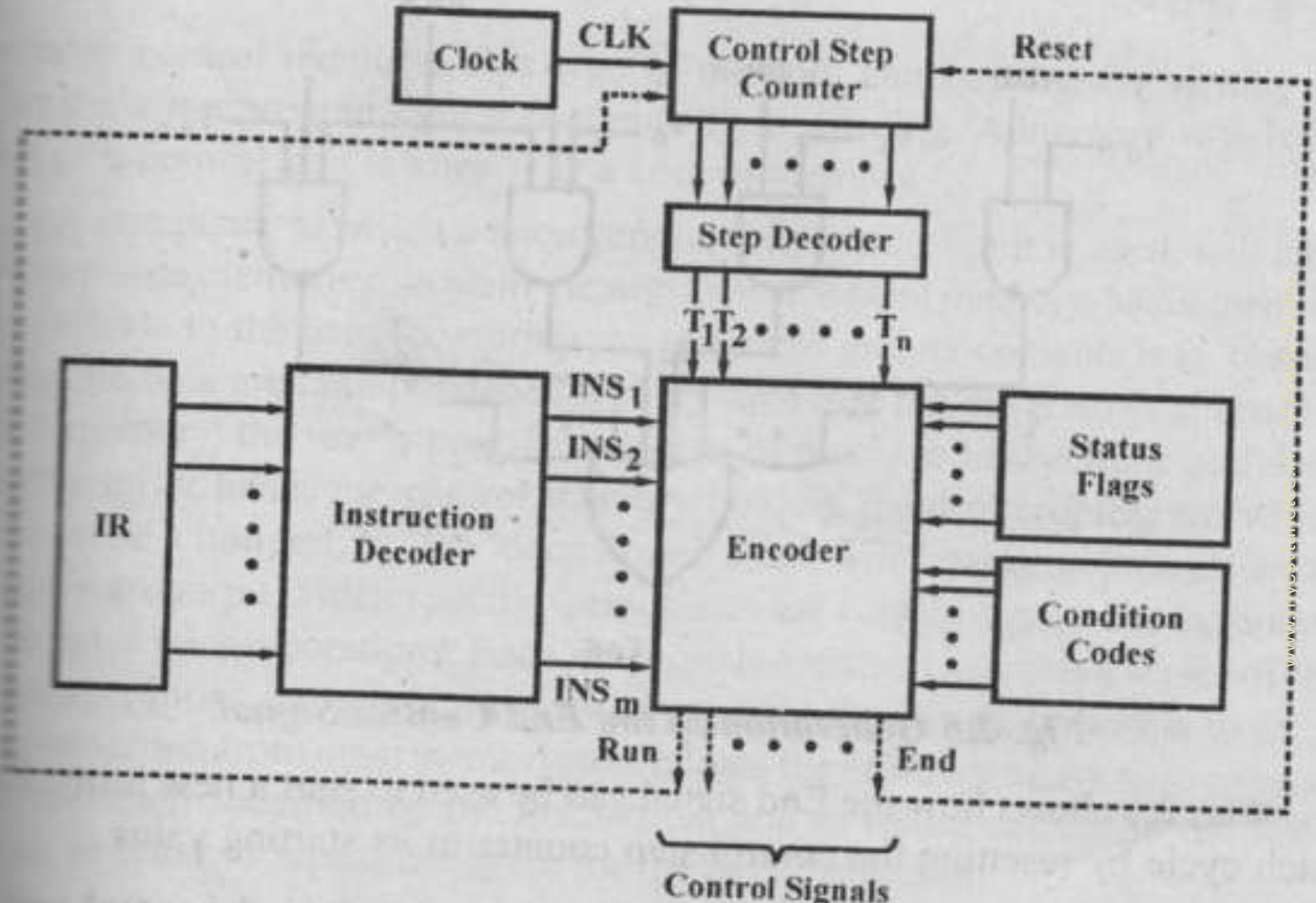


Fig. 2.4 Separation of the Decoding and Encoding Function

Advantages of Hardwired Control

- **Hardwired Control is more advantageous as compared to microprogram control.**
- **In hardware organization, the control logic is implemented with gates, flip-flops, decoders and other digital circuits**
- **Hardwired control provides highest speed.**
- **It can be optimized to produced a fast mode of operation.**
- **RISCs are implemented with hardwired control.**

Problems With Hard Wired Designs

- Complex sequencing & micro-operation logic
- Difficult to design and test
- Inflexible design
- Difficult to add new instructions

Deepak Bhalla (Asst. Professor (MCA & IT Dept.))

Micro Programmed Control Unit

- **The idea of microprogramming was introduced by M. V. Wilkes in 1951 as an intermediate level to execute computer program instructions.**
- **Microprograms were organized as a sequence of microinstructions and stored in special control memory.**
- **The algorithm for the microprogram control unit is usually specified by flow-chart description**

- The main advantage of the microprogram control unit is the simplicity of its structure.
- Outputs of the controller are organized in microinstructions and they can be easily replaced

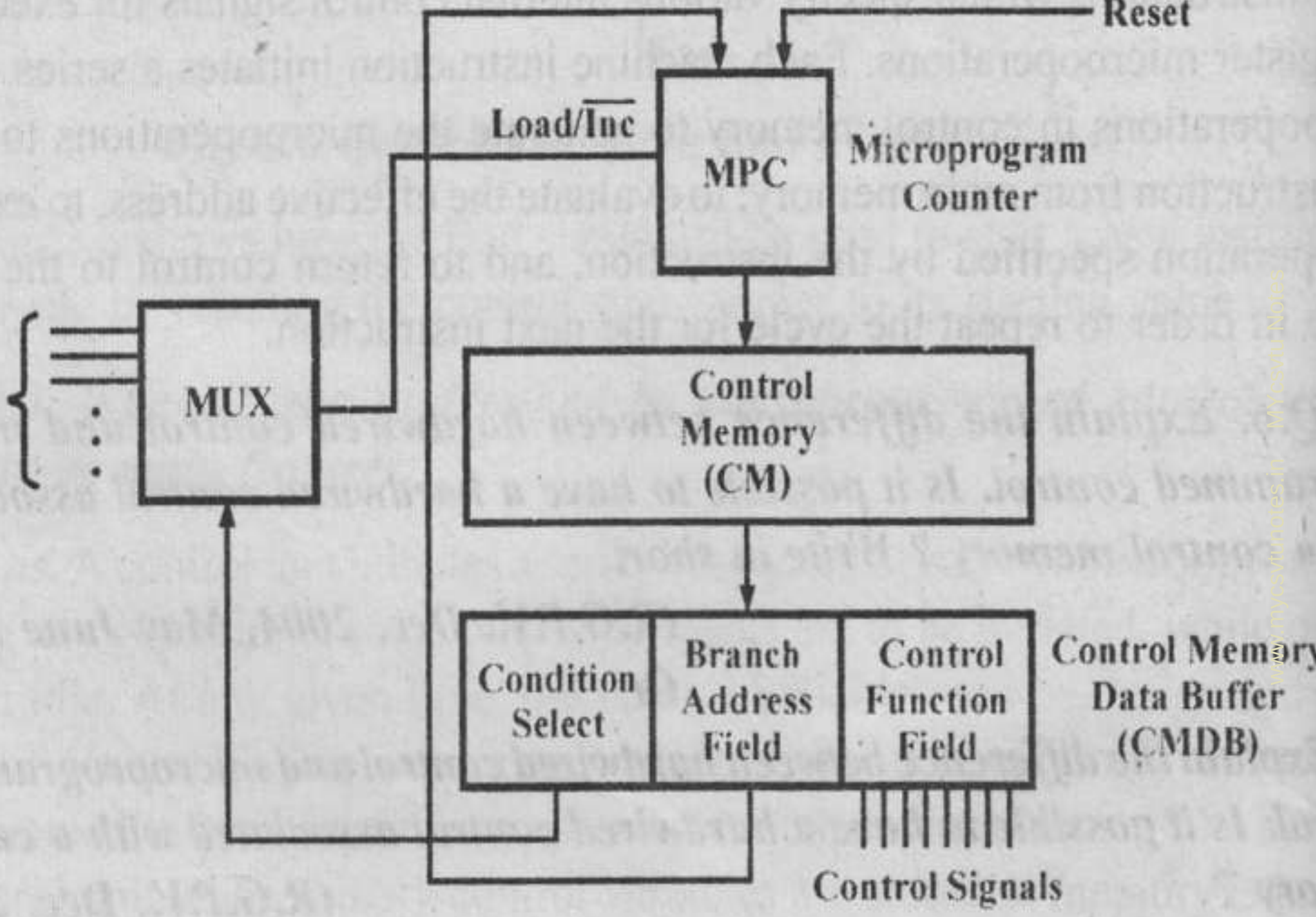


Fig. 2.7 Block Diagram of Microprogrammed Control Unit

- The summary of the use of various components included in this organization is given below:

- Control Memory Buffer Register(CMBR) :
  - **Control means buffer register functions the same as the memory buffer register of the main memory.**
  - **Basically it is a latch, and serves as a buffer from microinstructions retrieved from the control memory.**
  - **Typically, each microinstructions will have three distinct fields.**

RUNGTA

Condition Select	Branch Address Field	Control Function Field

- **The control select field chooses the external condition to be tested.**
- **In case the select condition is true the output of the MUX will be 1, because the output of the multiplexer is connected to the load input of the micro program counter(MPC), the MPC will be loaded with the address specified in the branch address field of the micro instruction**

RUNGTA

- **If the selected external condition is false, then microprogram counter will point to the next microinstruction to be executed.**
- **So this arrangement permits conditional branching.**
- **The control function field of the microinstruction may hold the control information in an encoded form.**

Micro Program Counter

- **The microprogram counter holds the address of the next microinstruction to be executed.**
- **Initially it is loaded from an external source to point to the starting address of the microprogram to be executed.**
- **From then on, the microprogram counter is incremented after each instruction fetch, and the instruction fetched is shifted to the CMBR.**

- **If a branch instruction is encountered , then the microprogram counter will be loaded with the contents of branch address field of the micro instruction that is held in the control memory buffer register.**

External Condition Select MUX

- **The MUX chooses one of the external condition according to the contents of the condition select field of the microinstruction.**
- **So, the condition to be chosen must be specified in an encoded form.**

Difference Between Hardwired & Micro Programmed Control

- **Hardwired control is more advantageous as compared to microprogram control.**
- **In the hardwire organization, the control logic is implemented with gates, flip-flop, decoders and other digital circuits.**
- **It has the advantage that it can be optimized to produce a fast mode of operation.**
- **Mostly computer based on the reduced instruction set computer (RISC) architecture concept use hardwired control, as the name implies, requires changing in the wiring among the various components, if the design has to be modified or changed.**

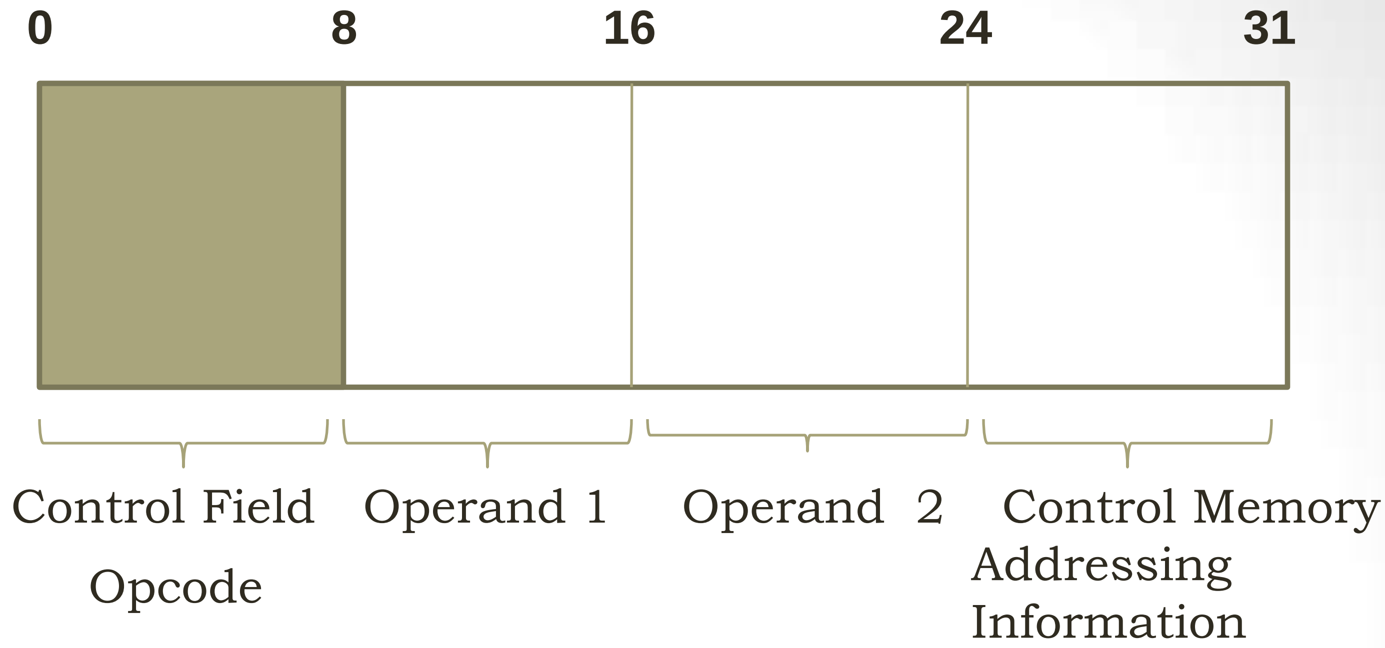
- **In the micro programmed organization, the control information is stored in control memory.**
- **The control memory is programmed to initiate the required sequence of micro operation.**
- **The main advantage of microprogrammed control unit is the fact that once the hardwired configuration is established. There should be no need for further hardware or wiring changes.**

Micro Instruction Format

1. **A micro instruction has two parts :**
   i. **A set of control fields that specify the control signals to be activated.**
   ii. **An address field that contains the address in control memory of the next microinstruction to be executed.**

Microinstruction length is determined by 3 factors:

- **Maximum number of simultaneous micro operations which must be specified , that is, the degree of parallelism required at the micro operation level.**
- **The manner in which control information is represented or encoded.**
- **The manner in which the next microinstruction is specified.**



Microinstruction Format of the IBM System/370 Model 145

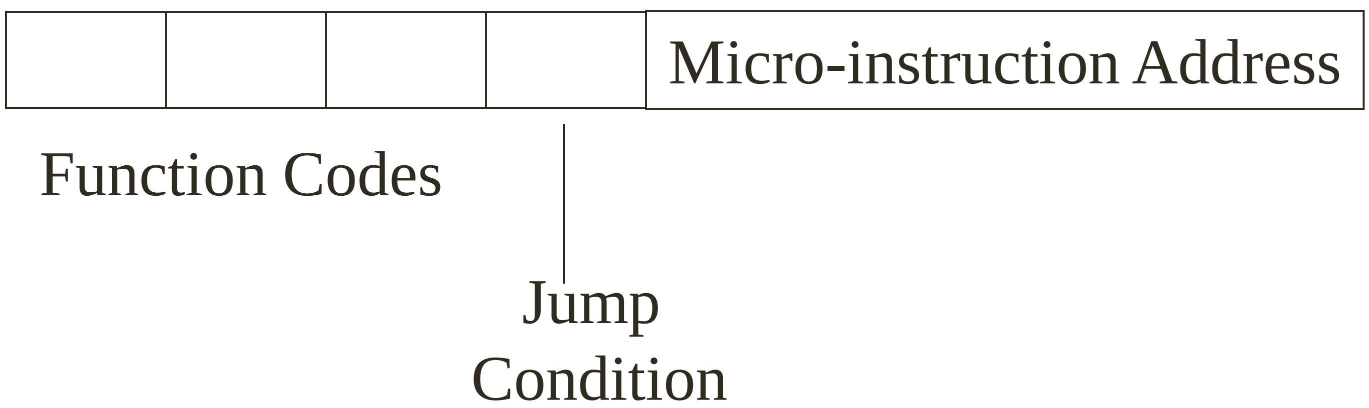
- **Microinstructions which specify a single micro operation are similar to conventional machine instructions**
- **They are relatively short, but owing to their lack of parallelism.**

- **The format of IBM system/370 model 145 which is shown in figure above is representative of this type of micro instruction**
- **It consists of 4 bytes.**
- **The leftmost byte (shaded) is an opcode which specifies the microinstruction to be performed.**
- **The next two bytes specify the operands which in most cases, are the address of CPU registers.**
- **The right most byte contains information used to construct the address of the next micro instruction.**

Micro-instruction Types

- Each micro-instruction specifies single (or few) micro-operations to be performed
  - (*vertical* micro-programming)
- Each micro-instruction specifies many different micro-operations to be performed in parallel
  - (*horizontal* micro-programming)

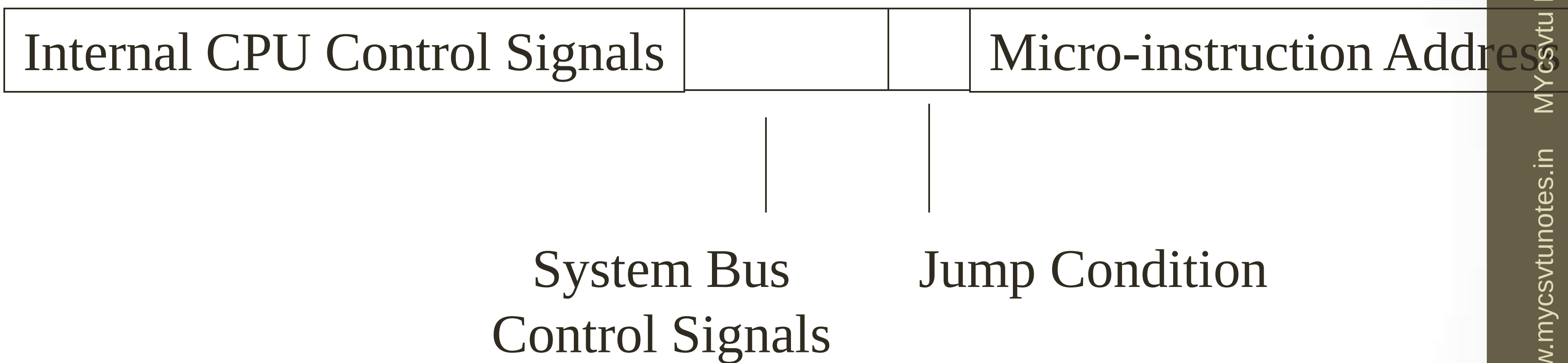
Vertical Micro-programming register



Vertical Micro-programming

- Short format
- Width is narrow
- n control signals encoded into $\log_2 n$ bits
- Limited ability to express parallelism
- Considerable encoding of control information.

Horizontal Micro-programmed diag



Horizontal Micro-programming

- Long Format
- Wide memory word
- Ability to express High degree of parallelism
- Little encoding of control information

Sequencing Techniques

- Based on current microinstruction, condition flags, contents of IR, control memory address must be generated
- Based on format of address information
  - Two address fields
  - Single address field
  - Variable format

Address Generation

- Explicit — Implicit
- Two-field — Mapping
- Unconditional Branch — Addition
- Conditional branch — Residual control

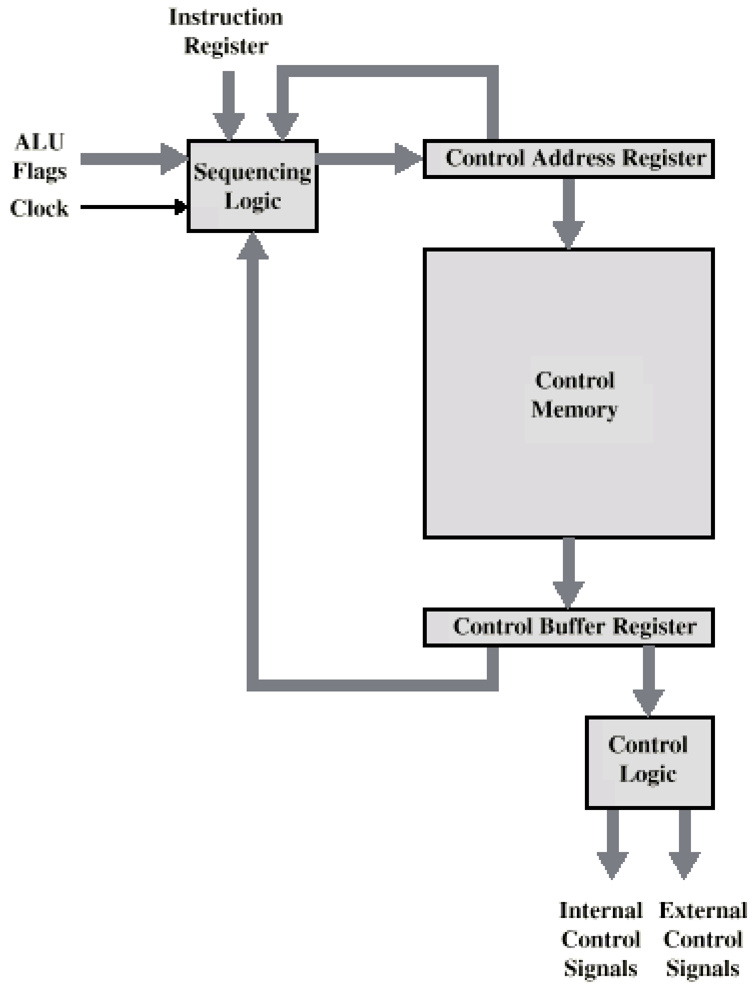
Execution

- The cycle is the basic event
- Each cycle is made up of two events
  - Fetch
    - Determined by generation of microinstruction address
  - Execute

Execute

- Effect is to generate control signals
- Some control points internal to processor
- Rest go to external control bus or other interface

Control Unit Organization



Deepak Bhalla (Asst. Professor (MCA & IT Dept.))