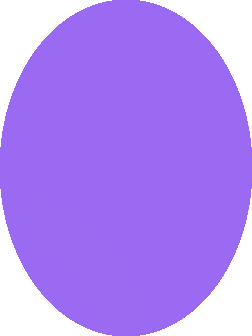
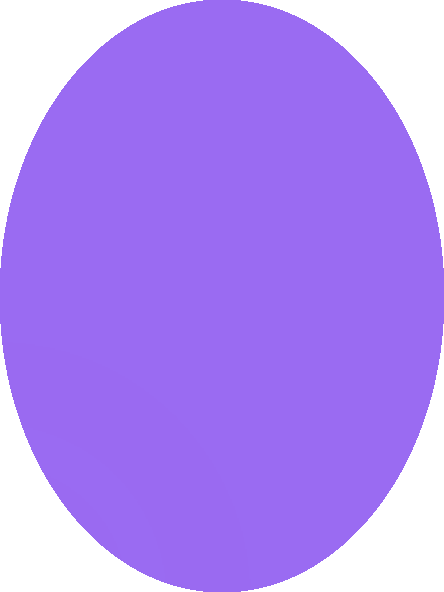
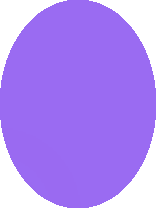
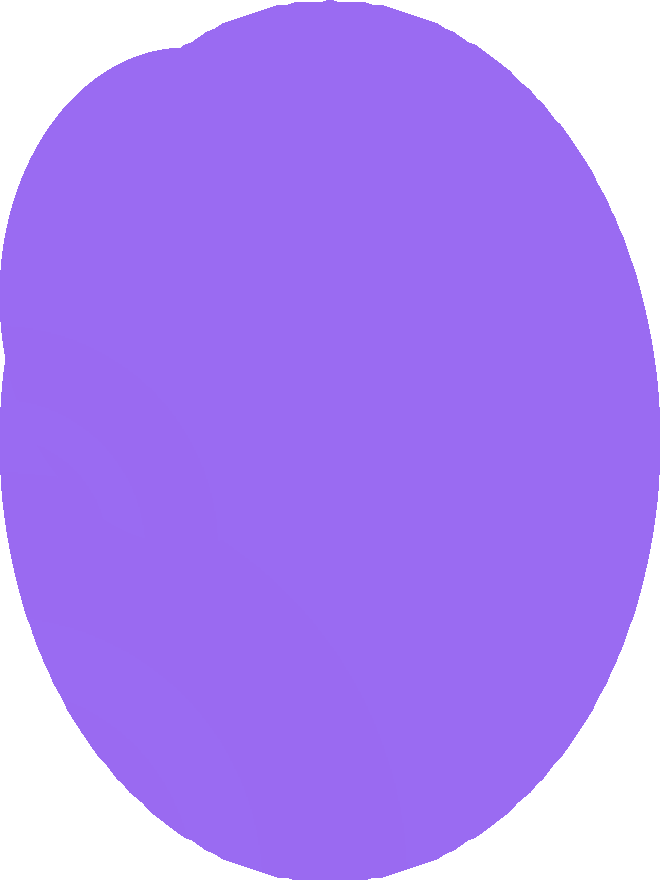
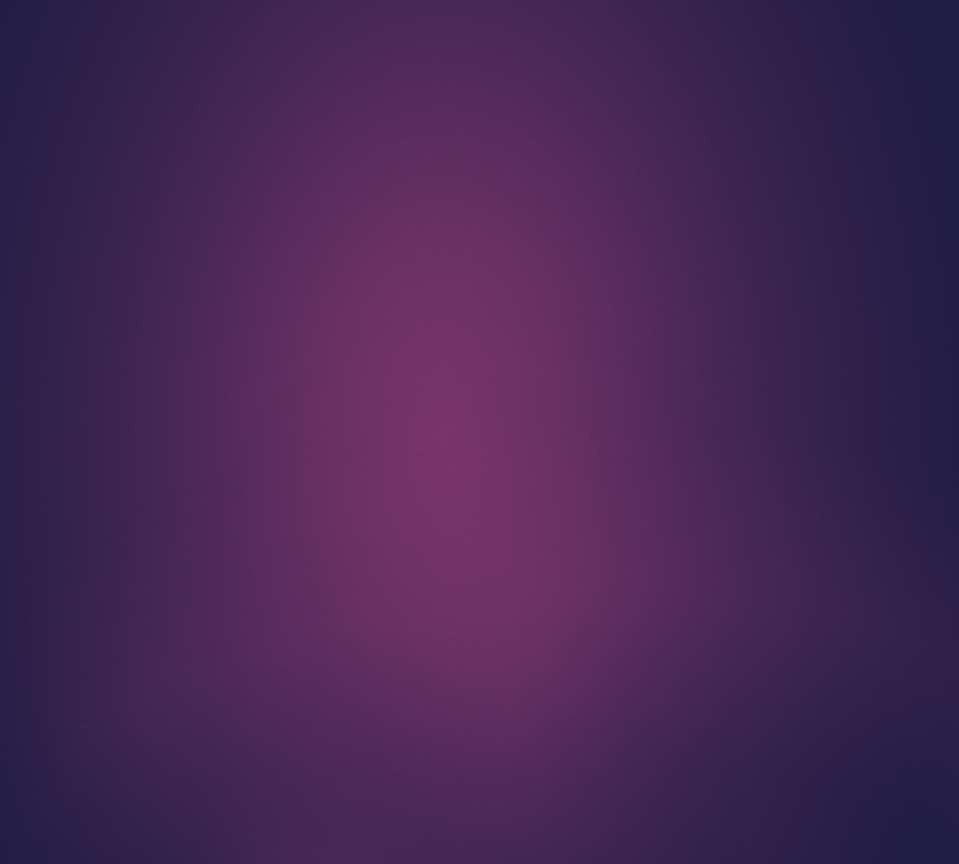


*COMPUTER ORGANIZATION*

*&*

*ARCHITECTURE*



# CONTROL UNIT

 Control unit generates timing and control signals for the operations of the computer.

 It’s the part of the CPU that initiates sequences of microoperations.

 It tells the computer’s memory, arithmetic & logic unit and input/output devices how to respond to a program’s instructions.

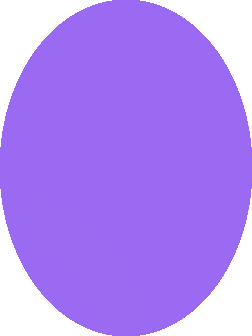
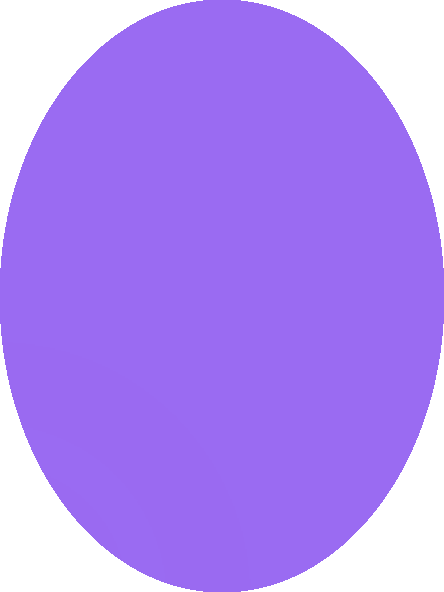
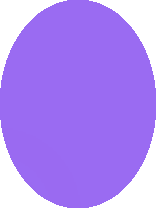
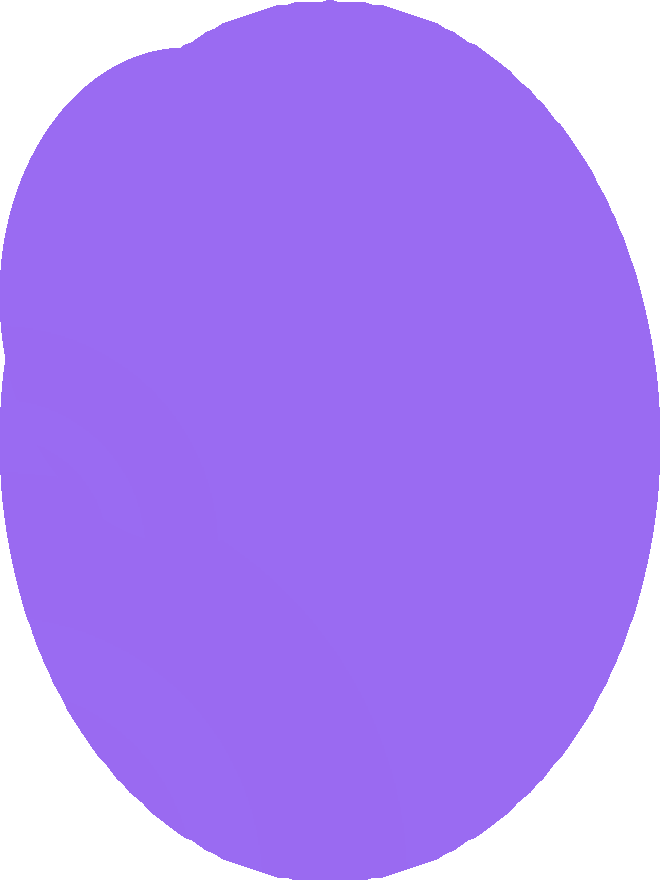
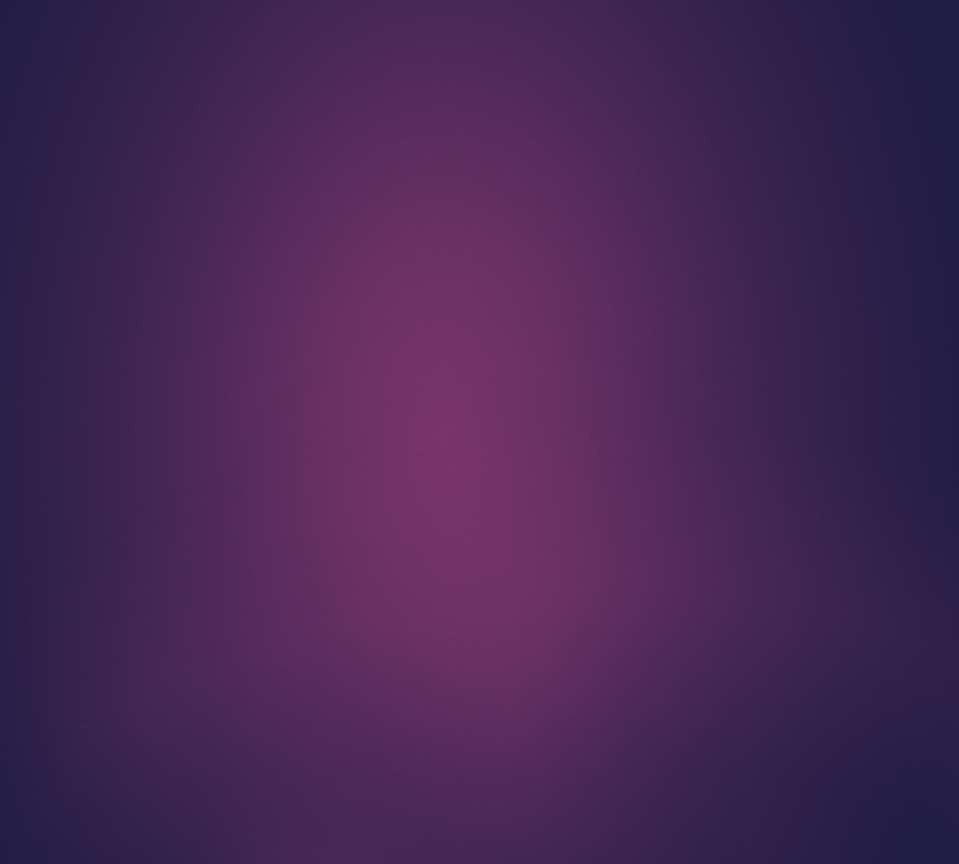
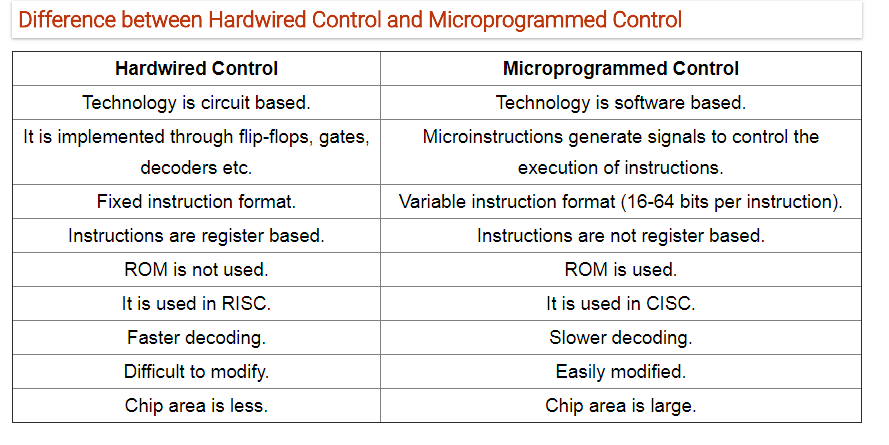
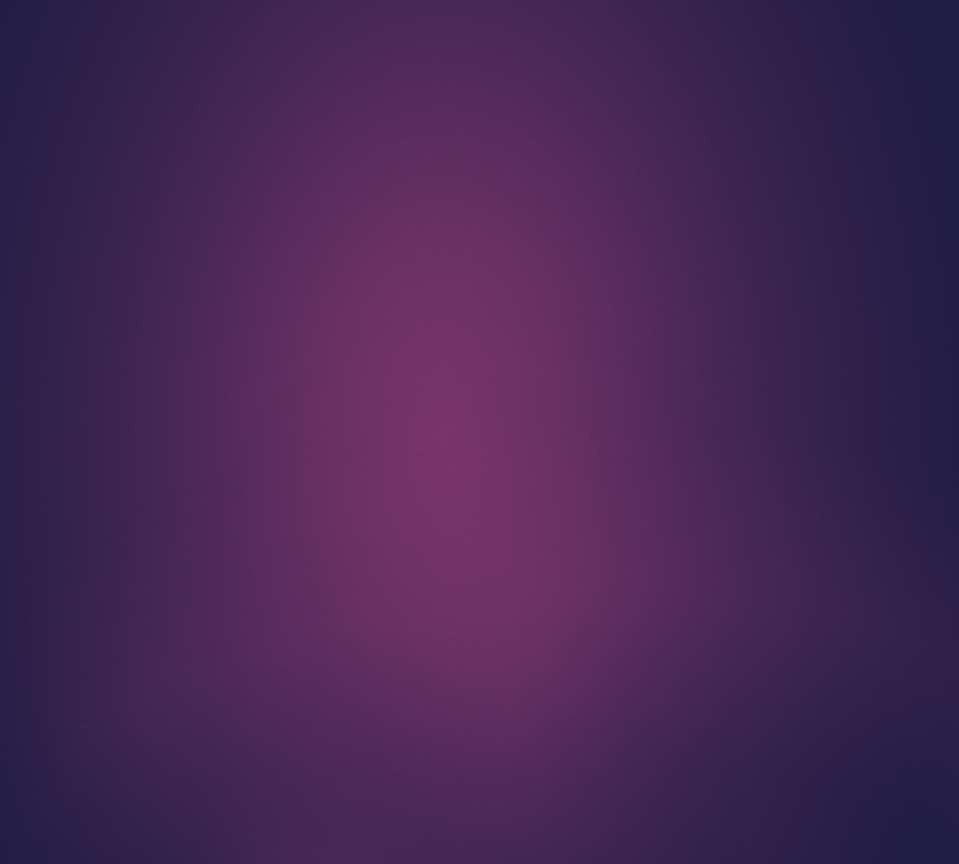
**TYPES :**

There are two methods to implement the control unit:

 **Hardwired Control Unit (**uses fixes instructions, combinational logic units

of AND/OR (logic gates), encoders, decoders, etc.)

 **Microprogrammed Control Unit (t**he logic of the control unit is specified by microprograms (consists of a sequence of instructions that specify microoperations).



# IMPORTANT TERMS

Control unit initiates a series of sequential steps of microoperations.

##  CONTROL WORD :

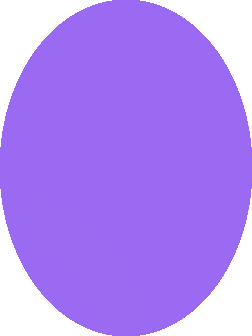
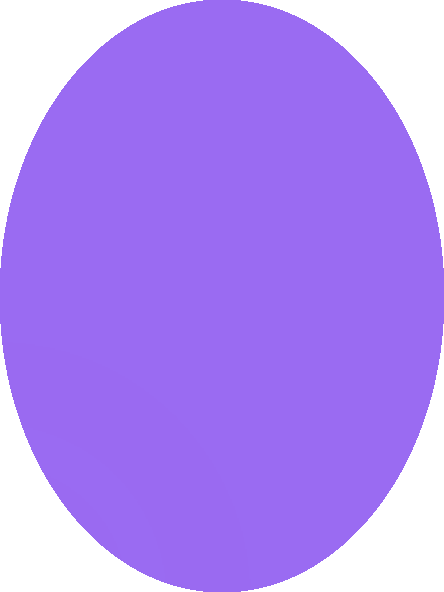
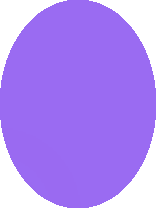
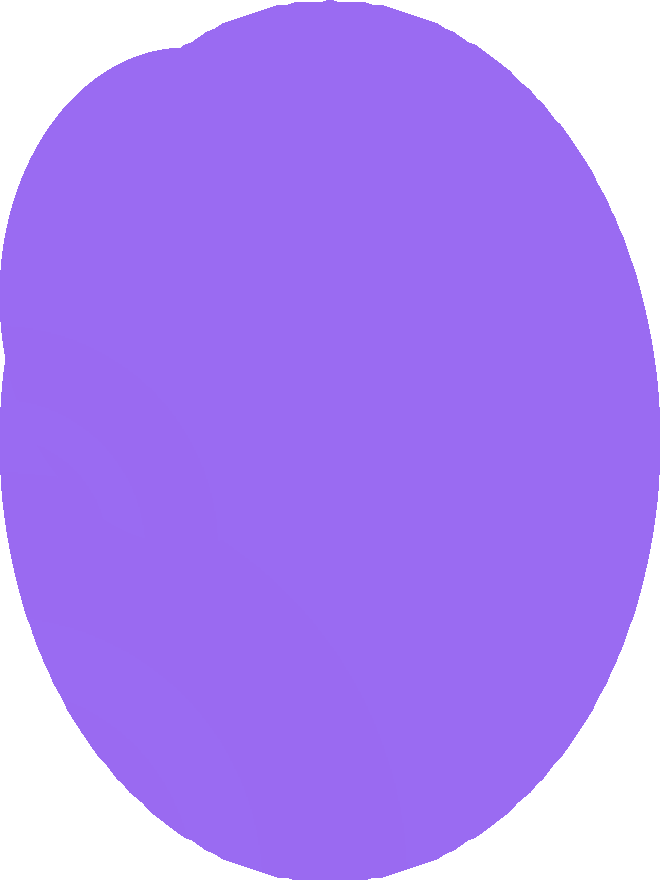
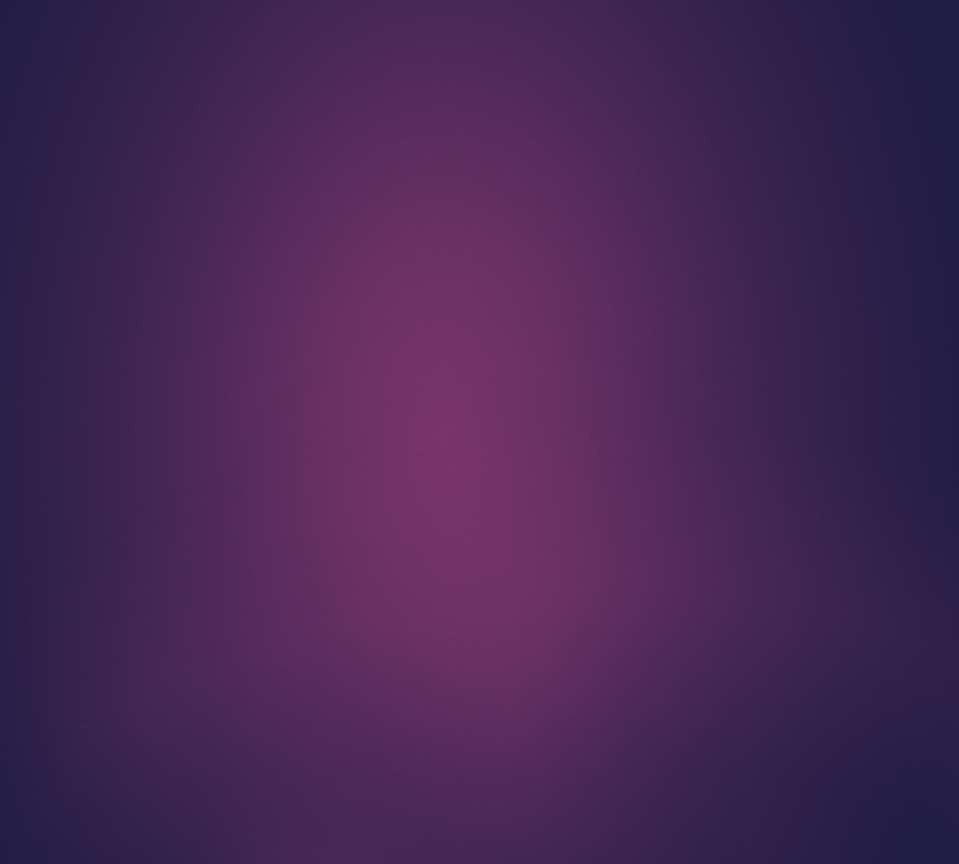
* The control variables at any time are represented by 1’s and 0’s, known as Control Word**.**
* Control words can be programmed to perform various operations.

##  MICROPROGRAMMED CONTROL UNIT :

* A control unit whose binary control variables are stored in the memory is called MICROPROGRAMMED CONTROL UNIT.

##  MICROINSTRUCTIONS :

* Each word in the control memory contains microinstructions .
* It specifies one or more microoperations for the system.



MICROPROGRAM

##  MICROPROGRAM :

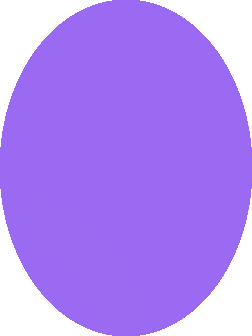
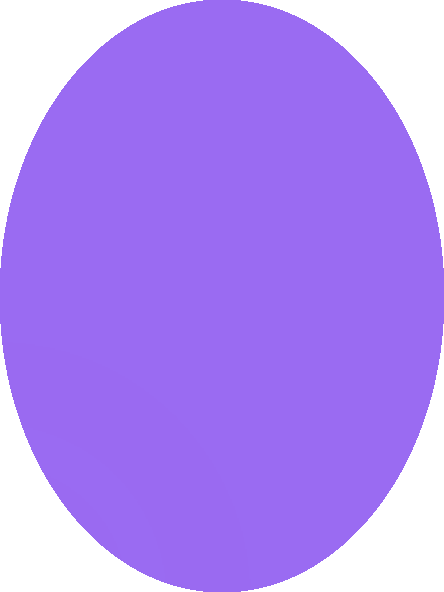
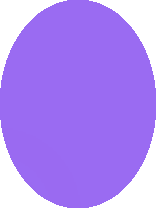
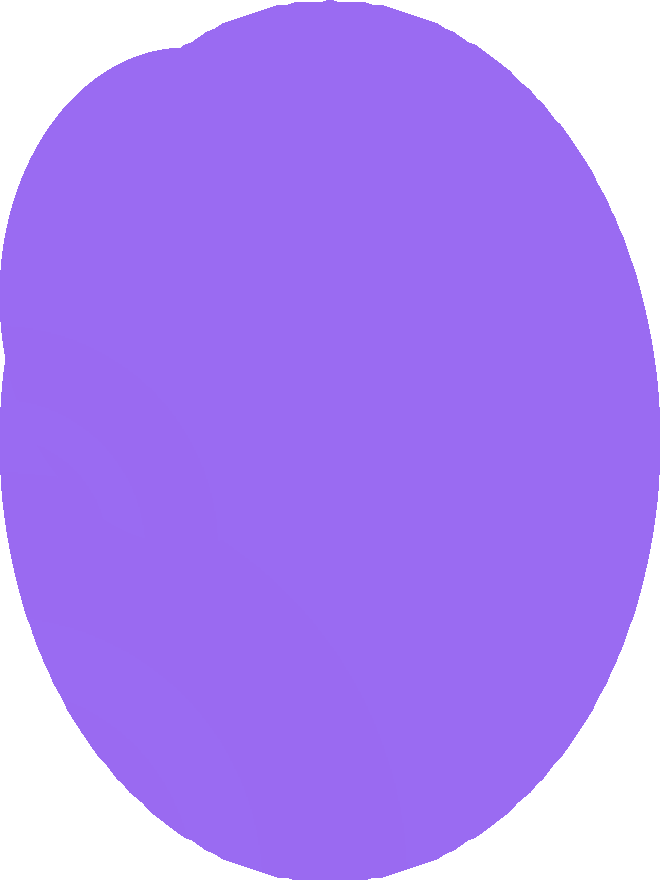
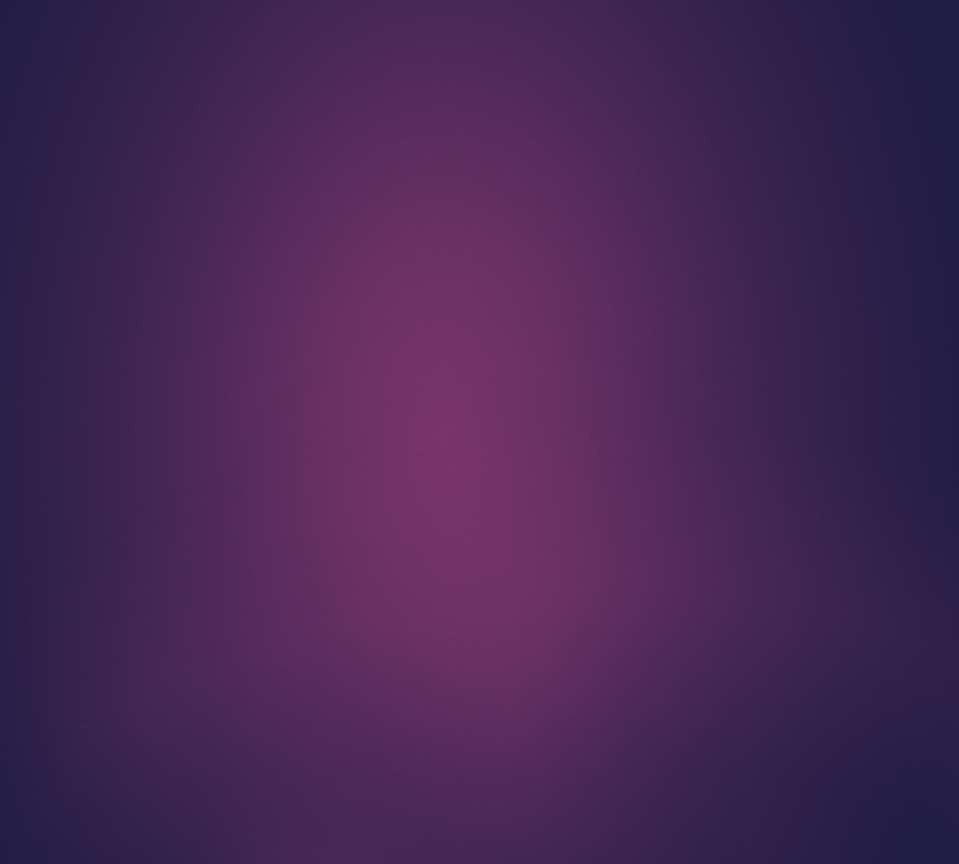
* Sequence of microinstructions constitutes a microprogram.
* The microprogram controls the function of the CPU.
* Alterations in microprograms are not needed once the control unit is in

operation. The control memory can be read-only memory (ROM).

* The Contents of words in **ROM** are fixed and cannot be altered, since there is no writing capacity in ROM.

##  DYNAMIC MICROPROGRAMMING :

* microprogram is loaded initially from an auxiliary memory (secondary memory), such as magnetic disk.
* employs a writable control memory which allows the user to change the microprogram, though it is mostly used for reading.



# TYPES OF MEMORY IN MPC

1. **MAIN MEMORY**

 The main memory is used for storing programs. The content of the main memory can be altered when data is manipulated and each time the program is changed.

 Program in main memory contains machine instructions and data.

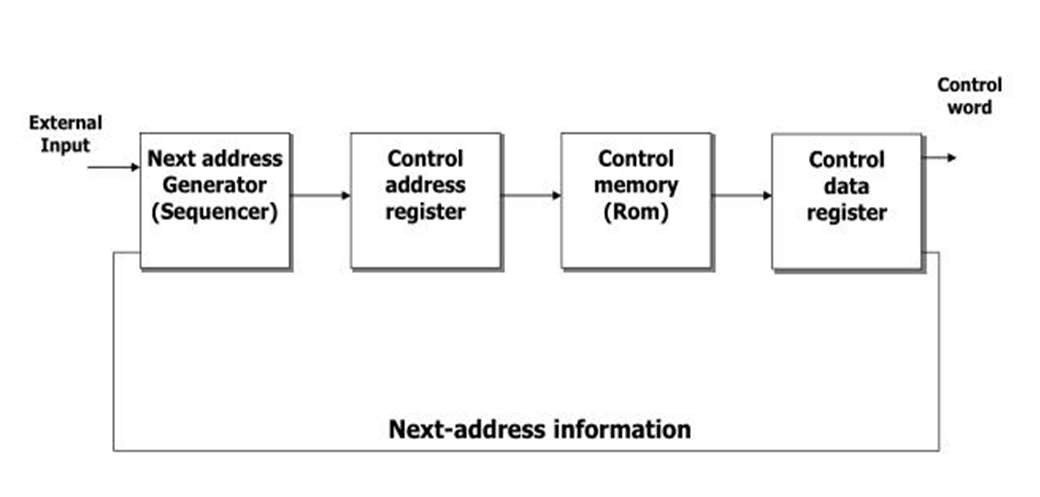
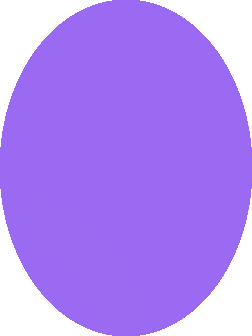
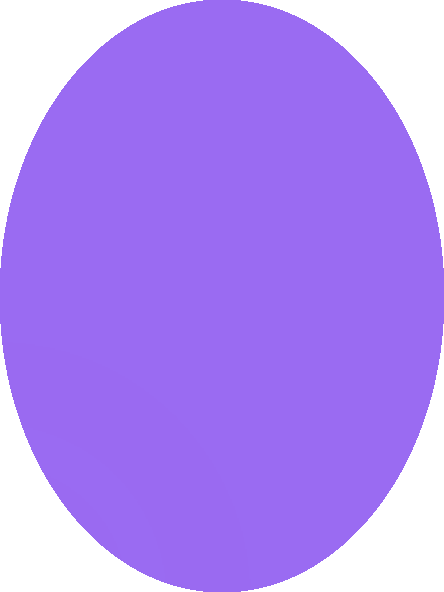
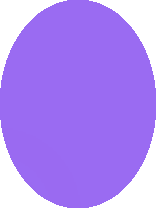
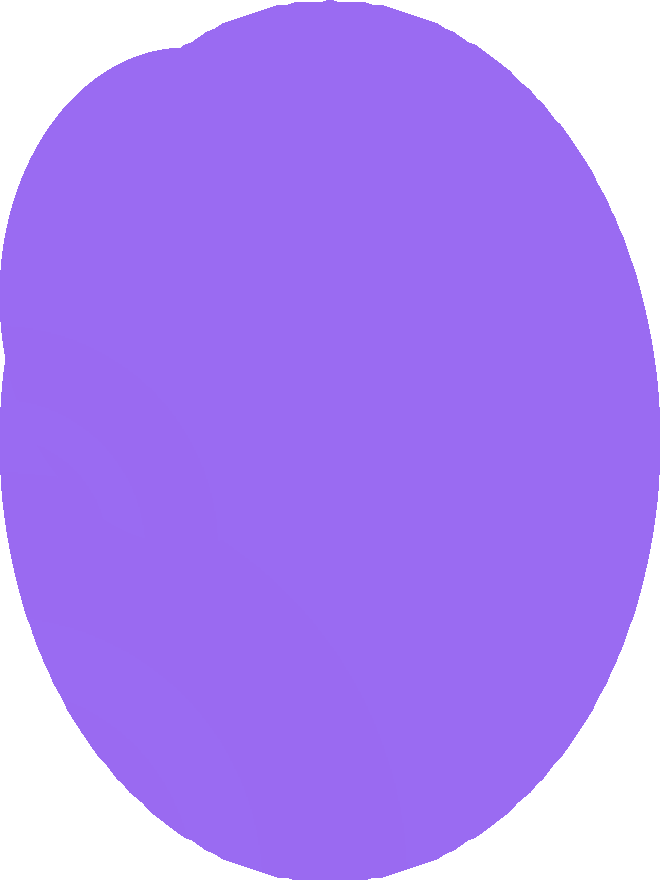
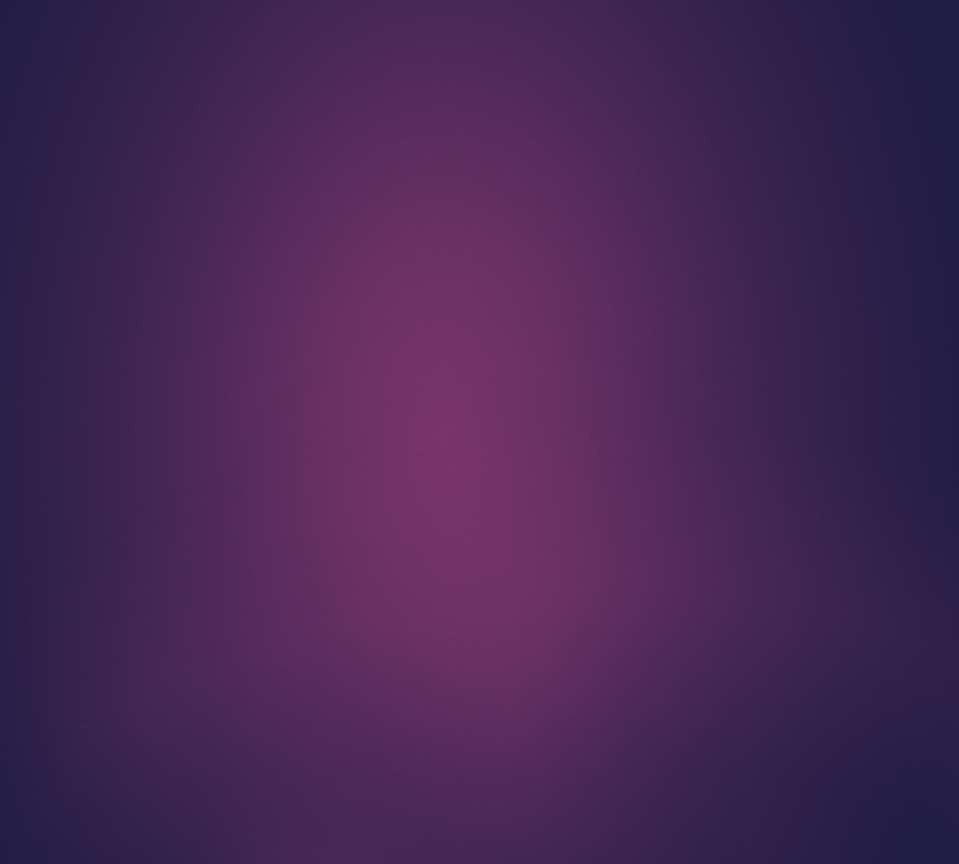
1. **CONTROL MEMORY**

 Memory that is a part of the control unit is called Control memory.

 Control memory holds microprograms that cannot be altered by the user. The microprogram consists of microinstructions to execute register microoperations.

 Machine instruction initiates a series of microinstructions in the control memory.

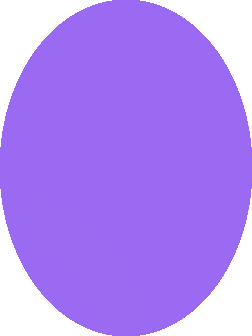
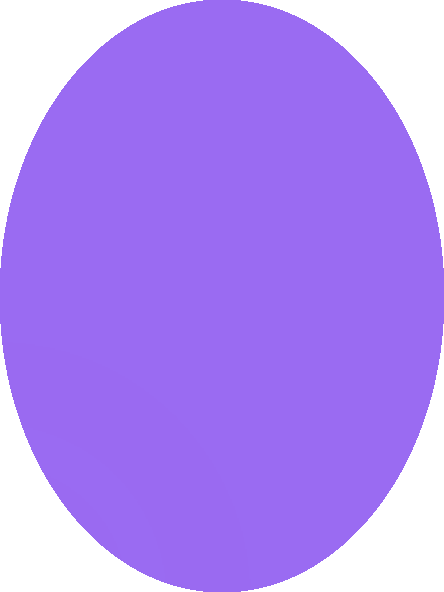
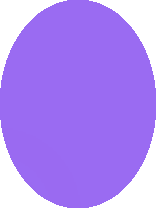
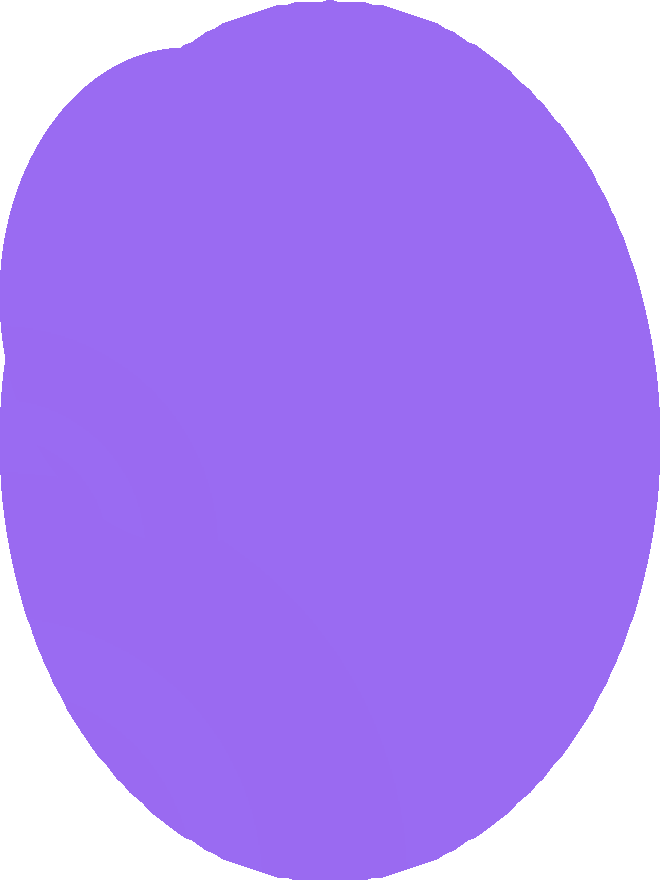
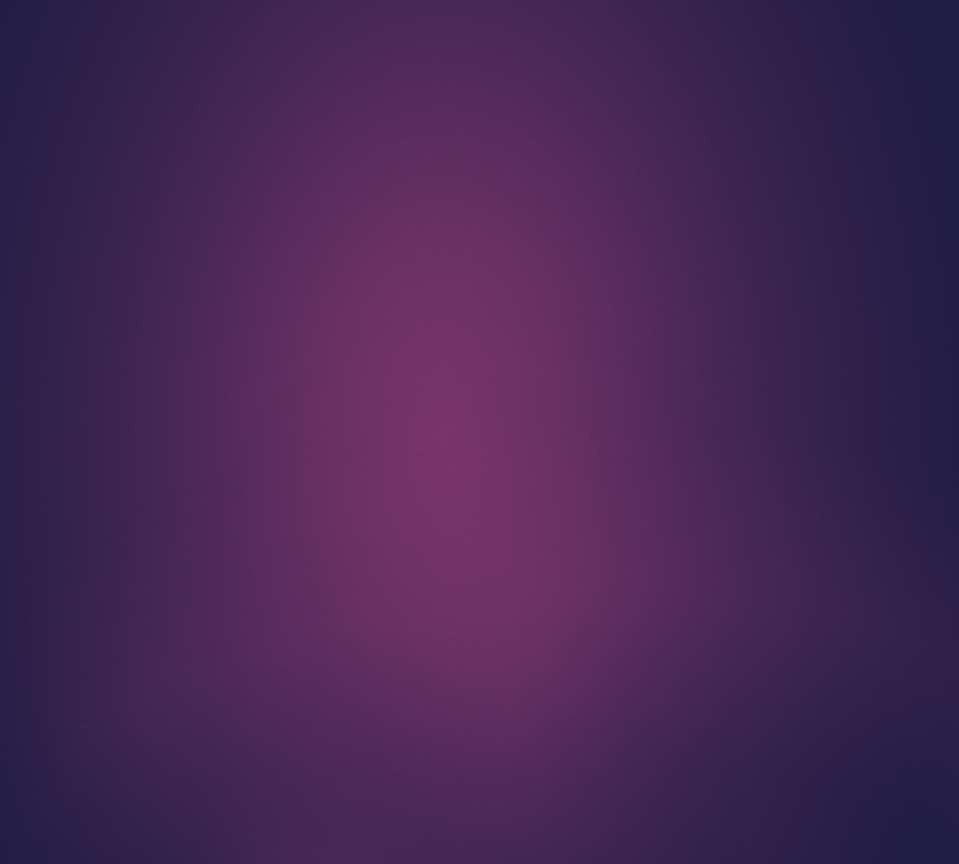
 The microinstruction generates microoperations to fetch instructions from main memory, to evaluate effective address, to execute the operations specified by instructions or to repeat the cycle for the next instruction.



# CONFIGURATION

The general configuration of microprogrammed control unit is demonstrated as follows:

**(ROM)**



# SEQUENCER AND PIPELINE REGISTER

 **SEQUENCER**:

The next address generator is called a sequencer, as it determines the

address sequence that is read from the control memory.

 FUNCTIONS:

* 1. Incrementing the control register
  2. Loading an address from control memory to CAR.
  3. Loading an initial address

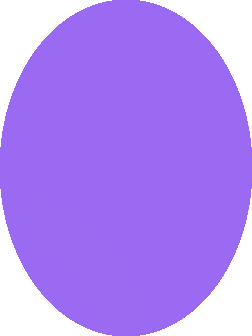
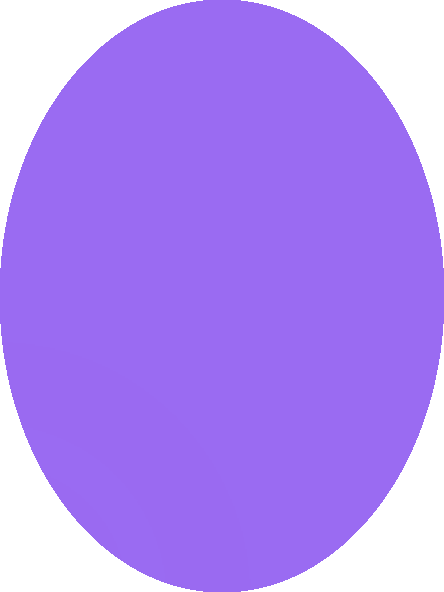
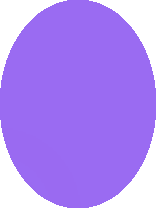
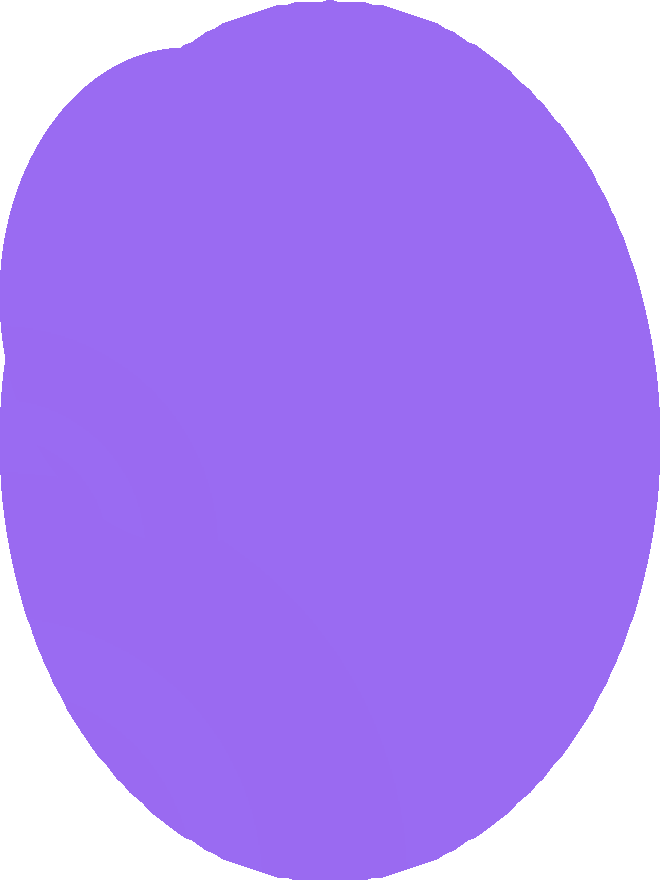
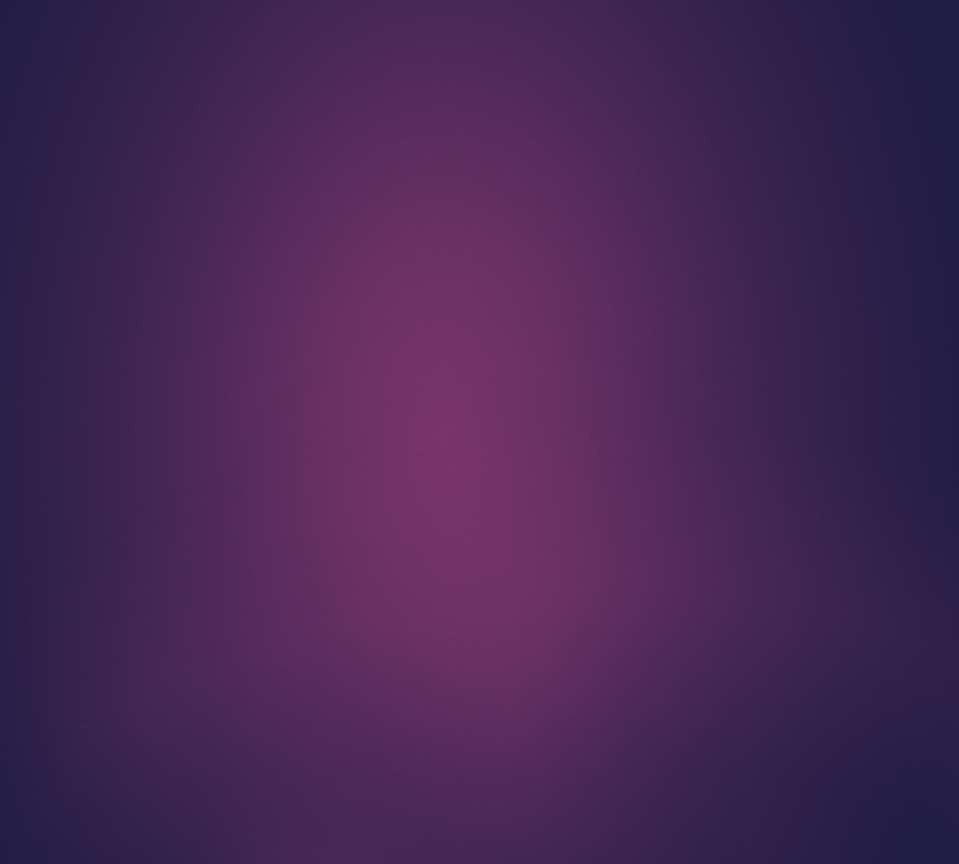
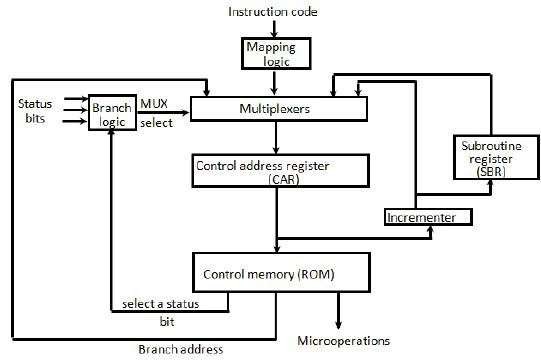
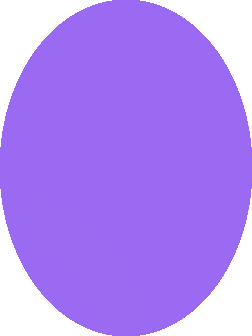
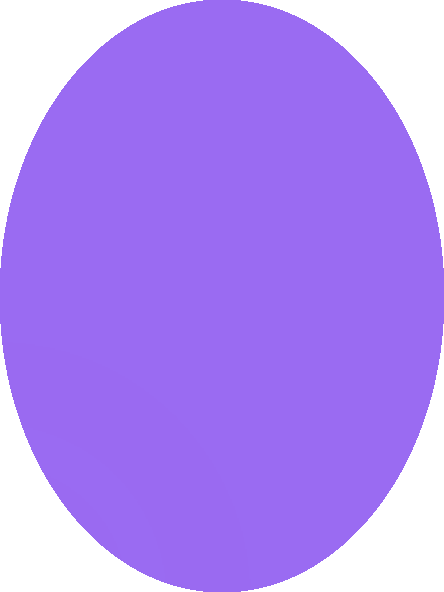
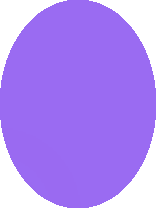
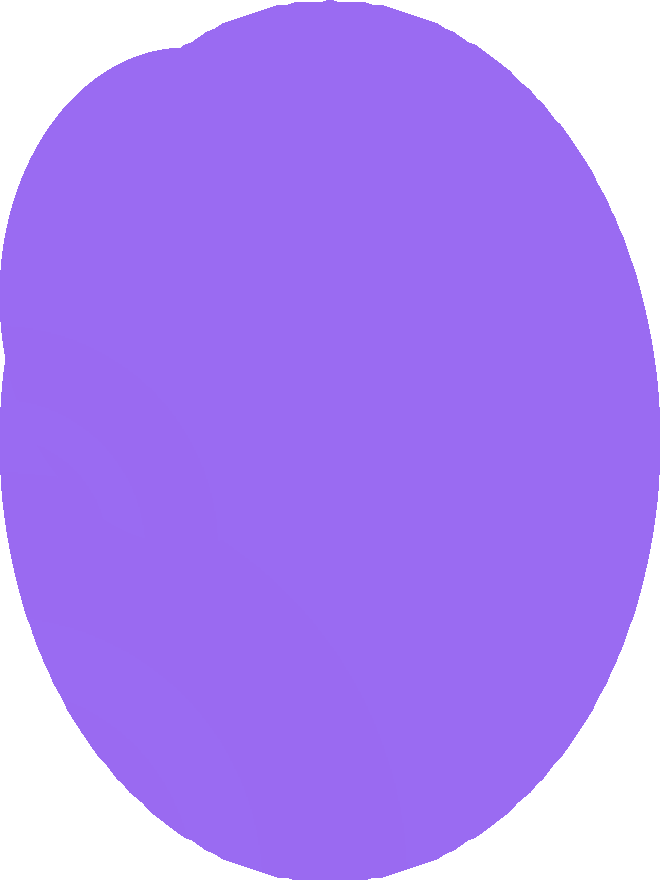
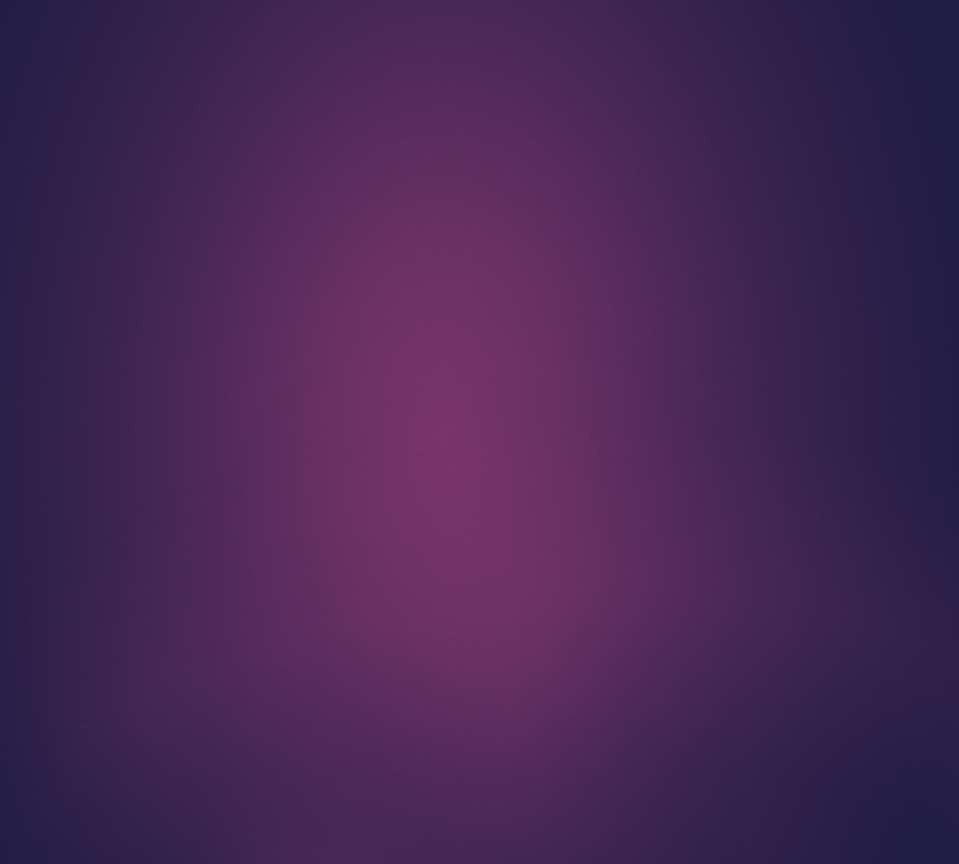
 **PIPELINE REGISTER** :

The control data register holds the present microinstruction while the next address is computed and read from the memory. This data register is called pipeline register.

 IMPORTANT ADVANTAGE OF MICROPROGRAMMED CONTROL :

No hardware or wiring changes is required.

# SELECTION OF ADDRESS FOR CONTROL MEMORY



ADDRESS SEQUENCING

 Micro instructions are stored in control memory in group with each group specifying a routine.

 An initial address is loaded into the control address register when power is turned on. The fetch routine may be sqenced by incrementing CAR. At end of fetch routine instructions is in IR .

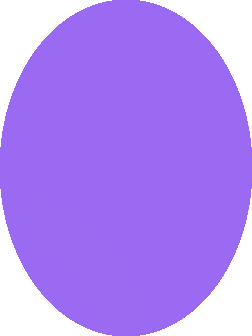
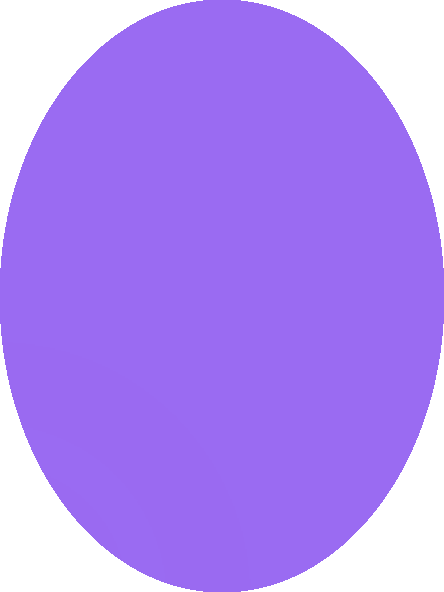
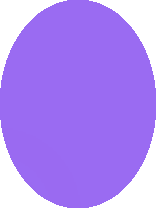
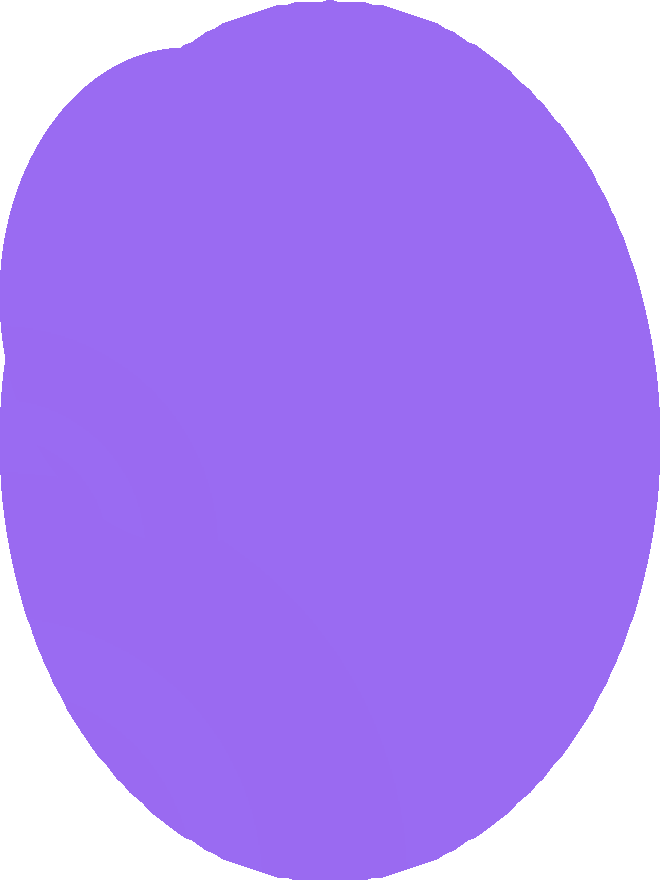
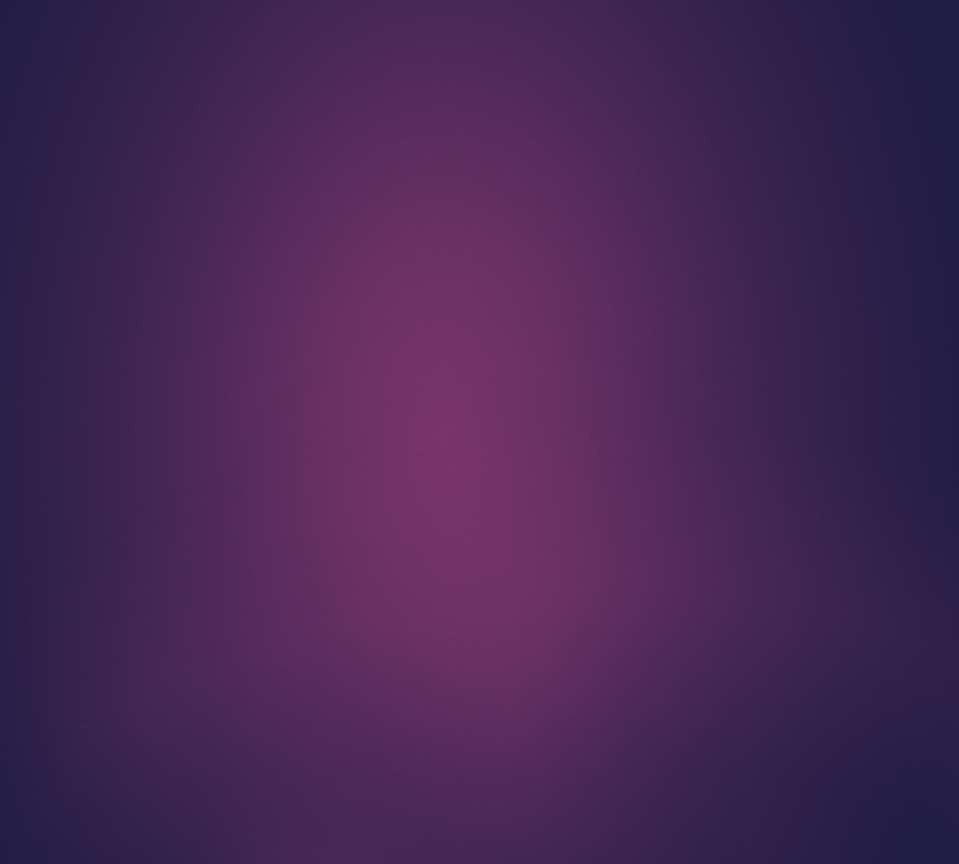
 The effective address computation routine in control memory can be reached through branch micro instructions which is conditioned on status mode bits of instructions.

 The micro operations steps to be generated in processor register depend

on opcode of instruction.

 Subroutine will require an external register for storing return address.

Return address cannot be stored in ROM because it has no writing capability.



# ADDRESS SEQUENCING

 The micro instructions in control memory contains a set of bits to initiate micro operations in registers and other bits to specify method by which next address is obtained.

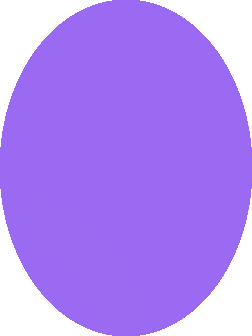
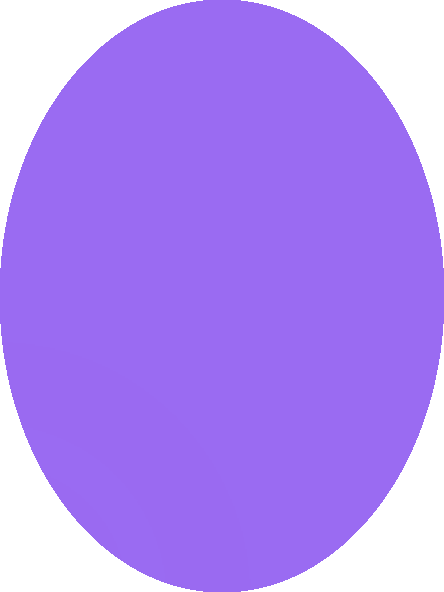
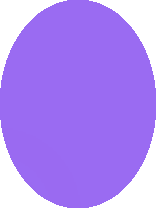
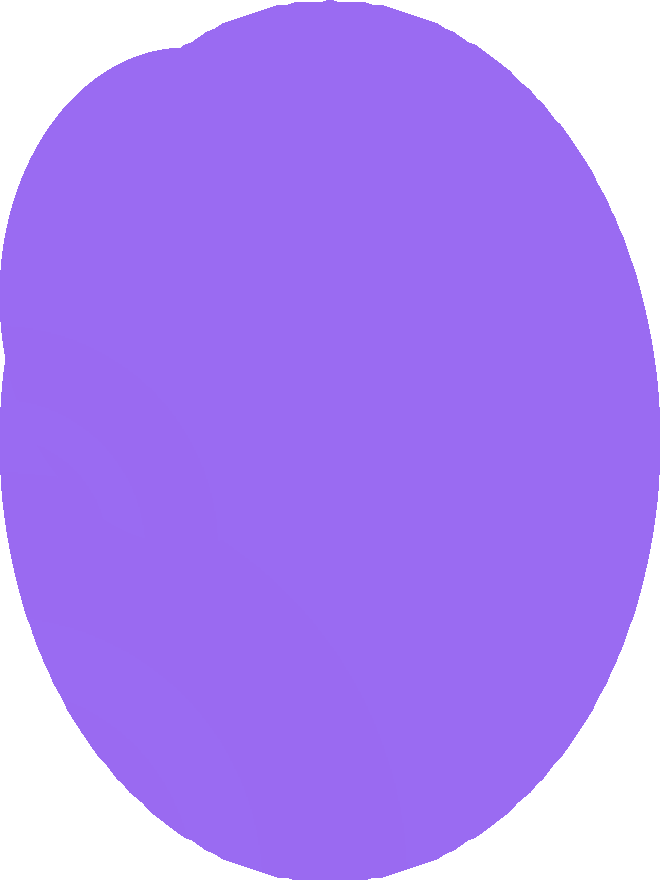
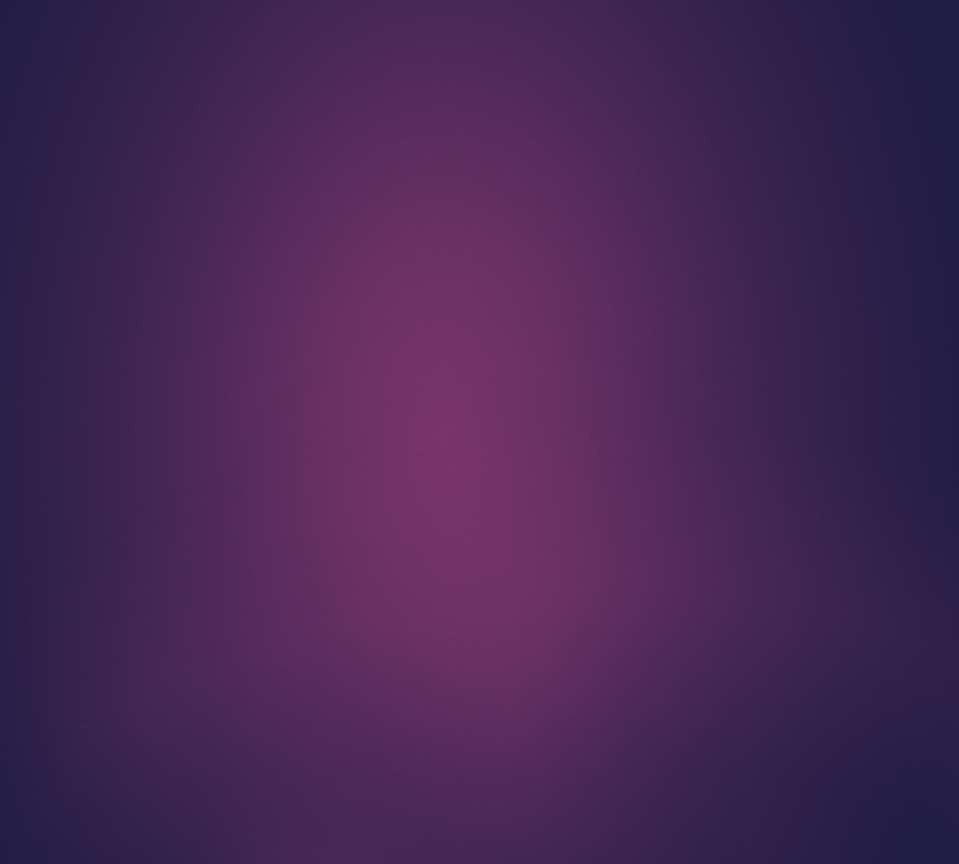
 The diagram shows four different paths from which the CAR receives the address.

 The incrementer increments the CAR content by one.

 Branching is achieved by specifying branch address in one of the fields of micro instructions.

 Conditional branching is obtained by status bit in order to determine it's condition.

 The return address for a sub routine is stored in special register (SBR) whose value is then used when microprogram wishes to return from the subroutine.



# STEPS IN SELECTION OF ADDRESS

 **CONDITIONAL BRANCHING**

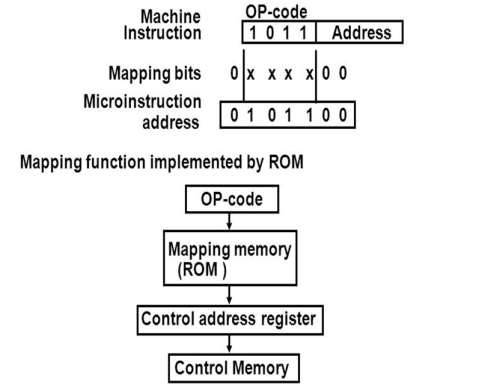
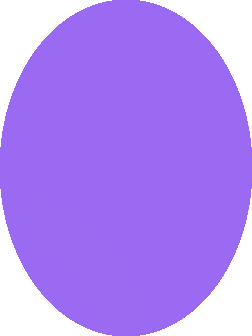
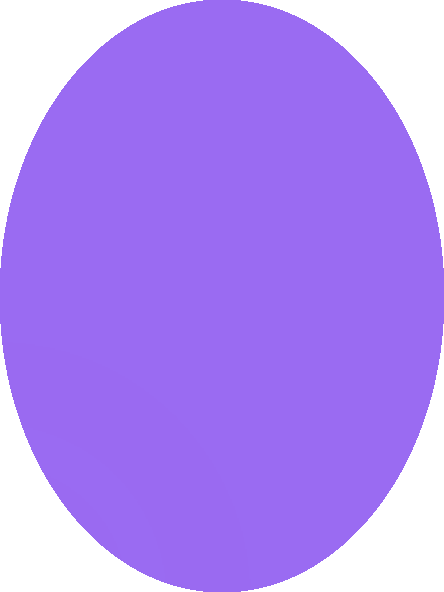
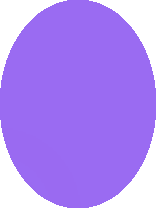
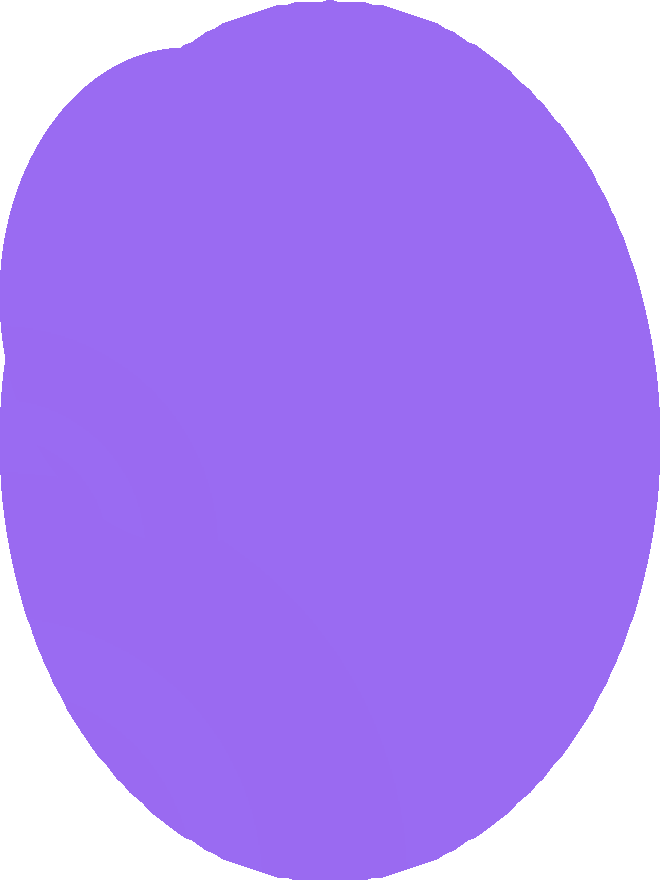
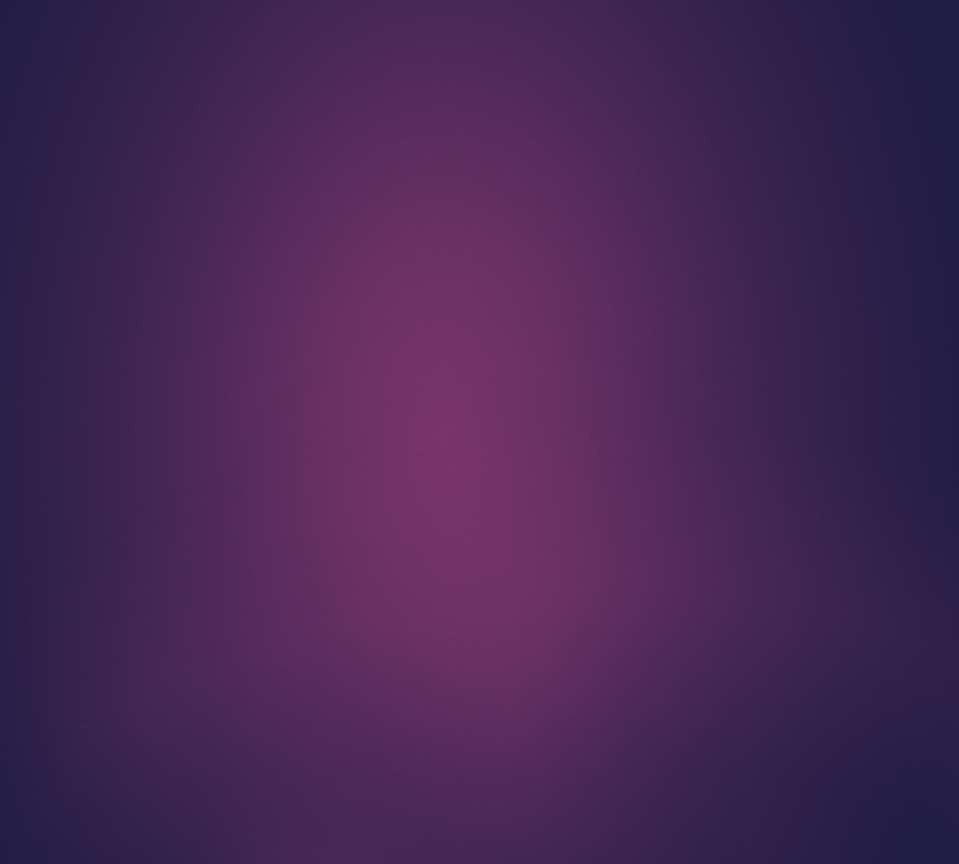
* branching from one routine to another depending on status bit conditions
* status bits provide parameter information.
* information in status bits are tested and actions are initiated based on their conditions:1or0

 **UNCONDITIONAL BRANCH**

* fix value of status bit to 1

 **SUBROUTINES**

* + - A set of common instructions that can be used in a program many times
    - Each time a subroutine is used in main program, a branch is made to the beginning of subroutine



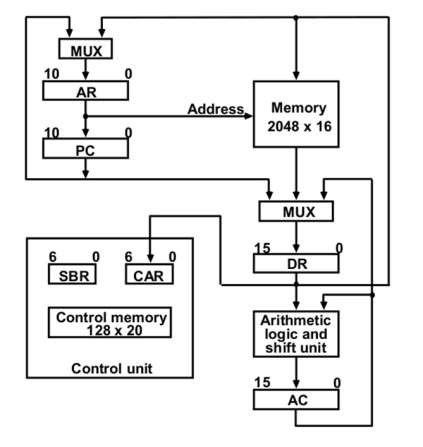
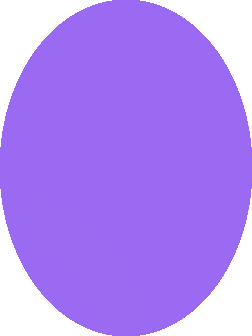
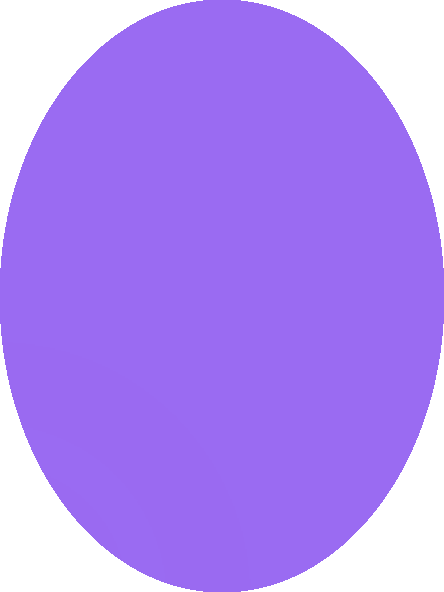
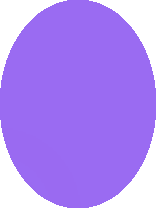
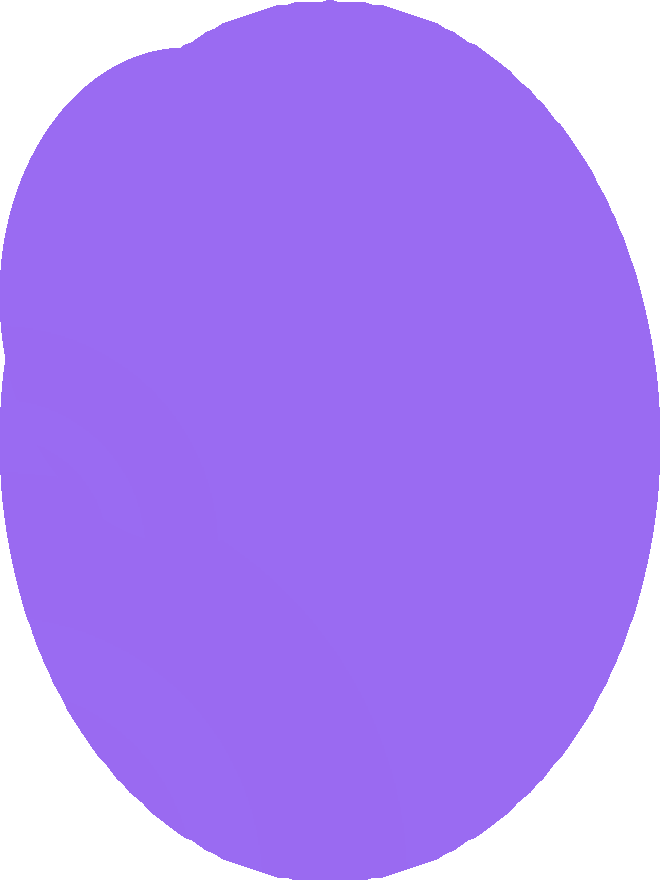
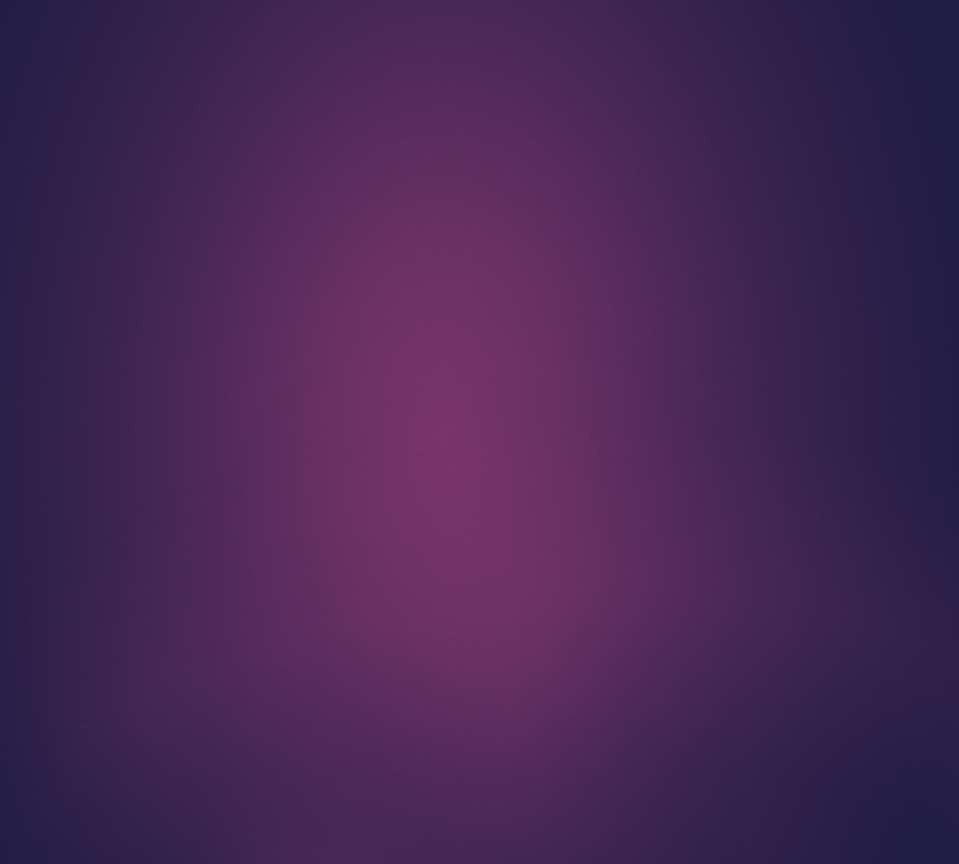
# MAPPING OF INSTRUCTION

 Each computer instruction has its own microprogram

 Routine stored in a given location of the control memory

 **MAPPING :**

* + - transformation from instruction code bits to microinstruction address in control memory where routine is located.



# COMPUTER HARDWARE CONFIGURATION

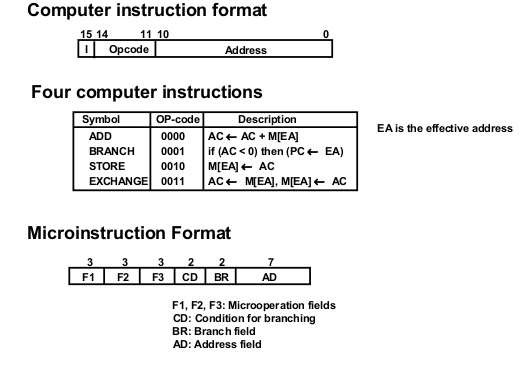
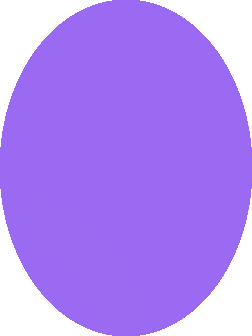
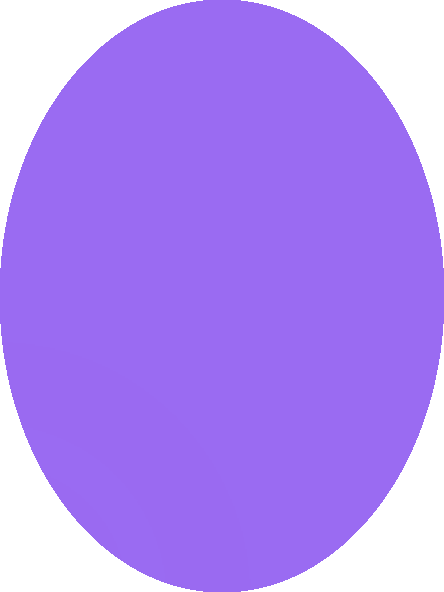
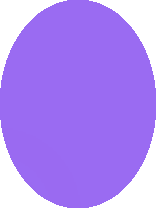
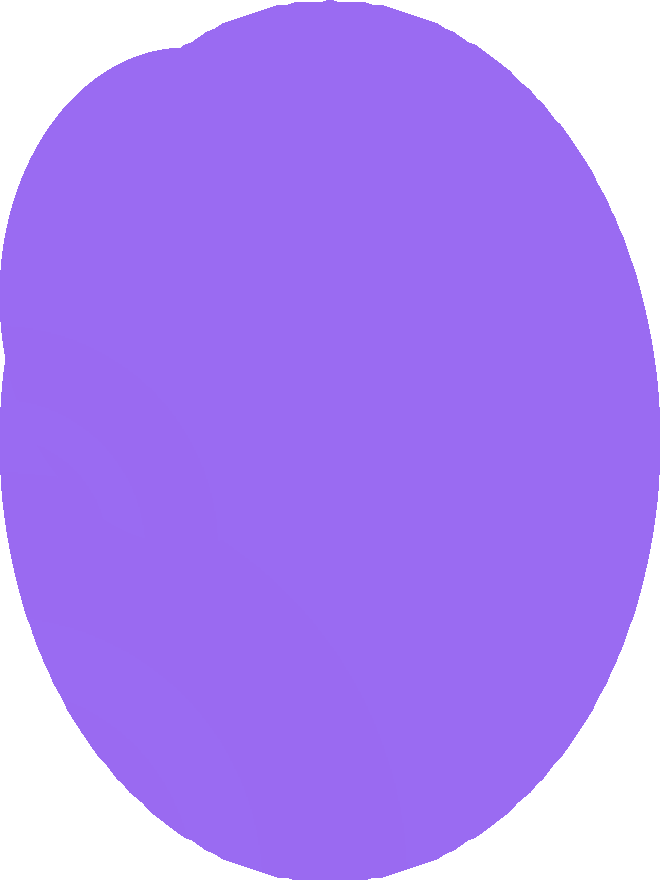
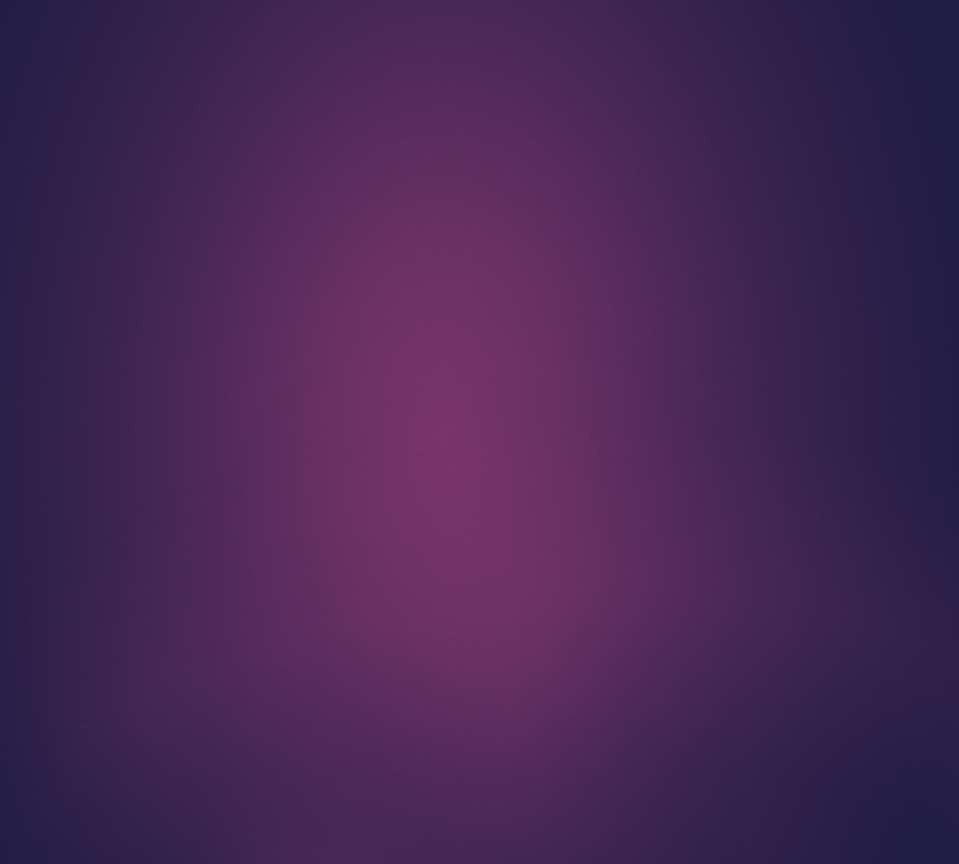
## MICROPROGRAM EXAMPLE:

 Two memory units: Main memory, control memory.

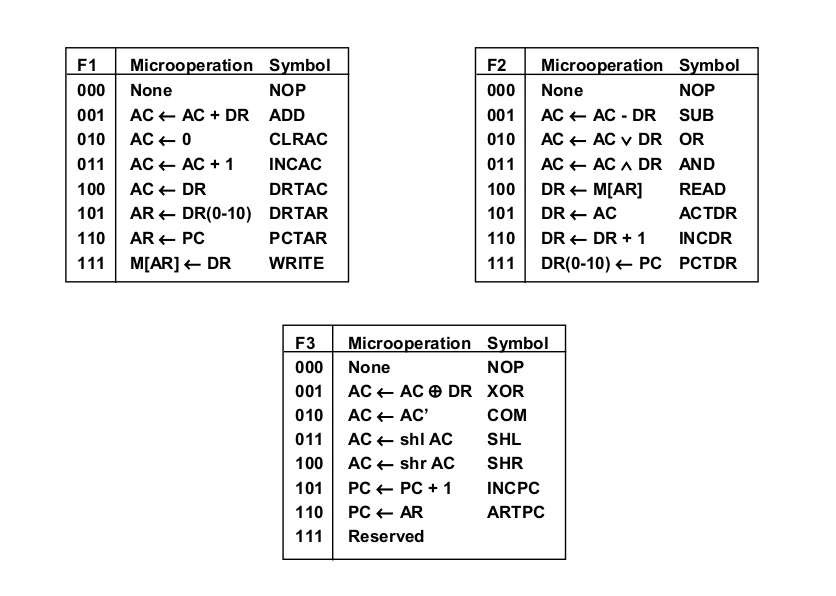
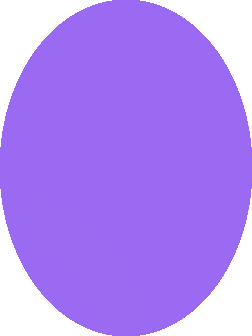
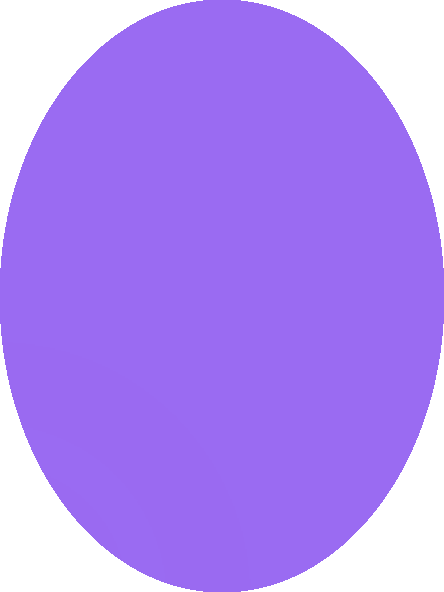
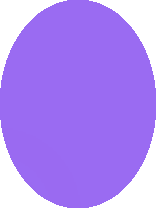
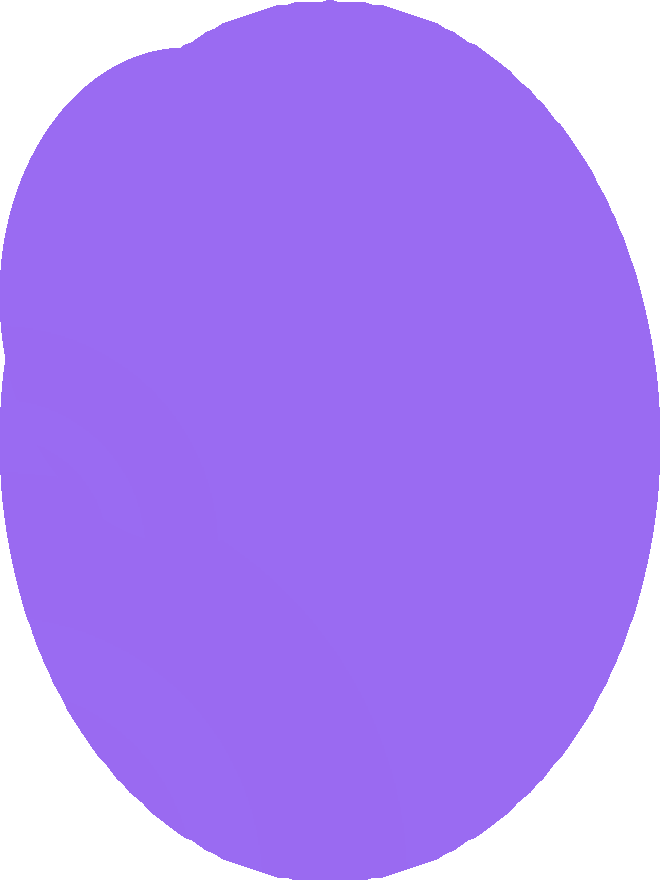
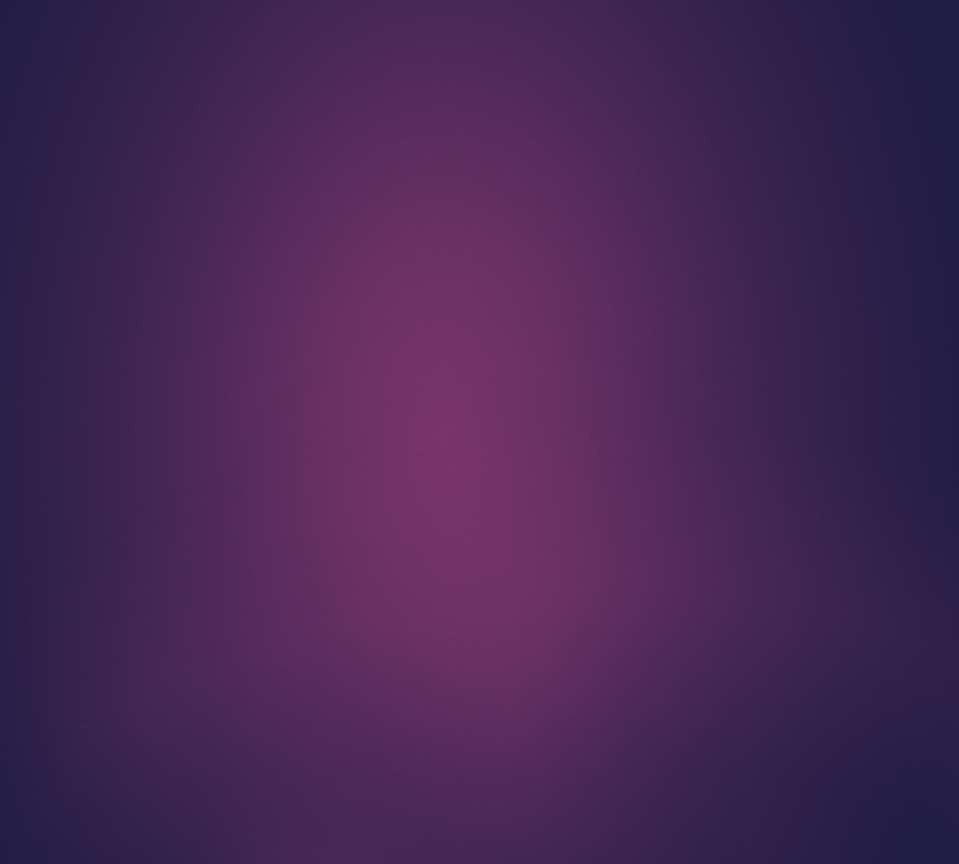
## - CONTROL MEMORY :

 4 registers are associated with processor unit (PC,AR,DR,AC)

 2 registers are associated with the control unit (CAR,SBR)

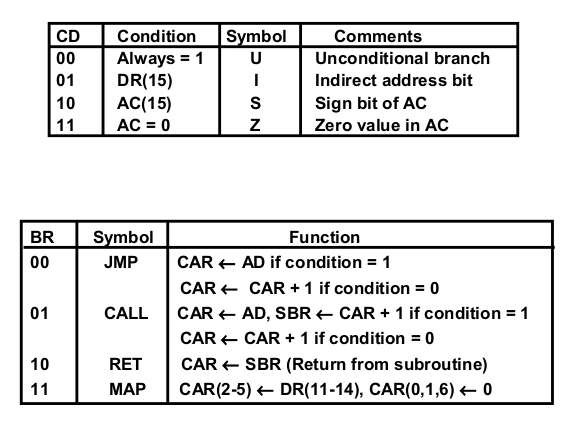
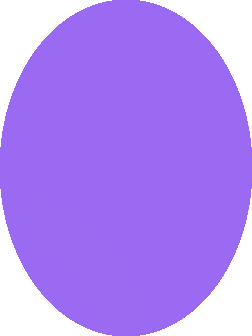
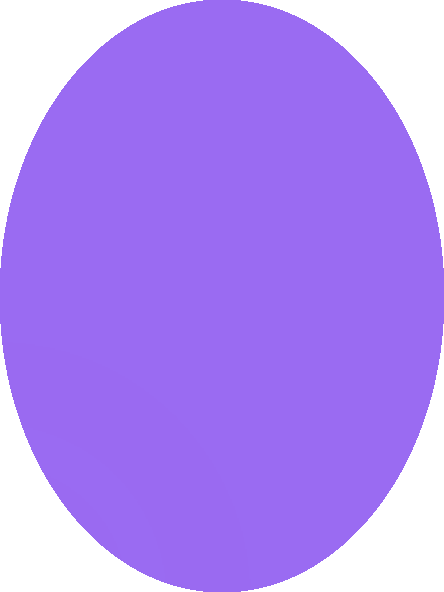
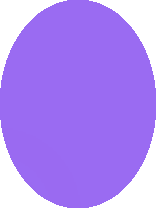
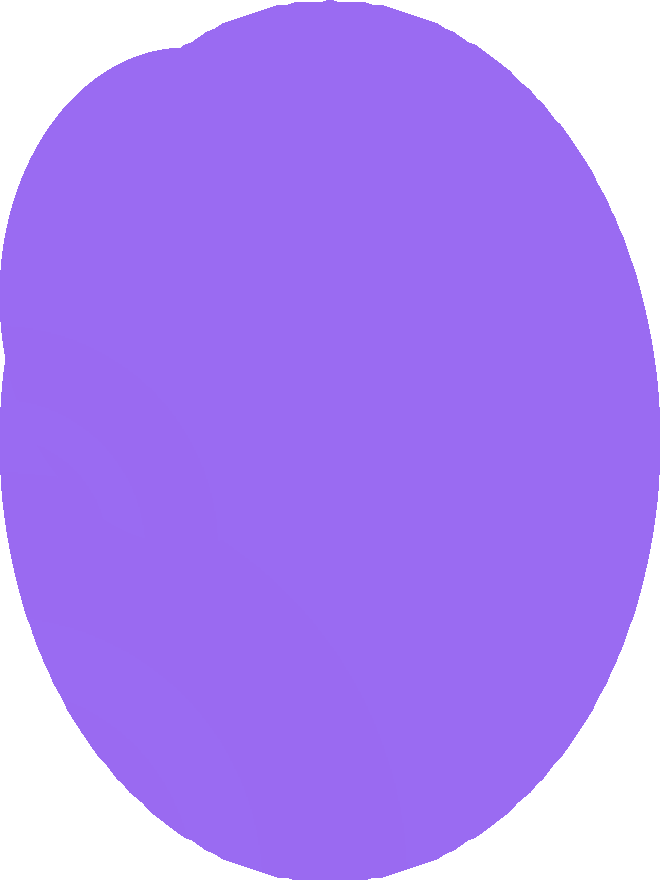
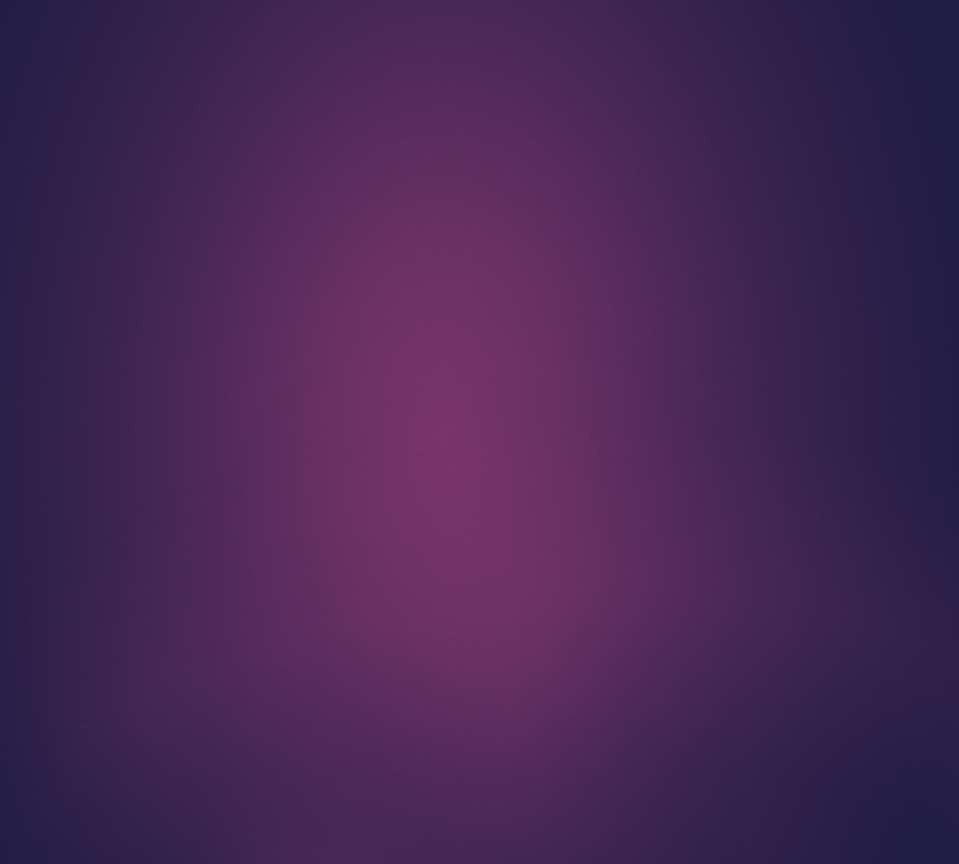


# MICROINSTRUCTION CODE FORMAT – 20 BITS

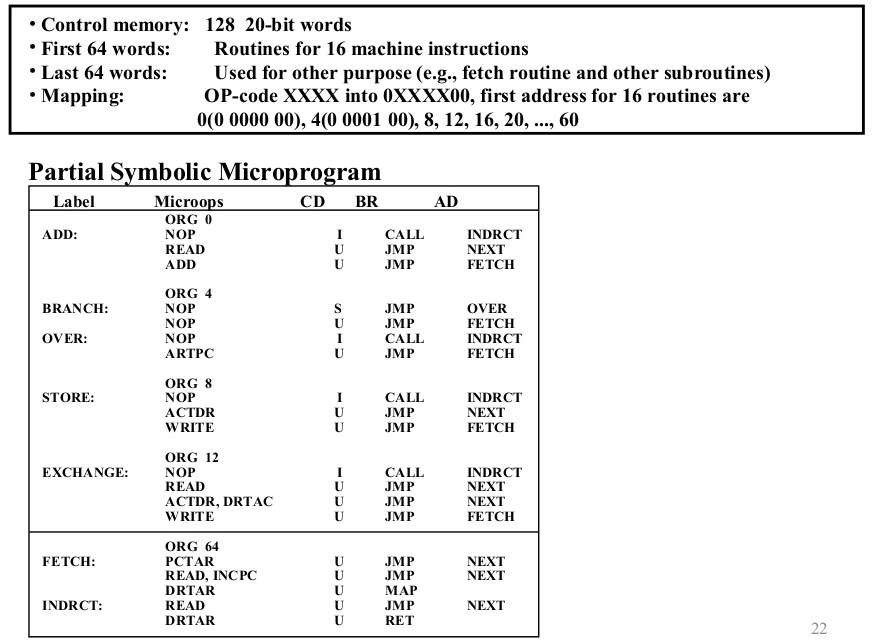
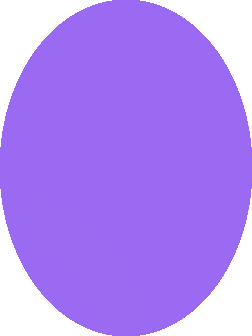
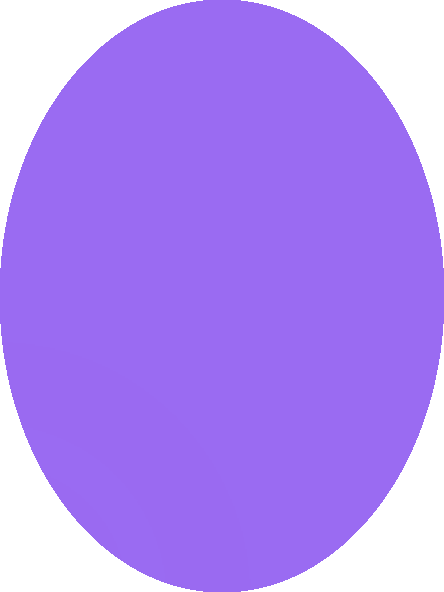
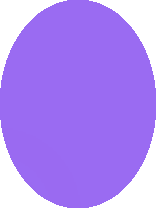
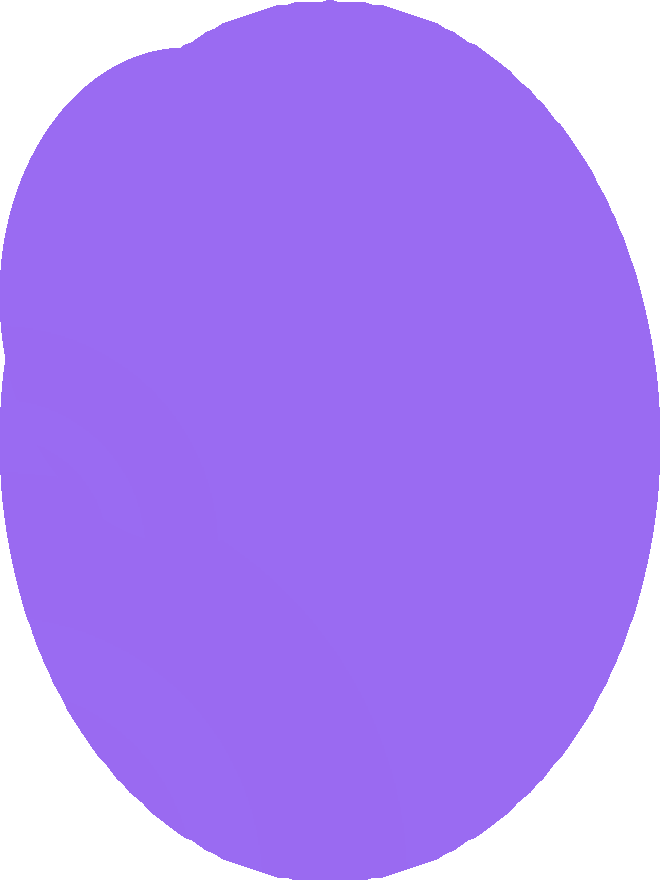
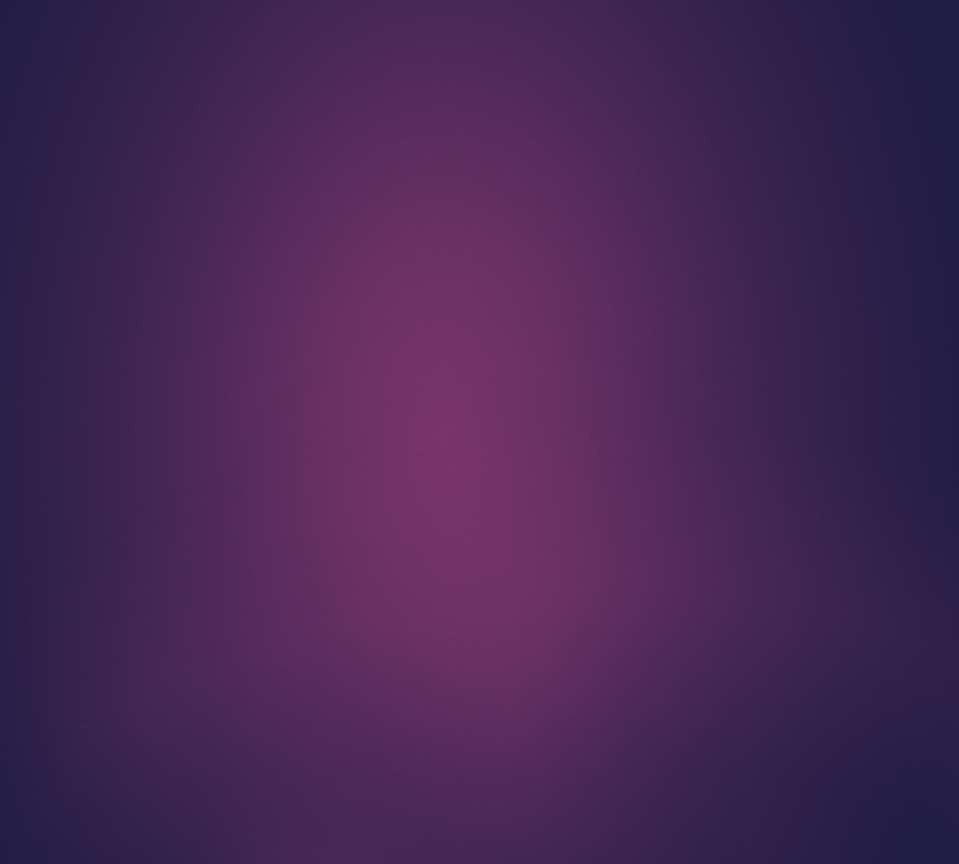


SYMBOLS & BINARY CODE – I F1,F2,F3

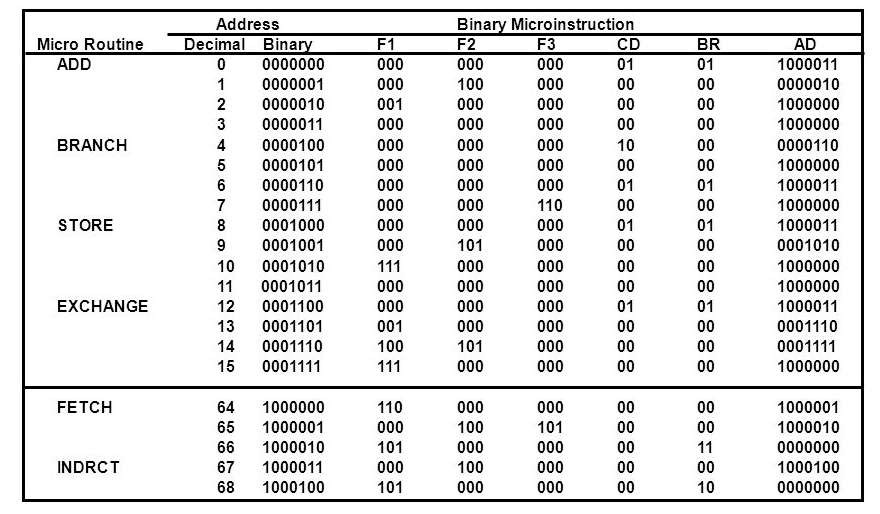
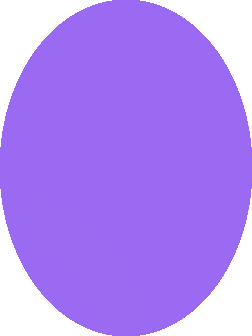
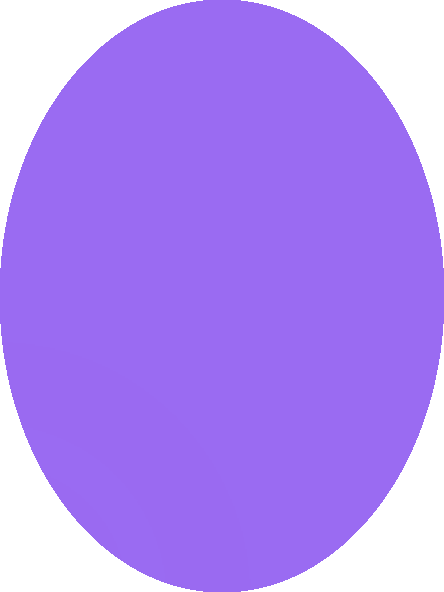
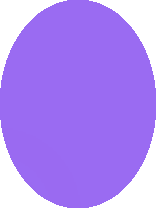
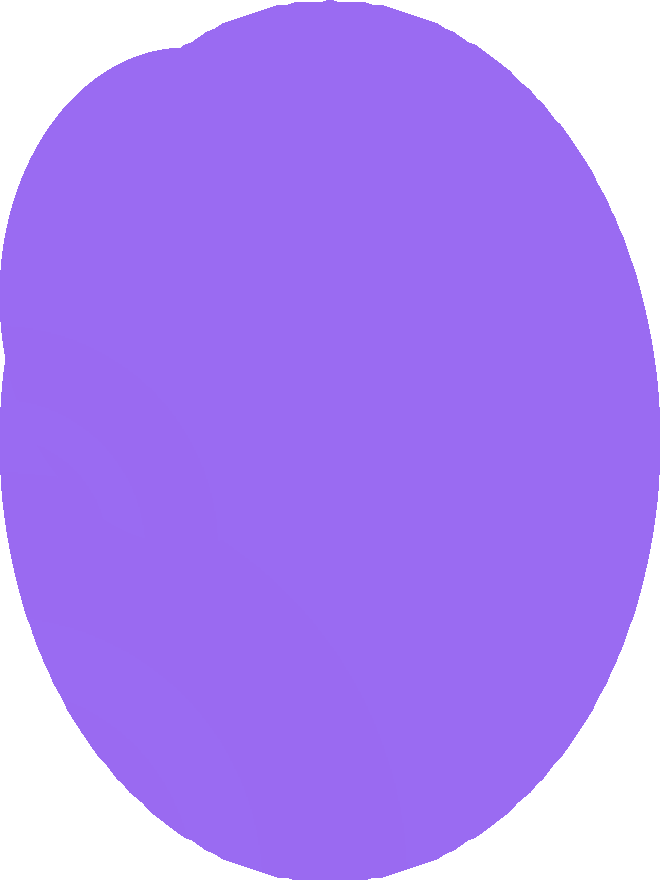
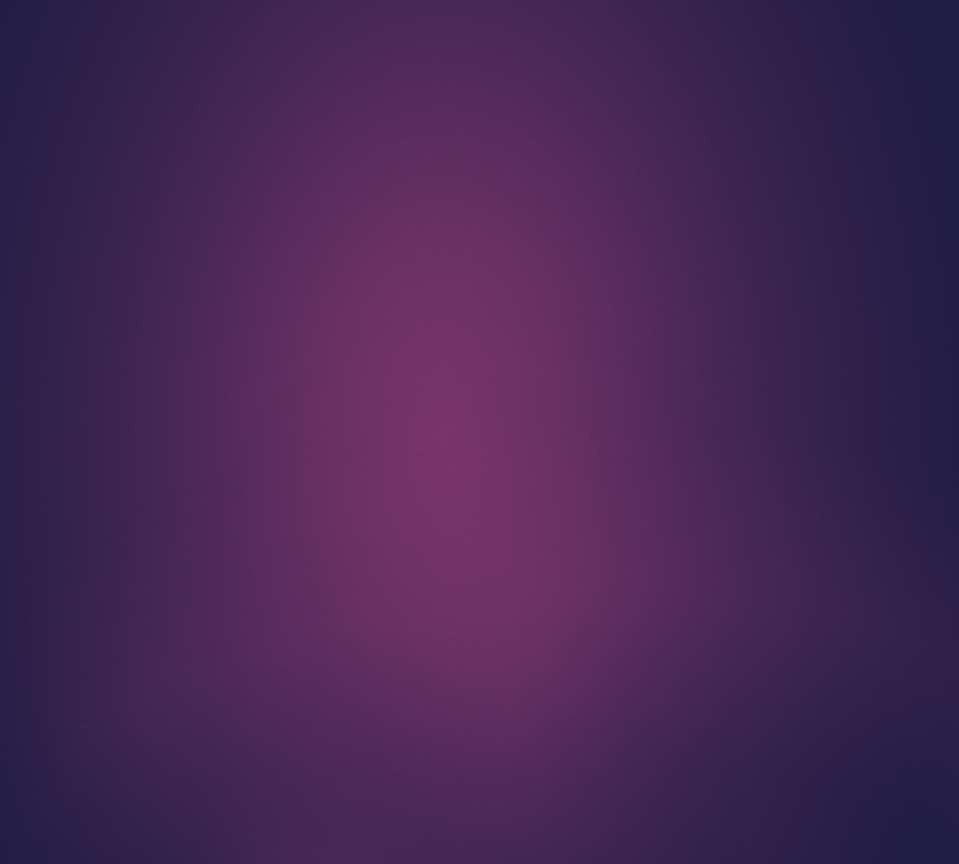
SYMBOLS & BINARY CODE – II



CD,BR



SYMBOLIC MICROPROGRAM



BINARY MICROPROGRAM

