

Computer Organization

Control Unit



Department of Computer Science
Missouri University of Science & Technology
hurson@mst.edu

Computer Organization

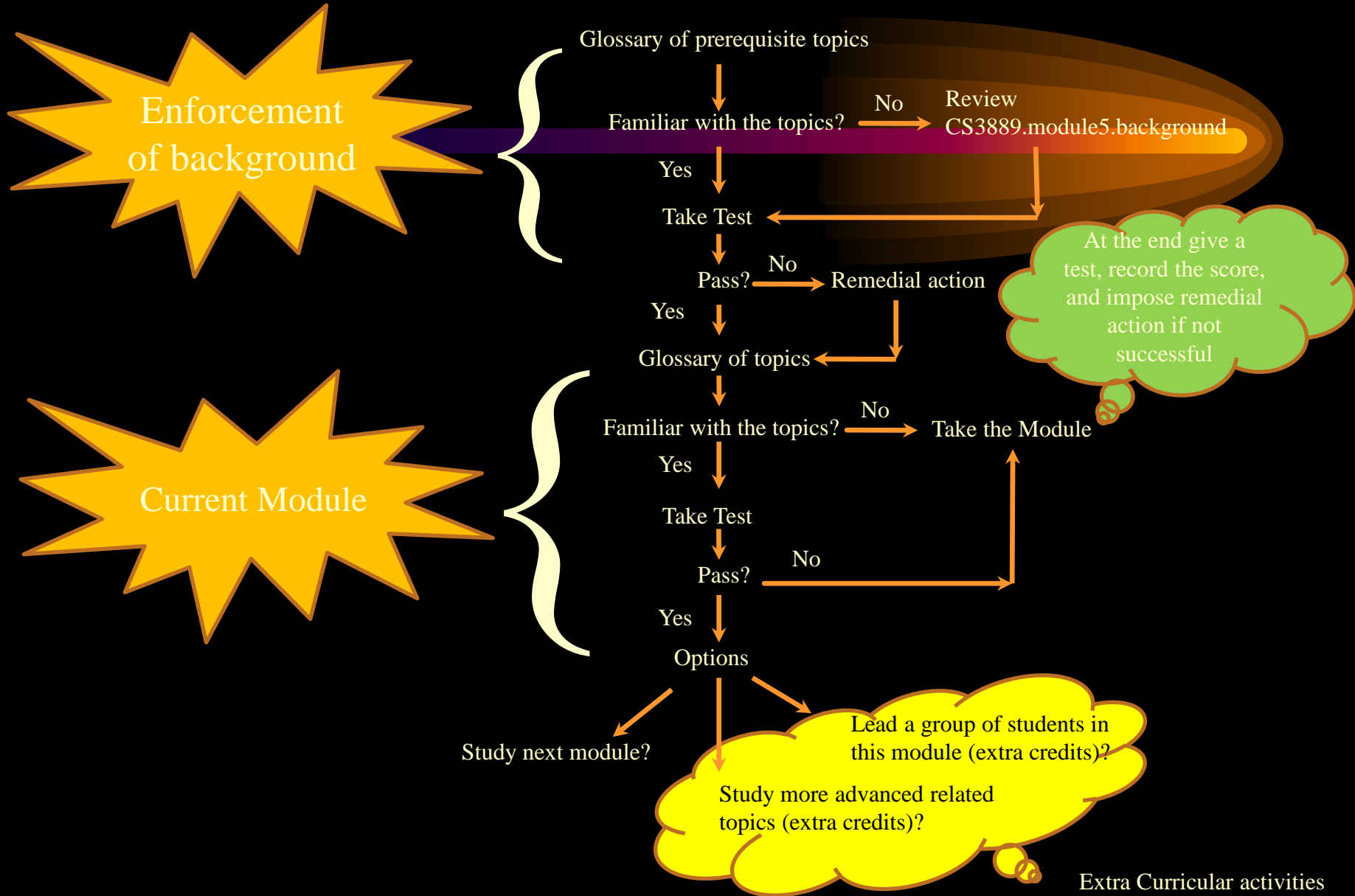


Note, this unit will be covered in three lectures. In case you finish it earlier, then you have the following options:

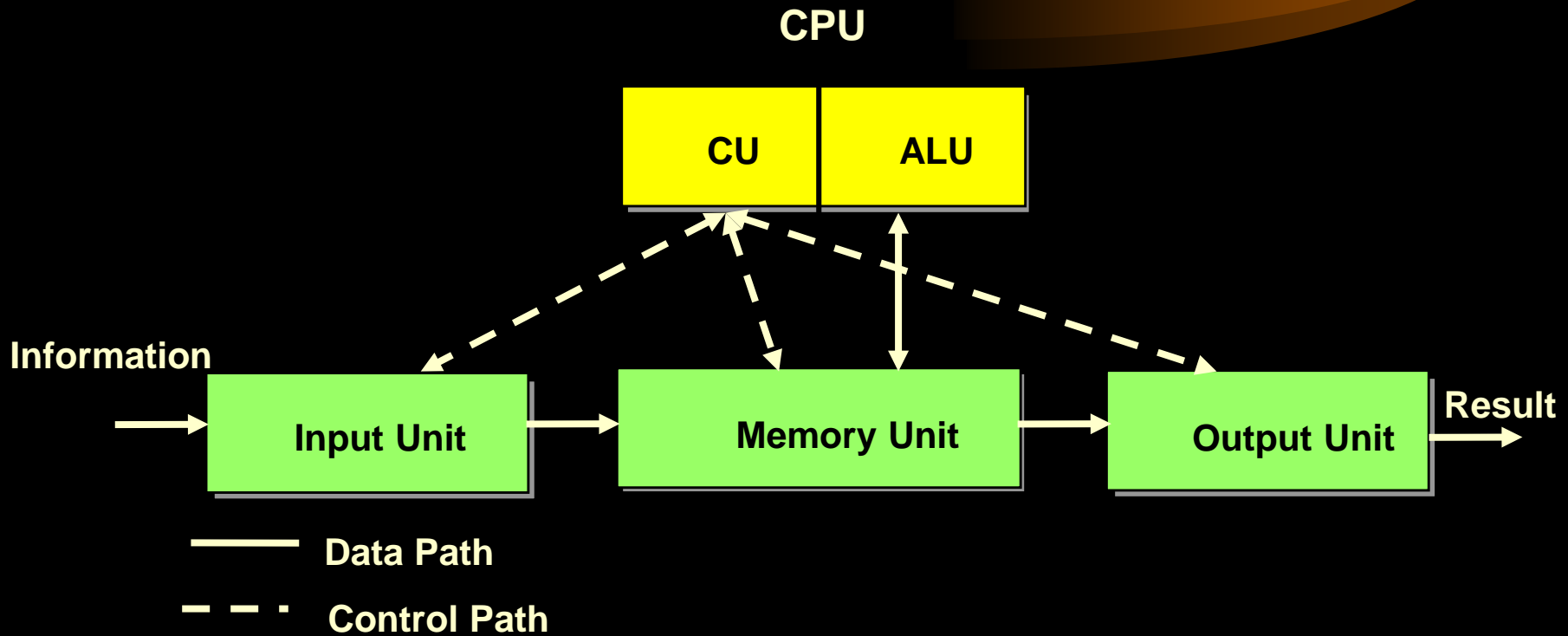
- 1) Take the early test and start CS3889.module6
- 2) Study the supplement module (supplement CS3889.module5)
- 3) Act as a helper to help other students in studying CS3889.module5

Note, options 2 and 3 have extra credits as noted in course outline.

Computer Organization



Computer Organization



Computer Organization



◆ **Control Unit**

★ In general control unit has two major tasks:

● **Instruction Sequencing:** the way instructions are selected.

● **Instruction Interpretation:** the way instructions are executed.

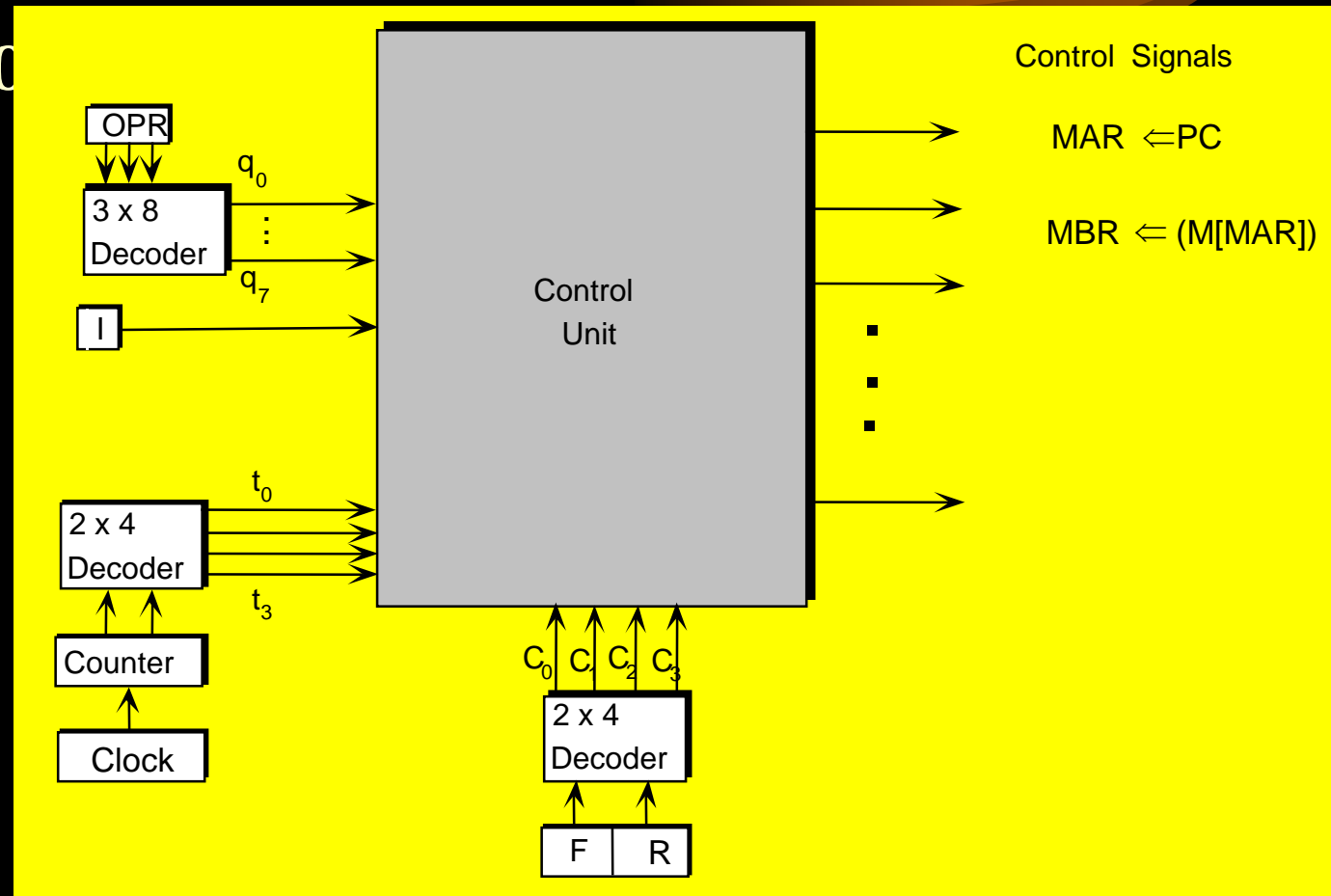
Computer Organization

◆ Control Unit — Instruction Interpretation

- ★ Each assembly (machine) level instruction is an **orderly sequence** of μ -operations.
- ★ Assembly (machine) level instructions are executed by means of **control functions** (control signals).
- ★ Each control function is a Boolean function which enables the execution of a μ -operation.
- ★ It is the task of control unit to generate control signals in an orderly fashion.
- ★ A control unit is the realization of control functions. It is simply a **decoder** that based on an input pattern, generates a proper output pattern, where each output signal is a control signal.

Computer Organization

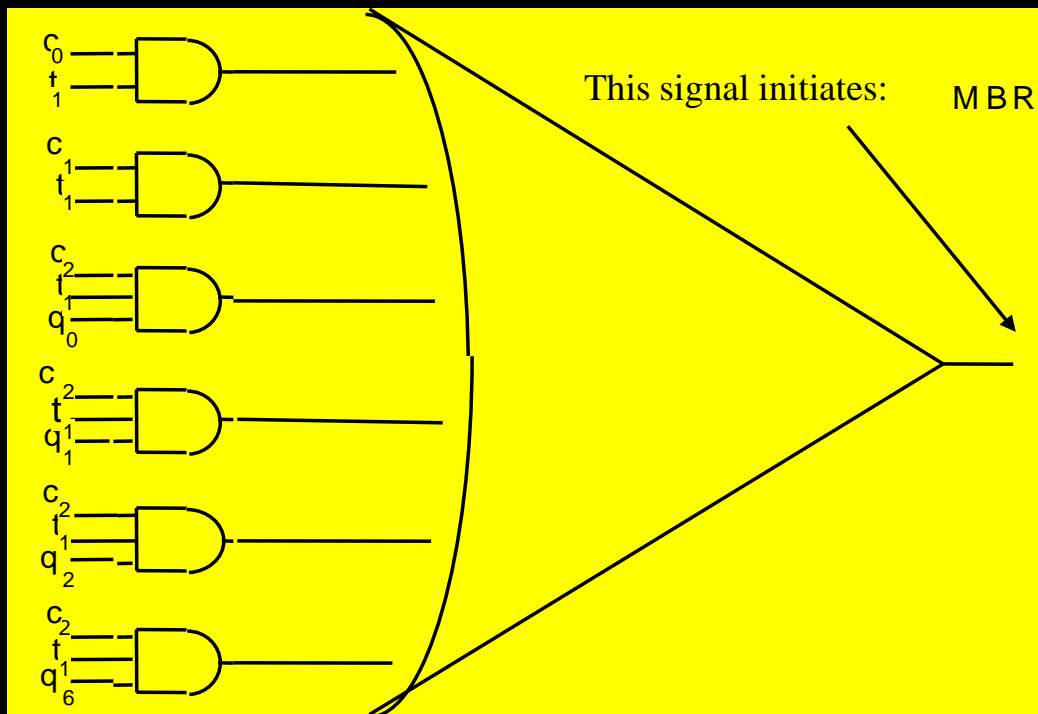
◆ Control



Computer Organization

◆ Control Unit – Simple machine

- ✳ In the simple machine, main memory has been repeatedly read in various routine.



$c_0t_0:$	$MAR \leftarrow PC$	
$c_1t_0:$	$MAR \leftarrow (MBR(\text{ADDRESS}))$	$C) + 1$
$q_0c_2t_0:$	$MAR \leftarrow (MBR(\text{address}))$	$\text{r}(I)$
$q_0c_2t_1:$	$MBR \leftarrow (M[MAR])$	
$q_0c_2t_2:$	$AC \leftarrow (AC) \wedge (MBR)$	
$q_0c_2t_3:$...	

Computer Organization

◆ Control Unit

- ★ Naturally, such an ad hoc approach is justifiable when dealing with simple organizations. In case of complex systems one needs to use a more **systematic approach** in the design of control unit. Two approaches have been proposed:
- ★ **Hardwired**: Realization of the control unit as a sequential device.
- ★ **μ-Programming**: Realization of the control unit by means of programming routines.

Computer Organization



- ◆ **Control Unit — Hardwired**
 - ★ State Table Method
 - ★ Delay Element Method
 - ★ Sequence Counter
 - ★ Programmable Logic Array

Computer Organization

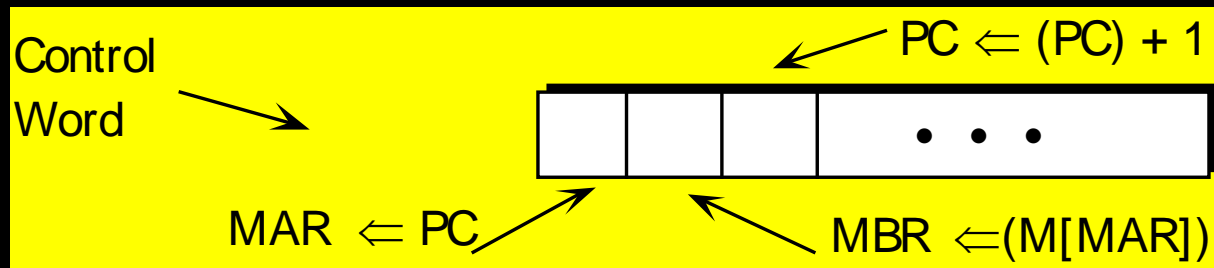
◆ Control Unit — μ -Programmed

- ★ As one can visualize, a hardwired control unit is a "big decoder" which, at each moment in time, based on the input pattern (system's status) generates a bit pattern as output (control signals). This bit pattern then activates the proper functional boxes.
- ★ A microprogrammed control unit is simply a mechanism which allows to:
 - i) Store all possible output bit patterns in a memory, and
 - ii) Read them in an orderly fashion.

Computer Organization

◆ Control Unit — μ -Programmed

- ★ Suppose we have a memory (control memory), like any other memory organization, composed of memory words (control word).
- ★ Further, each bit of a memory word is designated by a mnemonic name which represents a specific function:



Computer Organization

◆ Control Unit — μ -Programmed

- ★ It is also assumed that we have a facility to set up the control words by different patterns:

1	0	0	0	• • •	0
0	1	1	0	• • •	0
•					
•					
•					

Computer Organization

◆ **μ-Programmed Control Unit**

- ★ If one has a mechanism to fetch the control words and interpret their contents, then fetching the previous two memory words in proper order, generates the sequence of control functions needed to perform fetch cycle.

Computer Organization

◆ μ -Programmed Control Unit — Hardware Requirements

- ★ **Control Store** to store micro instructions. This storage could be a part of main memory or a dedicated storage medium.
- ★ **Hardware facility** to fetch, decode and execute micro instructions.
- ★ **Mechanism** to map the operation codes (at assembly level) to the proper addresses in the control store.

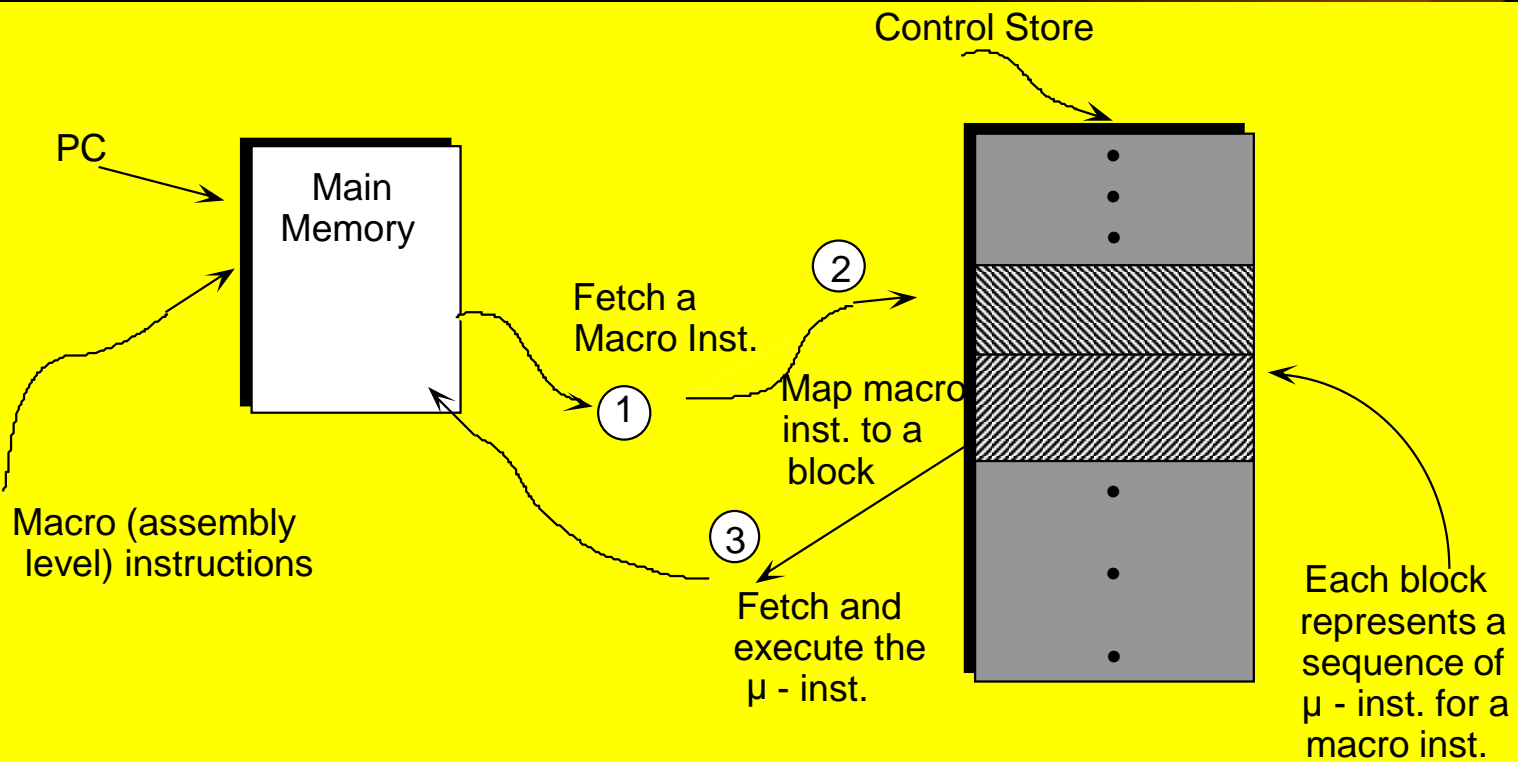
Computer Organization

◆ **μ-Programmed Control Unit – Flow of Control**

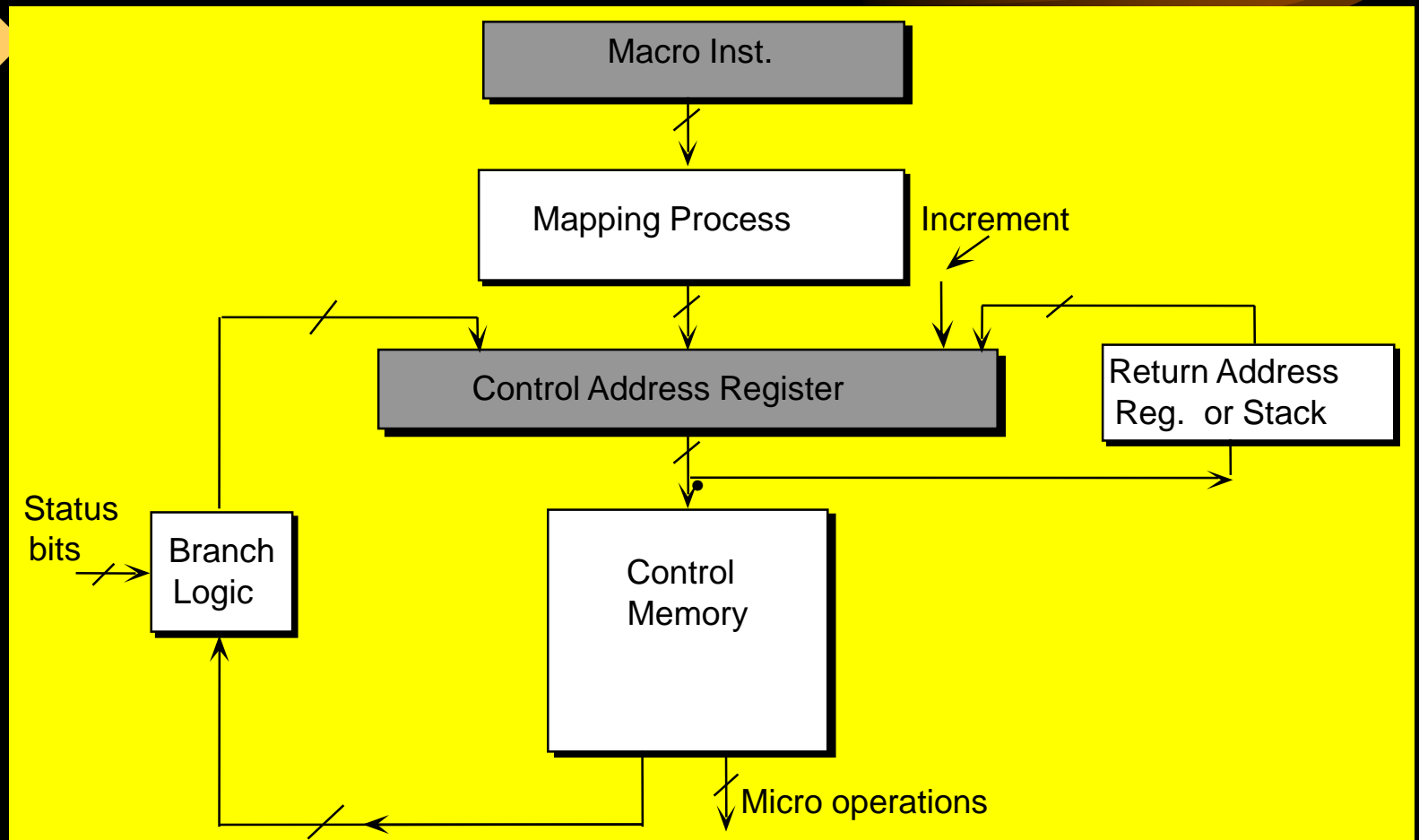
★ Micro instructions are stored in groups. Each group of micro instructions represents a macro instruction (assembly instruction). Similar to a "case statement", the operation code of a macro instruction transfers control to a designated collection of micro instructions.

- A macro instruction is fetched.
- The operation code is mapped into a block of the control store which represents that macro instruction.
- The designated micro instructions are fetched and executed in an orderly fashion.

Computer Organization



Computer Organization



Computer Organization

◆ μ -Programmed Control Unit — μ -Instruction Format

- ★ A μ -instruction (in general) is composed to two parts:
control field and **address field**.
 - Control field is used to activate functional units.
 - Address field is used to fetch μ -instructions in an orderly fashion. However, similar to the concept of the program counter, the address field can be eliminated at the expense of a register called micro program counter.
- ★ In some cases control words have additional field(s) called Emit-Field(s) - say to initialize a register.

Computer Organization

◆ Questions

- ★ Compare and contrast μ -programmed and hardwired control units against each other.
- ★ How can we reduce the length of the μ -instructions?
- ★ Discuss the factors that could effect the length of the μ -instructions.

Computer Organization



◆ μ -Programmed Control Unit — μ -Instruction Format

★ Based on the degree of parallelism and representation of μ -operations, μ -instructions can be classified into three groups:

- Horizontal
- Vertical
- Diagonal

Computer Organization

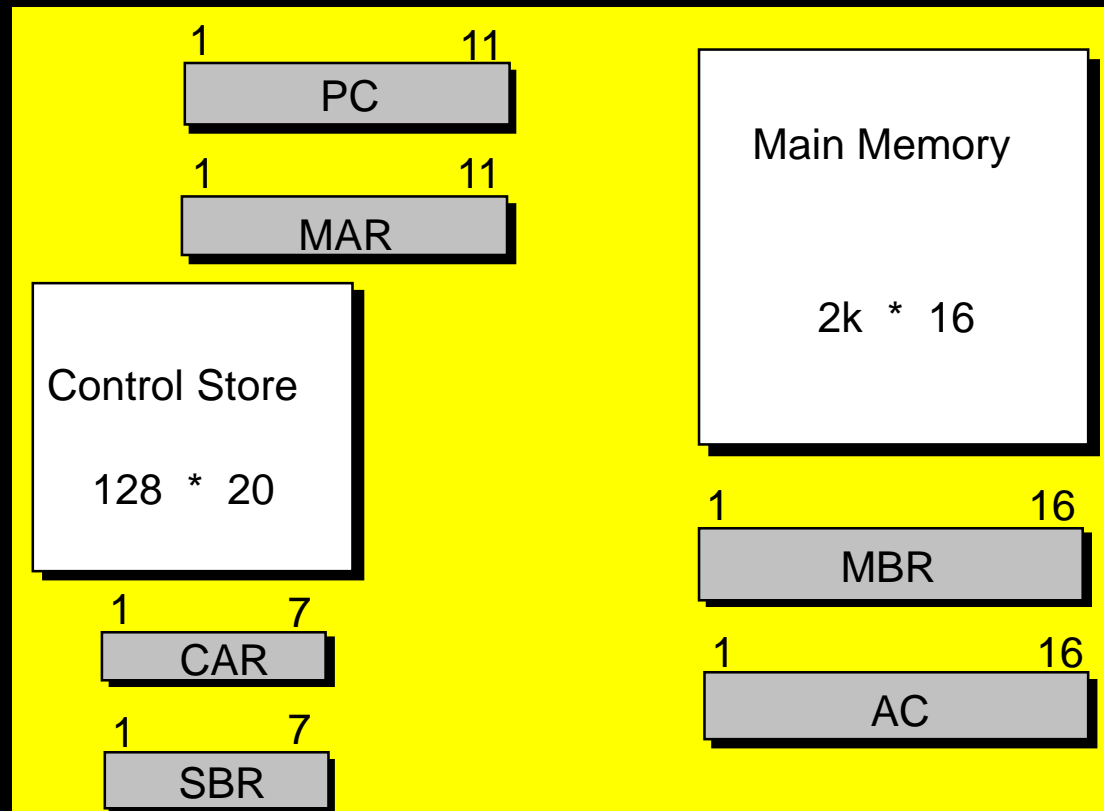
◆ μ -Programmed Control Unit — μ -Instruction Format

- * **Horizontal:** each bit of the control part represents a distinct μ -operation. A horizontal μ -instruction is characterized by:
 - Long format.
 - Ability to express a high degree of parallelism.
 - Little encoding of the control information.
- * **Vertical:** control part represents just one μ -operation. A vertical format is characterized by:
 - Short format
 - No degree of parallelism
 - Considerable encoding of the control information
- * **Diagonal:** it is a compromise between horizontal and vertical formats. It is characterized by:
 - Moderate length format
 - Some degree of parallelism
 - Some encoding of the control information

Computer Organization

◆ μ -Programmed Control Unit – An Example

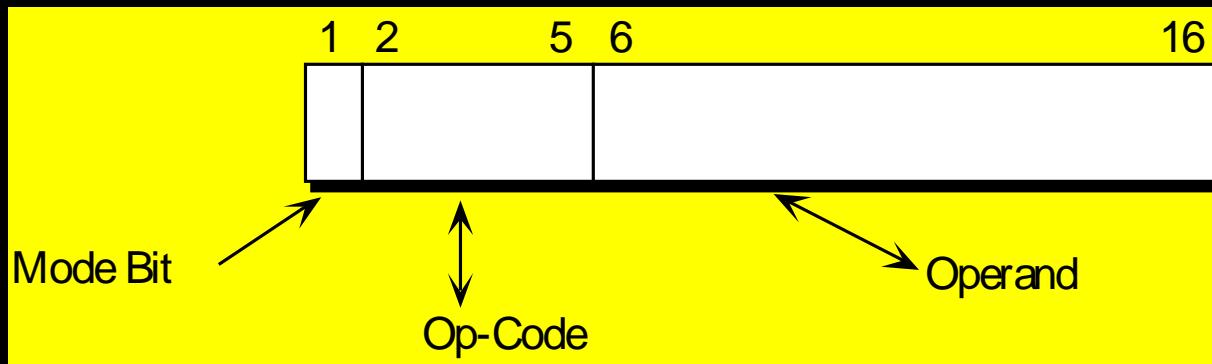
★ The computer configuration is as follows:



Computer Organization

◆ μ -Programmed Control Unit — An Example

★ Macro Instruction Format



Computer Organization

◆ μ -Programmed Control Unit – An Example

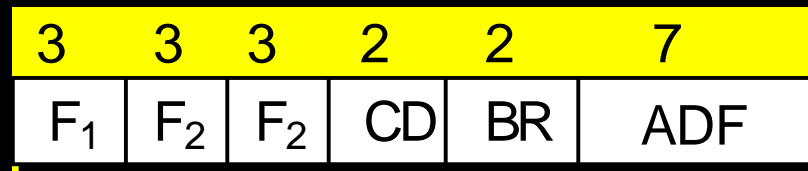
★ Sixteen possible instructions, such as

ADD	0000	$AC \leftarrow (AC) + (M)$
BRANCH	0001	if $AC < 0$ then $PC \leftarrow$ effective address
STORE	0010	$M \leftarrow (AC)$

Computer Organization

◆ μ -Programmed Control Unit – An Example

★ μ -Instruction Format



★ F₁, F₂, and F₃ fields are the control part. Having F₁, F₂, and F₃ as part of control part implies some degree of parallelism.

- 000 100 101 implies 2 micro operations
- F₁ = 000 \Rightarrow no operation
- F₂ = 100 \Rightarrow MBR \leftarrow (M[MAR])
- F₃ = 101 \Rightarrow PC \leftarrow (PC) + 1
- CD, BR and ADF fields collectively represent a (conditional) transfer control.

Computer Organization

◆ μ -Programmed Control Unit – An Example

★ CD is 2-bits long and defines status conditions:

CD			Condition
00	U	unconditional	1
01	I	Indirect mode bit	MBR(I)
10	S	Sign of AC	AC(S)
11	Z	Zero value in AC	AC = 0

Computer Organization

◆ μ -Programmed Control Unit – An Example

★ BR is 2-bits long and defines different types of branches:

BR

00	JMP	$CAR \leftarrow (ADF)$ if condition is true $CAR \leftarrow (CAR)+1$ otherwise
01	CALL	$CAR \leftarrow (ADF), SBR \leftarrow (CAR)+1$ if condition is true $CAR \leftarrow (CAR)+1$ otherwise
10	RET	$CAR \leftarrow (SBR)$
11	MAP	$CAR(2-5) \leftarrow (MBR(Op-Code)), CAR(1,6,7) \leftarrow 0$

★ ADF is 7-bits long and represents the address field.

Computer Organization

◆ μ -Programmed Control Unit – An Example

- ★ Similar to the assembly instructions, a symbolic notation can be used to represent each μ -instruction.
- ★ In our system, each μ -instruction is composed of six elements:
 - A label followed by semi-colons
 - μ -op-codes separated by commas
 - CD fields (U, I, S, Z)
 - BR field (JMP, CALL, RET, MAP)
 - ADF field (a label, NEXT, or blank)
 - Comments

Computer Organization

◆ μ -Programmed Control Unit — An Example

★ FETCH Cycle

- ★ The first 64 words in the control store represent the micro instructions for 16 macro instructions. For each macro instruction, a block of four μ -instructions is designated. Therefore, the micro code to represent the instruction fetch cycle is stored somewhere in the high order half part of the control store.

FETCH: PCTAR	U	JMP	NEXT
READ, INCPC	U	JMP	NEXT
BRTAR	U	MAP	

Computer Organization

◆ μ -Programmed Control Unit — An

Example

★ ADD

ADD:	NOP	I	CALL	INDRCT
	READ	U	JMP	NEXT
	ADD	U	JMP	FETCH

★ BRANCH

BRANCH:	NOP	S	JMP	OVER
	NOP	U	JMP	FETCH
OVER:	NOP	I	CALL	INDRCT
	BRTPC	U	JMP	FETCH

Computer Organization

◆ μ -Programmed Control Unit — An Example

★ STORE

STORE: NOP	I	CALL	INDRCT
ACTBR	U	JMP	NEXT
WRITE	U	JMP	FETCH

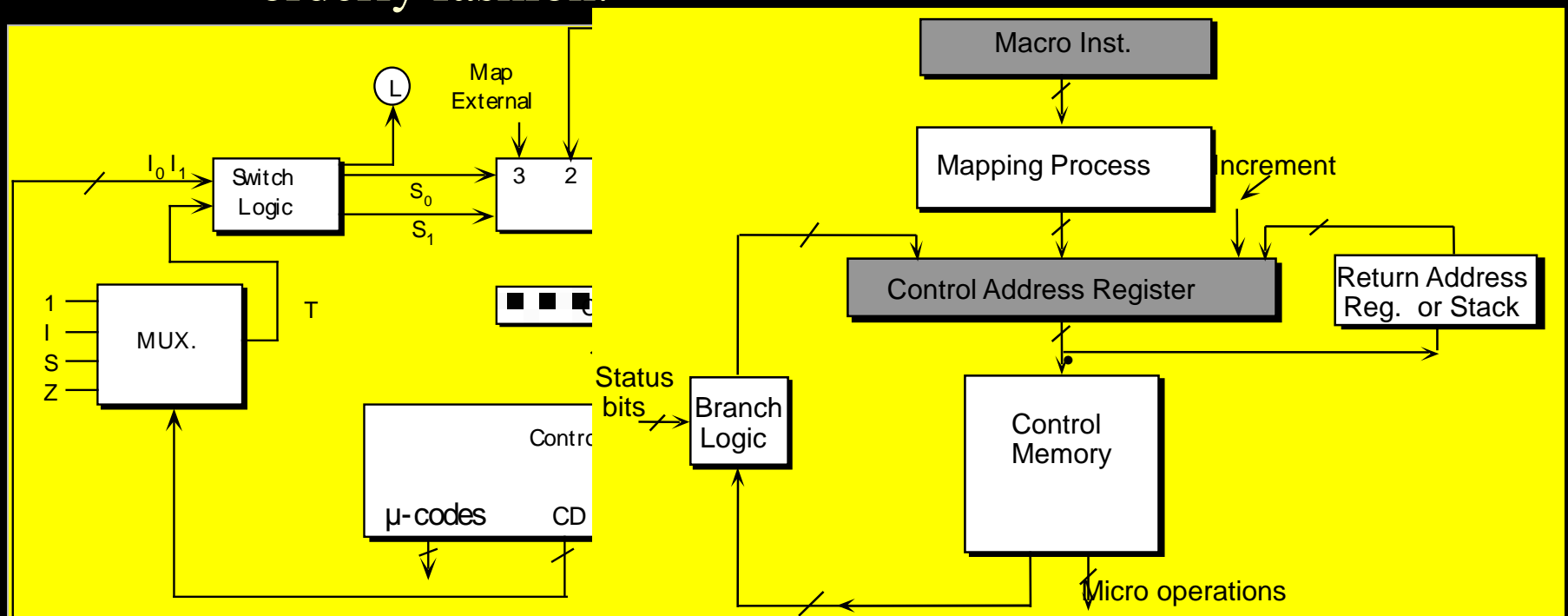
★ Indirect Cycle

INDRCT:	READ	U	JMP	NEXT
	BRTAR	U	RET	

Computer Organization

◆ μ -Programmed Control Unit – Microprocessor Sequencer

★ It is a unit which allows to fetch the μ -instructions in an orderly fashion.



Computer Organization

◆ μ -Programmed Control Unit – Microprocessor Sequencer

	BR	I ₀	I ₁	T	S ₀	S ₁	L
JMP	00	0	0	0	0	0	0
JMP	00	0	0	1	1	0	0
CALL	01	0	1	0	0	0	0
CALL	01	0	1	1	1	0	1
RET	10	1	0	X	0	1	0
MAP	11	1	1	X	1	1	0

$$S_0 = \bar{I}_0 \bar{I}_1 T + \bar{I}_0 I_1 T + I_0 \bar{I}_1 = \bar{I}_0 T (\bar{I}_1 + I_1) + I_0 \bar{I}_1 = \bar{I}_0 T + I_0 \bar{I}_1$$

$$S_1 = \bar{I}_0 I_1 + I_0 I_1 = I_0 (\bar{I}_1 + I_1) = I_0$$

$$L = \bar{I}_0 I_1 T$$

Computer Organization

◆ Control Unit

- * **Control unit:** A computer module with two major tasks: instruction sequencing and instruction interpretation.
- * **Instruction sequencing:** The way instructions are selected.
- * **Instruction Interpretation:** The way instructions are executed.
- * **Hardwired control unit:** A way to design the control unit, in which the control signals are generated by random logic.

Computer Organization

◆ Control Unit

- ★ **Micro programming:** A way to design the control unit, in which the control signals are generated by reading them from the control store.
- ★ **State table method:** A hardwired technique for the design of control unit.
- ★ **Delay elements method:** A hardwired technique for the design of control unit.
- ★ **Sequence counter method:** A hardwired technique for the design of control unit.
- ★ **Programmable logic array:** A hardwired technique for the design of control unit.

Computer Organization

◆ Questions

- ★ In a general purpose computer, what are the major tasks of a control unit?
- ★ Based on our earlier discussion about the simple machine, there are missing input signals to the control unit diagram of the simple machine. What are they?
- ★ What is a correct answer? Control unit:
 - Is a decoder which interprets the status of the computer at each moment of time.
 - Is a black box which generates control signals in an orderly fashion.
 - Initiates the execution of micro operations.
 - None of the above.