

COMPUTER SCIENCE

Computer Organization and Architecture

ALU & Control unit

Lecture_03



Vijay Agarwal sir





**TOPICS
TO BE
COVERED**

o1 Micro Operation

o2 Control Unit

ALU Data Path, Micro operation Micro Program & Control Unit.

- Component of the Computer.
- Working of Registers & MUX.
- ALU Data Path.
- Micro operation.
- Micro Program.

Micro operation.

↳ Fetch cycle

② Execute cycle

→ Direct AM

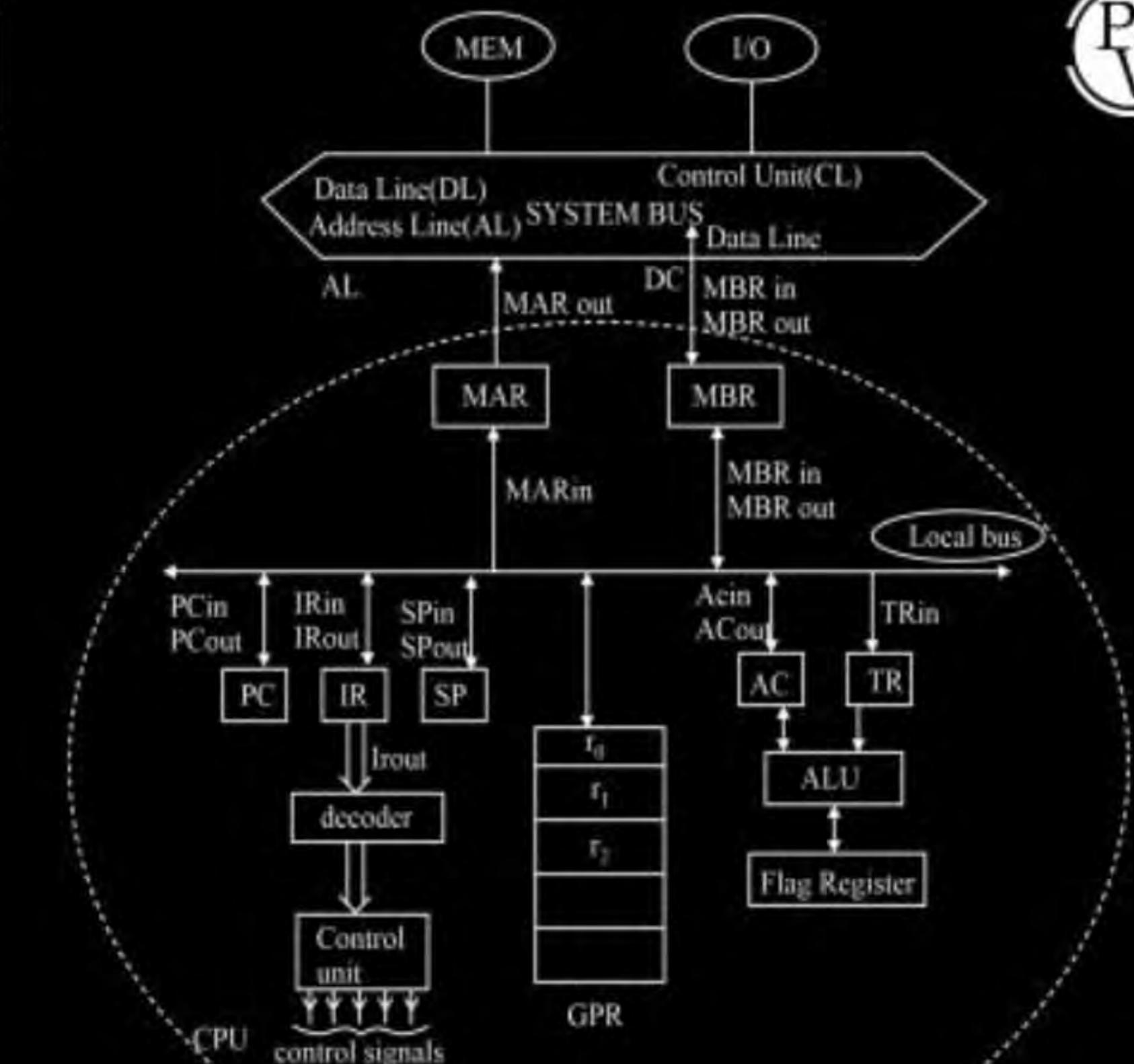
→ Indirect AM

STORE Instr

③ Interrupt cycle

Structure of Computer

- ① CPU org.
- ② Memory
- ③ I/O



Component of Computer

1. CPU , 2. Memory & 3. IO

- ✓ Memory
- ✓ Register
- ✓ ALU
- ✓ Timing Signals, Control signals
- ✓ Flags(PSW)

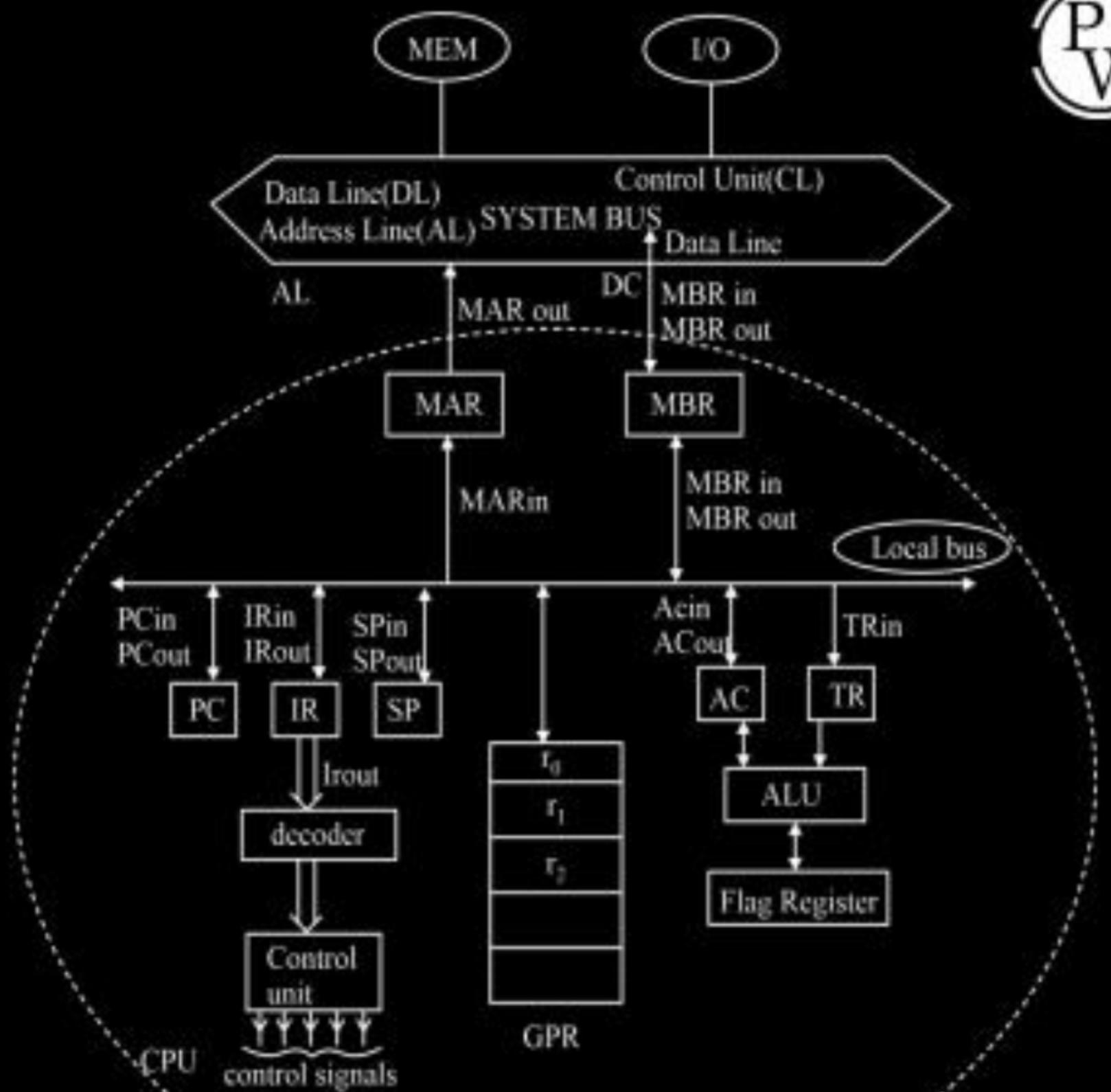
Common Bus

- ✓ multiplex
- ✓ Select line.

ALU, Datapath & Control Unit

- Micro Instruction
- Micro Program
- Control Unit Design

Structure of Computer



Component of Computer

1. CPU , 2. Memory & 3. IO

Memory

ALU

Register

Timing Signals, Control signals

Flags(PSW)

- In the Processor some general purpose & some special purpose register are available. All the register are connected to a common path called Data Path (Bus). Data Path is collection of functional units(ALU & MUX..)

Every Register has 2 switch Rin & Rout

R_{in} : R_{in} is set to 1, if the content of the bus loaded into Register R_i .

R_{out} : R_{out} is set to 1, the content from the register R_i will be placed on bus.

R_{in}
 R_{out}



Micro Operations

The functional, or atomic, operations of a processor

- ❑ Series of steps, each of which involves the processor registers
- ❑ Micro refers to the fact that each step is very simple and accomplishes very little
- ❑ The execution of a program consists of the sequential execution of instructions

Each instruction is executed during an instruction cycle made up of shorter sub cycles (fetch, indirect, execute, interrupt)

The execution of each sub cycle involves one or more shorter operations (micro-operations)

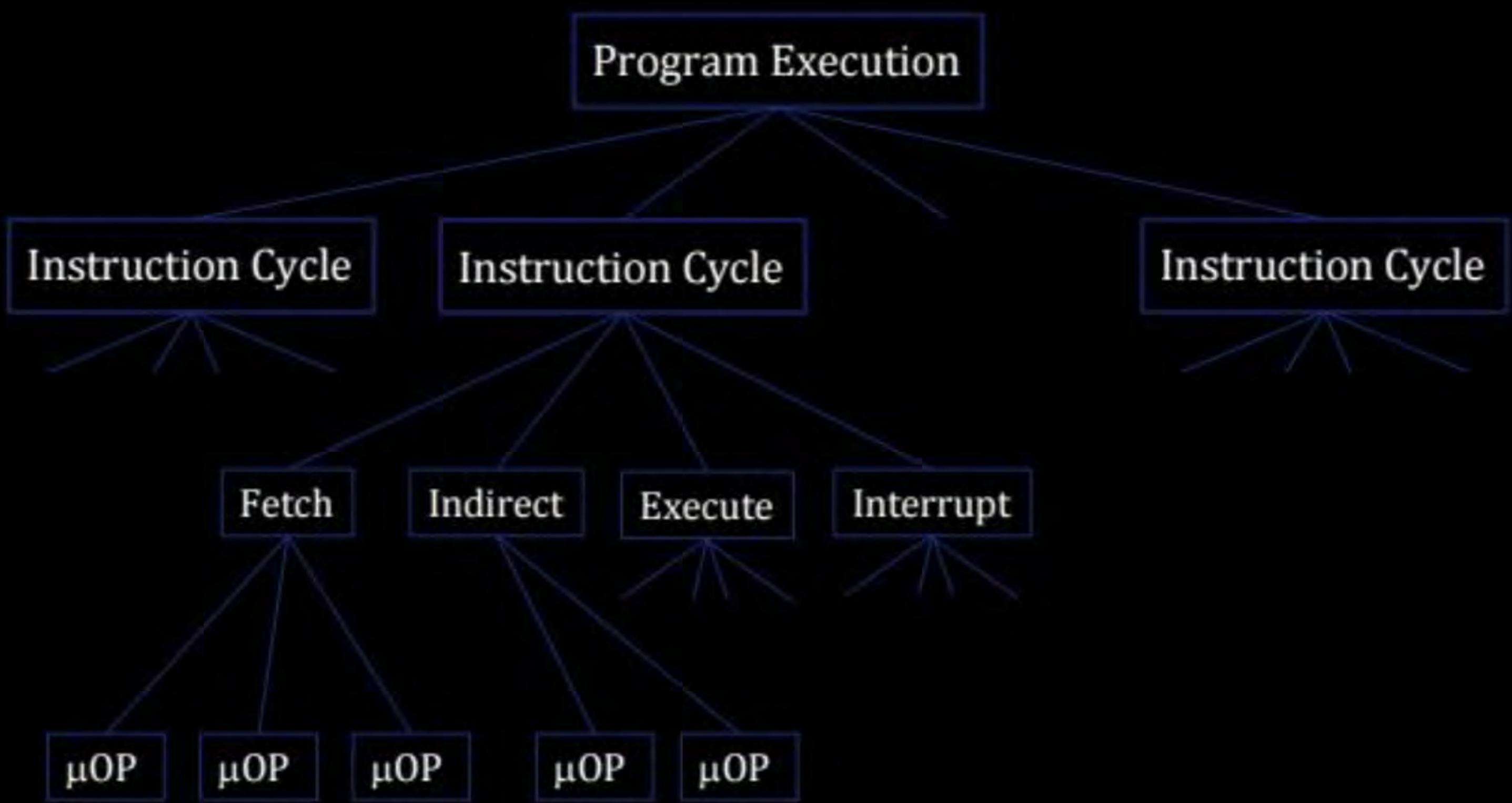
Micro Operation

- Micro Operation is a elementary operation in the Hardware
Example: Register to register transfer operation is a one kind of micro operation.
- Micro operation Consume 1 cycle to complete the execution.
- Control Signal are required to execute the micro operation.

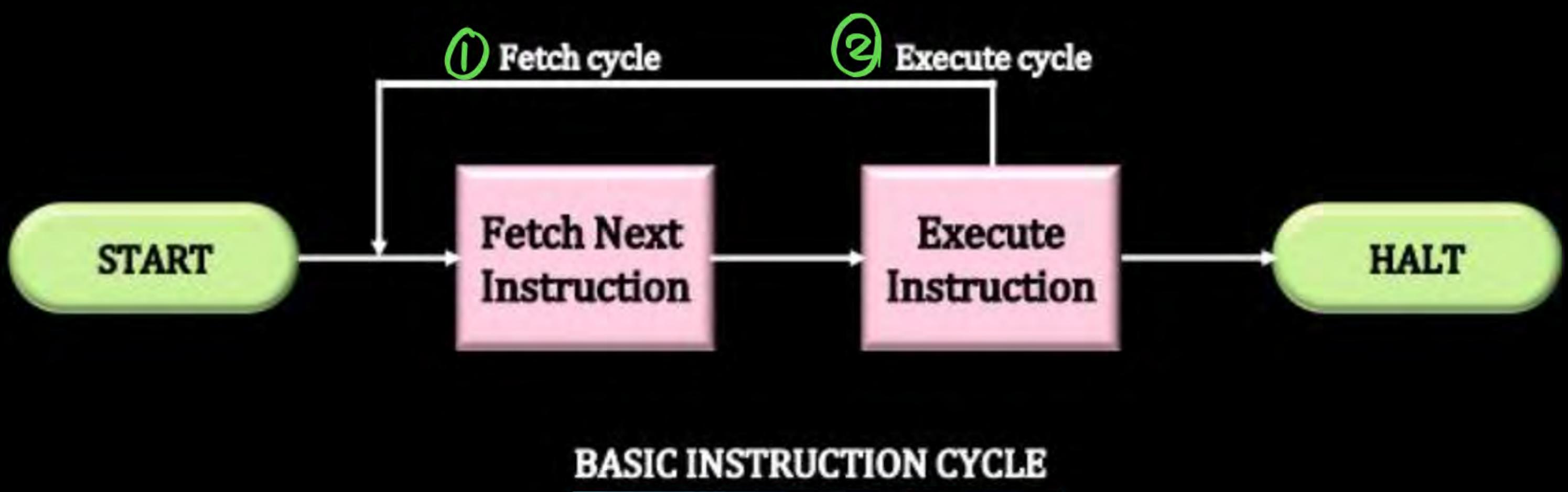
Machine Instruction: $MOV r_0, r_1$

RTL (Register Transfer Language): $r_0 \leftarrow r_1$

Micro operation: $T_1: r_1\text{out } r_0\text{in}$



Constituent Elements of a Program Execution



Micro Program (μ - Program)

- Sequence of Micro operation [μ operation] to perform Some operation in the Hardware level is called microprogram.

The Fetch Cycle

- Occurs at the beginning of each instruction cycle and causes an instruction to be fetched from memory

Four registers are involved:

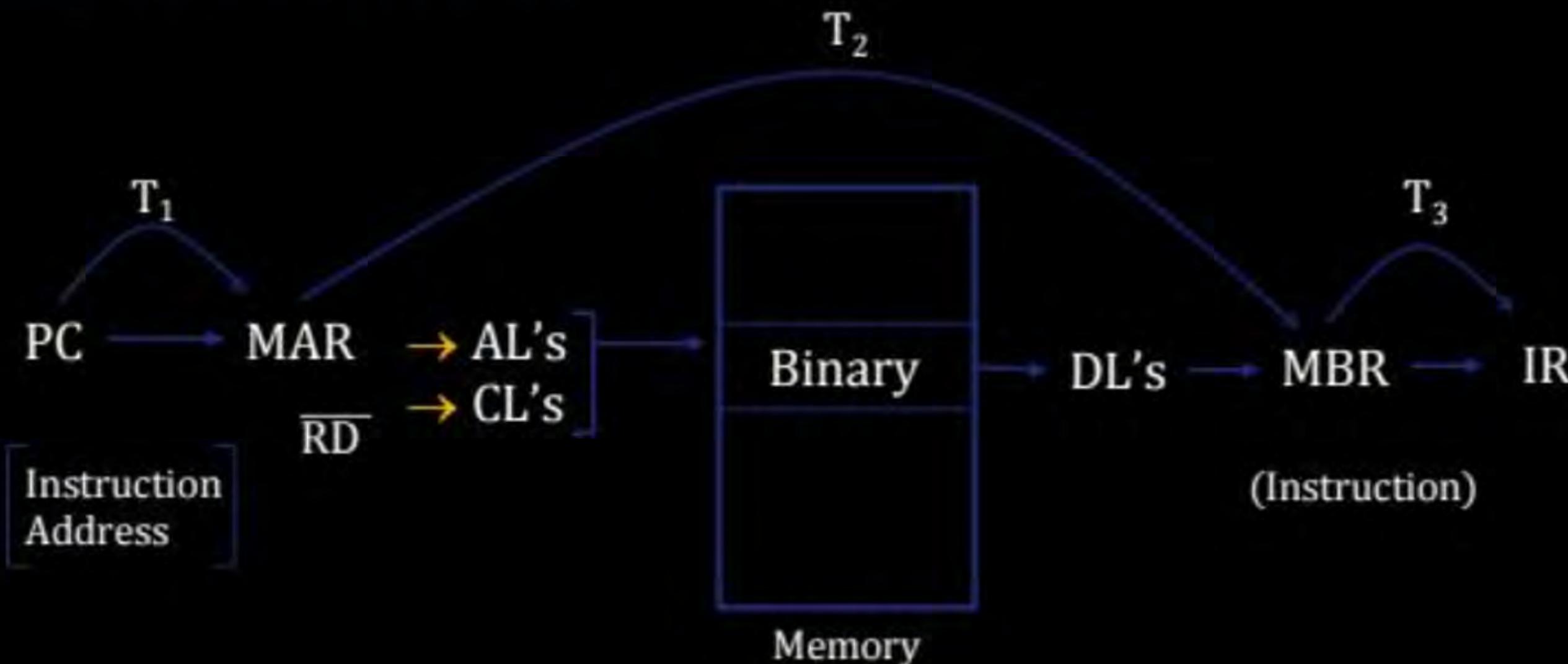
- ❖ Memory Address Register (MAR)
 - Connected to address bus
 - Specifies address for read or write operation
- ❖ Memory Buffer Register (MBR)
 - Connected to data bus
 - Holds data to write or last data read
- ❖ Program Counter (PC)
 - Holds address of next instruction to be fetched
- ❖ Instruction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

Instruction Cycle

(1) Fetch Cycle: Instruction Fetch.

Hardware Design(H/W Design)

AL: Address Line
DL: Data Line
CL: Control Line



Microprogram

T1: PC → MAR;

PC_{out} MAR_{in}

T2: M[MAR] → MBR; MAR_{out} MBR_{in} System Bus

PC + I → PC

PC_{out} PC_{in} Local Bus

Increment

T3: MBR → IR

MBR_{out} IR_{in}

tMAR	
MBR	
PC	0000000001100100
IR	
AC	

(a) Beginning (before t_1)

MAR	0000000001100100
MBR	
PC	0000000001100100
IR	
AC	

(b) After first step

MAR	0000000001100100
MBR	0001000000100000
PC	0000000001100101
IR	
AC	

(c) After second step

MAR	0000000001100100
MBR	0001000000100000
PC	0000000001100101
IR	0001000000100000
AC	

(d) After third step

Sequence of events, fetch cycle

Rules for Micro Operations Grouping

Proper sequence must be followed

- $\text{MAR} \leftarrow (\text{PC})$ must precede $\text{MBR} \leftarrow (\text{memory})$

Conflicts must be avoided

- Must not read and write same register at same time
- $\text{MBR} \leftarrow (\text{memory})$ and $\text{IR} \leftarrow (\text{MBR})$ must not be in same cycle

Execute Cycle

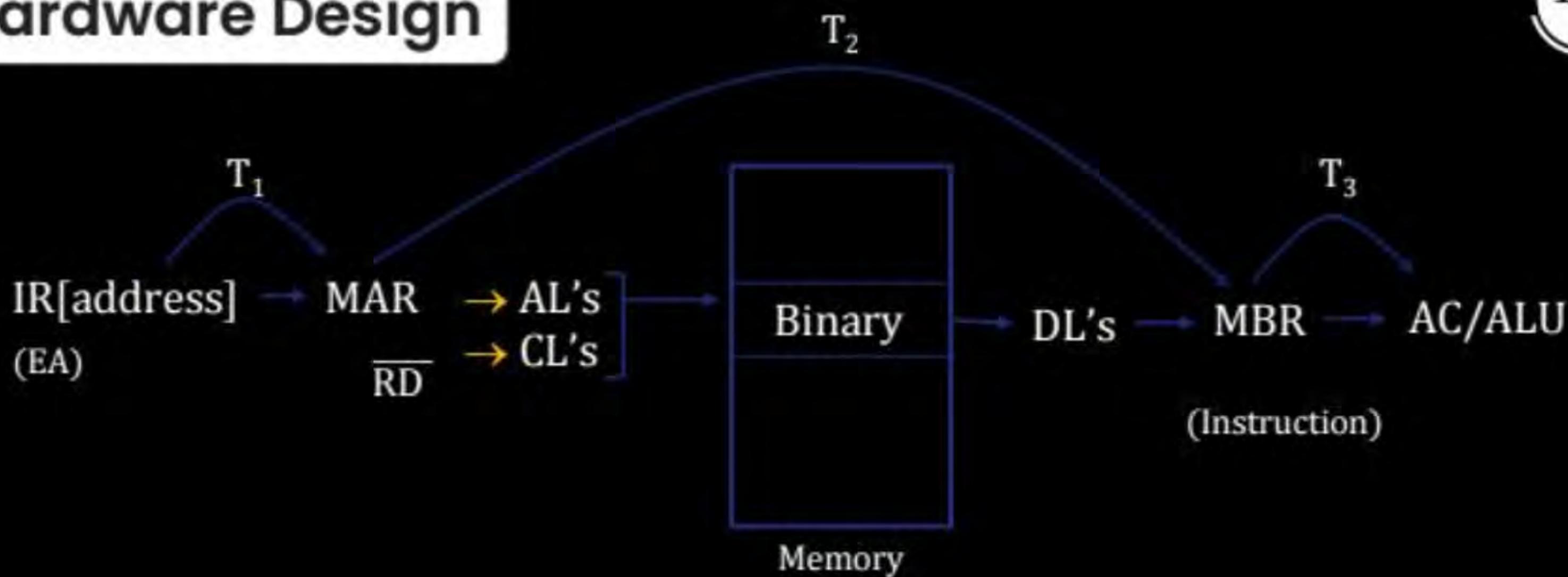
- ❑ Because of the variety of opcodes, there are a number of different sequences of micro-operations that can occur.
- ❑ Instruction decoding
 - ❖ The control unit examines the opcode and generates a sequence of micro-operations based on the value of the opcode
- ❑ A simplified add instruction:
 - ❖ ADD R1, X (which adds the contents of the location X to register R1)
 - ❖ In the first step the address portion of the IR is loaded into the MAR
 - ❖ Then the referenced memory location is read
 - ❖ Finally the contents of R1 and MBR are added by the ALU
 - ❖ Additional micro-operations may be required to extract the register reference from the IR and perhaps to stage the ALU inputs or outputs in some intermediate registers

Execute Cycle

- (a) ID Stage: Enable the Hardwire to perform the operation
(Instruction Decode)
- (b) OF Stage: AM's (addressing mode) are required to access.
(Operand Fetch)

Example: Direct AM Memory Read
 LOAD [4000]
 AC \leftarrow m[4000]

Hardware Design



T1:

IR(Address field) \rightarrow MAR:

IR_{out}

MAR_{in}

T2:

$M[MAR]$ \rightarrow MBR:

MAR_{out}

MBR_{in}

T3:

$MBR \rightarrow AC/ALU$

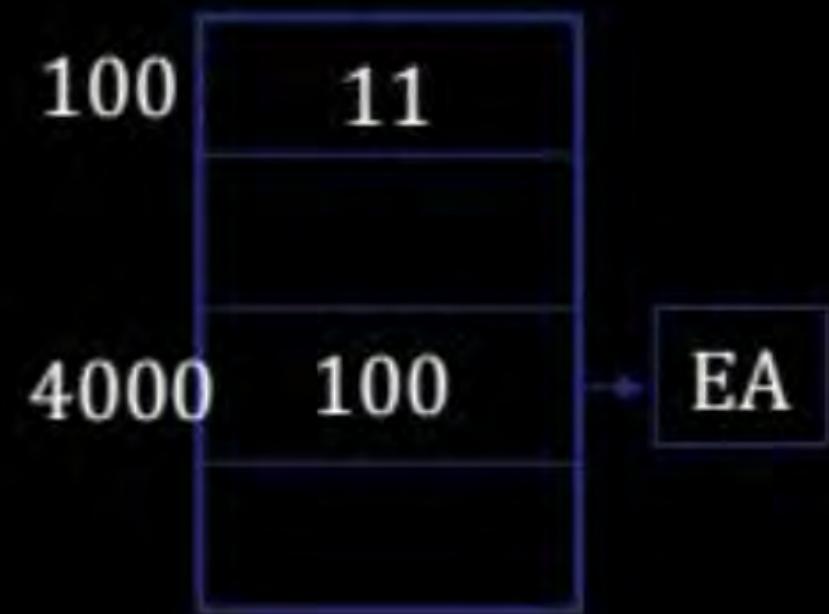
MBR_{out}

AC_{in}/ALU_{in}

Example: Indirect AM

Load @4000

$AC \leftarrow M[[4000]]$



Microprogram:

- T₁: IR[Address] → MAR: IR_{out} MAR_{in}
- T₂: M[MAR] → MBR(EA): MAR_{out} MBR_{in}
- T₃: MBR → MAR: MBR_{out} MAR_{in}
- T₄: M[MAR] → MBR: MAR_{out} MBR_{in}
- T₅: MBR → AC/ALU: MBR_{out} AC_{in}/ALU_{in}

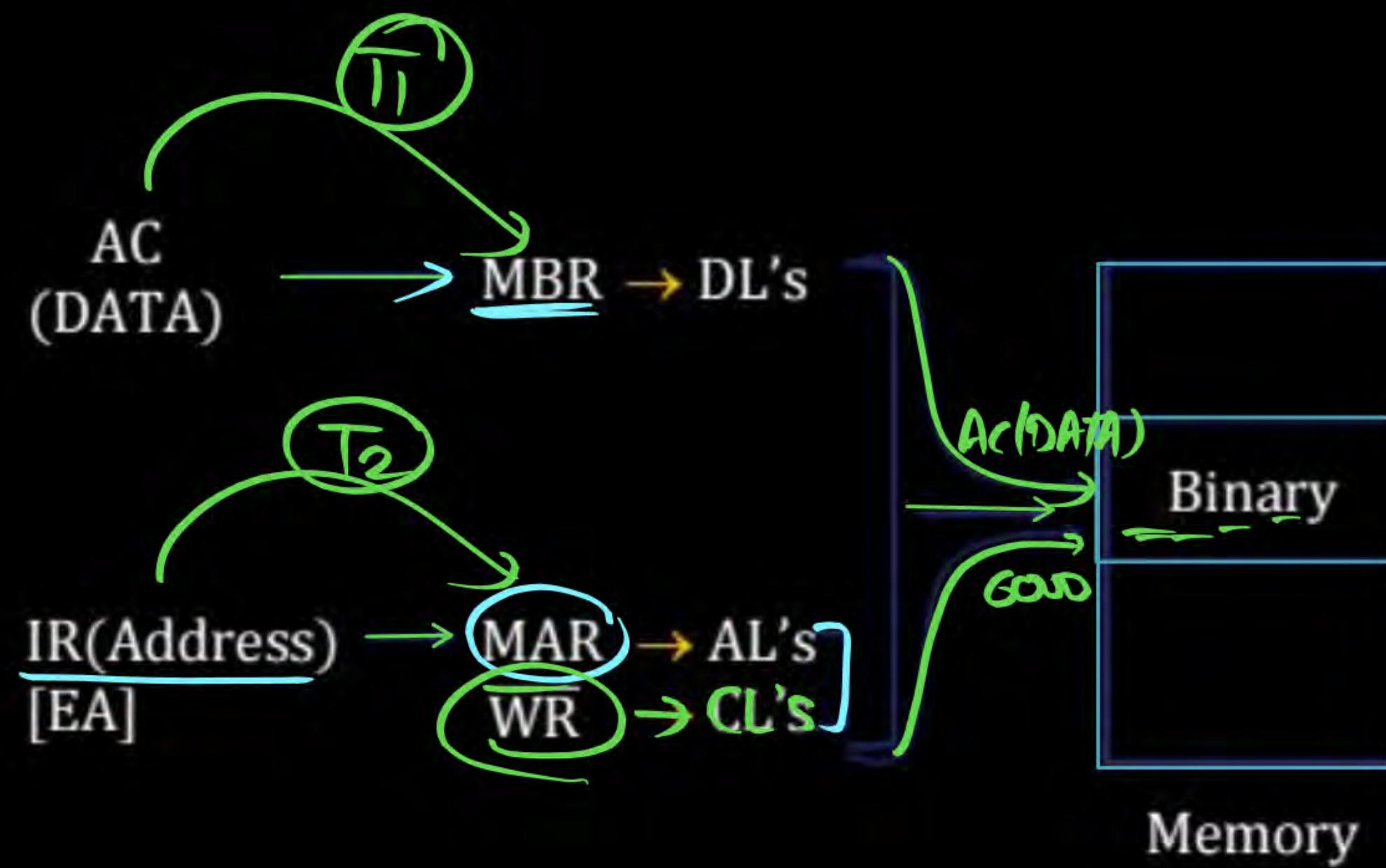
IR:
OPCODE AF

STORE	[6000]
-------	--------

STORE [6000]
 $m[6000] \leftarrow AC$

LOAD: Memory Read.
Store: Memory Write!

P
W



Example:

Memory write

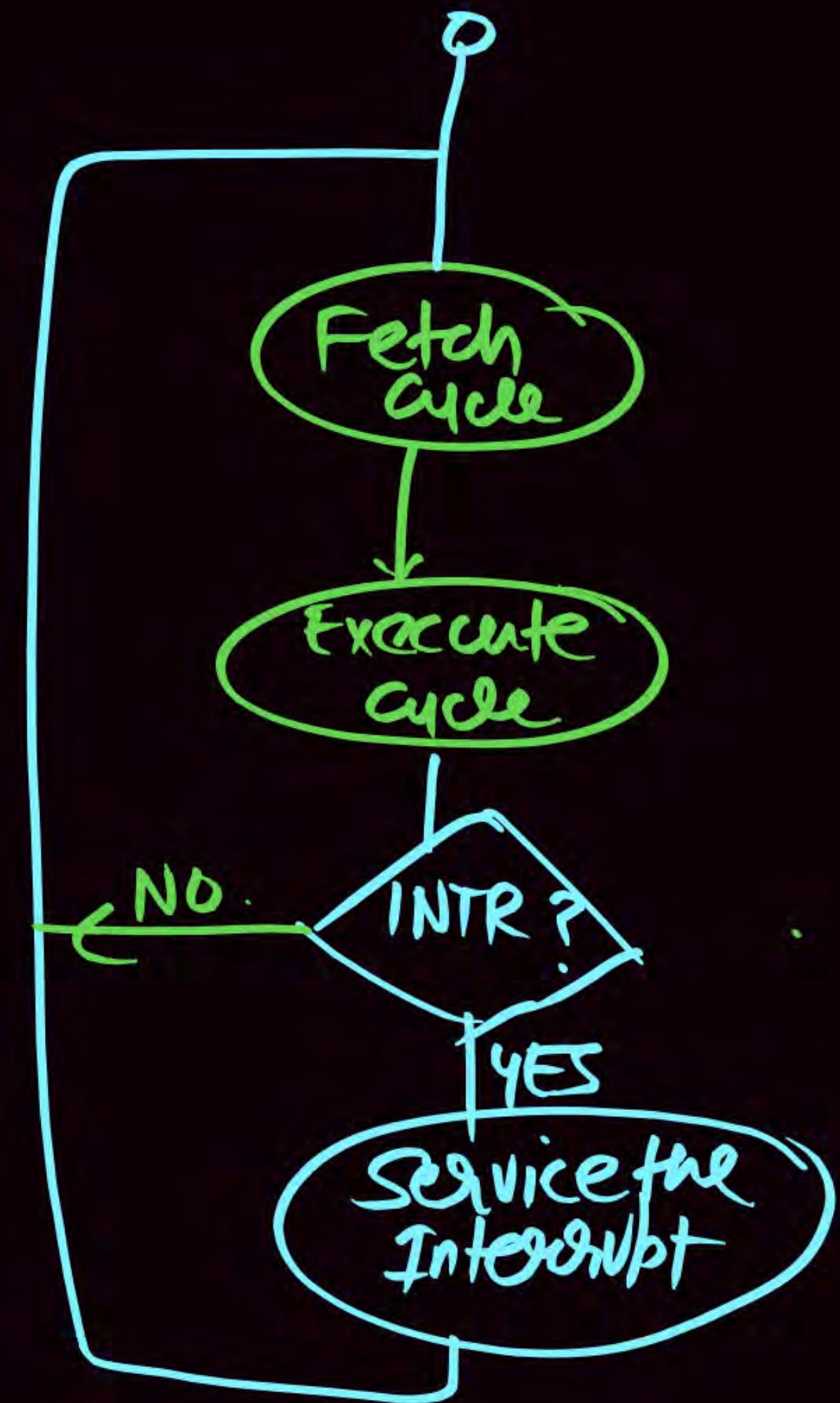
Store [6000]

 $M[6000] \leftarrow AC$

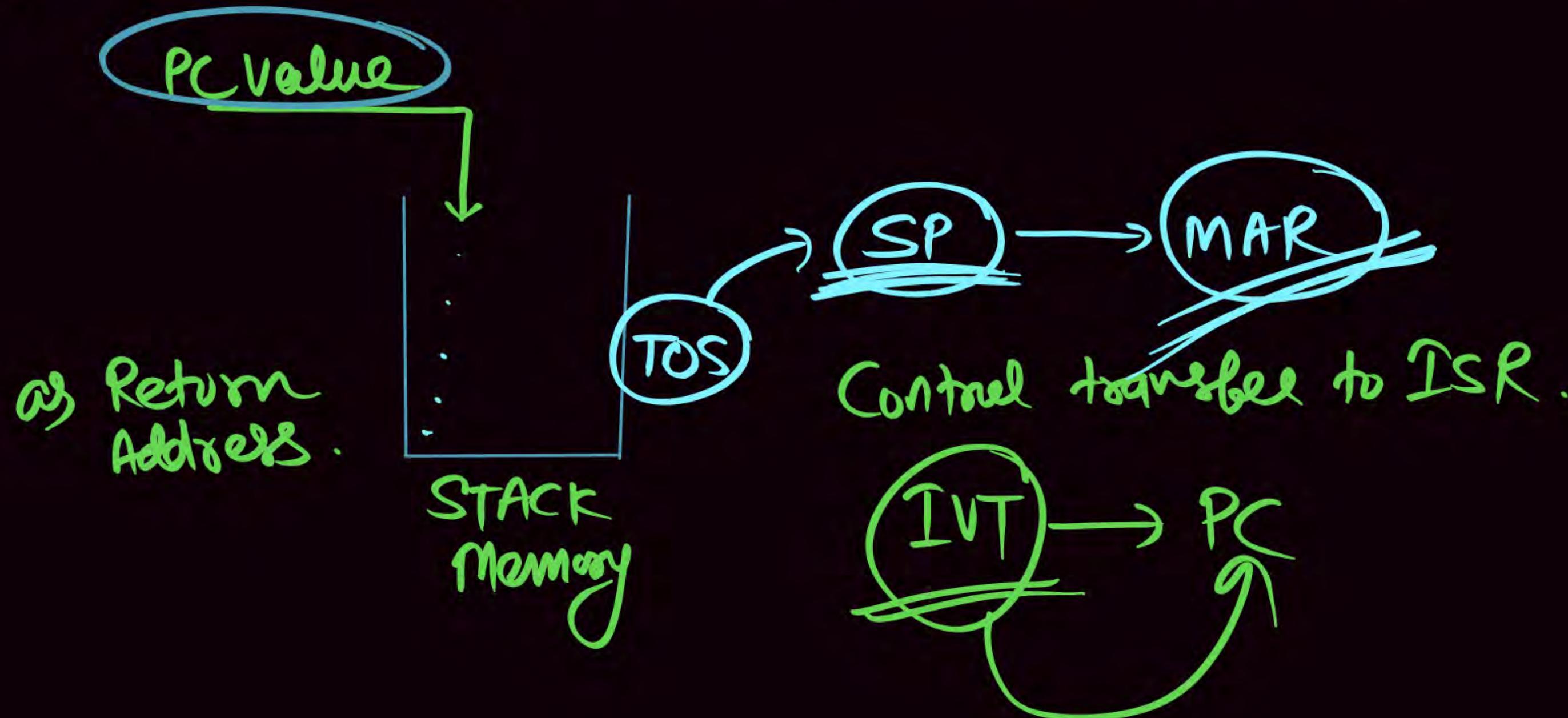
Micro Program

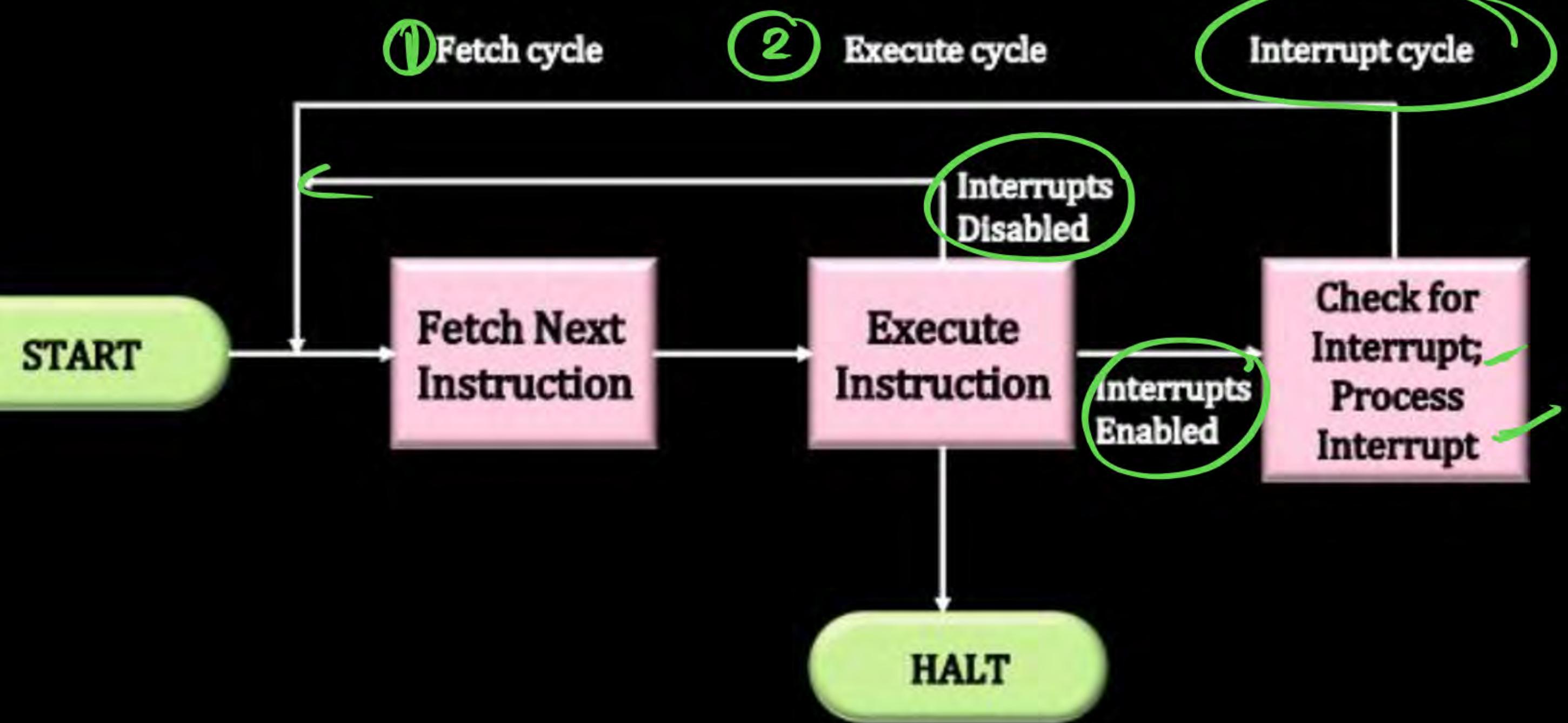
 $T_1: AC \rightarrow MBR:$ $T_2: IR(Address) \rightarrow MAR$ $T_3: MBR \rightarrow M[MAR]$ AC_{out} IR_{out} MBR_{out} MBR_{in} MAR_{in} MAR_{in}

Interrupt cycle



When Interrupt encountered in the System then
After Completion of Current Instruction Execution





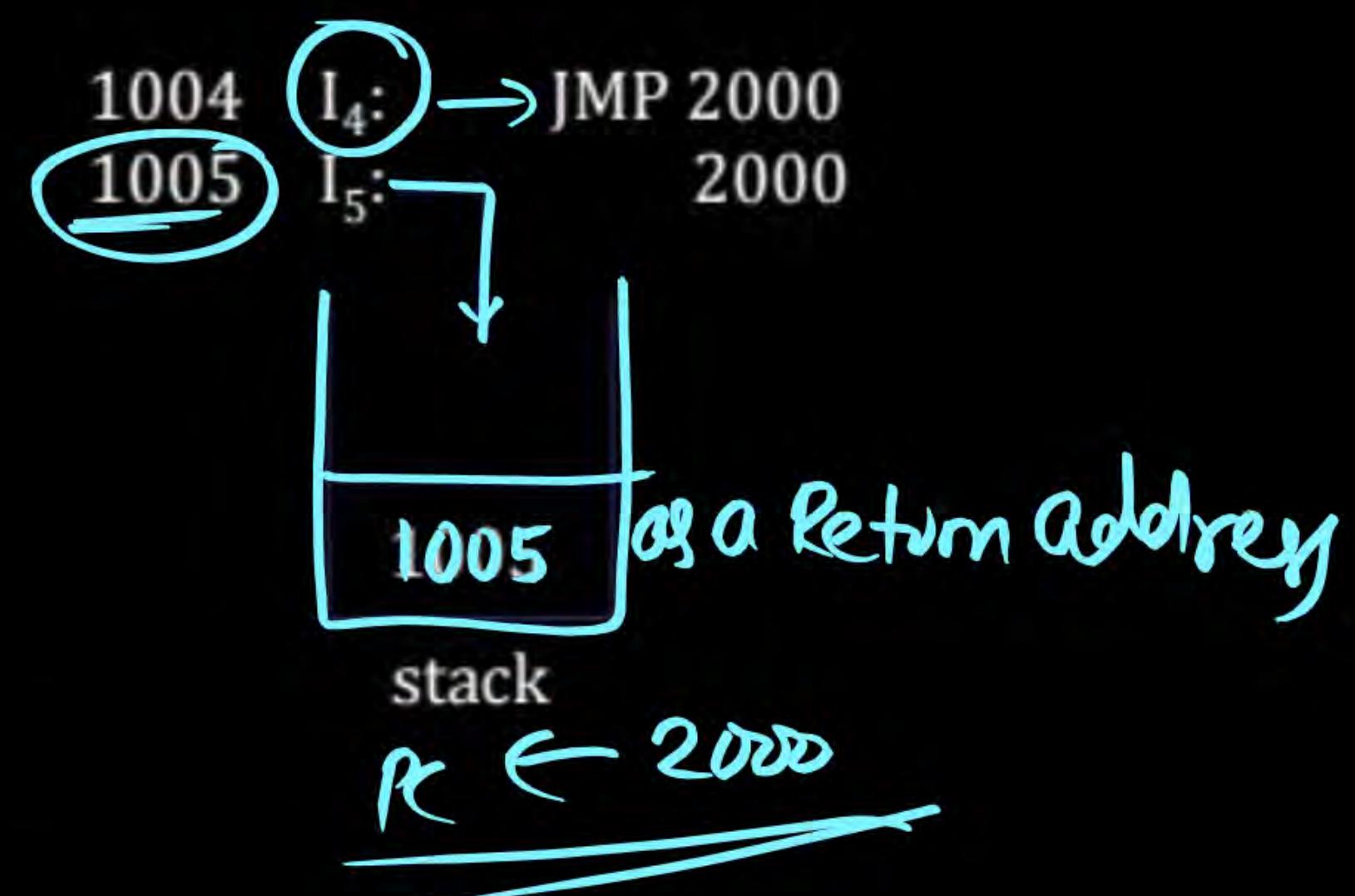
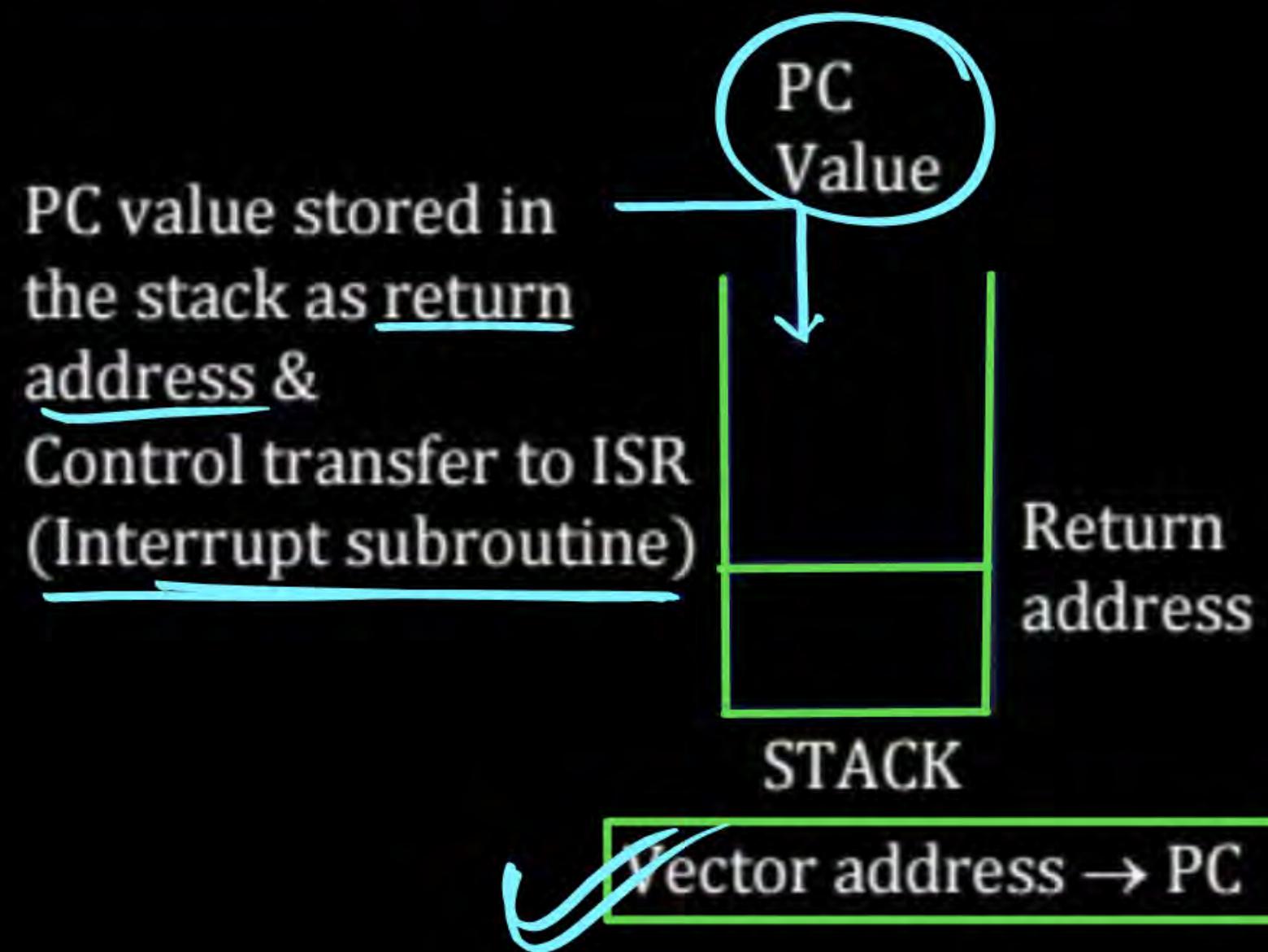
Instruction cycle with Interrupts

Interrupt cycle:

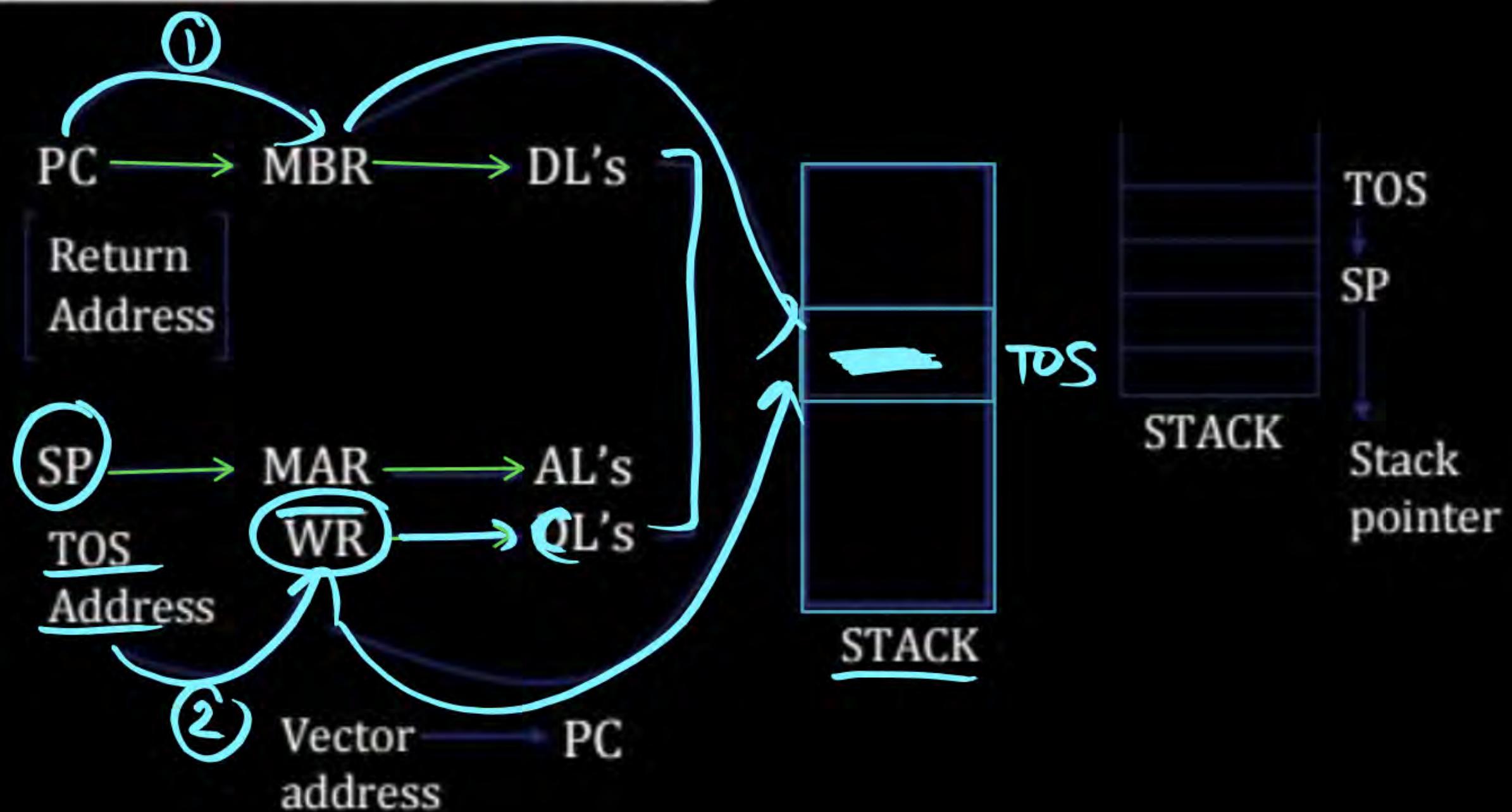
- At the completion of the execute cycle, a test is made to determine whether any enabled interrupts have occurred, and if so, the interrupt cycle occurs ✓
- The nature of this cycle varies greatly from one machine to another
- In a simple sequence of events:
 - * ① In the first step the contents of the PC are transferred to the MBR so that they can be saved for return from the interrupt
 - ② Then the MAR is loaded with the address at which the contents of the PC are to be saved, and the PC is loaded with the address of the start of the interrupt processing routine
 - ③ These two actions may each be a single micro-operation
 - ④ Because most processors provide multiple types and/or levels of interrupts, it may take one or more additional micro-operations to obtain the Save Address and the Routine Address before they can be transferred to the MAR and PC respectively
 - ⑤ Once this is done, the final step is to store the MBR, which contains the old value of the PC, into memory
 - ⑥ The processor is now ready to begin the next instruction cycle

Interrupt cycle:

Whenever Interrupt occur after the completion of current Instruction execution interrupt will be serviced.



Hardware Design



Microprogram μ Program

- T1: PC \rightarrow MBR
- T2: SP \rightarrow MAR
- T3: MBR \rightarrow M[MAR]: System Bus

Vector Address \rightarrow PC: Local Bus

MCQ

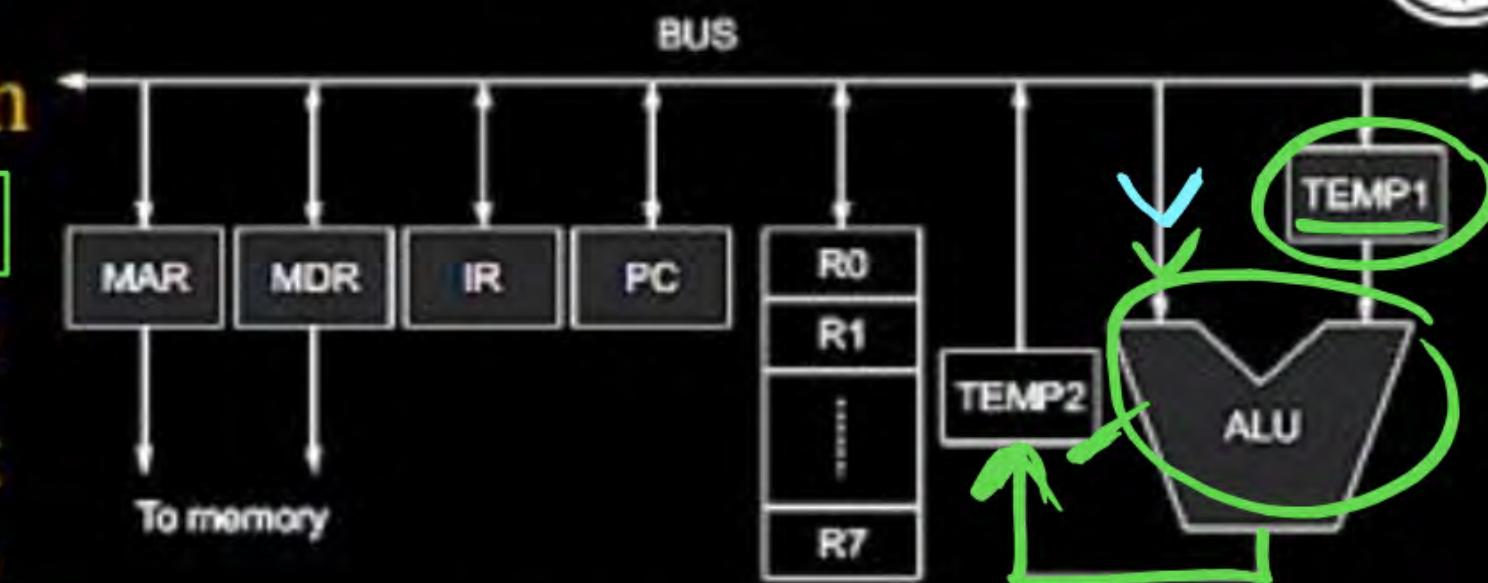
Consider the following data path diagram

Consider an instruction: $R0 \leftarrow R1 + R2$.

The following steps are used to execute it over the given data path. Assume that PC is incremented appropriately. The subscripts r and w indicate read and write operations, respectively.

1. $R2_r$, $\underline{\text{TEMP1}_R}$, $\underline{\text{ALU}_{\text{add}}}$, $\underline{\text{TEMP2}_w}$
2. $R1_r$, TEMP1_w ,
3. PC_r , MAR_w , MEM_r
4. TEMP2_R , RO_w
5. MDR_r , IR_w

Ans (C).



[GATE-2020-CS: 1M]

Which one of the following is the correct order of execution of the above steps?

- A 3, 5, 1, 2, 4
- B 2, 1, 4, 5, 3
- C 3, 5, 2, 1, 4
- D 1, 2, 4, 3, 5

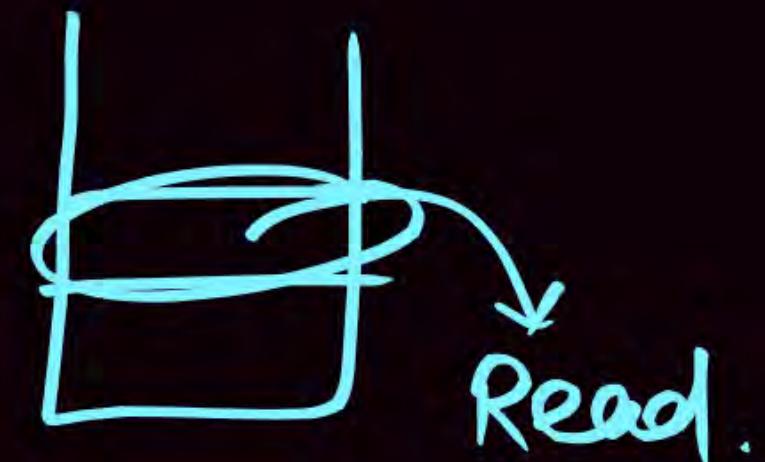
$$I_1: R_0 \in R_1 + R_2.$$

3. $PCR \rightarrow MAR_w \rightarrow MEM_R ; PC \rightarrow MAR$
5. $MBR_R \rightarrow IR_w ; MBR \rightarrow IR$.
2. $R_{IR} \rightarrow Temp_{1w} ; R_I \rightarrow Temp_L$.
1. $R_{2x} \rightarrow Temp_{Lx} \rightarrow ALU_{Add} \rightarrow Temp_{2w} ; Temp_2 \in R_2 \oplus Temp_L$
4. $Temp_{2x} \rightarrow R_{0w} ; R_0 \in Temp_2$

3, 5, 2, 1, 4

Ans

$$R_0 \in R_1 + R_2.$$



...

MCQ

The following are some events that occur after a device controller issues an interrupt while process L is under execution.

- (2) (P) The Processor pushes the process status of L onto the control stack.
- (1) (Q) The processor finishes the execution of the current instruction.
- (4) (R) The processor executes the interrupt service routine.
- (5) (S) The processor pops the process status of L from the control stack.
- (3) (T) The processor loads the new PC value based on the interrupt.

Which one of the following is the correct order in which the events above occur?

[GATE-2018-CS: 1M]

A QPTRS

Ans (A).

B PTRSQ

C TRPQS

D QTPRS

MCQ

Consider the following sequence of micro -operations.

$\text{MBR} \leftarrow \text{PC}$

$\text{MAR} \leftarrow \text{X}$

$\text{PC} \leftarrow \text{Y}$

Memory $\leftarrow \text{MBR}$

Which one of the following is a possible operation performed by this sequence

[GATE-2013-CS: 2M]

A Instruction fetch

~~Arg(D)~~

C Conditional branch

B operand fetch

D Initiation of interrupt service

$PC \rightarrow MBR$.

(SP)
tos [X] $\rightarrow MAR$

(ISR) y $\rightarrow PC$

$MBR \rightarrow MAR$
(PC value \rightarrow stack)
 $\xrightarrow{\text{as}} \text{Return address}$

Interrupt

@ Instruction Fetch

$PC \rightarrow MAR \rightarrow \text{Memory} \rightarrow MBR \rightarrow IR$.

b) Operand Fetch.

$IR(AF) \rightarrow MAR \rightarrow \text{Memory} \rightarrow MBR \rightarrow AC$

Working of Computer .

ALU Data Path

Micro operation

Micro Program.

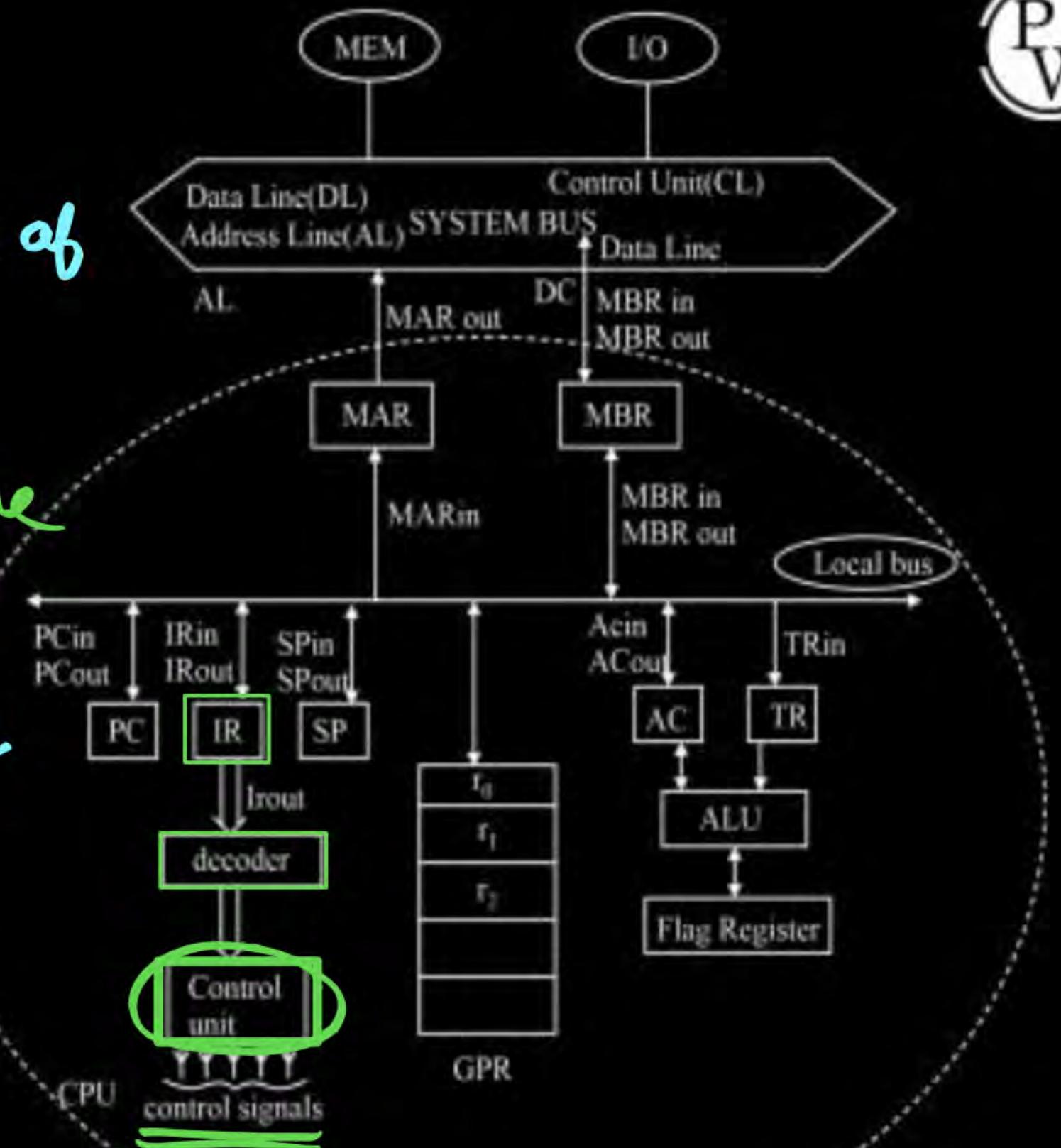
Control-unit

- Control Unit is the supervisor in the System that Control Each & Every Activity.
- Control Unit takes Various input but Produce only One Output [control signals]
- These Control Signal are Required to execute the Micro Operation.

Structure of Computer

Q) Why Control Unit After Decoder?
 In Fetch & Decode same type of operation performed

After the Decode, input given to the Control Unit, So According to the Operations Control Signal generated, to perform the Micro operation.



$PC \rightarrow MAR$; $PC_{out} \leftarrow MAR_{in}$

Control Unit Functional Requirements

- By reducing the operation of the processor to its most fundamental level we are able to define exactly what it is that the control unit must cause to happen
- Three step process to lead to a characterization of the control unit:
 - ① Define basic elements of processor ✓
 - ② Describe micro-operations processor performs ✓
 - ③ Determine the functions that the control unit must perform to cause the micro-operations to be performed .
- The control unit performs two basic tasks:
 - ① Sequencing
 - ② Execution

$T_1: PC \rightarrow MAR$

$T_2: M(MAR) \rightarrow MBR$

$T_3: MBR \rightarrow IR$.

Control Unit

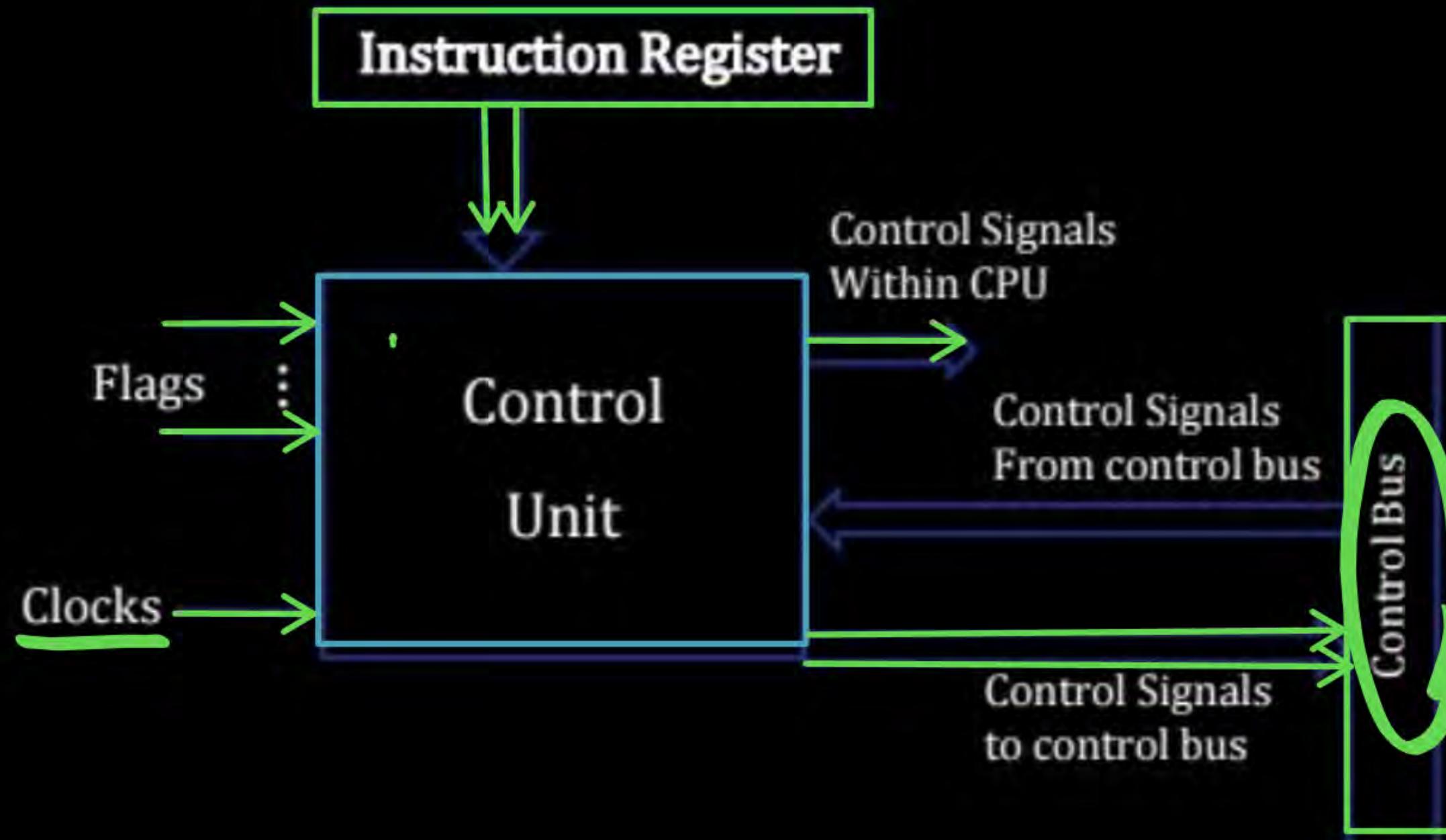
Control unit is the Supervisor in the System that control each & every activity.

- Control Signals are implement in a Control Unit.
- Control Signal are Required to execute the micro operation.
- Micro operation is the elementary operation in the hardware.
- Control unit generates the sequence of control Signal.
- Control Signal are Directly executed on a Base Hardware (H/W)

So H/W generate the desired Response.

Computer System Functionality is Program Execution.

Block Diagram of the Control Unit



Computer System \Rightarrow Program Execution

System



Prog \Rightarrow Instruction

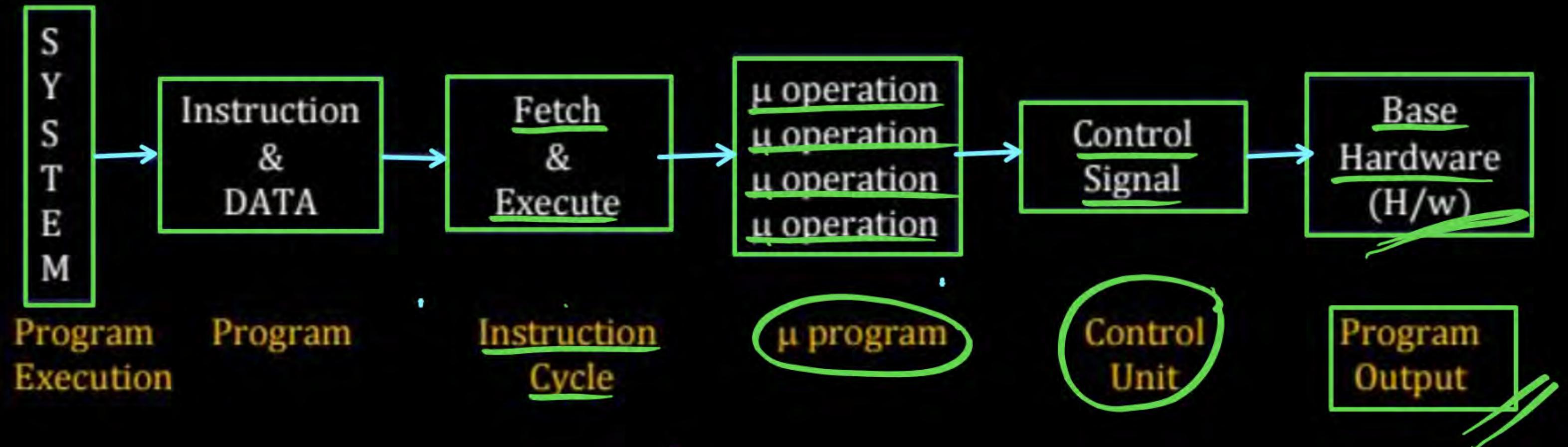


Instruction Cycle · (Fetch & Execute cycle)

Micro operation

Control Signal

Output



Control Unit : Control Word.

T_L: PC → MAR ; PC_{out} MAR_{in}.

(8)

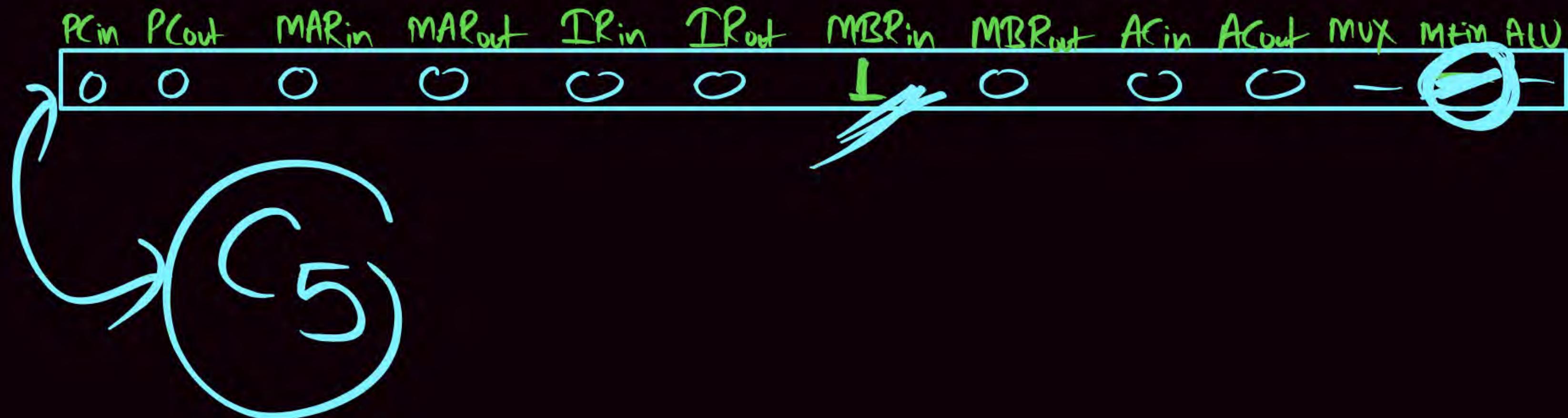
PC _{in}	PC _{out}	MAR _{in}	MAR _{out}	IR _{in}	IR _{out}	MBR _{in}	MBR _{out}	AC _{in}	AC _{out}	MUX	M _{in} ALU
0	1	1	0	0	0	0	0	0	0	-	-

PC → MAR ; PC_{out} MAR_{in}

L Control Word for L Micro operation

Control Unit : Control Word.

Memory → MBR; **MBRin**



Control Unit : Control Word.

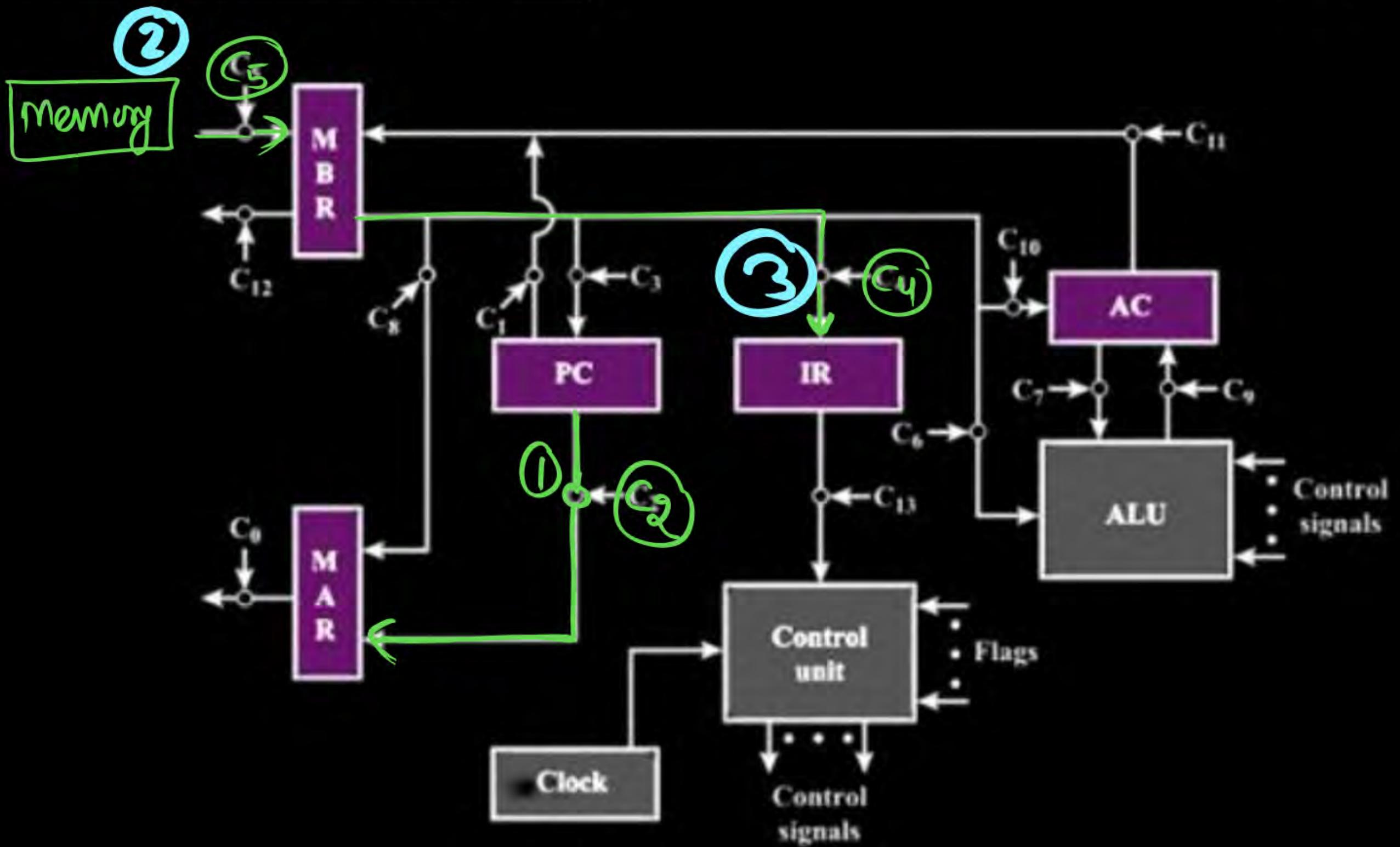
MBR \rightarrow DR; MBRout DRin

PCin	PCout	MARin	MARout	DRin	IRout	MBRin	MBRout	ACin	ACout	MUX	Mtin ALU
0	0	0	0	1	0	0	1	0	0	-	-

Micro-operations & Control Signals

	Micro-operations	Active-control Signals
Fetch:	$T_1: MAR \leftarrow PC$ (or) $PC \rightarrow MAR$ <i>PCout MARin</i>	<i>C₂</i>
	$T_2: MBR \leftarrow Memory$ <i>Memory → MBR</i> $PC \leftarrow (PC) + 1$ <i>PCout PCin</i>	<i>C₅, C_R</i>
	$T_3: IR \leftarrow MBR$ <i>MBR → IR; MBRout IRin</i>	<i>C₄</i>
Indirect:	$T_1: MAR \leftarrow IR(Address)$	C_8
	$T_2: MBR \leftarrow Memory$	C_5, C_R
	$T_3: IR(Address) \leftarrow MBR(Address)$	C_4
Interrupt:	$T_1: MBR \leftarrow PC$	C_1
	$T_2: MAR \leftarrow Save-address$ $PC \leftarrow Routine-address$	
	$T_3: Memory \leftarrow MBR$	C_{12}, C_W

Data Paths & Control Signals



Control Unit : Control Word.

T_L: PC → MAR ; PC_{out} MAR_{in}.

(3)

PC _{in}	PC _{out}	MAR _{in}	MAR _{out}	DR _{in}	DR _{out}	MBR _{in}	MBR _{out}	AC _{in}	AC _{out}	MUX	MT _{in}	ALU
0	1	1	0	0	0	0	0	0	0	-	-	-

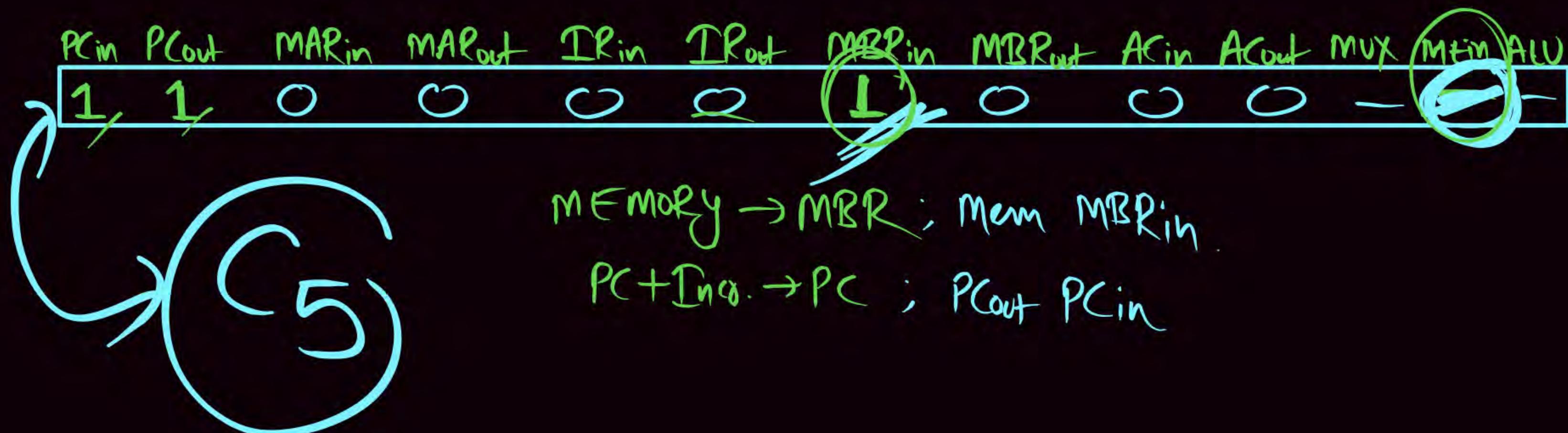
PC → MAR; PC_{out} MAR_{in}

1 Control Word for L Microoperation



Control Unit : Control Word.

Memory → MBR; **MBRin**

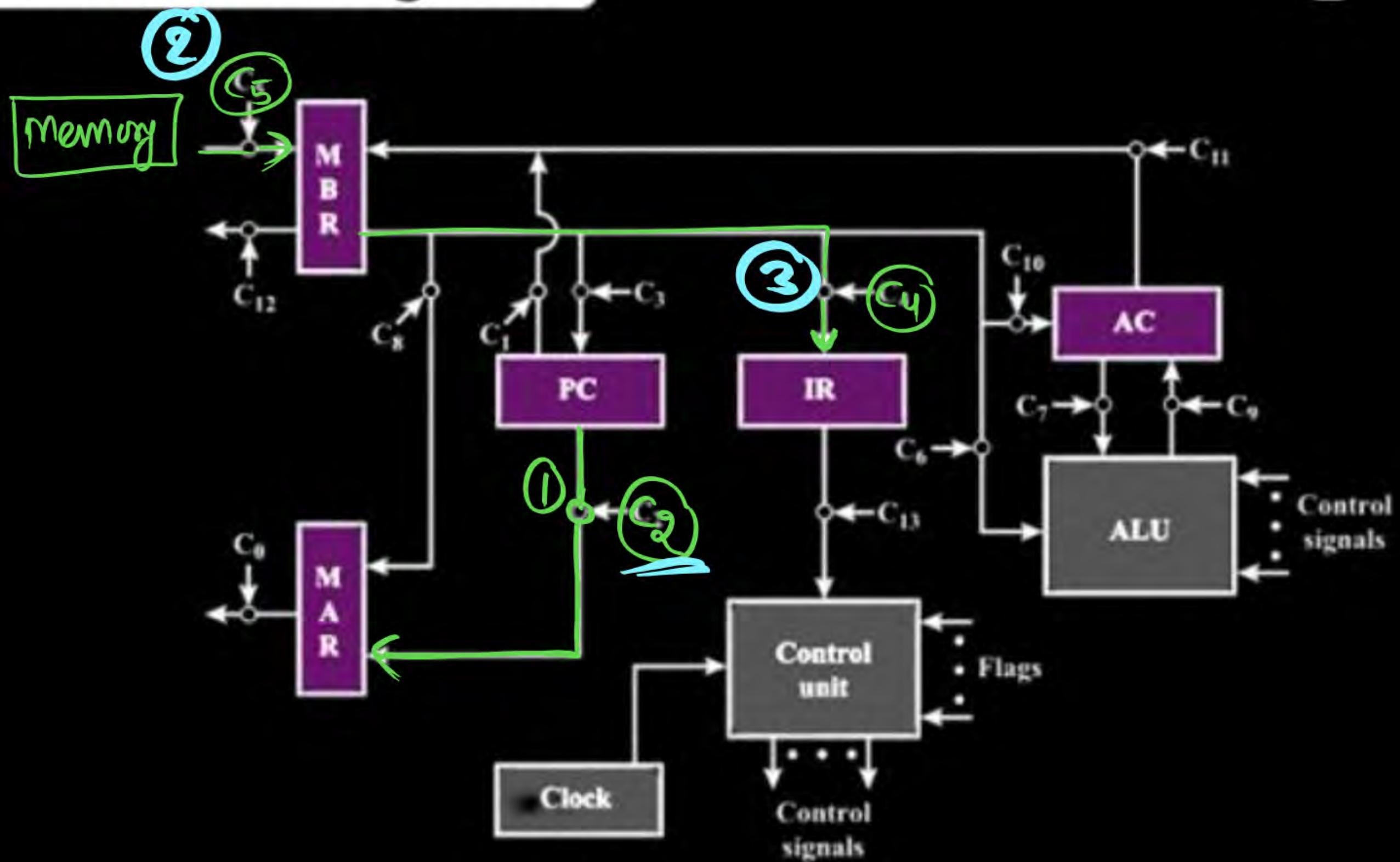


Control Unit : Control Word.

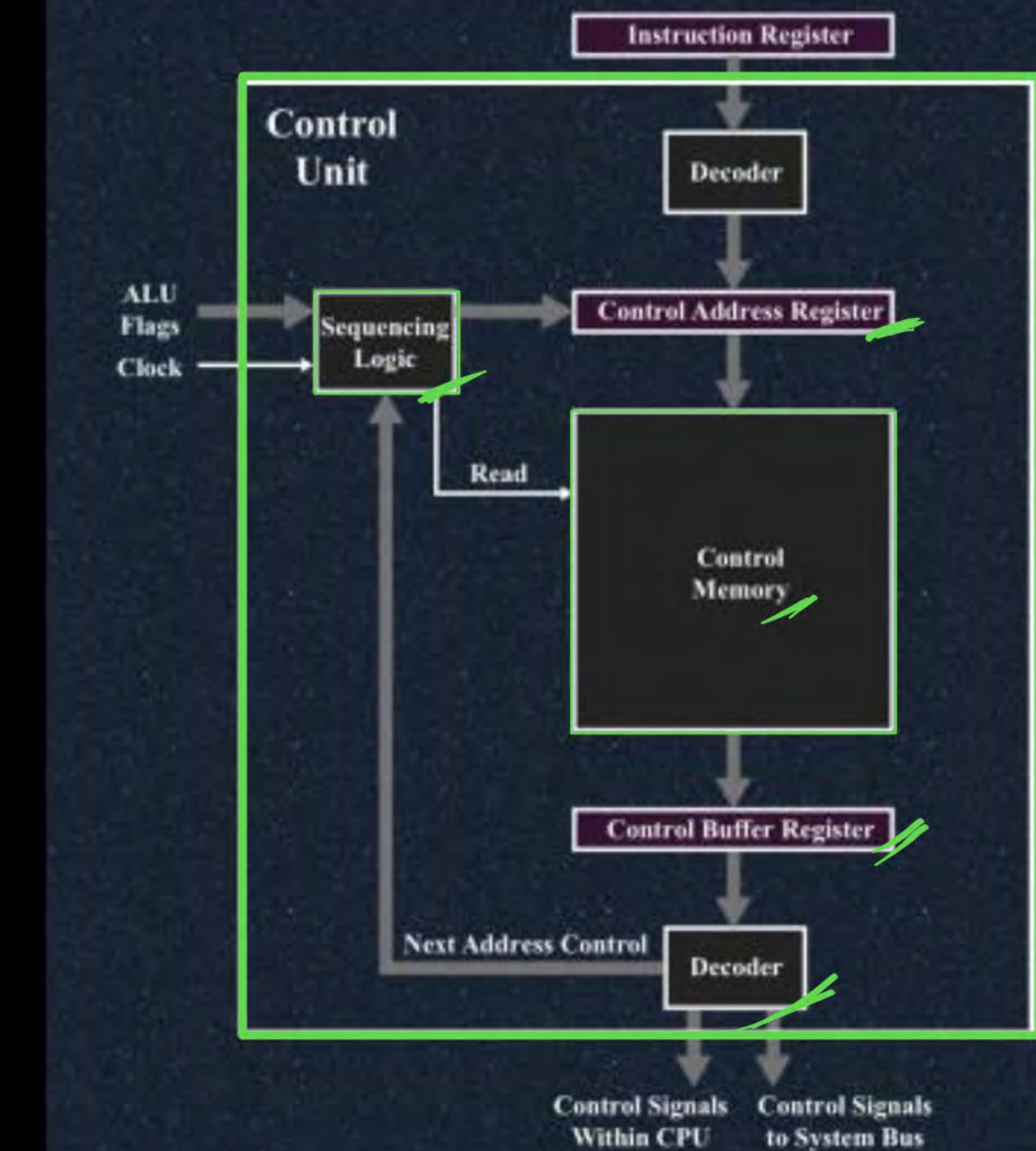
$MBR \rightarrow IR$; $MBR_{out} DR_{in}$



Data Paths & Control Signals



Functioning of Microprogrammed Control Unit



Pre Requirement of the CU Design is as follow

- 1) How many Control Lines are present in the Hardware [S0, S1, S2, S3...]
- 2) How many Instruction are implemented in the Hardware [I1, I2, I3....]
- 3) How many Micro operation are required for each Instruction [T1, T2, T3...]
- 4) What the control Signal Required for each micro operation for each Instruction.

Control Signals will be Implemented into the Control Unit by using following Approach:

- 1) HARDWIRED CU Design
- 2) MICRO-PROGRAMMED CU Design

Q.

Consider the following hypothetical CPU which uses 3 data register A, B & C and supports 3 instruction I_1 , I_2 & I_3 . Obtain the logic function that will generate the hardwired control for the signal A_{in} & B_{out} with the following data.

	I_1	I_2	I_3
T_1	A_{in}, B_{out}	A_{in}, C_{in}, B_{out}	B_{in}, B_{out}
T_2	B_{in}, C_{in}, A_{out}	A_{in}, A_{out}	A_{in}, B_{in}, C_{out}
T_3	B_{in}, B_{out}	B_{in}, B_{out}	B_{in}, B_{out}
T_4	C_{in}, A_{out}	B_{in}, A_{out}	A_{in}, A_{out}
T_5	End	End	End

Step 1 : Search where the control signals Ain & Bout are present.

Step 2 : Options are in I.T format or T.I format.

Step 3 : For any particular time interval. Is the control signal presents for all the instructions?

Step 1 : Search where the control signals A_{in} & B_{out} are present.

Step 2 : Options are in I.T format or T.I format.

Step 3 : For any particular time interval. Is the control signal presents for all the instructions?

$$A_{in} = T_1 I_1 + (T_1 + T_2) I_2 + (T_2 + T_4) I_3$$

$$B_{out} = T_1 + T_3$$

↓

B_{out} is present for all instruction during T_1 & T_3

Q.

A hardwired CPU uses 10 control signals S1 to S10 in various time steps T1 to T5 to implement 4 instructions I1 to I4 as shown below.

	T1	T2	T3	T4	T5
I1	S1, S3, S5	S2, S4, S6	S1, S7	S10	S3, S8
I2	S1, S3, S5	S8, S9, S10	S5, S6, S7	S6	S10
I3	S1, S3, S5	S7, S8, S10	S2, S6, S9	S10	S1, S3
I4	S1, S3, S5	S2, S6, S7	S5, S10	S6, S9	S10

Which of the following pairs of expressions represent the circuit for generating control signals S5 and S10 respectively [$(IJ + Ik) \cdot Tn$ indicates that the control signal should be generated in time step Tn if the instruction being executed is [IJ to IK]]?

- (a) $S5 = T1 + I2 \cdot T3$ and $S10 = (I1 + I3) \cdot T4 + (I2 + I4) \cdot T5$
- (b) $S5 = T1 + (I2 + I4) \cdot T3$ and $S10 = (I1 + I3) \cdot T4 + (I2 + I4) \cdot T5$
- (c) $S5 = T1 + (I2 + I4) \cdot T3$ and $S10 = (I2 + I3 + I4) \cdot T2 + (I1 + I3) \cdot T4 + (I2 + I4) \cdot T5$
- (d) $S5 = T1 + (I2 + I4) \cdot T3$ and $S10 = (I2 + I3) \cdot T2 + I4 \cdot T3 + (I1 + I3) \cdot T4 + (I2 + I4) \cdot T5$

Q.

A CPU has only three instructions I1, I2 and I3, which use the following signals in time steps T1—T5:

I1: T1 : Ain, Bout, Cin
T2 : PCout, Bin
T3 : Zout, Ain
T4 : PCin, Bout
T5 : End

I2: T1 : Cin, Bout, Din
T2 : Aout, Bin
T3 : Zout, Ain
T4 : Bin, Cout
T5 : End

I3: T1: Din, Aout
T2 : Ain, Bout
T3 : Zout, Ain
T4 : Dout, Ain
T5 : : End

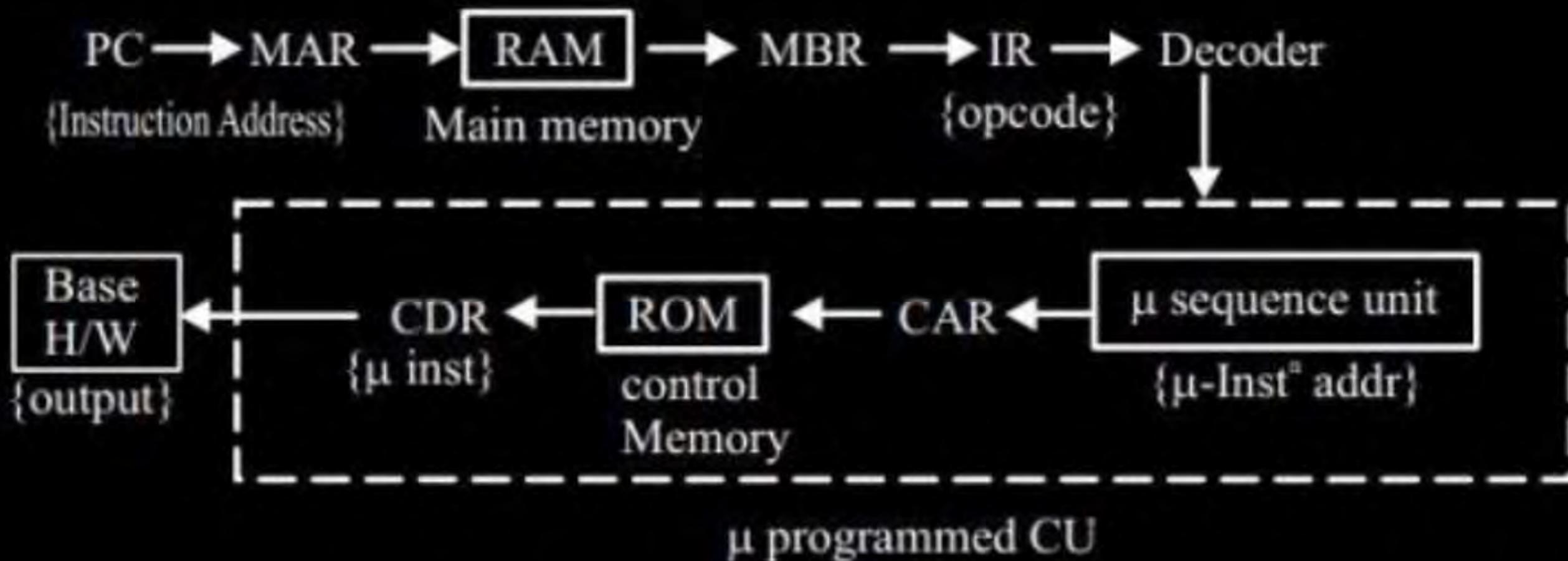
P
W

Which of the following logic functions will generate the hardwired control for the signal Ain?

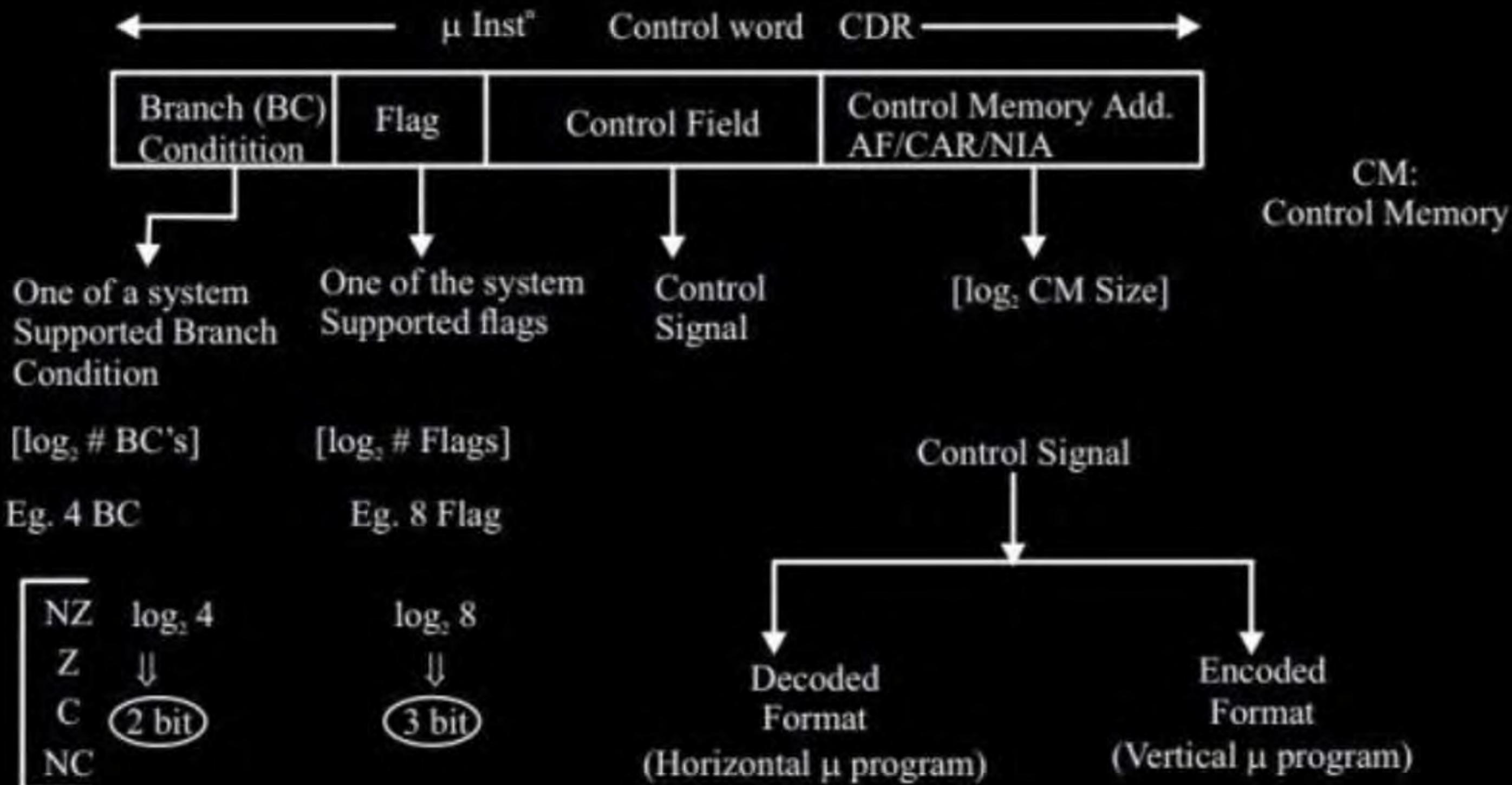
[GATE CSE 2004]

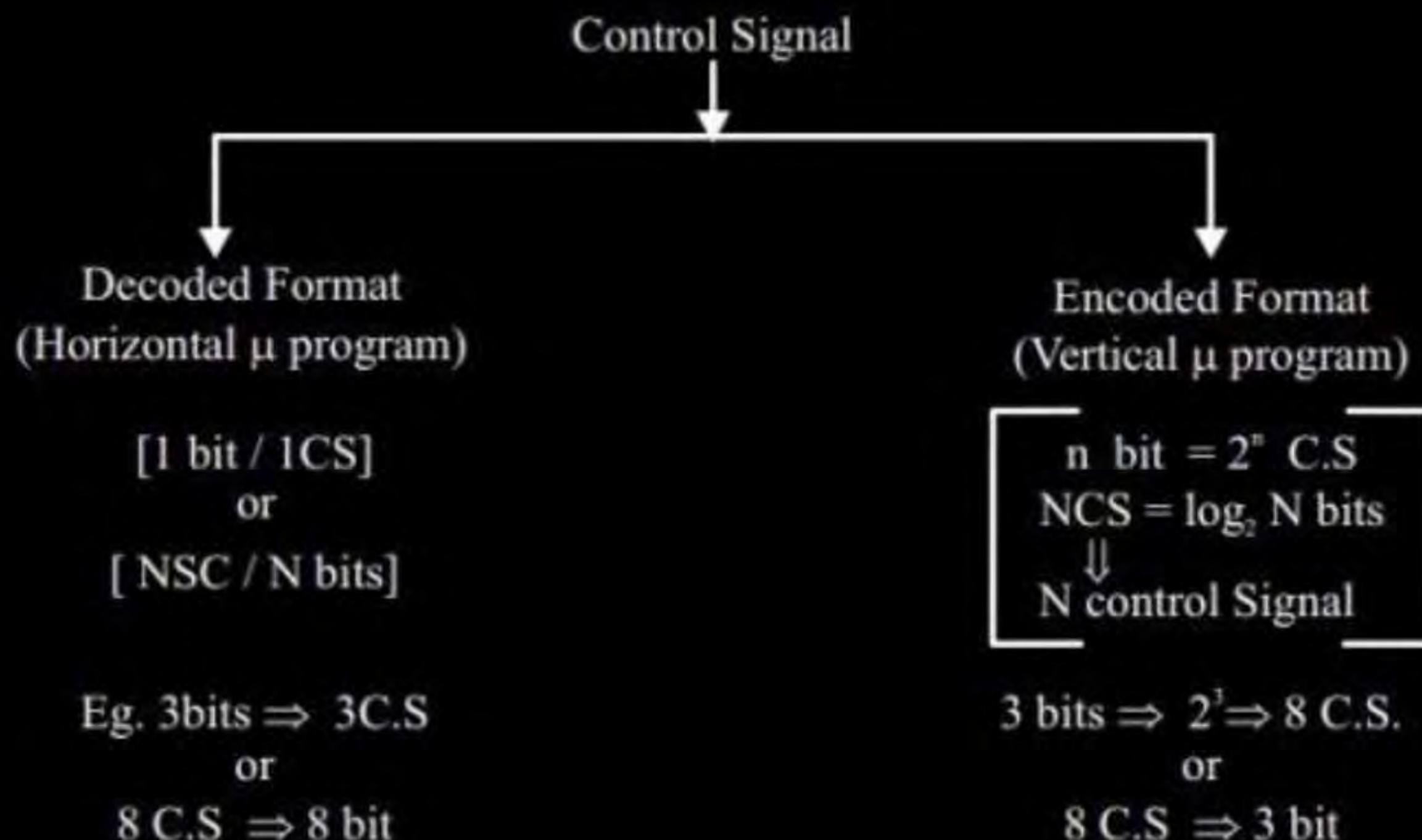
- (a) $T1 \cdot I1 + T2 \cdot I3 + T4 \cdot I3 + T3$
- (b) $(T1 + T2 + T3) \cdot I3 + T1 \cdot I1$
- (c) $(T1 + T2) \cdot I1 + (T2 + T4) \cdot I3 + T3$
- (d) $(T1 + T2) \cdot I2 + (T1 + T3) \cdot I1 + T3$

MICRO-PROGRAMMED CU DESIGN



MICRO INSTRUCTION FORMAT





Q.

P
W

Arrange the following configuration for CPU in decreasing order of operating speeds: Hardwired control, vertical micro-programming, horizontal micro-programming.

- (a) Hardwired control, vertical micro programming, horizontal micro programming.
- (b) Hardwired control, horizontal micro programming, vertical microprogramming
- (c) Horizontal micro programming, vertical micro programming. Hardwired control
- (d) Vertical micro programming, horizontal micro programming, hardwired control

Q.

Horizontal microprogramming.

P
W

- (a) does not require use of signal decoders
- (b) Results in larger sized microinstructions than vertical microprogramming
- (c) use one bit each control signal
- (d) All of the above

Q.

An instruction set of a processor has 125 signals which can be divided into 5 groups of mutually exclusive signals as follows:

P
W

Group 1 : 20 signals. Group 2 : 70 signals, Groups 3 : 2 signals.
Groups 4 : 10 signals, Groups 5 : 23 signals.

How many bits of the control words can be saved by using vertical microprogramming over horizontal microprogramming?

- (a) 0
- (b) 103
- (c) 22
- (d) 55

Q.

Consider a Hypothetical CU it has 1024 control words memory. Is support 48 control signals & 16 Flags. What is the size of the control words in bits & control memory in byte using

P
W

- (i) Horizontal Programming?
- (ii) Vertical Programming?

Q.

Control field of a micro Instruction support 2 groups of control signal in which Group 1 Indicate None/One of 400 control signal & Group 2 (Horizontal) Indicate 6 signals. Hardwire contain 16 Flags & 32 Branch condition.

If CAR Register size is 20 bit then what is CDR in bits & control memory in bits?

P
W

Q.

Consider a CPU where all the instructions require 7 clock cycles to complete execution. There are 140 instructions in the instruction set. It is found that 125 control signals are needed to be generated by the control unit. While designing the horizontal micro-programming control unit, single address field format is used for branch control logic. What is the minimum size of the control word and control address register?

P
W

- (a) 125, 7
- (b) 125, 10
- (c) 135, 7
- (d) 135, 10

[GATE IT 2008]

Q.

A micro program control unit is required to generate a total of 25 control signals. Assume that during any microinstruction at most two control signals are active. Minimum number of bits required in the control word to generate the required control signal.

P
W

- (a) 2
- (b) 2.5
- (c) 10
- (d) 12

[GATE CSE 1996]

Q.

A Hypothetical processor contain word length of 12 bits. It support 1 word opcode. Each Instruction takes 8 cycles to complete the execution. Processor contain 256 control signal & 16 flags. Processor used Horizontal Control Unit. 64 Branch is used then what is the size of CAR & CDR in bits of control memory?

P
W

Q.

RISC Reduced Instruction set computer	CISC Complex Instruction set computer
1. It support less number of addressing Mode (AM)	1. It support more number of AM.
2. It support smaller Instruction set	2. It support larger Instruction set.
3. It support more number of Register	3. It support less number of Register
4. It support fixed length Instruction	4. It support variable length Instruction
5. It support 1 Instruction per cycle (CPI=1) (Cycle per Instruction =1)	5. It support number 1 Instruction Per cycle (CPI + 1)
6. It support pipeline successfully	6. It support unsuccessful Pipeline
7. It is the expensive processor used in Real Time application	7. It is the low expensive processor
8. It is a super computer	8. General Purpose computer
9. It uses hardwired control unit. (Motorola processor, power processer, ARM processor)	9. It uses microprogrammed (vertical) control unit (Pentium processer)

**THANK
YOU!**

