

COMPUTER SCIENCE

Computer Organization and Architecture

ALU & Control unit

Lecture_04



Vijay Agarwal sir





TOPICS
TO BE
COVERED

o1 Control Unit

o2 Control Unit Design

Working of Computer.

- DATA PATH
- How DATA is transferred from One Register to Another Reg.
- Role of Multiplexer
- How AR, PC, TR & other Register Work.
- Working of Common (Internal) BUS.
- Timing Signal & Control Signal

Micro operation

- Fetch cycle
 - Execute cycle
 - Interrupt cycle
- } Control word will be generated.

Working | Flow of Micro operation.

Control Unit

Micro Program (μ - Program)

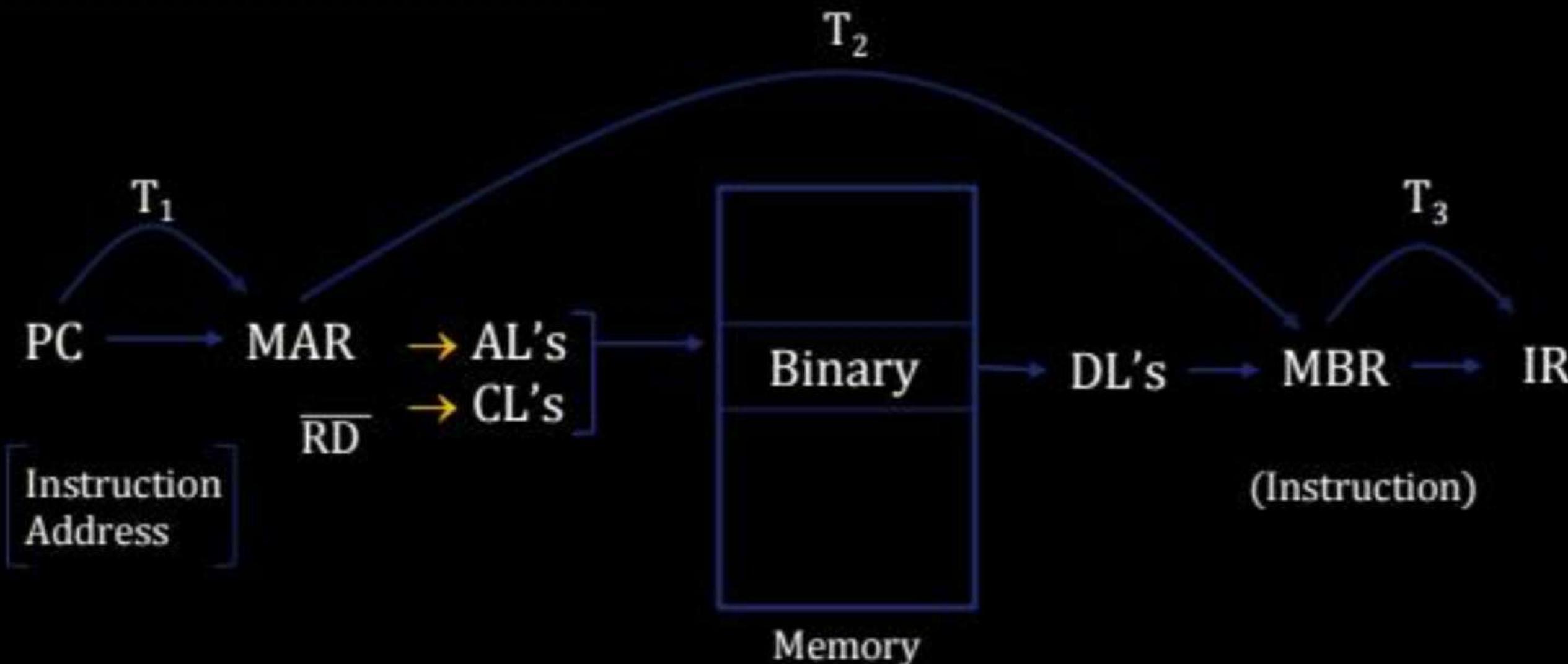
- Sequence of Micro operation [μ operation] to perform Some operation in the Hardware level is called microprogram.

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

- MAR \leftarrow (PC) must precede MBR \leftarrow (memory)

Conflicts must be avoided

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

Execute Cycle

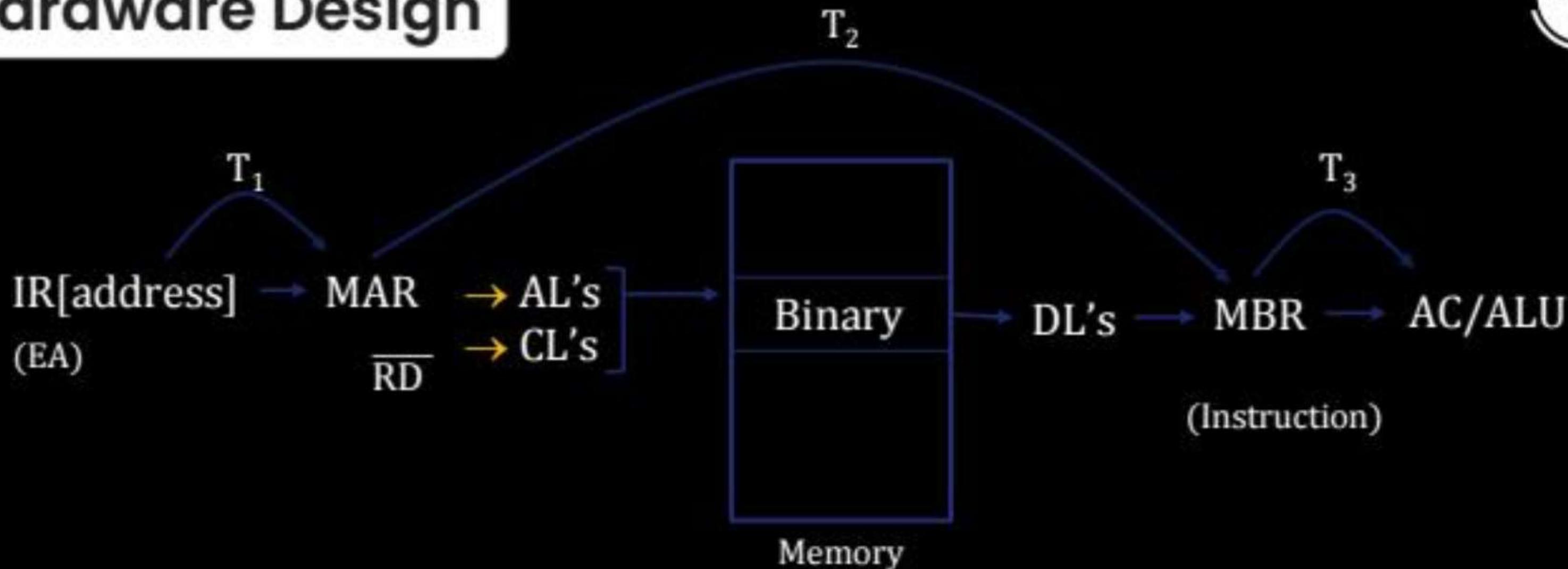
- ❑ Because of the variety of opcodes, there are a number of different sequences of micro-operations that can occur.
- ❑ Instruction decoding
 - ❖ The control unit examines the opcode and generates a sequence of micro-operations based on the value of the opcode
- ❑ A simplified add instruction:
 - ❖ 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 ← m[4000]

Hardware Design



T1:

IR(Address field) → MAR:

IR_{out}

MAR_{in}

T2:

M[MAR] → MBR:

MAR_{out}

MBR_{in}

T3:

MBR → AC/ALU

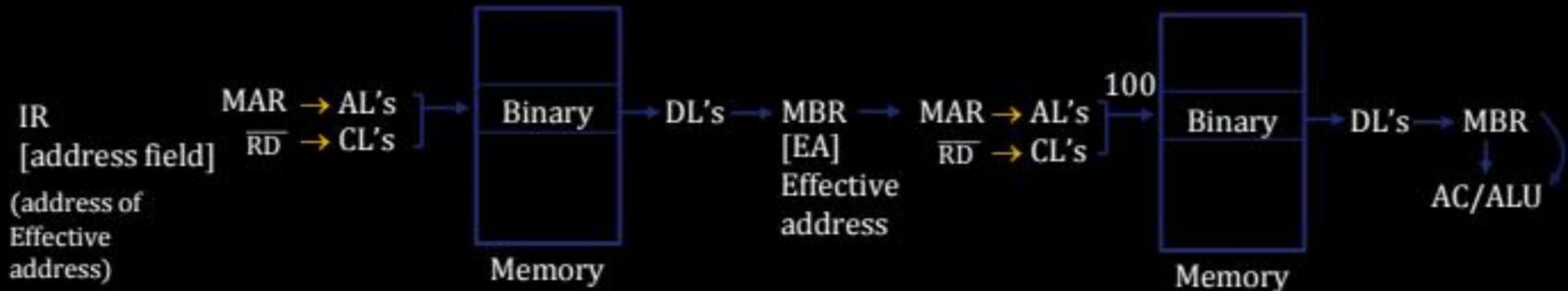
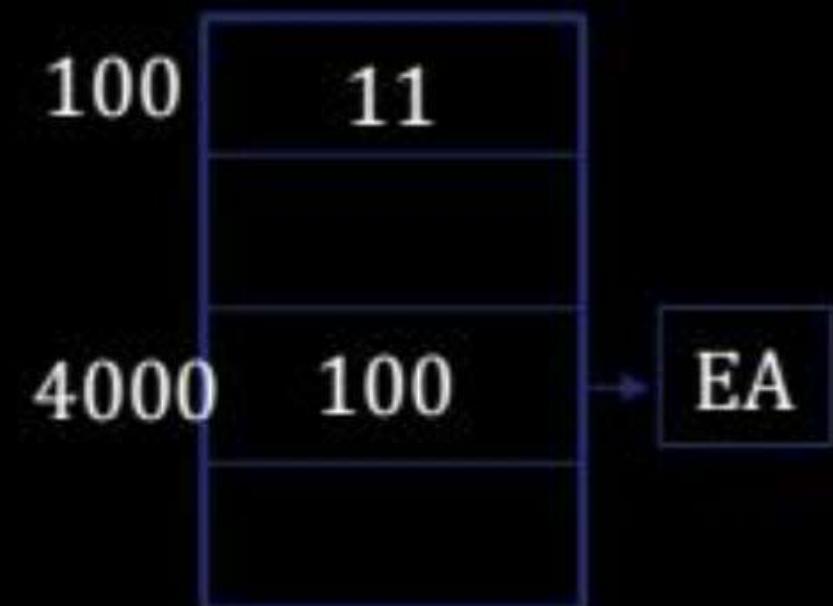
MBR_{out}

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}

Example:

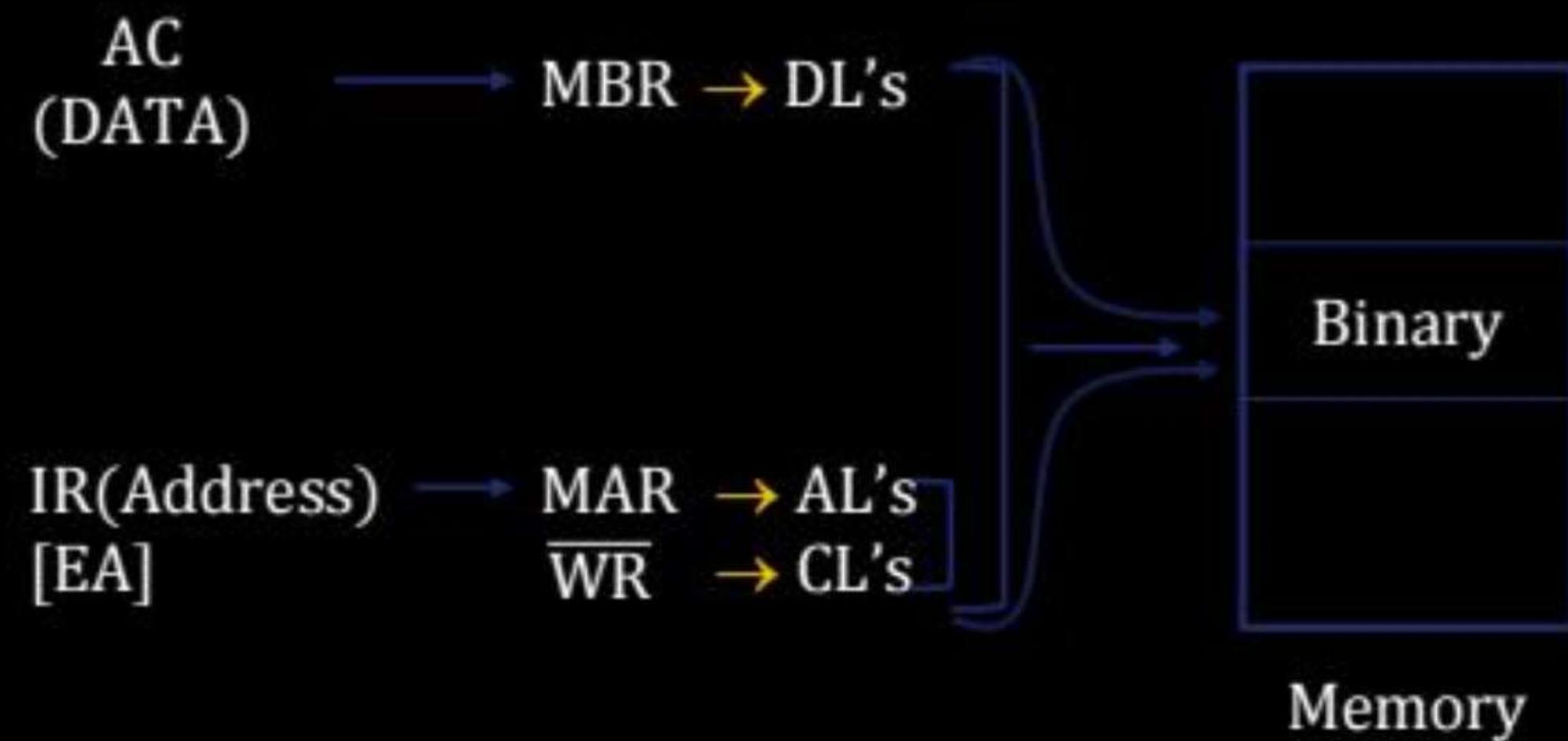
Memory write

Store [6000]

 $M[6000] \leftarrow AC$

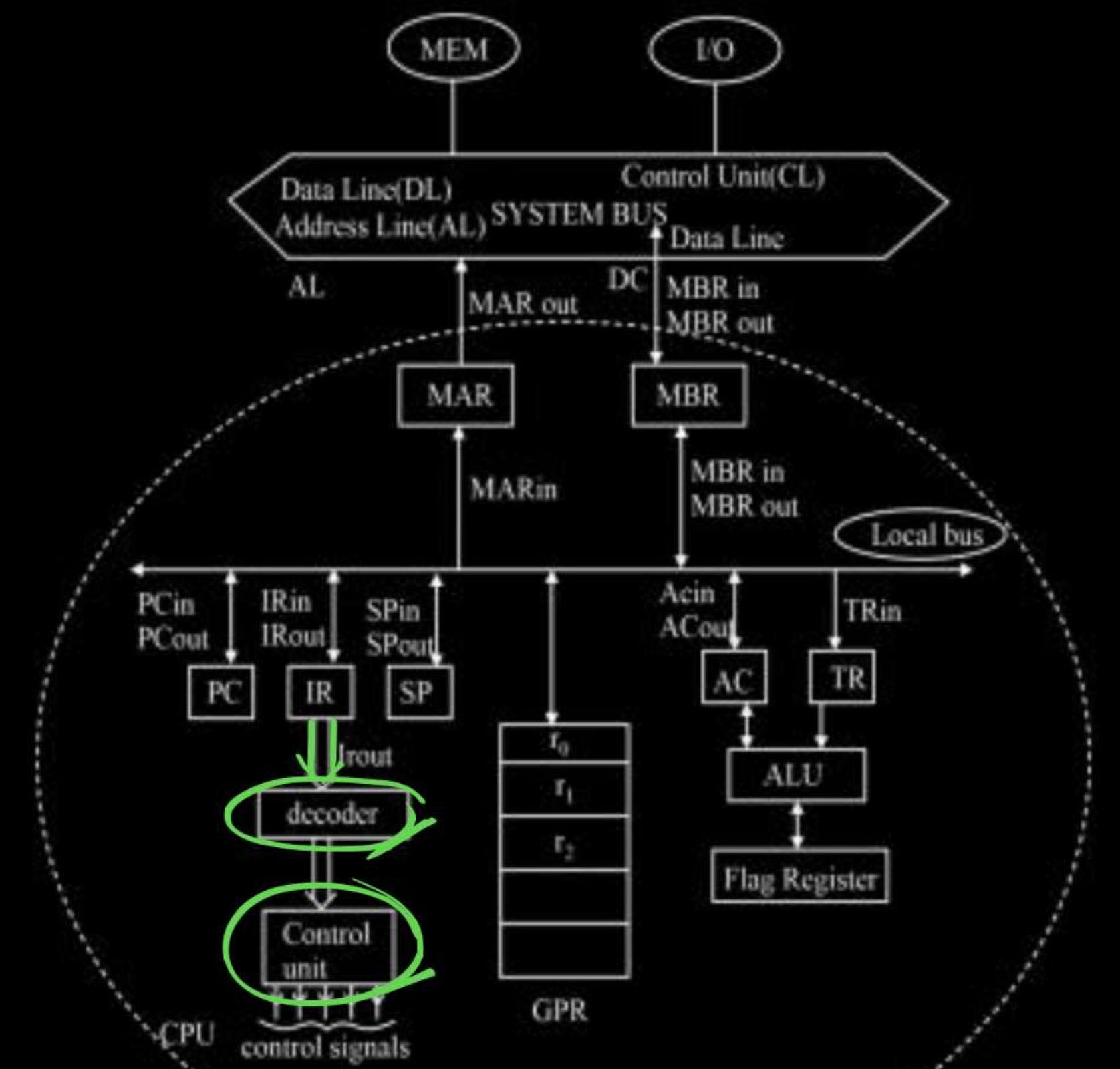
Micro Program

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



Control Unit

- Control Unit is the Supervisor in the System that Control each & every Activity.
- Control Unit takes several input But Produce Control Signals These Control Signal are Required to Perform the Micro operation.



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:
 - ◆ Define basic elements of processor
 - ◆ Describe micro-operations processor performs
 - ◆ Determine the functions that the control unit must perform to cause the micro-operations to be performed
- The control unit performs two basic tasks:
 - ❖ Sequencing → $T_1 : PC \rightarrow MAR$ $PC_{out} MAR_{in}$
 - ❖ Execution

PC _{in}	PC _{out}	MAR _{in}	MAR _{out}	Others

T

PC \rightarrow MAR ; PCout MARin

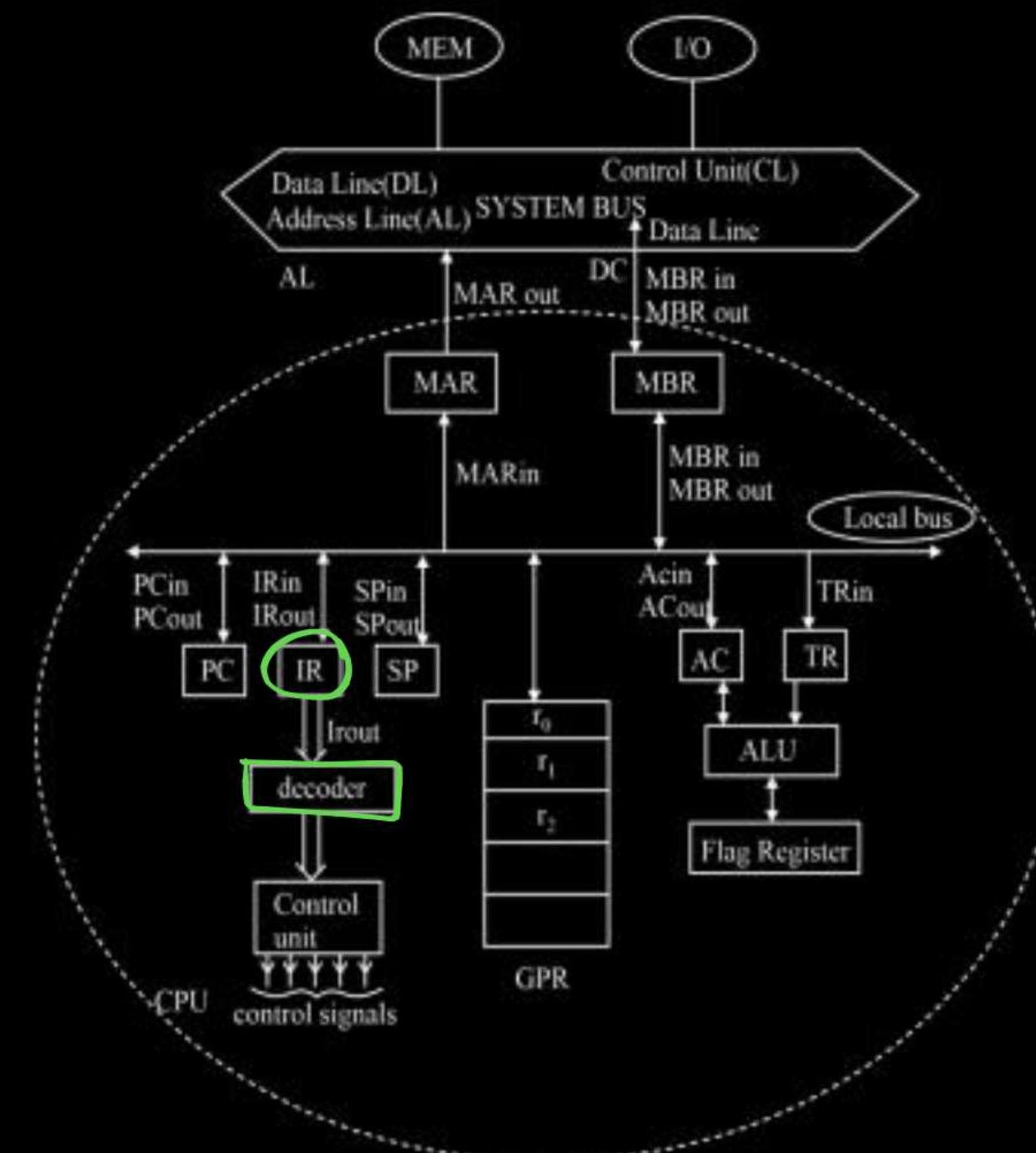
L Control Word

PCin	PCout	AC	IRin	IRout	MBRin	MBRout	ALU	MUX	MARin	MARout
0	1	0	0	0	0	0	-	-	1	0

L Control Word for L Micro operation .

In Fetch & Decode, Same type of operation are performed.

After the Decode, IIP given to Control Unit According to the operation, Control Signal generated to Perform the Micro operation.



Fetch Cycle

IR \leftarrow MEM .

then go to Decoder .

After the Decoder

Control unit generate
Control Signal to Perform
Respective operation .

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:
 - ❖ Define basic elements of processor
 - ❖ Describe micro-operations processor performs
 - ❖ Determine the functions that the control unit must perform to cause the micro-operations to be performed
- ❑ The control unit performs two basic tasks:
 - ❖ Sequencing
 - ❖ Execution

Control Unit

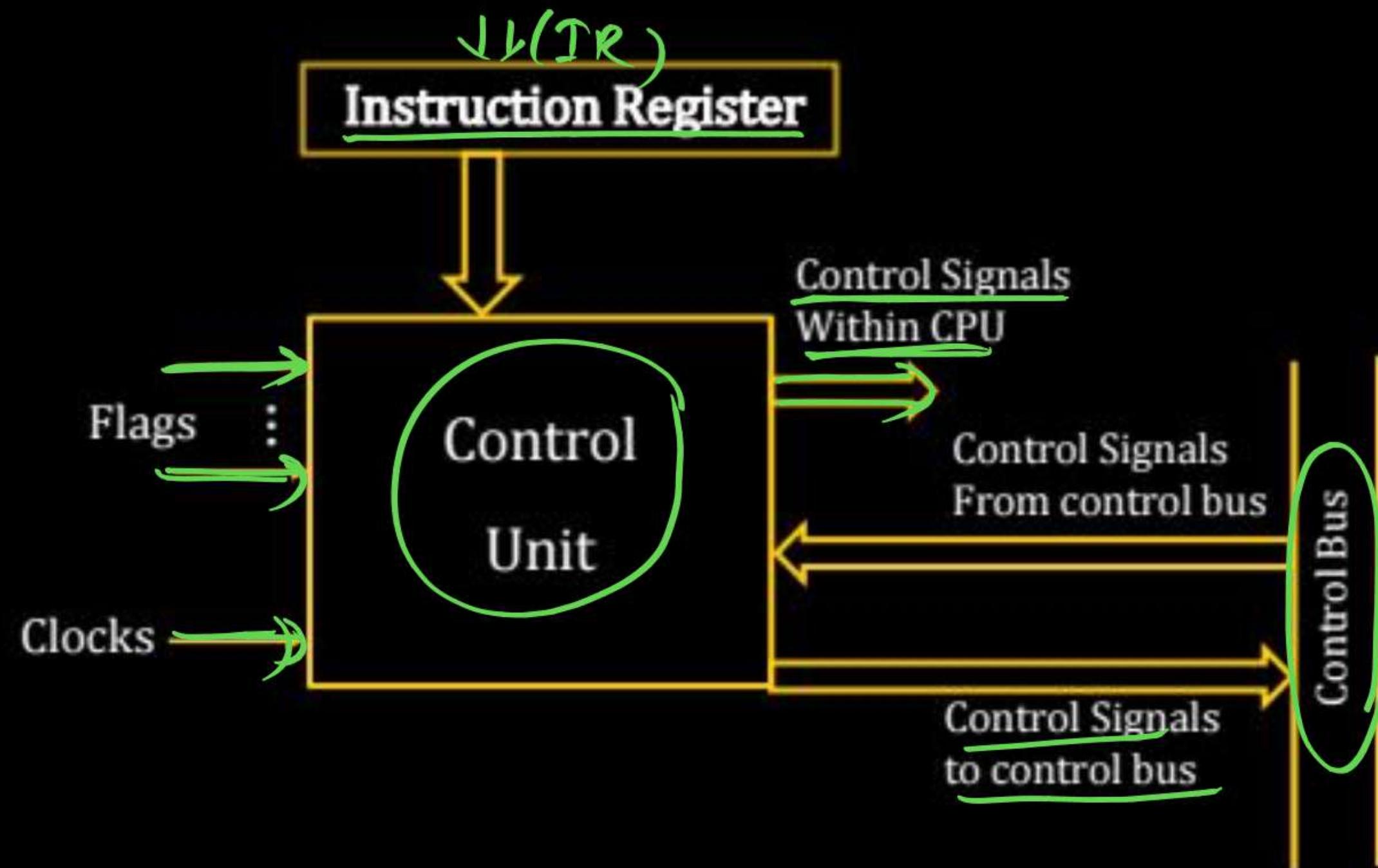
Control unit is the Supervisor in the System that control each & every activity.

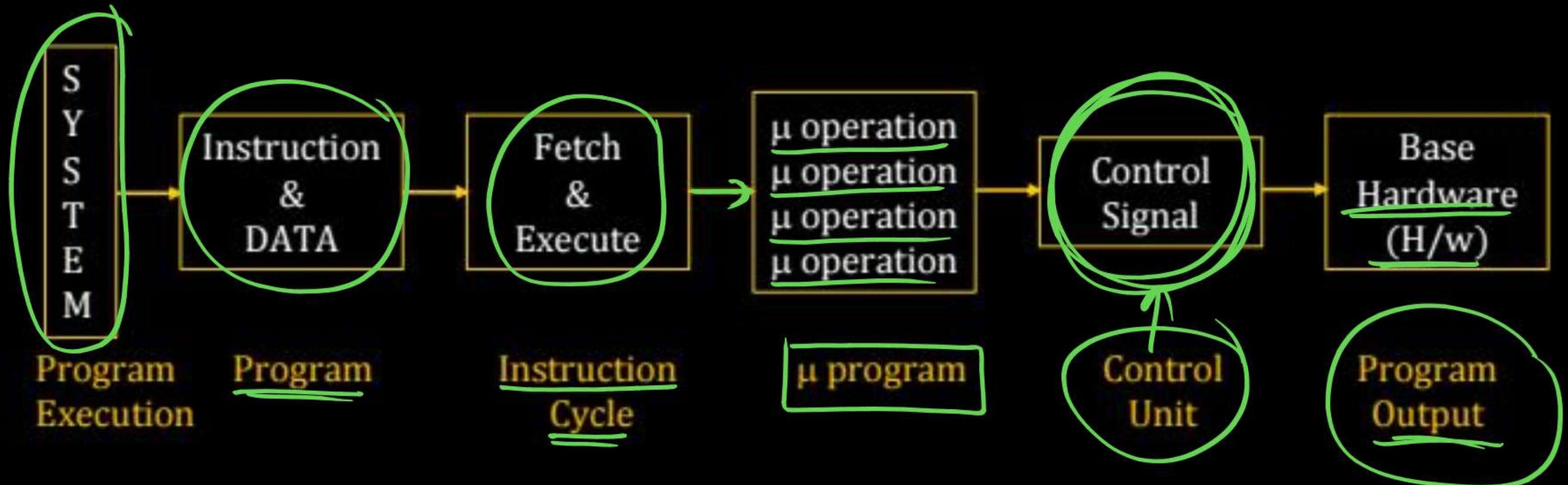
- Control Signals are implemented in a Control Unit.
- Control Signals are Required to execute the micro operation.
- Micro operation is the elementary operation in the hardware.
- Control unit generates the sequence of control Signals.
- Control Signals are Directly executed on a Base Hardware (H/W)

So H/W generate the desired Response.

Computer System Functionality is Program Execution.

Block Diagram of the Control Unit





Control Unit

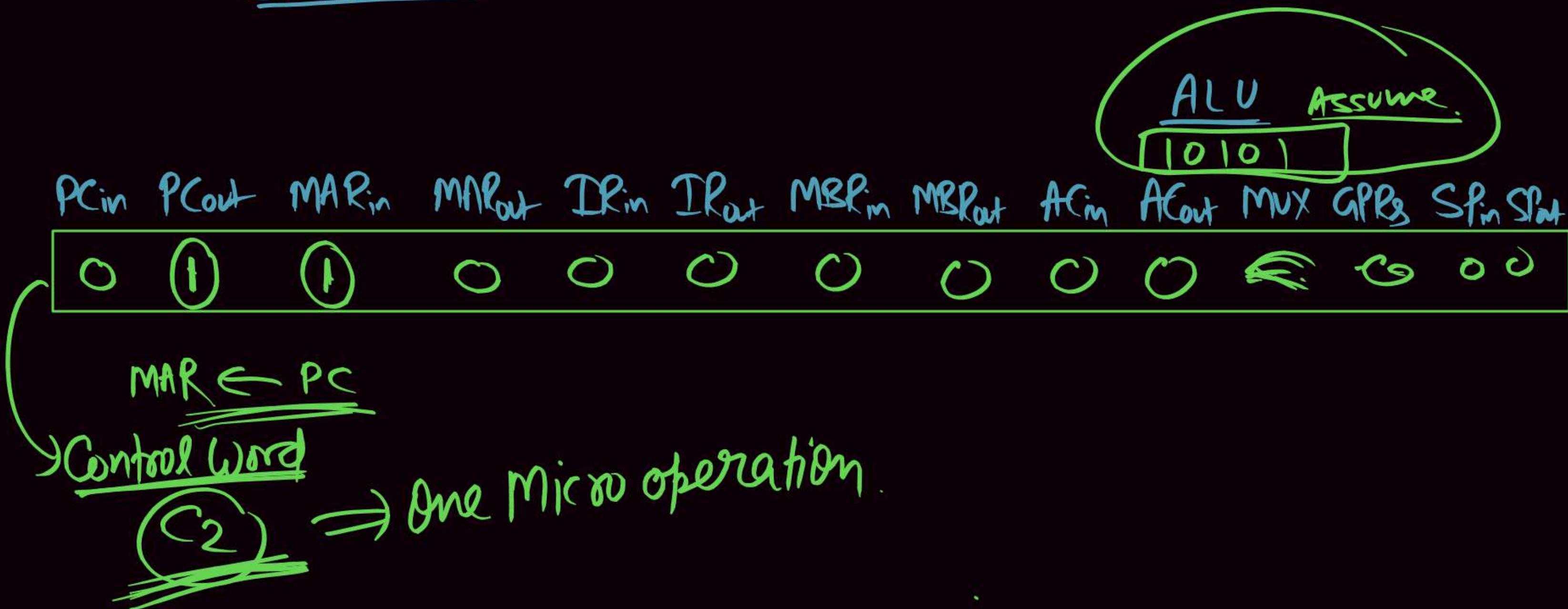
ALU

PC_{in} PC_{out} MAR_{in} MAR_{out} DR_{in} DR_{out} MBR_{in} MBR_{out} AC_{in} AC_{out} MUX GPR₃ SP_{in} SP_{out}

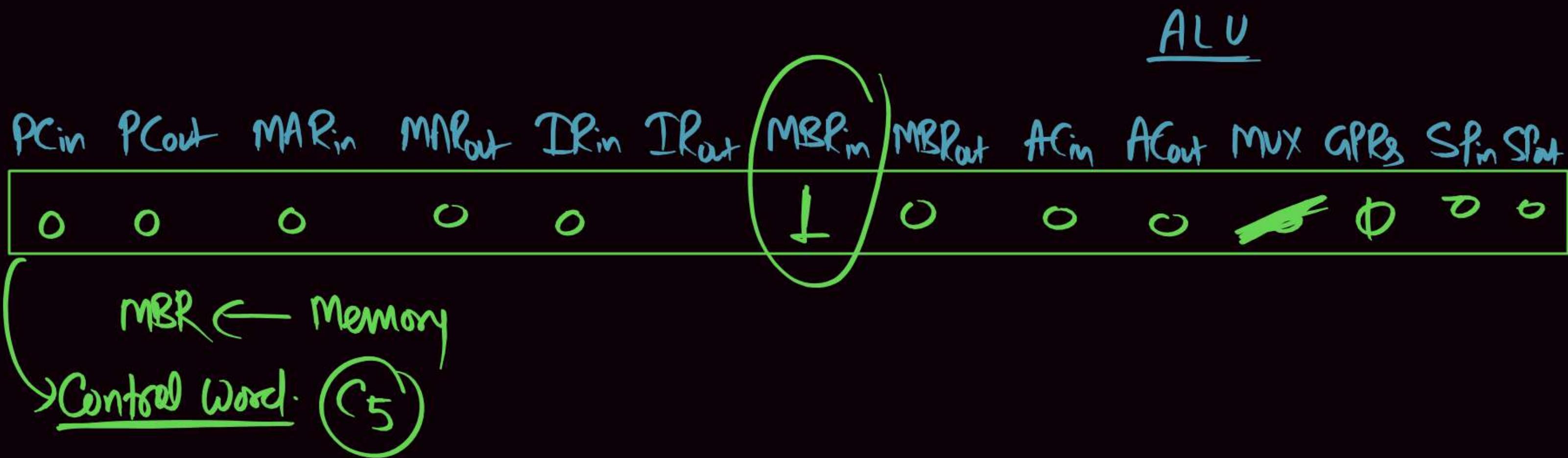
Control Word : Whatever Component

Registers
MUX
BUS
etc.

Control Unit



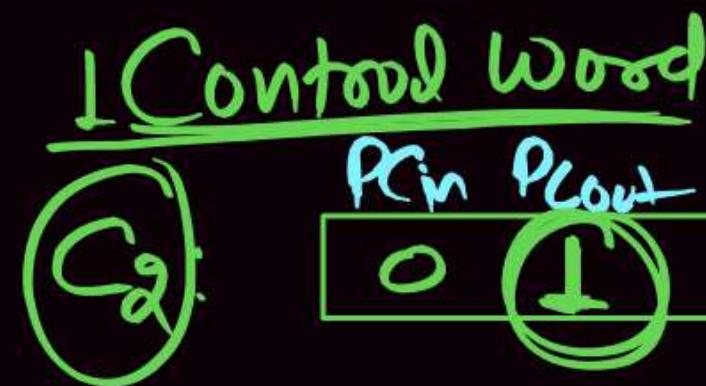
Control Unit





T₂: $M[MAR] \rightarrow MBR ; MAR_{out}, MBR_{in}$

PC _{in}	PC _{out}	AC	IR _{in}	IR _{out}	MBR _{in}	MBR _{out}	MAR _{in}	MAR _{out}
0	0	0	0	0	0	0	1	0

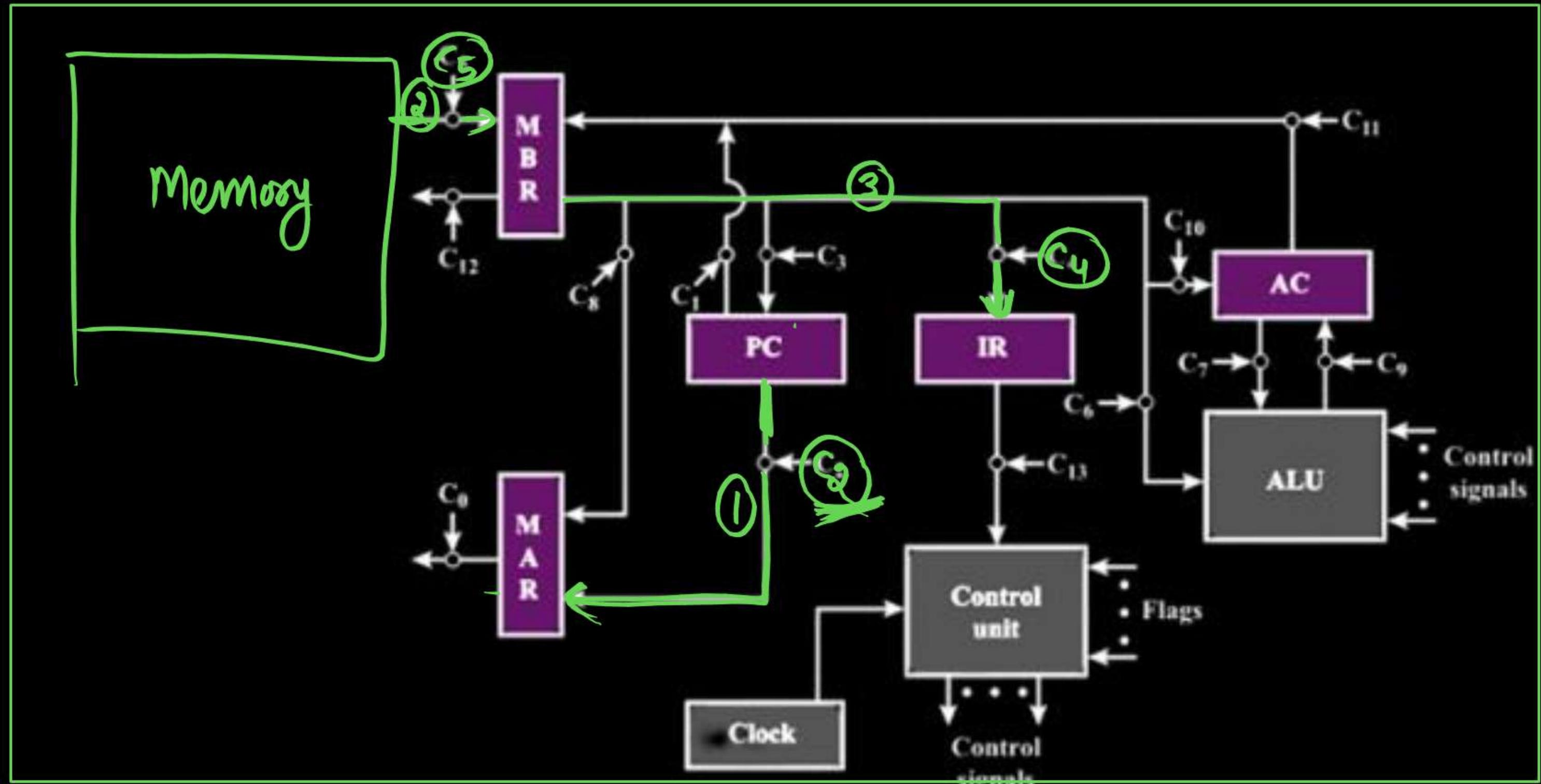


L Control Word: for L Micro operation.

Micro-operations & Control Signals

	Micro-operations	<u>Active-control Signals</u>
Fetch:	T ₁ : MAR \leftarrow PC (or) PC \rightarrow MAR	C ₂
	T ₂ : MBR \leftarrow Memory <u>Mem \rightarrow MBR</u> PC \leftarrow (PC) + 1	C ₅ , C _R
	T ₃ : IR \leftarrow (MBR)	C ₄
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

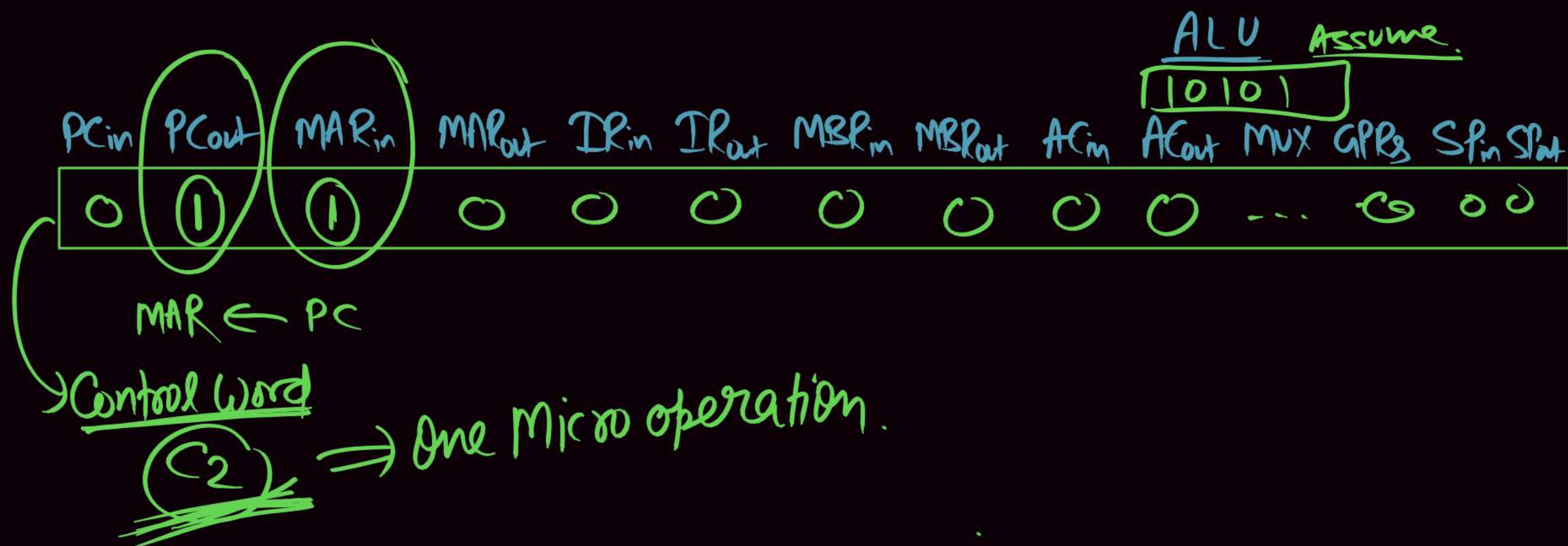
Data Paths & Control Signals



(c): PC \rightarrow MAP

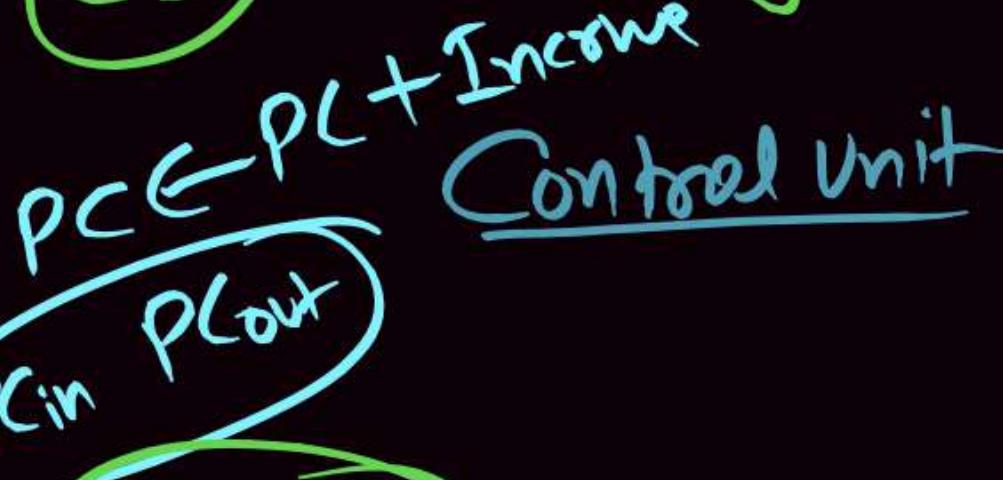
PCout MARin

Control Unit



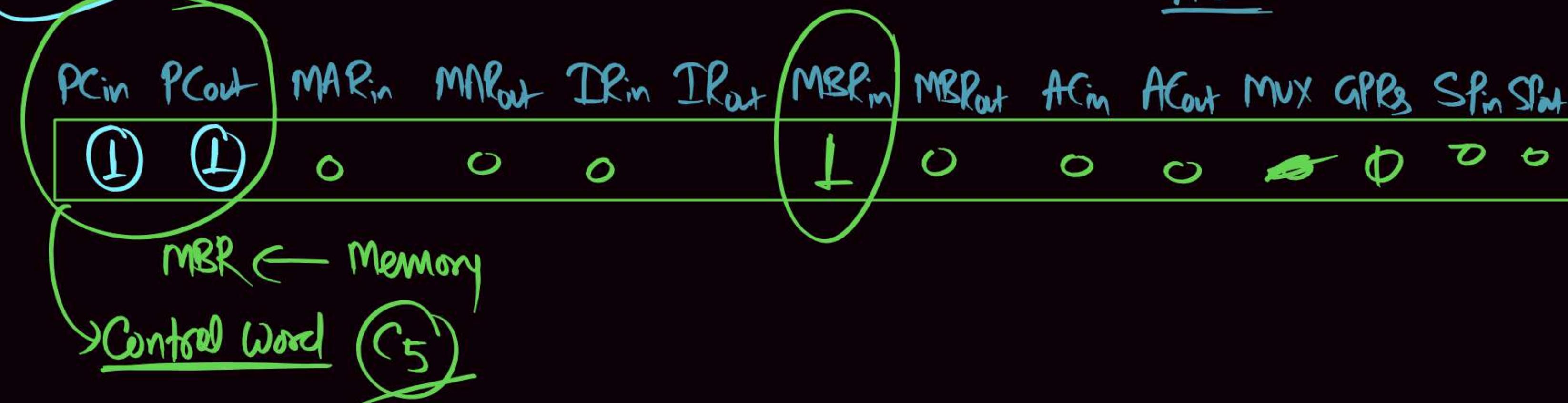
(5)

Memory \rightarrow MBR; MBR_{in} is set to 1



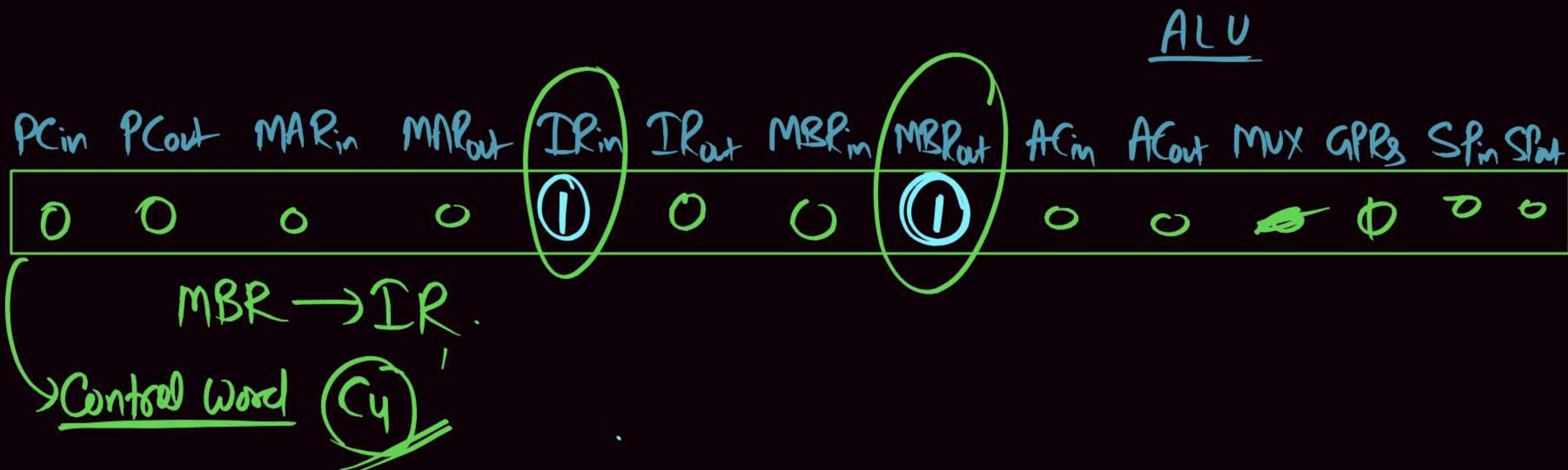
\bar{R} : Read Control Signal

ALU

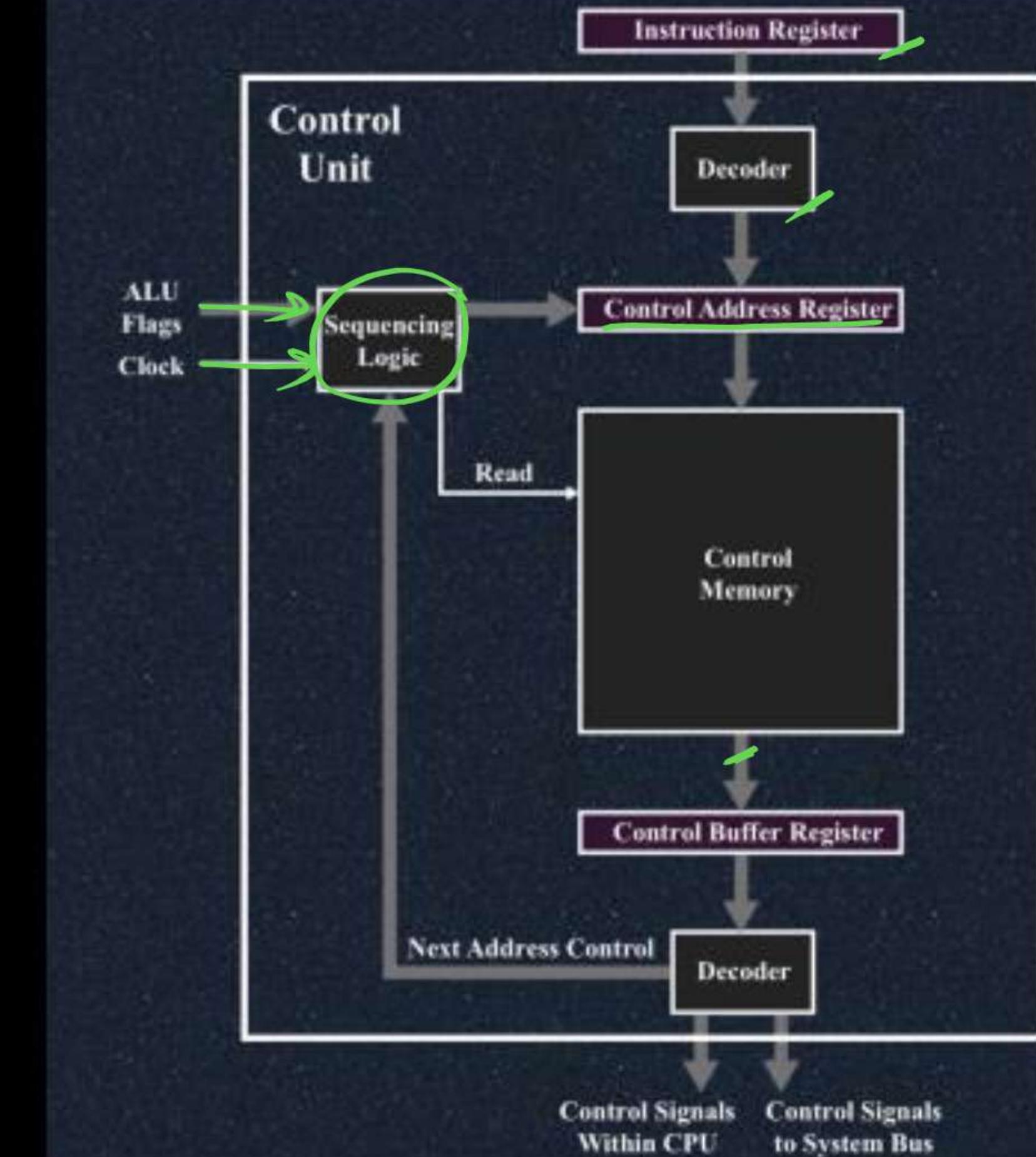


C₄

MBR → DR ; MBR_{out} DR_{in}



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.

<u>Fetch</u>	<u>Execute</u>	<u>Indirect AM</u>
T_1 T_2 T_3	Direct T_1 T_2 T_3	T_1 T_2 T_3 T_4 T_5

Working

- Control Unit generate the Control Signals. These are same for Fetch & Decode.
- After the Decode Different Control Signal (Control Word) are generated According to Different type of the operation.
- At the Control Unit Design time, Designer Decide Which Control Signal are generated in which cycle (T_1, T_2, \dots, T_n) of Different Instruction. That Stored in a Table.

for eg : P.T.O  ↳ After that We Make Boolean function.
(logic)
for implementing.

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 Ain & $Bout$ with the following data.

P
W

for D_1
 $T_1: A \leftarrow B$

$T_1: Bout \leftarrow Ain$

	I_1	I_2	I_3
T_1	<u>$Ain, Bout$</u>	$Ain, Cin, Bout$	$Bin, Bout$
T_2	$Bin, Cin, Aout$	$Ain, Aout$	$Ain, Bin, Cout$
T_3	$Bin, Bout$	$Bin, Bout$	$Bin, Bout$
T_4	$Cin, Aout$	$Bin, Aout$	$Ain, Aout$
T_5	End	End	End

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

for I1

In Time T1 for I1 : S1, S3, S5 Control signal generated.

for I1 Time T2 : S2 S4 S6 is Enable
 (Set = 1) [Enable Set = 1].

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

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

HARDWIRED CU DESIGN

① Control Signal are expressed in S.O.P [Sum of Product] form. they are Directly realized on Hardware.

- In Hardwired CU they used ^{fixed} logic Circuit to Interpret the Instⁿ then Generate the Control Signal .

Advantage : Hardwired CU is Fastest CU.

(RISC) is the Hardwired CU.

HARDWIRED CU DESIGN

DisAdvantage

- It is Not flexible.
- Minor Modification Require Redesigning & Rewiring.
- It does not Support New operation (once designed)

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 Ain & Bout with the following data.

$$Bout =$$

$$T_1 I_1 + T_1 I_2 + T_1 I_3$$

$$T_2 I_1 + T_2 I_2 + T_2 I_3$$

$$\Rightarrow T_1(I_1 + I_2 + I_3)$$

$$+ T_3(I_1 + I_2 + I_3)$$

	I_1	I_2	I_3
T_1	Ain, Bout	Ain, Cin, Bout	Bin, Bout
T_2	Bin, Cin, Aout	Ain, Aout	Ain, Bin, Cout
T_3	Bin, Bout	Bin, Bout	Bin, Bout
T_4	Cin, Aout	Bin, Aout	Ain, Aout
T_5	End	End	End

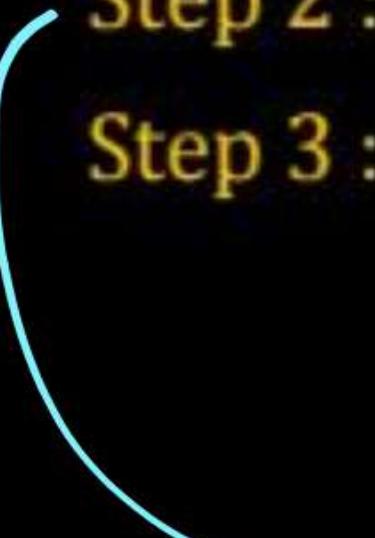
$$Bout = T_1(1) + T_3(1)$$

$$\begin{aligned}Ain &= T_1 I_1 + T_1 I_2 + T_2 I_2 + T_2 I_3 + T_4 I_3 \\&\Rightarrow T_1 I_1 + (T_1 + T_2) I_2 + (T_2 + T_4) I_3.\end{aligned}$$

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?


$$\tau_L I_2 \otimes I_2 \tau_I$$

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?

$$\boxed{\text{Ain} = T_1 I_1 + (T_1 + T_2) I_2 + (T_2 + T_4) I_3}$$

$$\boxed{\text{Bout} = T_1 + T_3}$$

↓ ↓

Bout 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, <u>S5</u>	S2, S4, S6	S1, S7	<u>S10</u>	S3, S8
I2	S1, S3, <u>S5</u>	S8, S9, <u>S10</u>	<u>S5</u> S6, S7	S6	<u>S10</u>
I3	S1, S3, <u>S5</u>	S7, S8, <u>S10</u>	S2, S6, S9	<u>S10</u>	S1, S3
I4	S1, S3, <u>S5</u>	S2, S6, S7	<u>S5</u> , <u>S10</u>	S6, S9	<u>S10</u>

$$S_5 = \underbrace{T_1(I_1 + I_2 + I_3 + I_4)}_{\textcircled{1}} + T_3(I_2 + I_4) \Rightarrow S_5 = T_L + (I_2 + I_4) T_3$$

$$S_5 = T_2 I_2 + T_2 I_3 + T_3 I_4 + T_4 I_1 + T_4 I_3 + T_5 I_2 + T_5 I_4$$

$$S_{10} = \underline{(I_2 + I_3) T_2} + \underline{I_4 T_3} + \underline{(I_1 + I_3) T_4} + \underline{(I_2 + I_4) T_5}$$

Which of the following pairs of expressions represent the circuit for generating control signals S_5 and S_{10} respectively $[(I_J + I_k) \cdot T_n]$ indicates that the control signal should be generated in time step T_n if the instruction being executed is [IJ to IK]?

- (a) $S_5 = T_1 + I_2 \cdot T_3$ and $S_{10} = (I_1 + I_3) \cdot T_4 + (I_2 + I_4) \cdot T_5$
- (b) $S_5 = \underline{T_1} + \underline{(I_2 + I_4)} \cdot T_3$ and $S_{10} = (I_1 + I_3) \cdot T_4 + (I_2 + I_4) \cdot T_5$
- (c) $S_5 = \underline{T_1} + \underline{(I_2 + I_4)} \cdot T_3$ and $S_{10} = (I_2 + I_3 + I_4) \cdot T_2 + (I_1 + I_3) \cdot T_4 + (I_2 + I_4) \cdot T_5$
- (d) ~~$S_5 = \underline{T_1} + \underline{(I_2 + I_4)} \cdot T_3$ and $S_{10} = (I_2 + I_3) \cdot T_2 + \underline{I_4 \cdot T_3} + (I_1 + I_3) \cdot T_4 + (I_2 + I_4) \cdot T_5$~~

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

$$Ain = I_1 T_1 + I_1 T_3 + I_2 T_3 + I_3 T_2 + I_3 T_3$$

$$\Rightarrow I_1 T_1 + I_1 T_3 + I_2 T_3 + I_3 T_3 + I_3 T_2 + I_3 T_4 + I_2 T_4$$

$$\Rightarrow I_1 T_1 + T_3(I_1 + I_2 + I_3) + I_3 T_2 + I_3 T_4$$

$$= I_1 T_1 + I_3 T_2 + I_3 T_4 + T_3.$$

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) $(T1 + T2) \cdot I2 + (T1 + T3) \cdot I1 + T3$

Microprogrammed CU : In Microprogrammed CU Control Word are stored in the Control Memory.

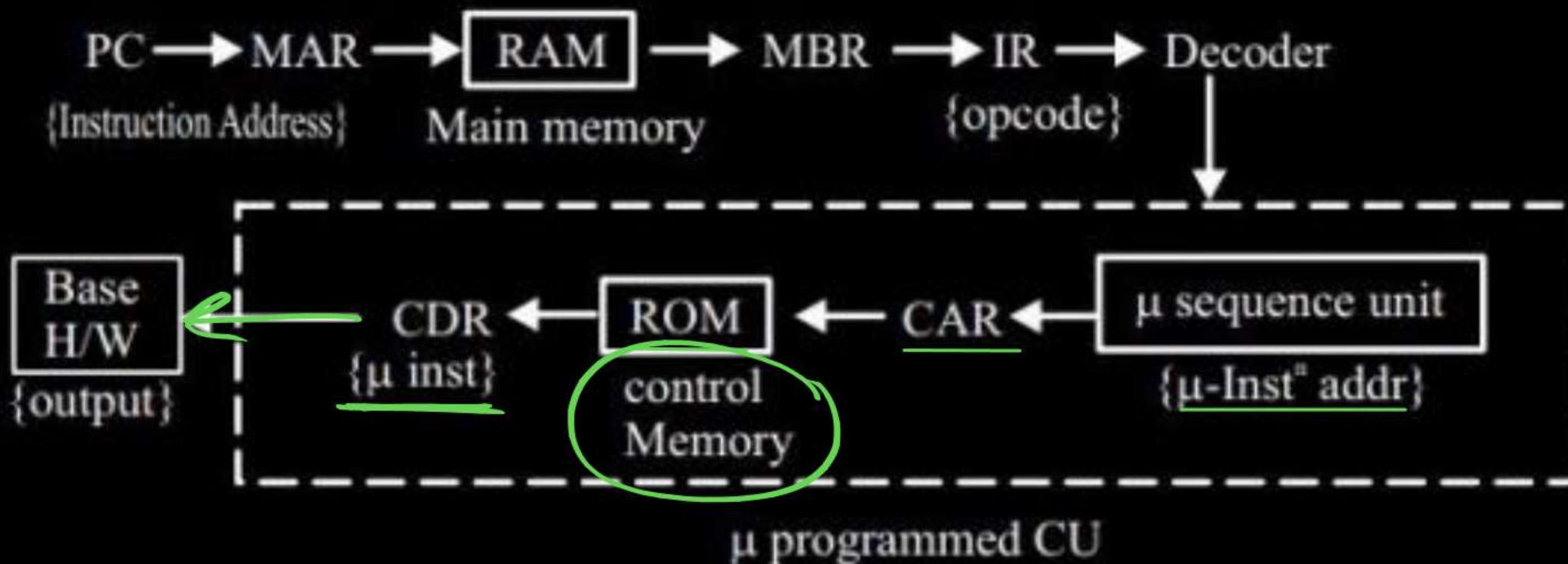
According to the type of operation Control Signal are generated.

Control Memory is Associated with CAR & CDR to Contain the Control Memory Address & DATA Respectively

CAR : Control Address Register

CDR : Control Data Register

MICRO-PROGRAMMED CU DESIGN



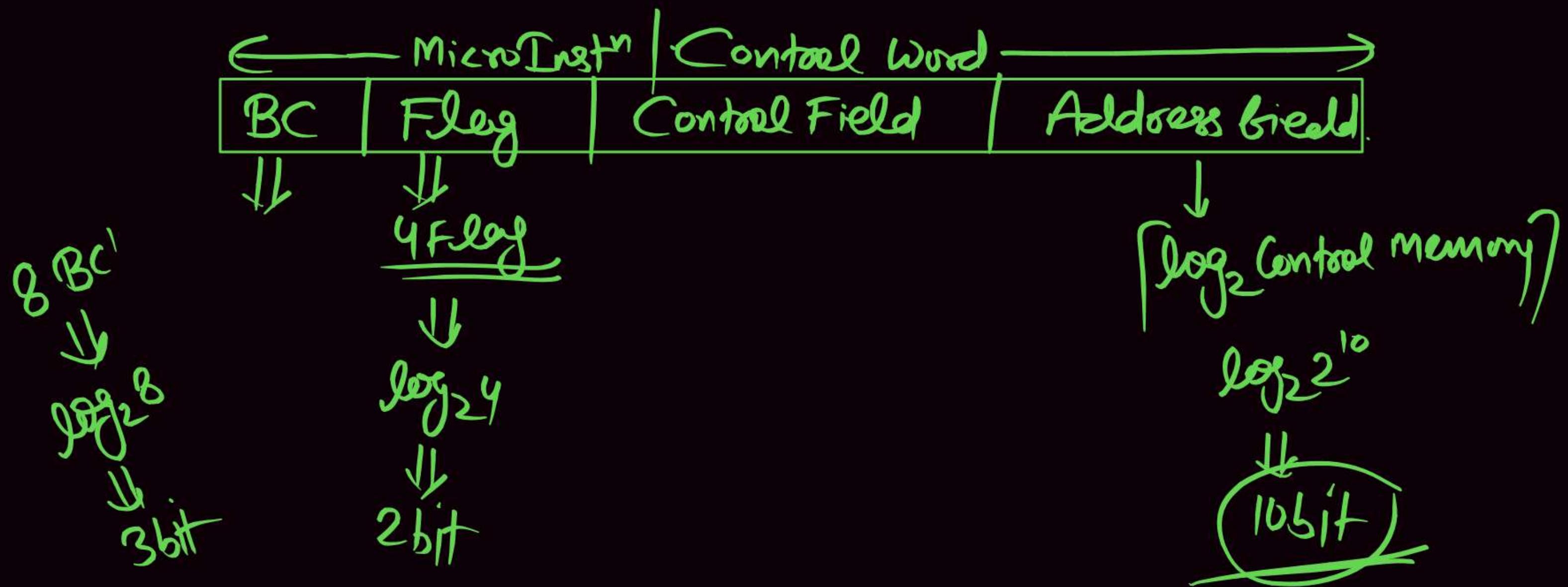
MICRO-PROGRAMMED CU DESIGN

Till Now

Read : Instruction Format

Now we Micro Instruction Format

$$\textcircled{B} \quad CM = 1KB$$



Control field



Control Signal

Decoded
Format

(Horizontal uprogramming)

(1 bit | LCS)
 $N_{CS} \Rightarrow N_{bit}$

Encoded
Format

(Vertical uprogramming)

($n_{bit} = 2^{\log_2 CS}$
 $N_{CS} \Rightarrow \log_2 n_{bits}$).

**THANK
YOU!**

