COMPUTER SCIENCE

Computer Organization and Architecture

ALU & Control unit

Lecture_01

Vijay Agarwal sir

01 Micro Operation

1. Introduction of COA
2. M/c Instruction & AM's
3. Floating Point Representation
4. Micro operation, Program, Data PATH, Control Unit

Micro operation, Micro Program. Data Path, Control Unit

Micro operation.

Instruction cycle.

subcycle.

1. Fetch cycle
2. Execute cycle
   - Decode
   - Execute.

Fetch cycle: To Fetch the Instn from Memory to CPU (IR).

Micro operation:

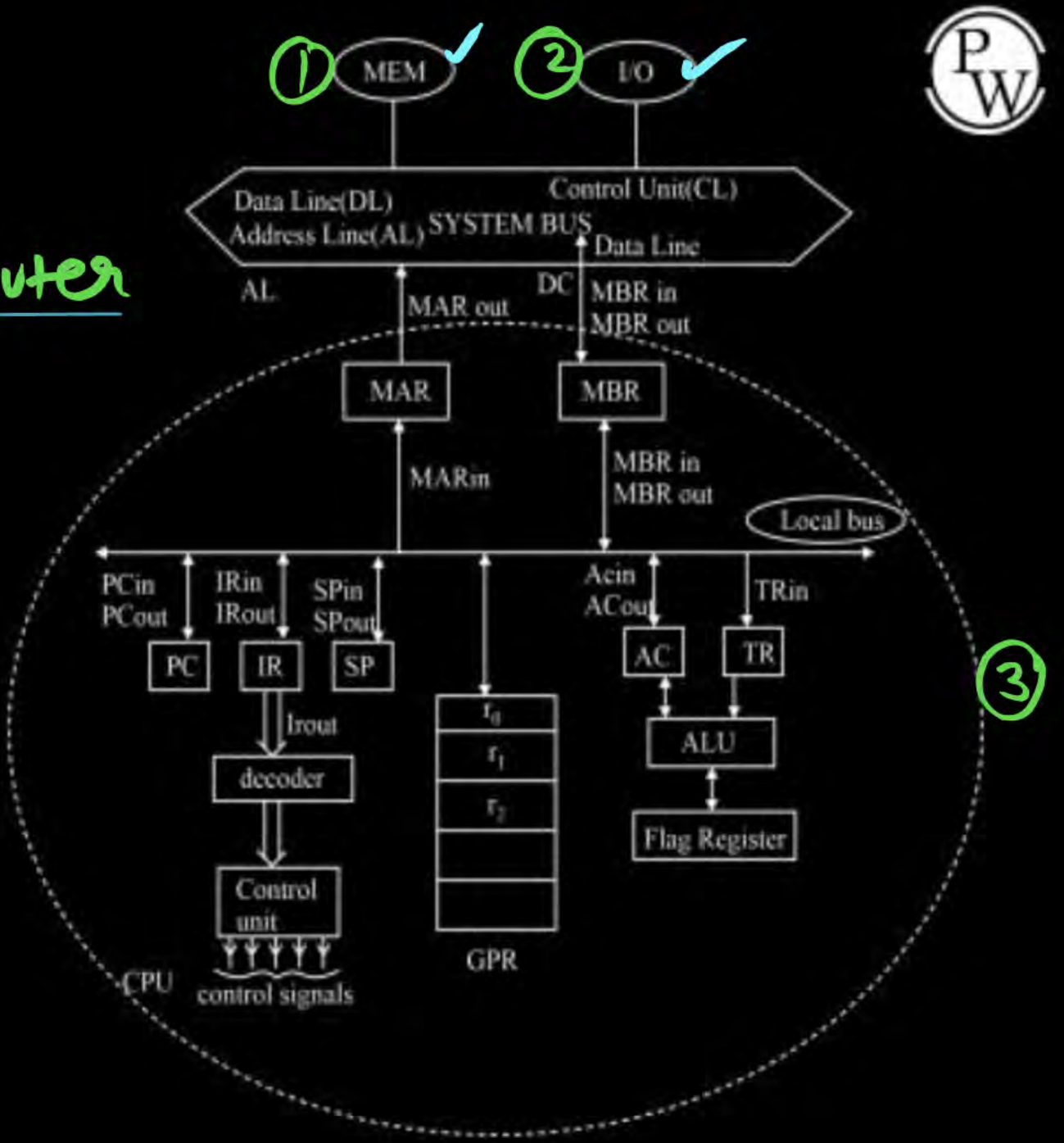
- PC → MAR
- MAR → Memory
- Memory → MBR
- MBR → IR

Working

Structure of Computer

Component of the Computer

1. Memory
2. CPU
3. I/O



Structure of Computer

CPU org.

Registers

ALU

Control Unit

PC

MAR/AR

MBR

IR

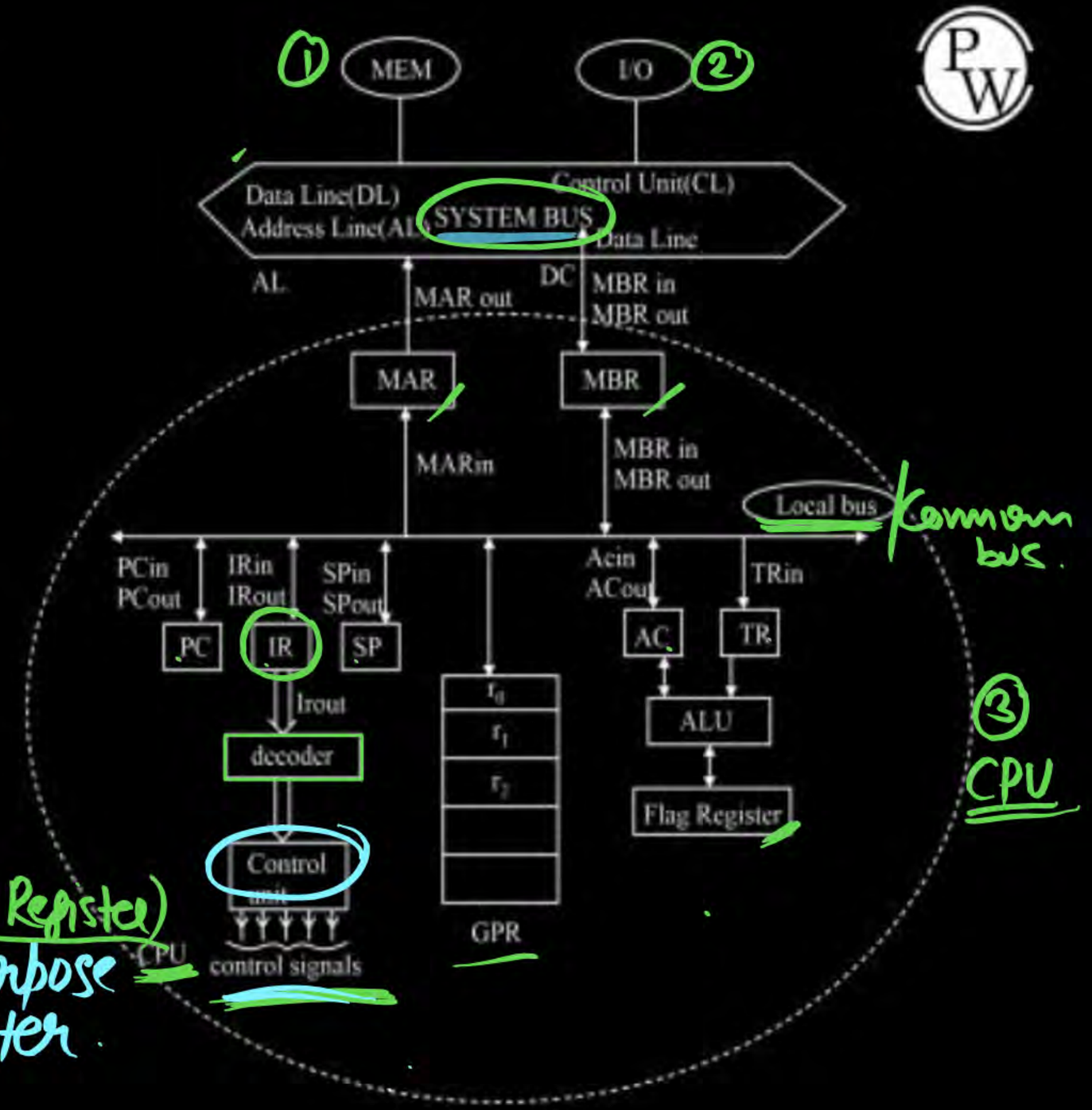
AC

TR

SP

PSW (Flag Register)

General Purpose Register.

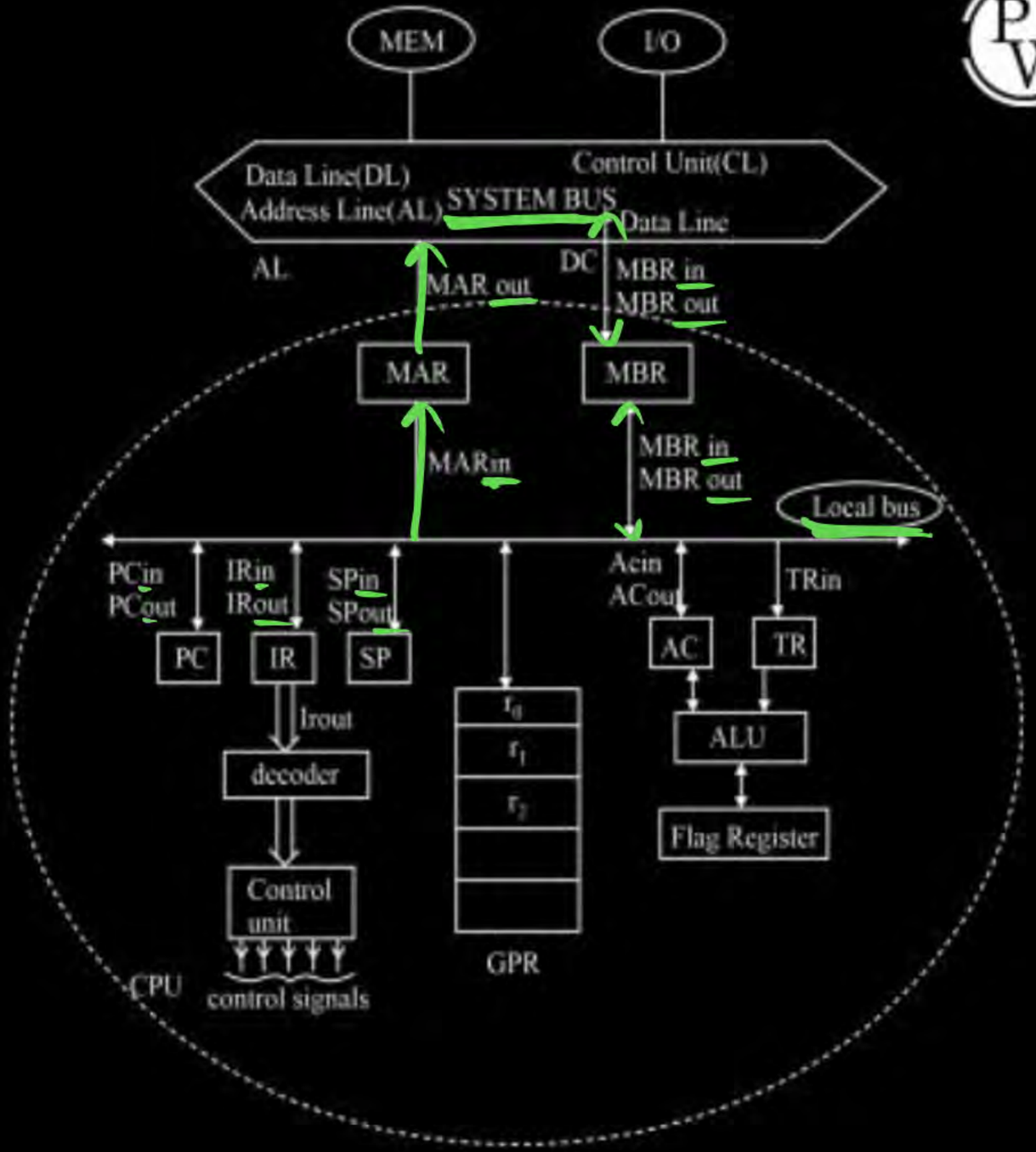


MAR | AR : Connected to the Address line of the System Bus.

MBR | MDR | DR : Connected to the Data line of the System Bus.

System BUS : Which Provide the Communication between Major Component of the Computer (CPU, I/O, Memory).

Structure of Computer



Component of Computer

1. CPU , 2. Memory & 3. IO

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

Common Bus.

CPU

Registery

ALU

Control Unit

PC
MAR
MBR
IR
AC
TR
SP
PSW
GPR

Register : [Flip Flop] : Collection of bits | Sequence of bits stored in Flip-Flop.

↳ Register is Temporary Storage.

Registers ⇒ 

↳ Special Purpose Register [GPR]

↳ General Purpose Register [SPR].

LD: Load

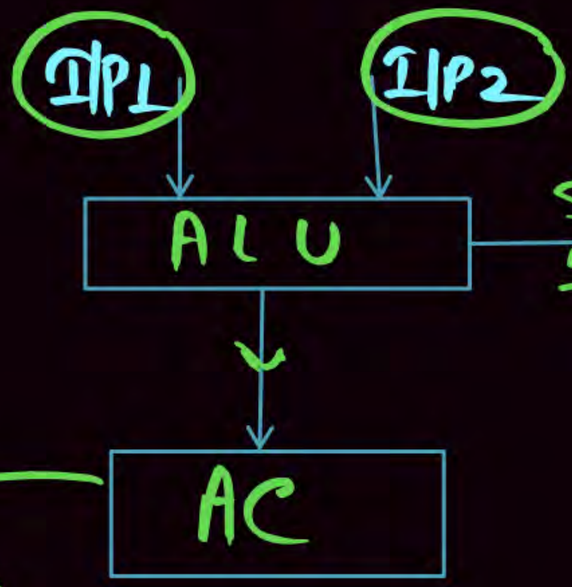
INC: Increment (Binary Counter)

CLR: Clear.

ALU

[Airthmatic & Logical unit]

- Hardware.
- Perform Airthmatic & Logical operation, Comparsion, Condition checking
- Perform Multiple type operation.



Status of Instruction

Flag Register [PSW] (Program Status Word)

1. Carry
2. Sign
3. Parity
4. Auxilliary Carry
5. Zero
6. Overflow.

from the AC Result will be stored in the Respective Destination.

Control Unit

Timing Signal & Control Signal.

Timing Signal: To execute the operation in Proper Sequence.

Control Unit: How & WHAT to Do.
[each & every thing Candidate].

eg
1. Fetch
2. Decode
3. Execute.

Fetch

T_1: PC → MAR

T_2: M[MAR] → memory
mem → MBR
MBR → IR.

eg Non Technical example

IV. Result
III. Exam writing
II. Admit Card
I. enrollment (Registration)

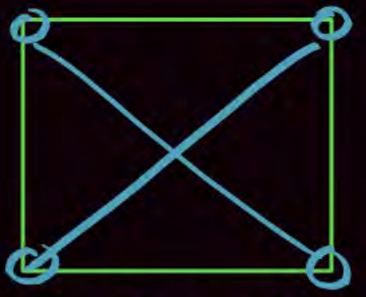
T_1: Enrollement (Registration)
T_2: Admit Card
T_3: Exam writing
T_4: Result.

WHY Common Bus ?..

(eg) If we have 16 Registers, 1 ALU, 1 Memory, 1 PSW & 1 other component

ie Total 20 Component.

$^{20}C_2 \Rightarrow$ 200+ Connections.



So the Solution instead of using 200+ Connection,

Connect all Component to a Common Bus. (Internal Bus)

But only 2 Parts [2 Component] Can Communicate At a time.

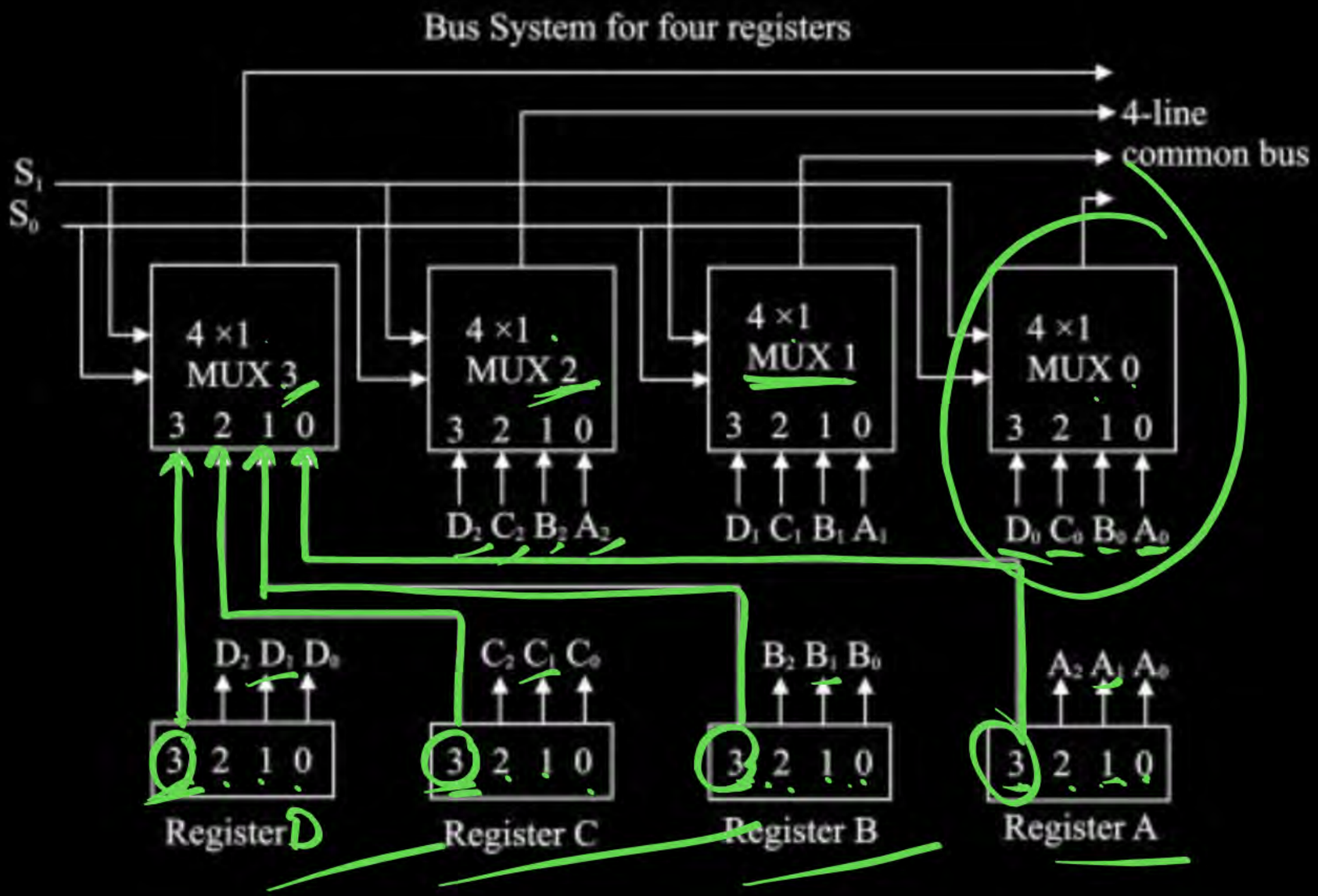
(Which Part[Component] communicate, for that Control Signal is Required.

Working of Register :

$$R_A \longrightarrow R_B$$

4 Register R_A, R_B, R_C & R_D.

Each Register Size = 4bit


Bus System for four registers
4-line
common bus
S1
S0
4 ×1
MUX 3
3 2 1 0
4 ×1
MUX 2
3 2 1 0
4 ×1
MUX 1
3 2 1 0
4 ×1
MUX 0
3 2 1 0
D2 C2 B2 A2
D1 C1 B1 A1
D0 C0 B0 A0
D2 D1 D0
C2 C1 C0
B2 B1 B0
A2 A1 A0
3 2 1 0
3 2 1 0
3 2 1 0
3 2 1 0
Register D
Register C
Register B
Register A

(eg) 4 Register A, B, C & D.
each Register size is 4 bits.

V.V.V.Imp

The Number of Multiplexer = Number of bits in the Register (Size of Register)

Size of Multiplexer = (Number) of Registers.

(eg) 4 Registers & each Register size is 4 bits
then Number of Multiplexer = 4
Size of Multiplexer = 4X1 = 4

Q If we have 32 Register & Each Register size is 8 bit then

Number of Multiplexer(MUX) = 8 (# bits in the Register)

Size of MUX = 32 (32X1) (Number of Register)

Note

If we have m Registers & each Register Size is 'N' bits then

Number of MUX = N (# bits in Register)

Size of MUX = M (mx1) (# Register).

(eg) 4 Register A, B, C & D.
each Register size is 4 bits.

V.V.V.Imp

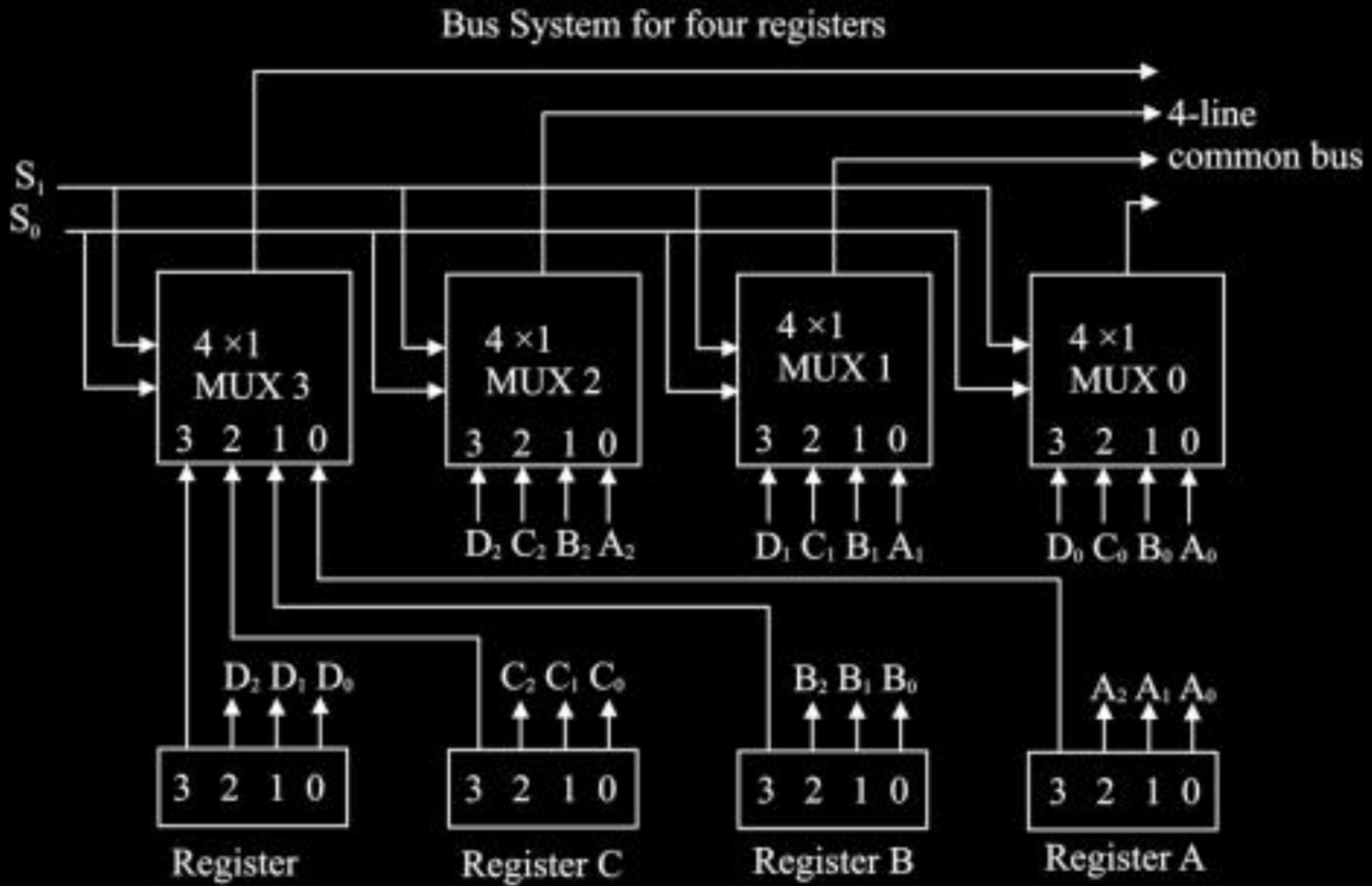
The Number of Multiplexer = Number of bits in the Register. (Size of Register)

Size of multiplexer = Number of Registers.

(eg) 4 Registers & each Register size is 4 bits

then Number of Multiplexer = 4

Size of multiplexer = 4X1 = 4


Bus System for four registers
4-line
common bus
S1
S0
4 ×1
MUX 3
3 2 1 0
4 ×1
MUX 2
3 2 1 0
4 ×1
MUX 1
3 2 1 0
4 ×1
MUX 0
3 2 1 0
D2 C2 B2 A2
D1 C1 B1 A1
D0 C0 B0 A0
D2 D1 D0
C2 C1 C0
B2 B1 B0
A2 A1 A0
3 2 1 0
3 2 1 0
3 2 1 0
3 2 1 0
Register
Register C
Register B
Register A

In & OUT.

Q.1 Multiplexer.

Q.2 Common bus

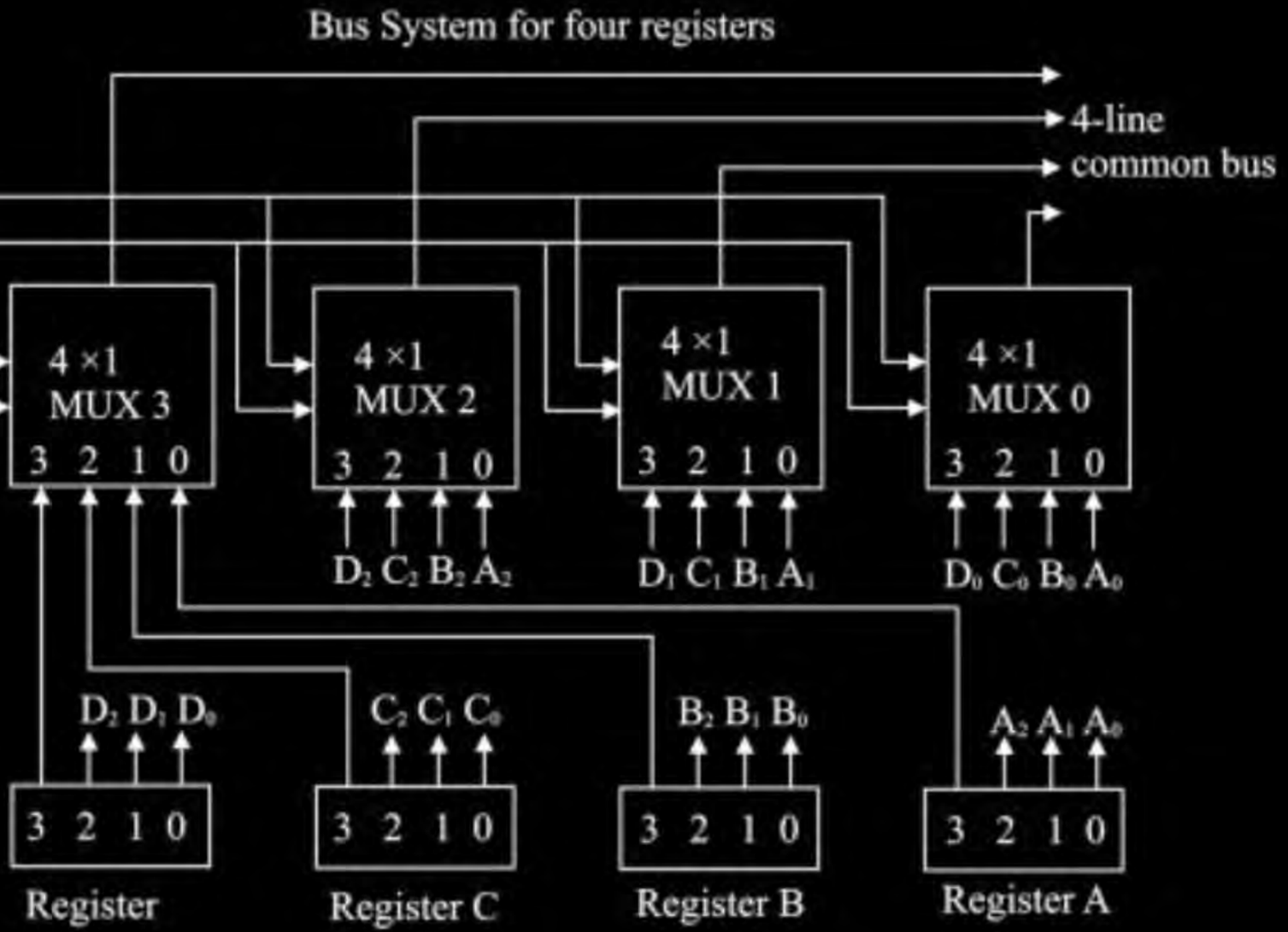
Q.3 Select lines.

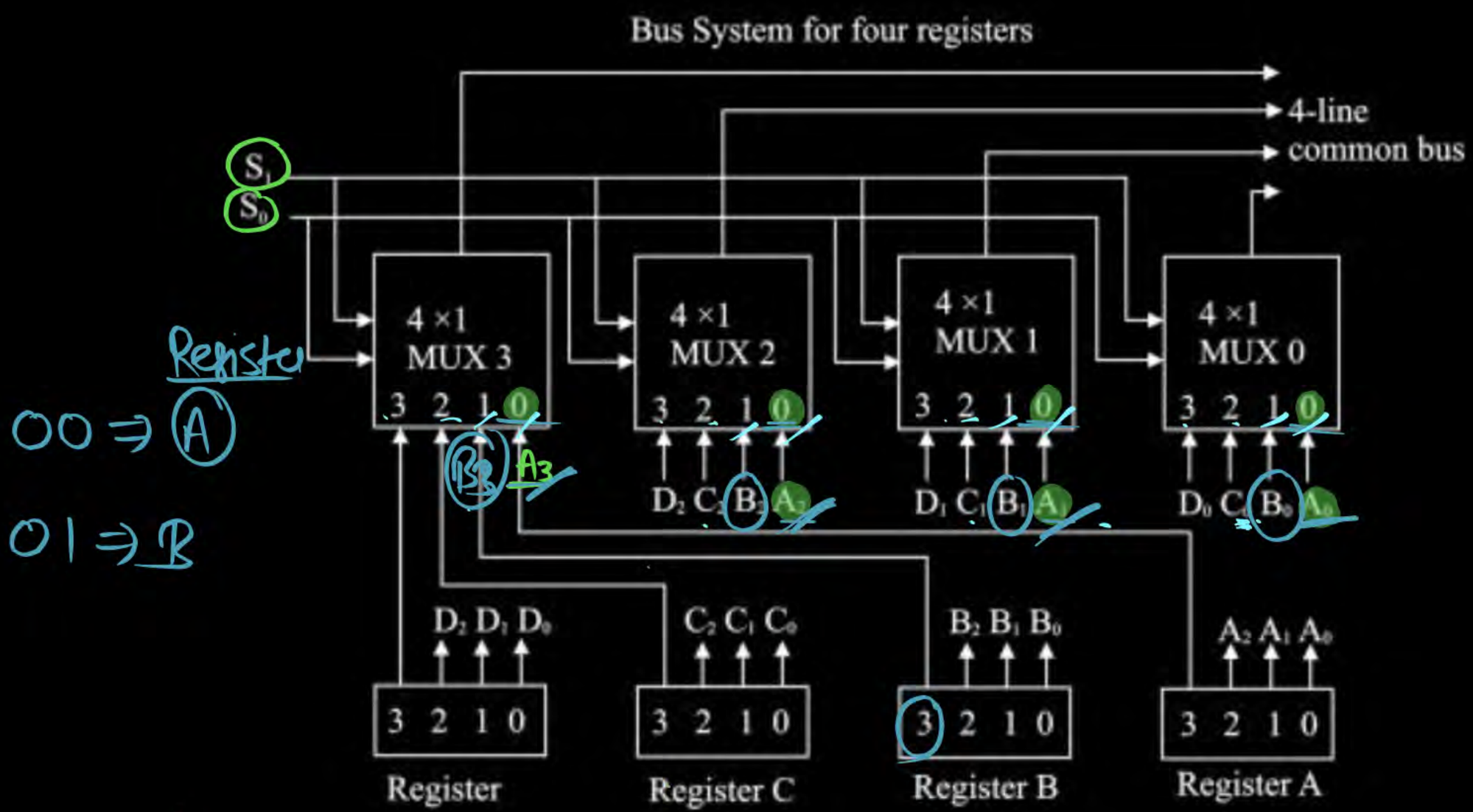
Q.4 How DATA Transferred

4 multiplexer then Select line.

2 Select line

S_1	S_0
0	0
0	1
1	0
1	1




Bus System for four registers
4-line
common bus
S1
S0
4 ×1
MUX 3
4 ×1
MUX 2
4 ×1
MUX 1
4 ×1
MUX 0
3 2 1 0
D2 C2 B2 A2
D1 C1 B1 A1
D0 C0 B0 A0
D2 D1 D0
C2 C1 C0
B2 B1 B0
A2 A1 A0
3 2 1 0
Register
Register C
Register B
Register A

$$R_A \rightarrow R_B.$$

Register A: 00 ⇒ '0'(A)

Function Table for Bus of Fig.

S_1	S_0	Register selected
0	0	A ($A_3A_2A_1A_0$)
0	1	B ($B_3B_2B_1B_0$)
1	0	C ($C_3C_2C_1C_0$)
1	1	D ($D_3D_2D_1D_0$)

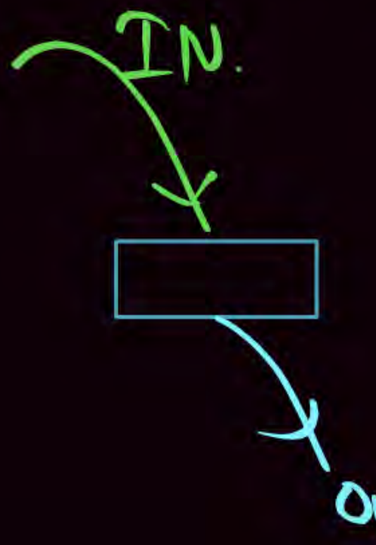
Function Table for Bus of Fig.

S_1	S_0	Register selected
0	0	A
0	1	B
1	0	C
1	1	D

eg

$R_1 \rightarrow R_2$

R_{1out} R_{2in}



OUT.

IN.

How Data is Transferred ?

Register A to Register B

R_{Aout} R_{Bin}.

$R_A \rightarrow R_B$.

R_{Aout} R_{Bin}.

Process : Register A Content given to the MUX (Multiplexer), MUX to Common Bus then Common Bus to Load into Register B. $[R_B]$.

00 means input '0' will be Select across all MUX.

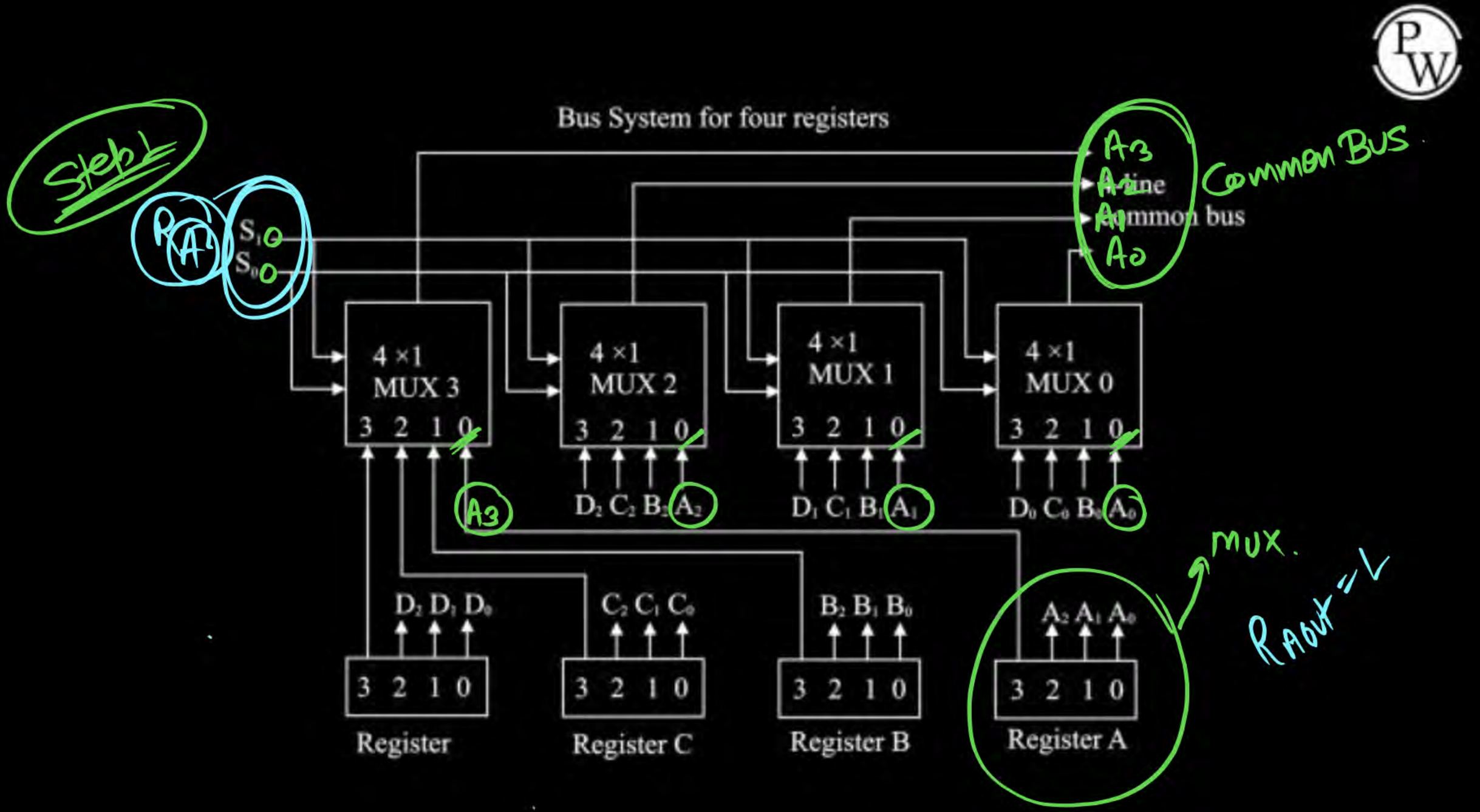
01 means input '1' will be Select across all MUX.

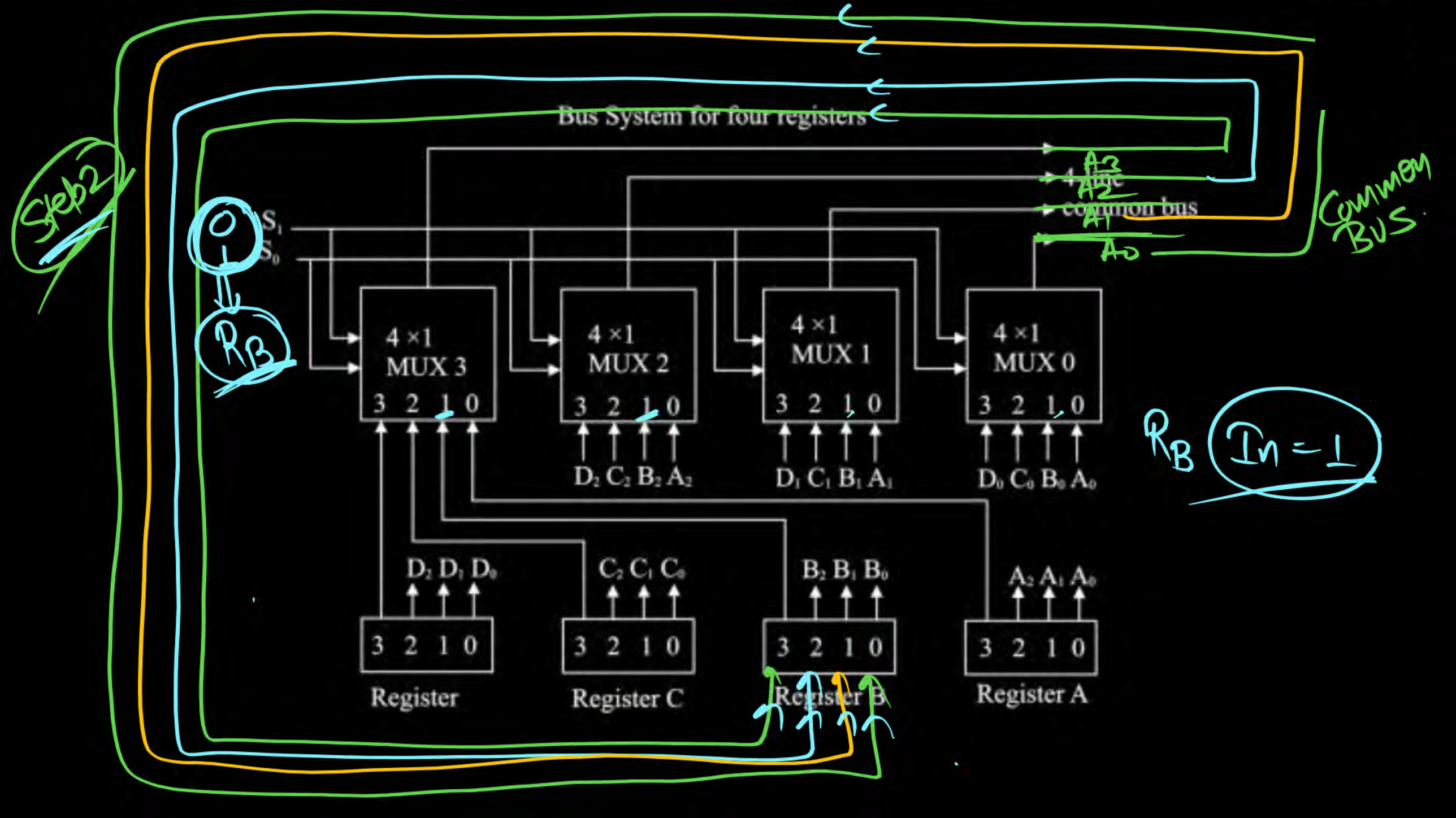
10 means input '2' will be Select across all MUX.

11 means input '3' will be Select across all MUX.

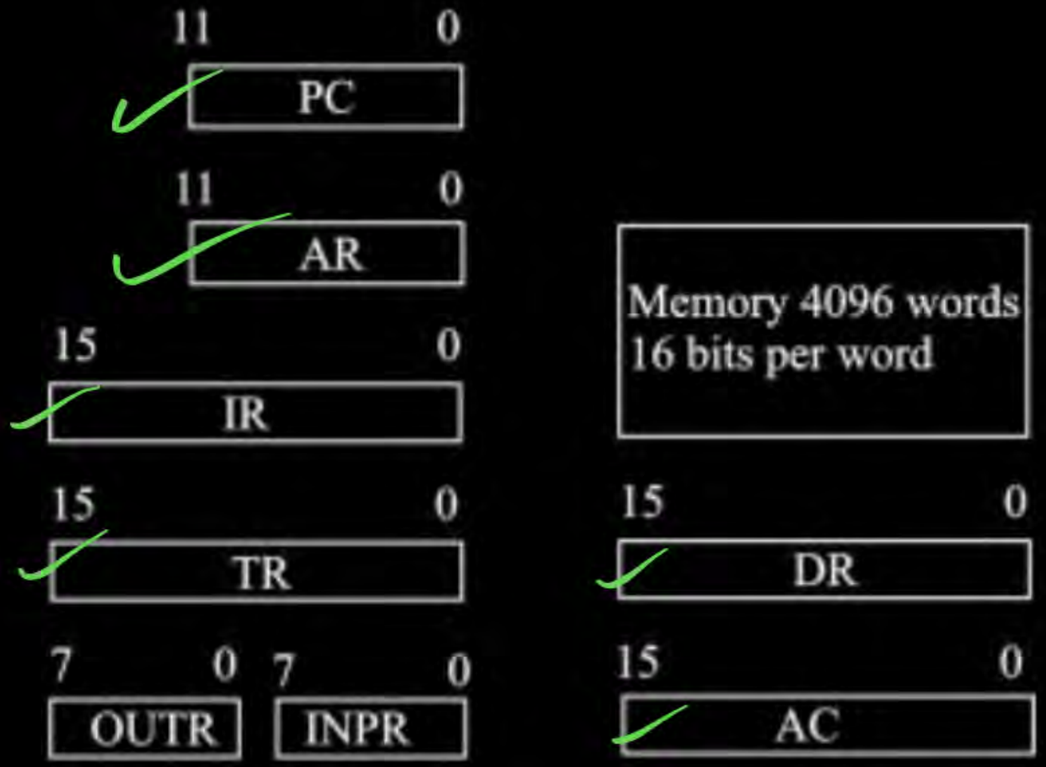
$R_A \rightarrow R_B$.

Select line $\quad 0 \quad 0 \Rightarrow A \; (A_3 A_2 A_1 A_0)$


Bus System for four registers
Step 1
RA1
S1 0
S0 0
A3
A2
A1
A0
Common Bus
Common bus
4 ×1
MUX 3
4 ×1
MUX 2
4 ×1
MUX 1
4 ×1
MUX 0
3 2 1 0
A3
D2 C2 B2 A2
D1 C1 B1 A1
D0 C0 B0 A0
D2 D1 D0
C2 C1 C0
B2 B1 B0
A2 A1 A0
Register
Register C
Register B
Register A
mux.
RAout = L


Bus System for four registers
S_1
S_0
R_B
4 ×1 MUX 3
4 ×1 MUX 2
4 ×1 MUX 1
4 ×1 MUX 0
3 2 1 0
$D_2\ C_2\ B_2\ A_2$
$D_1\ C_1\ B_1\ A_1$
$D_0\ C_0\ B_0\ A_0$
common bus
A_3
A_2
A_1
A_0
$D_2\ D_1\ D_0$
$C_2\ C_1\ C_0$
$B_2\ B_1\ B_0$
$A_2\ A_1\ A_0$
Register
Register C
Register B
Register A
Step2
Common Bus
R_B In = 1

Design a Small Computer



Basic computer registers and memory

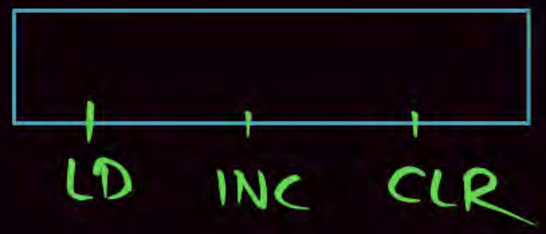
4096 X 16

2^{12} X 16

Address = 12 bit

Data = 16 bit

Register

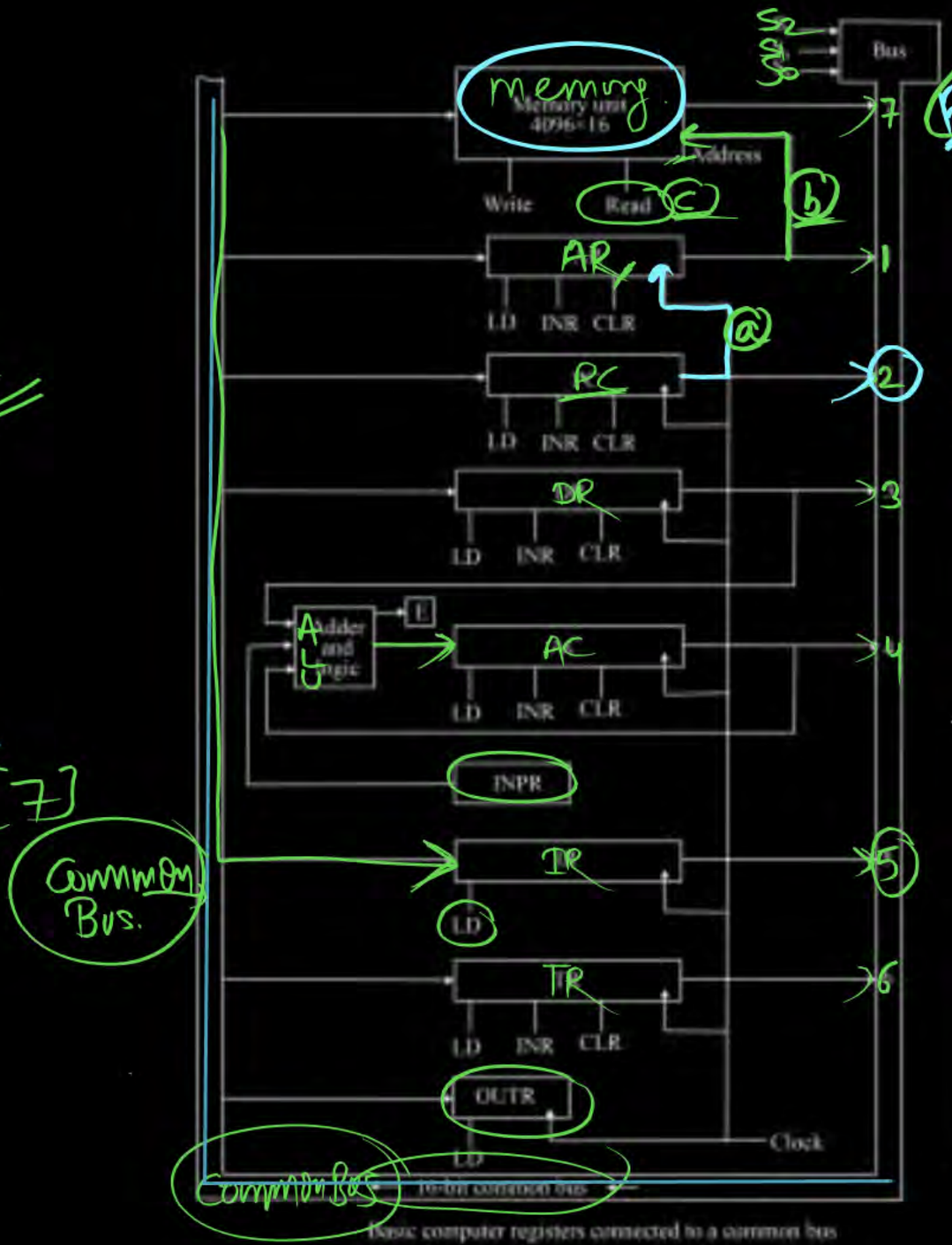


Working of Common bus, & other Components.

7 Component

3 Select line

S_2	S_1	S_0	
			memory ← [7]
1	1	1	



Common Bus.

Fetch cycle

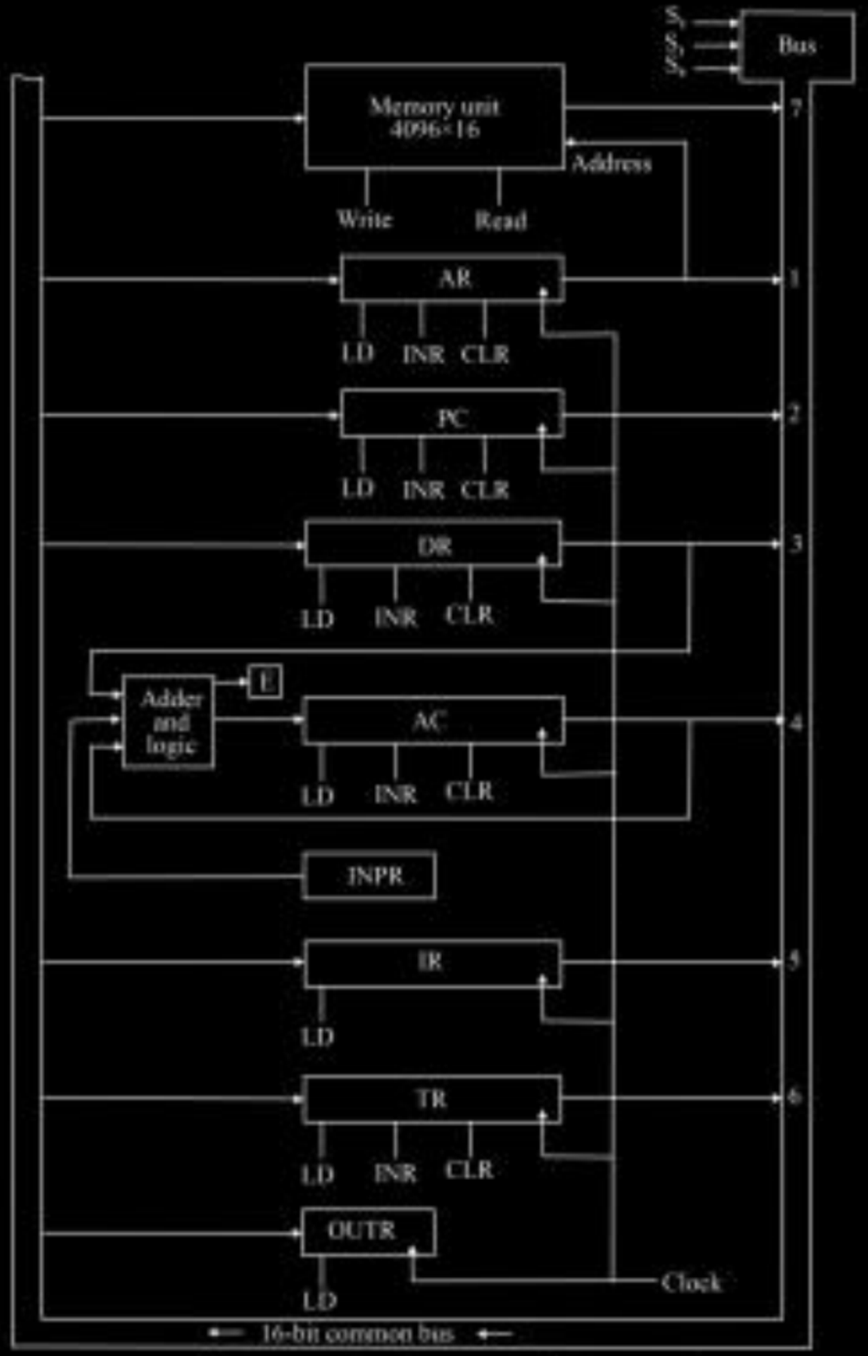
Read Control Signal

Load From memory

S_2	S_1	S_0	
1	1	1	⇒ 7 (memory enabled)

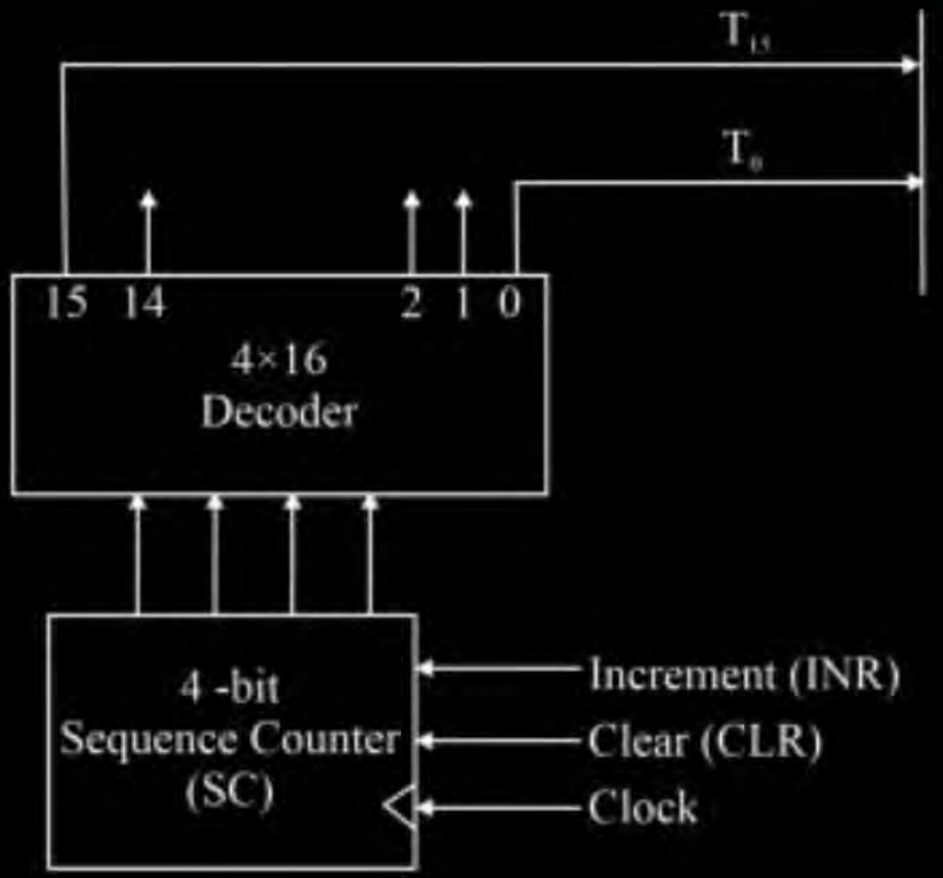
In a Memory we have 'n' Location. (multiple Location) which memory Address Content is load into BUS.

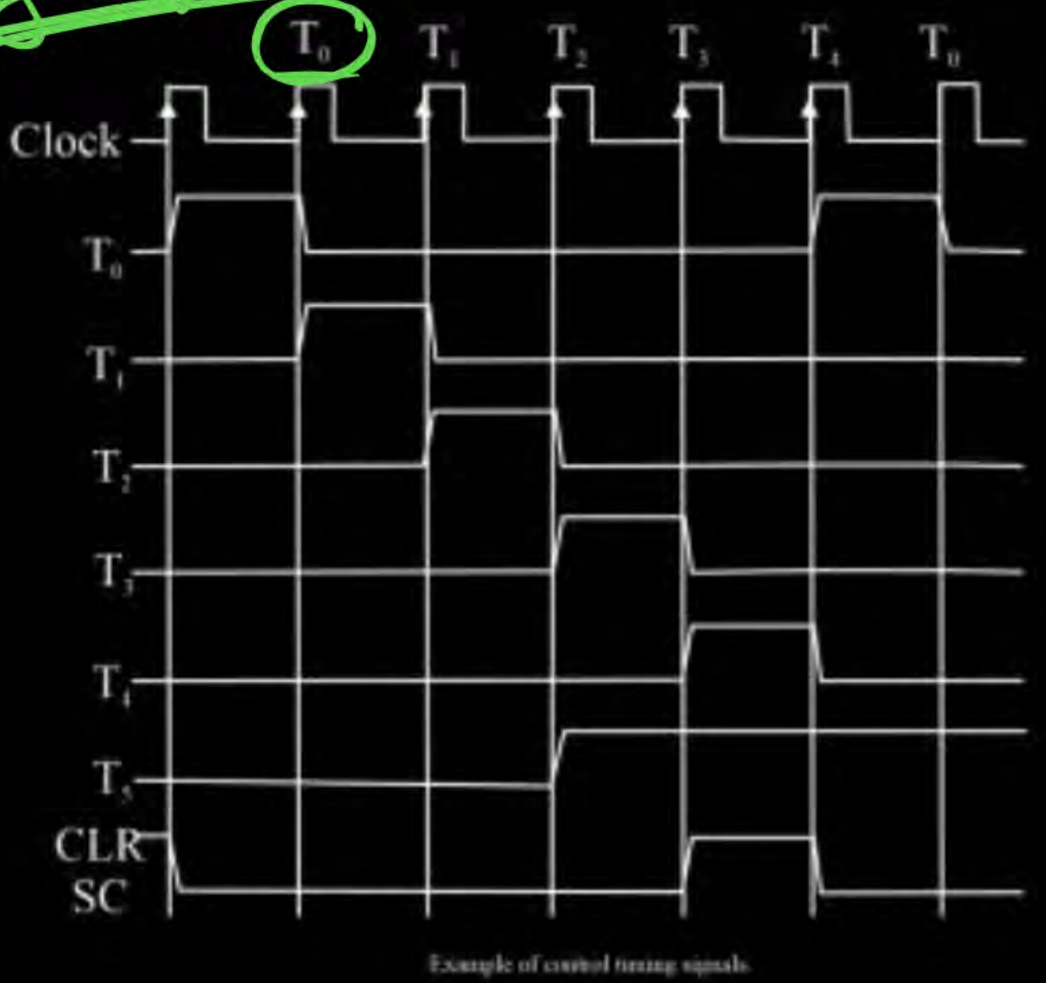
S_2	S_1	S_0	
1	1	1	⇒ 7 (Memory)
0	1	0	= 2 (PC.)
1	0	1	= 5 (IR)



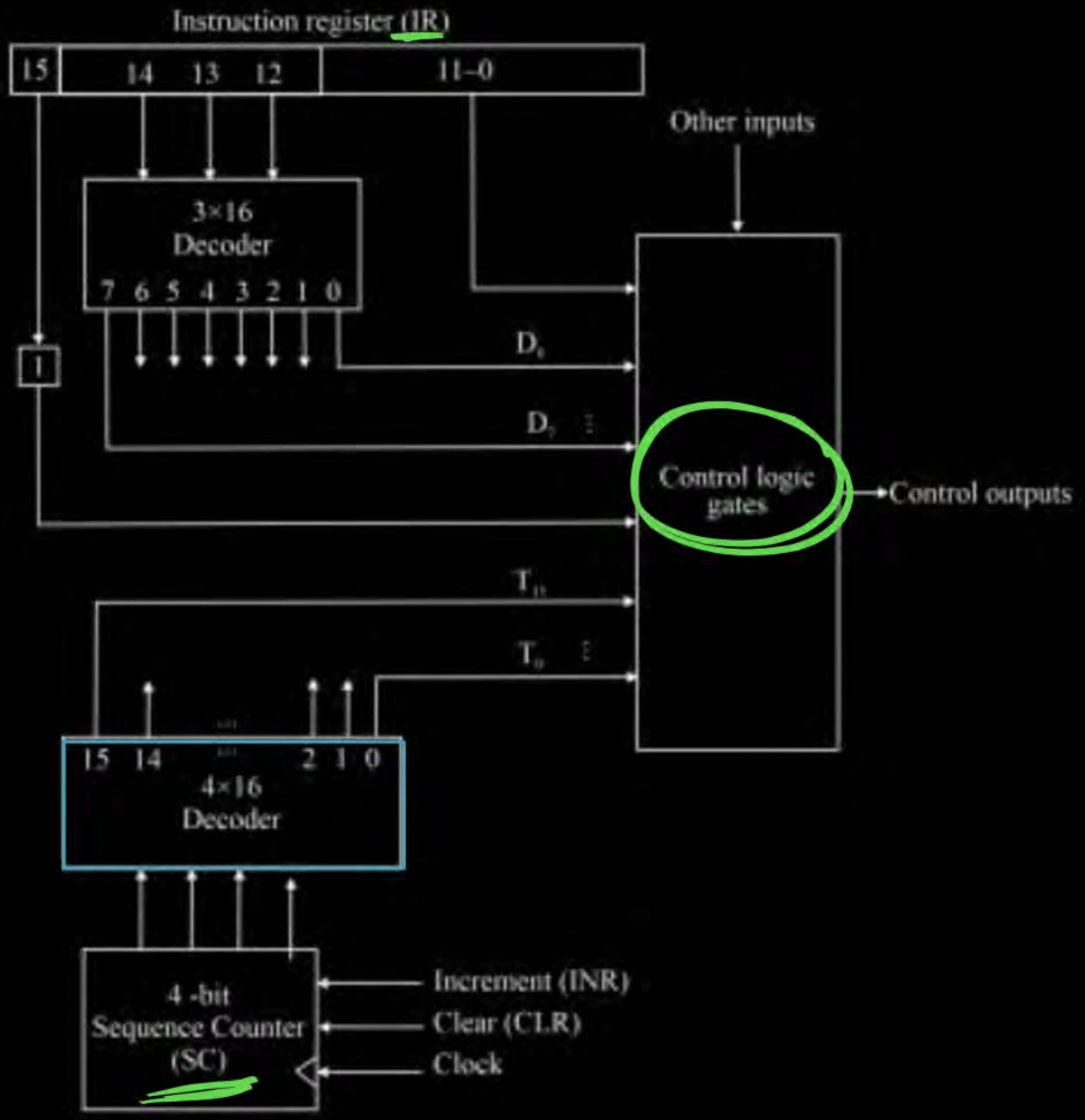
Basic computer registers connected to a common bus

Timing Signal

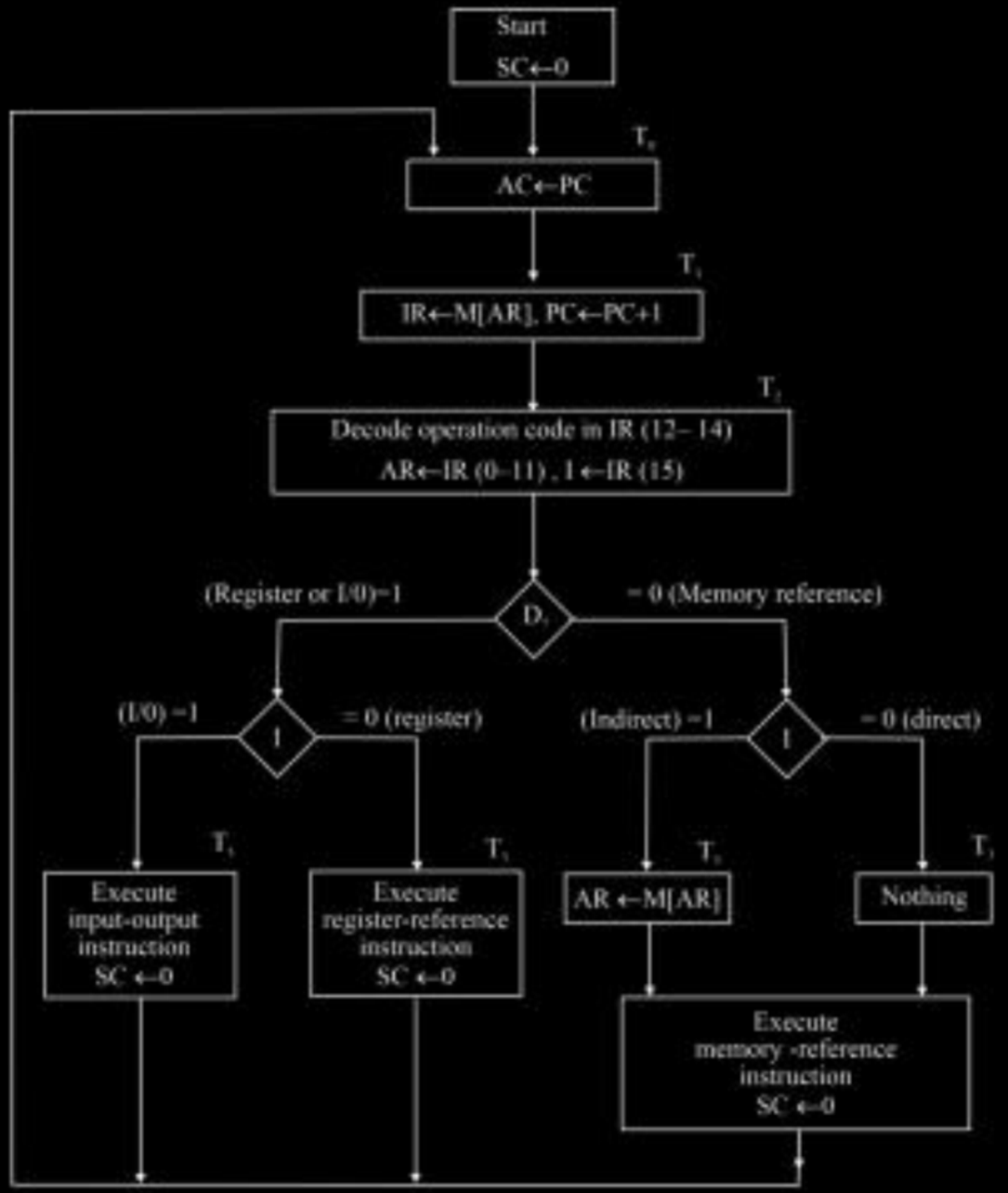




Example of control timing signals



Control unit of basic computer



Flow chart for instruction cycle (initial configuration)

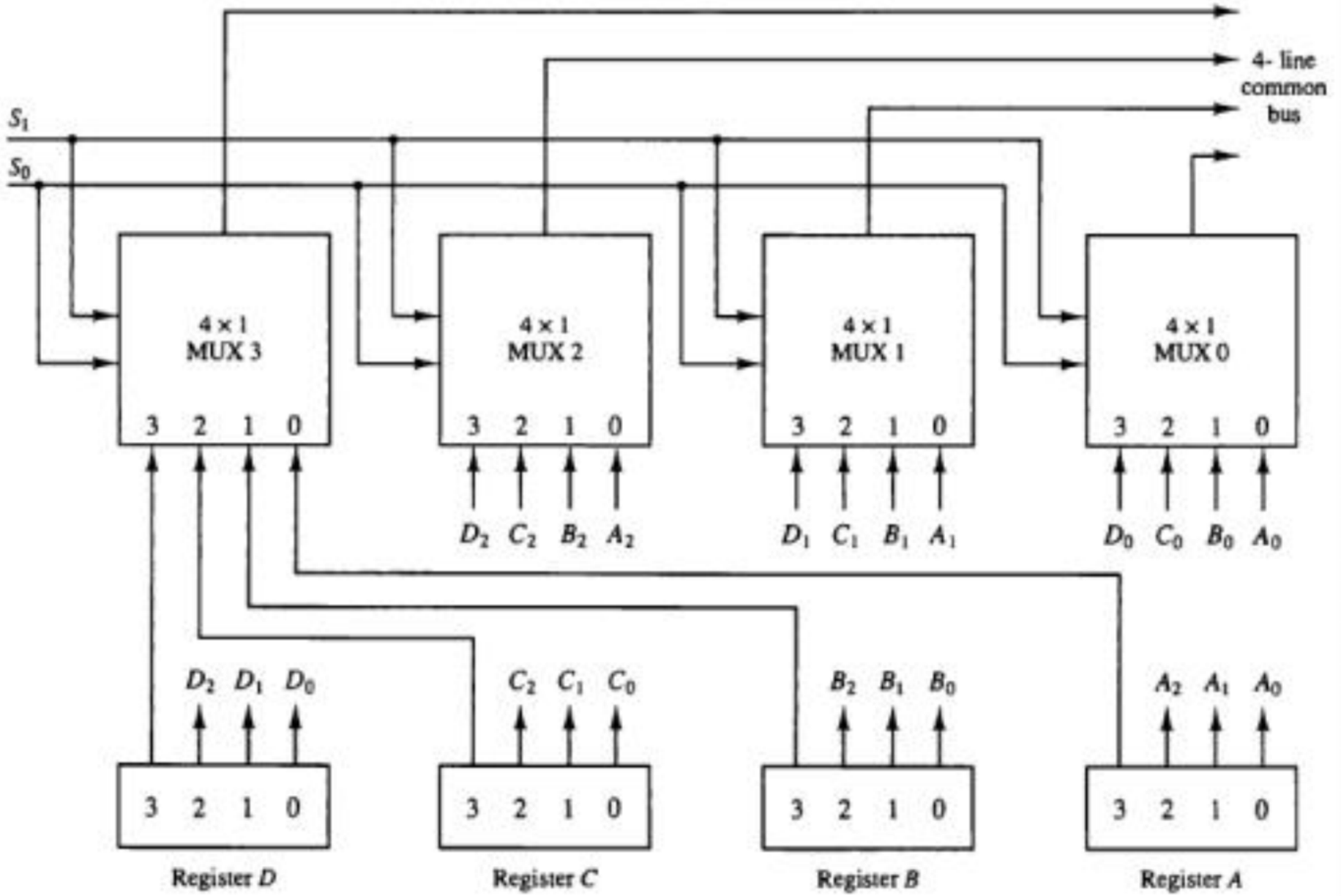
How Data is Transferred ?

Register A to Register B

TABLE 4-2 Function Table for Bus of Fig. 4-3

S_1	S_0	Register selected
0	0	*A*
0	1	*B*
1	0	*C*
1	1	*D*

Figure 4-3 Bus system for four registers.



Computer Design

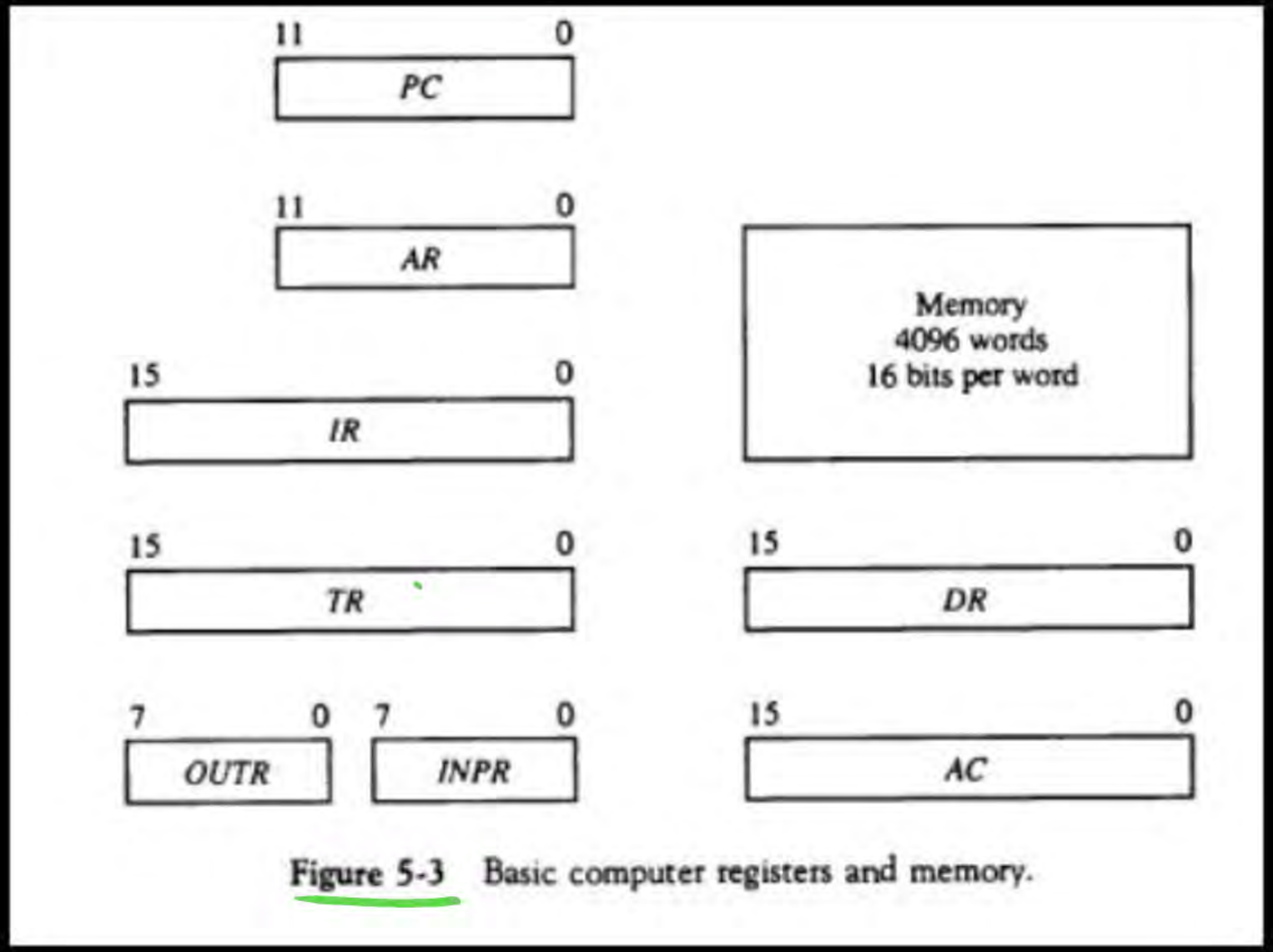


Figure 5-3 Basic computer registers and memory.

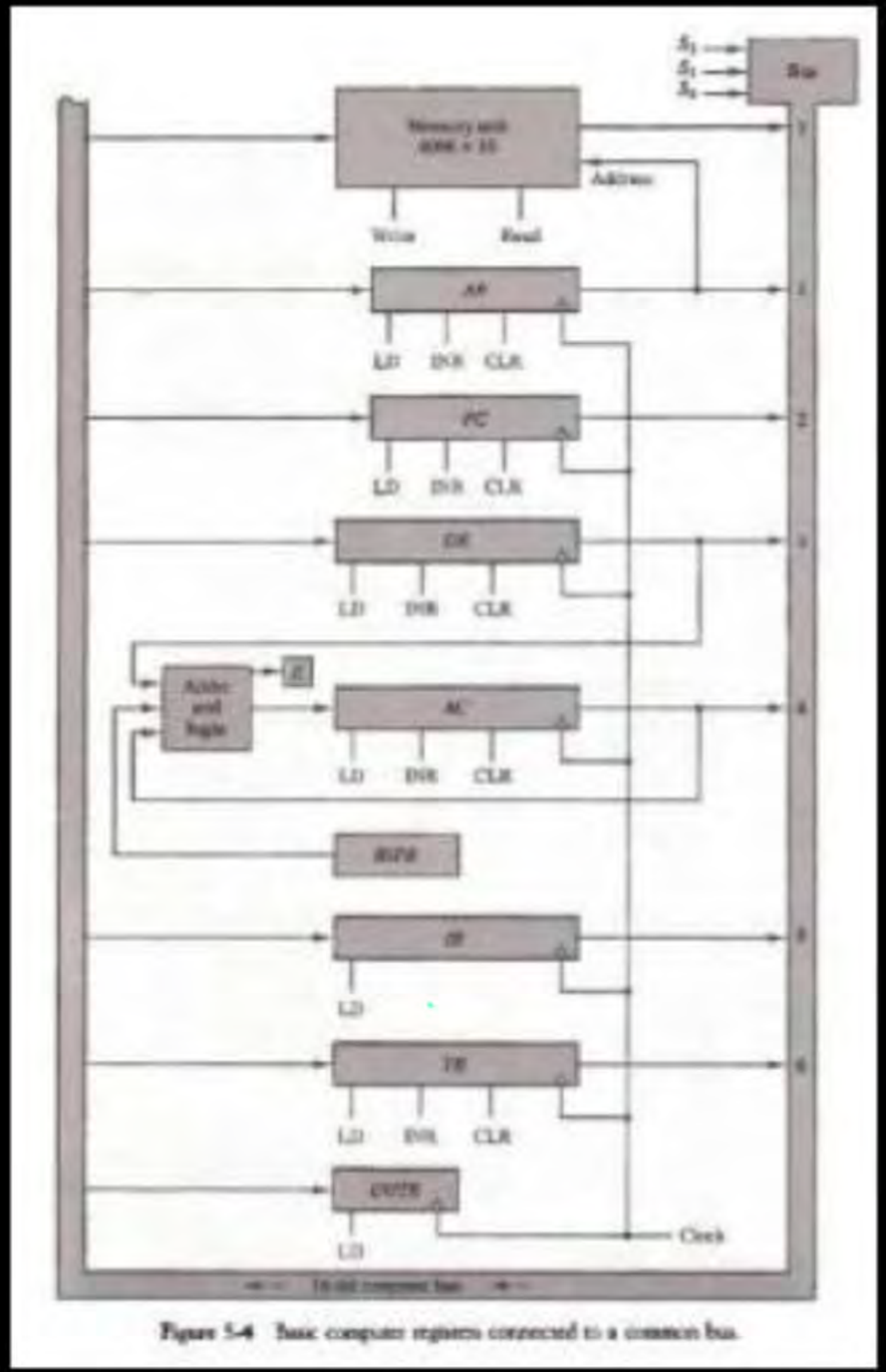


Figure 5-4 Basic computer registers connected to a common bus.

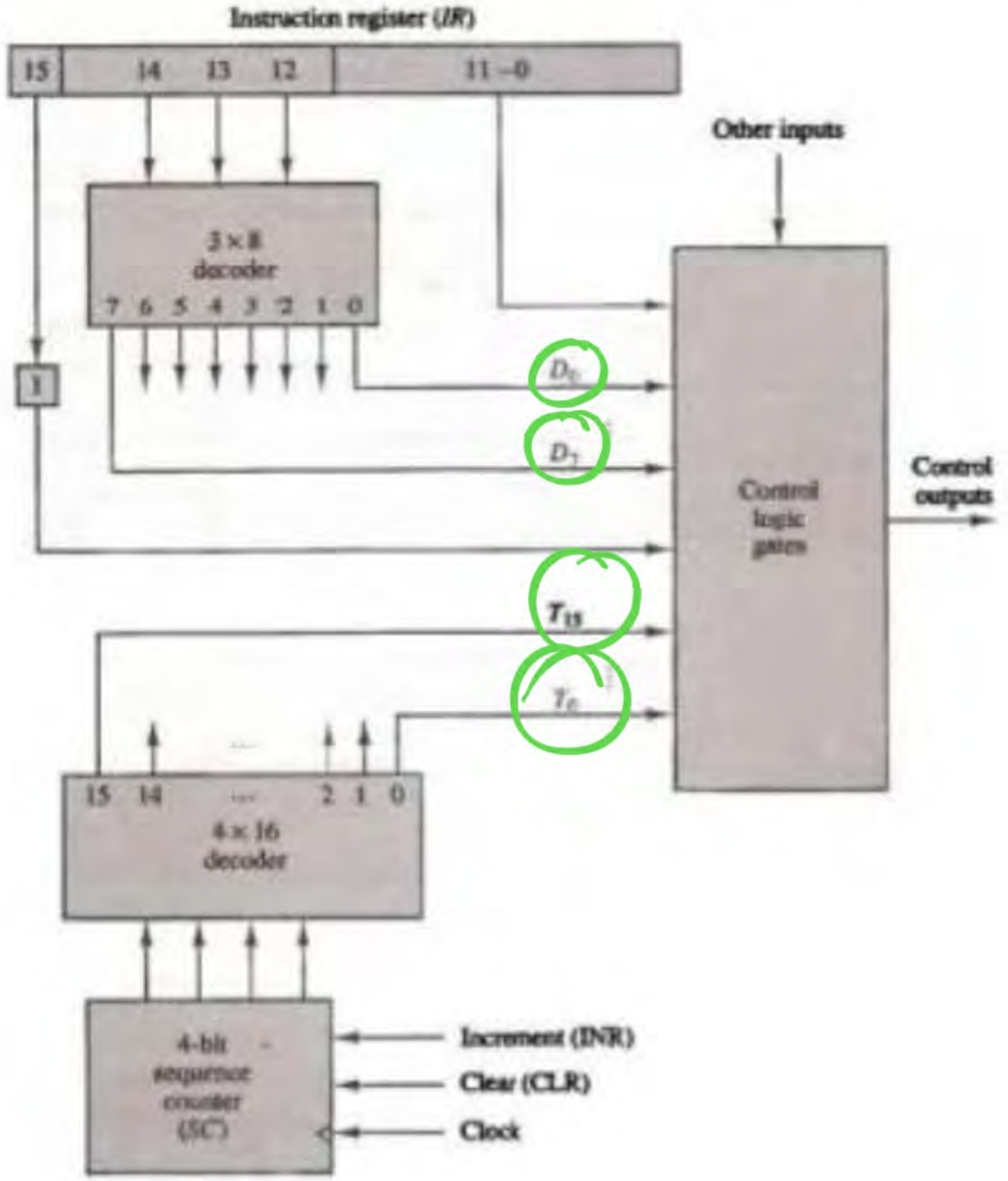


Figure 5-6 Control unit of basic computer.

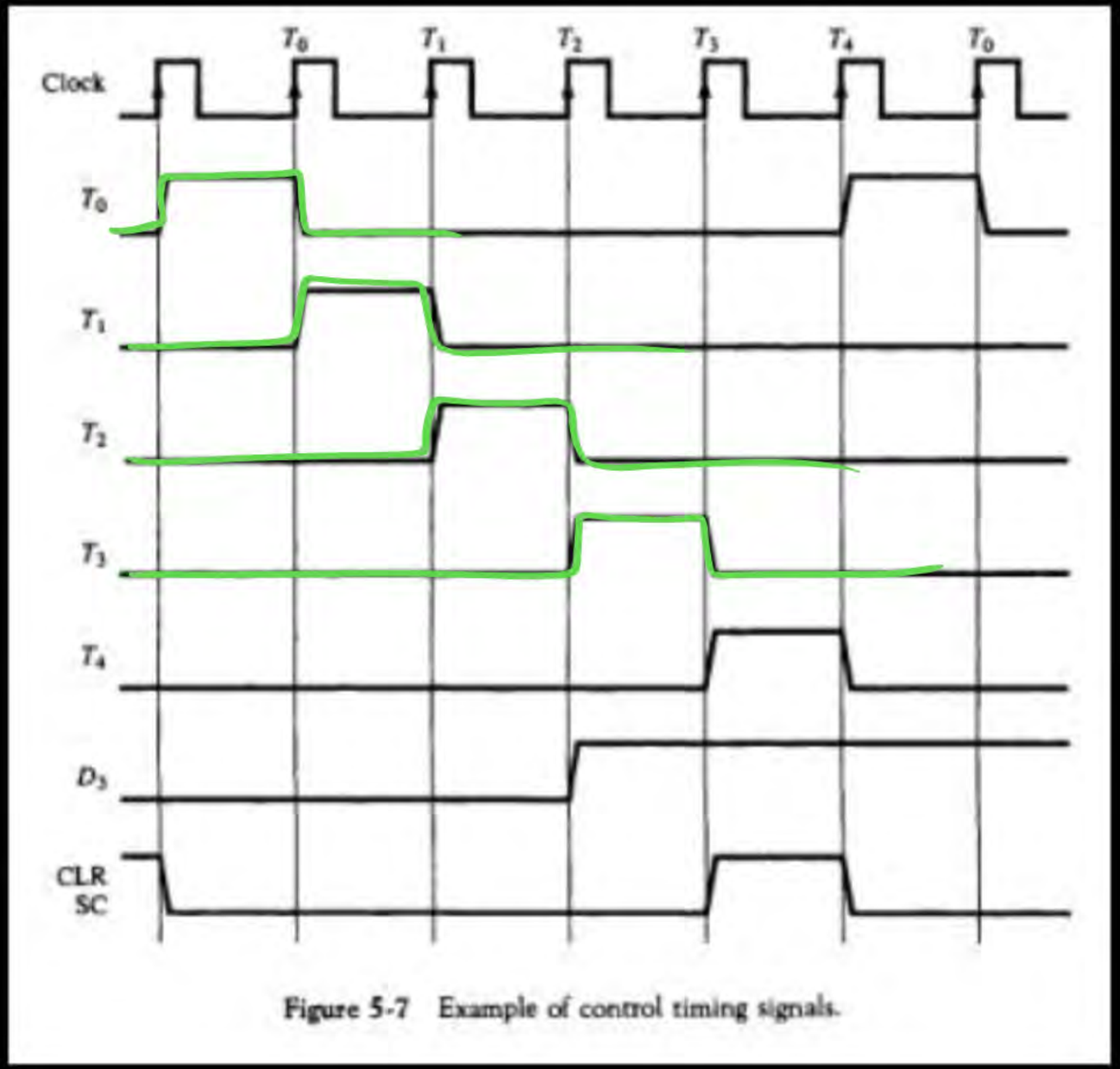


Figure 5-7 Example of control timing signals.

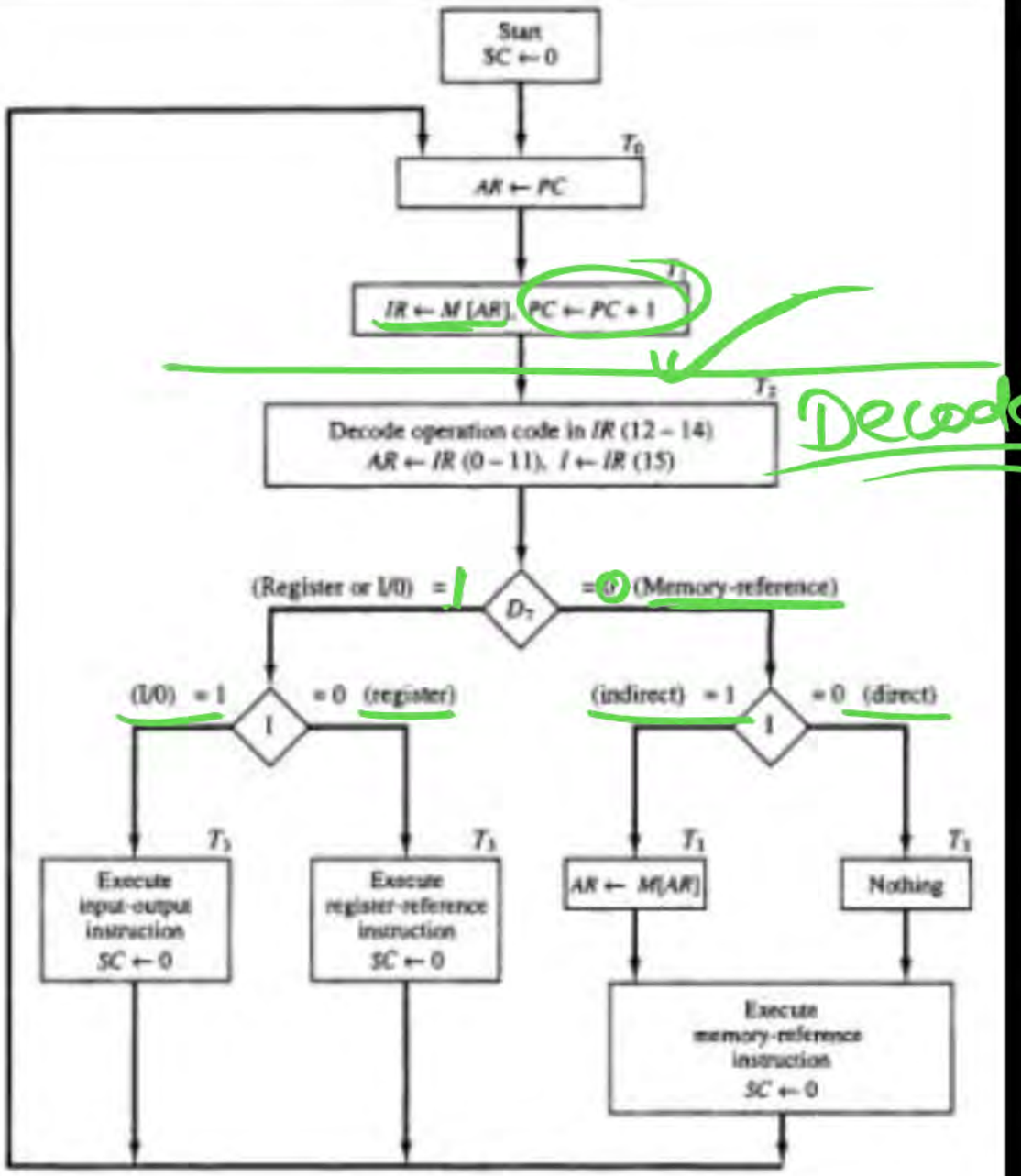


Figure 5-9 Flowchart for instruction cycle (initial configuration).

Register
ALU
Common Bus
Timing Signal & Control Signal.
Register Working

ALU DATA PATH

THANK YOU!