

# DESIGN OF

## MINI-8085

Processor Design (EE 739)

Submitted by-

AYUSH RANJAN

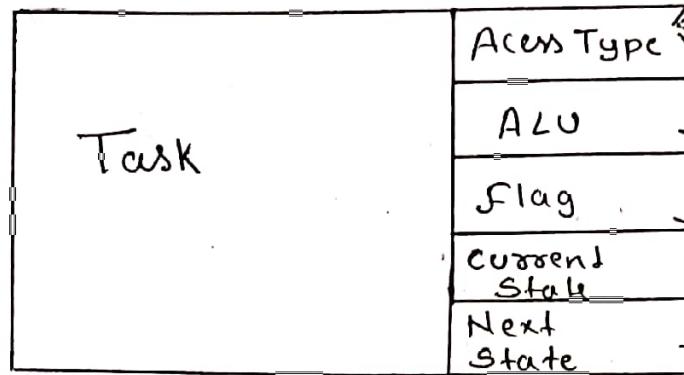
MTech (Solid State Device),

193070080

Indian Institute Of Technology, Bombay.

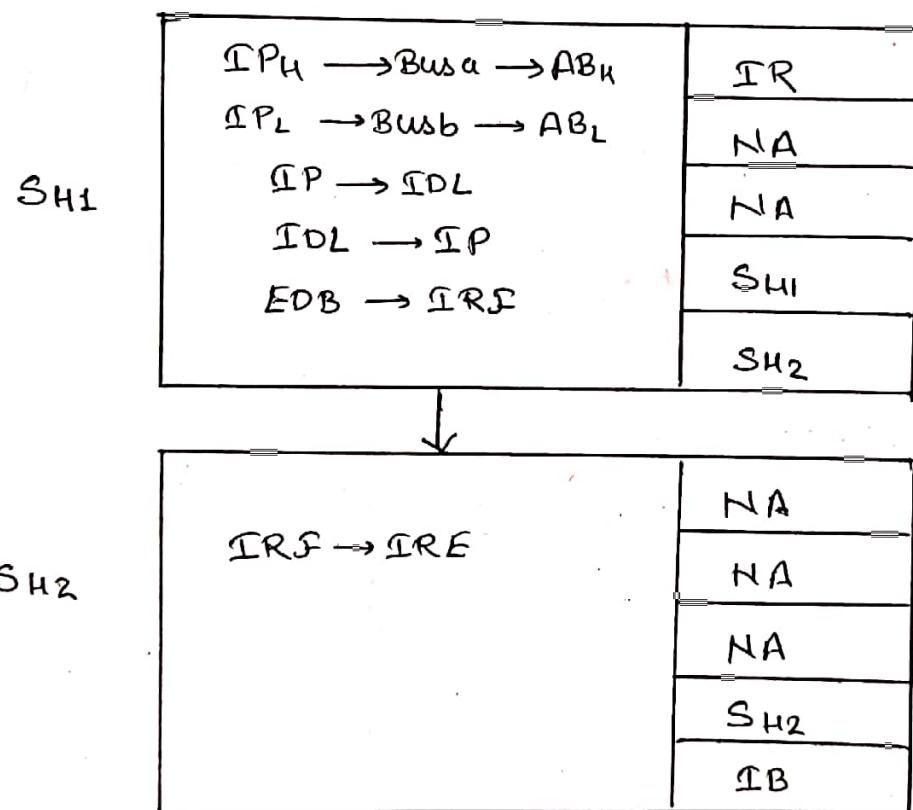
## Level-2 Hardware Flow Chart

Format for Level-2 flow chart



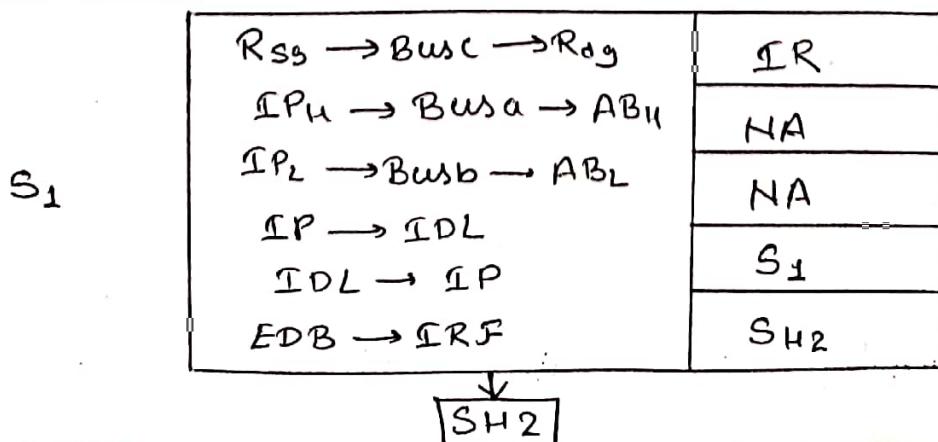
DR - Data Read  
 DW - Data Write  
 IR - Instruction Read  
 NA - No Access  
 → ALU operation (ADC, SBB, AND)  
 Flag affected NA  
 BC - Branch Control  
 IB - Inst Branch  
 StakeID → Direct Branch

House Keeping Task → Update IP and Instruction Fetch



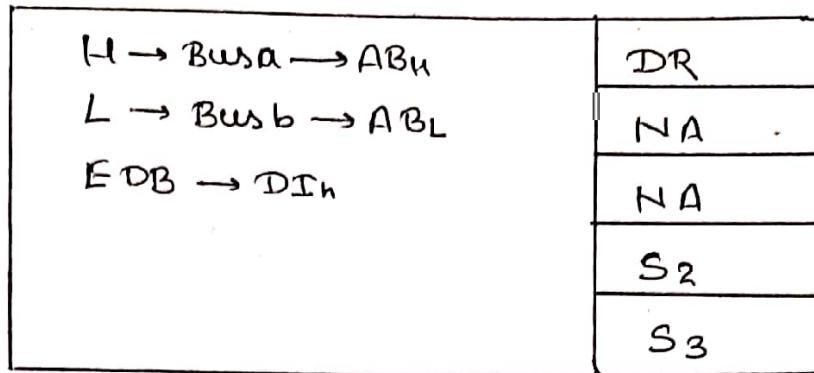
House keeping Task will be merged with Execution Task whenever it is possible.

MOV Rg, Rg

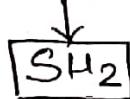
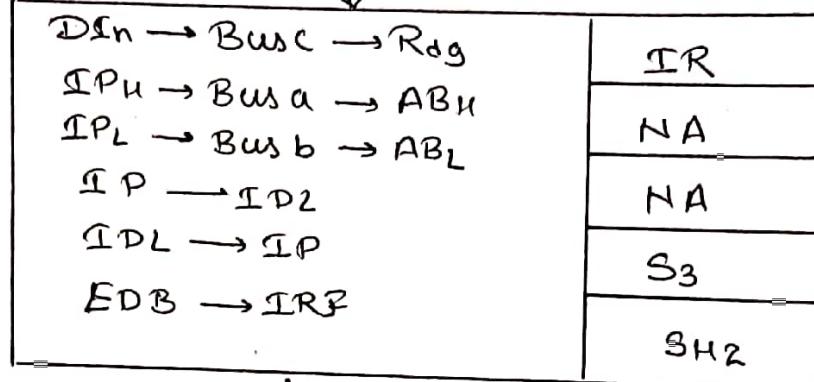


2) MOV R<sub>9</sub>, M

S<sub>2</sub>



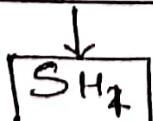
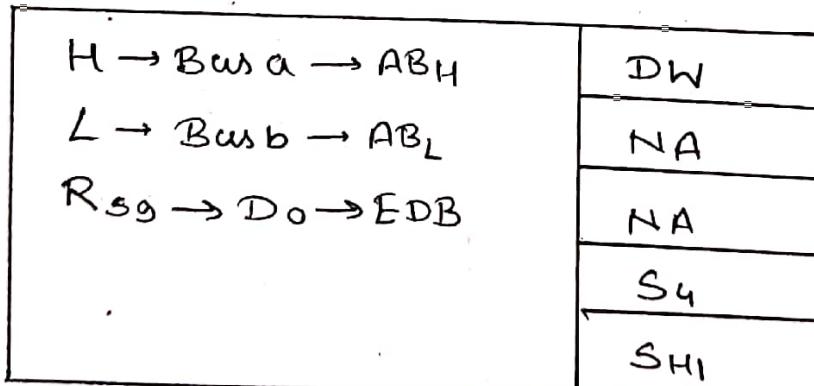
S<sub>3</sub>



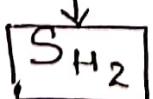
SH<sub>2</sub>

3) MOV M, R<sub>9</sub>

S<sub>4</sub>



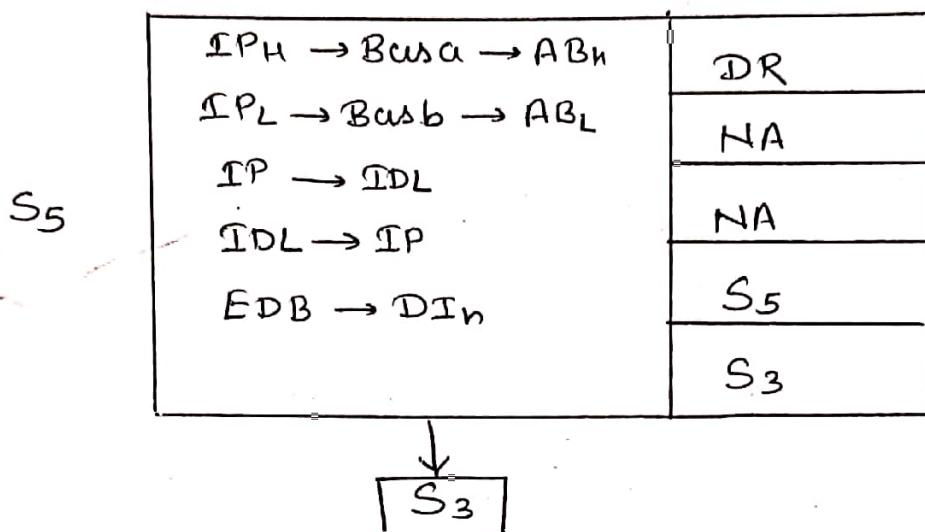
SH<sub>1</sub>



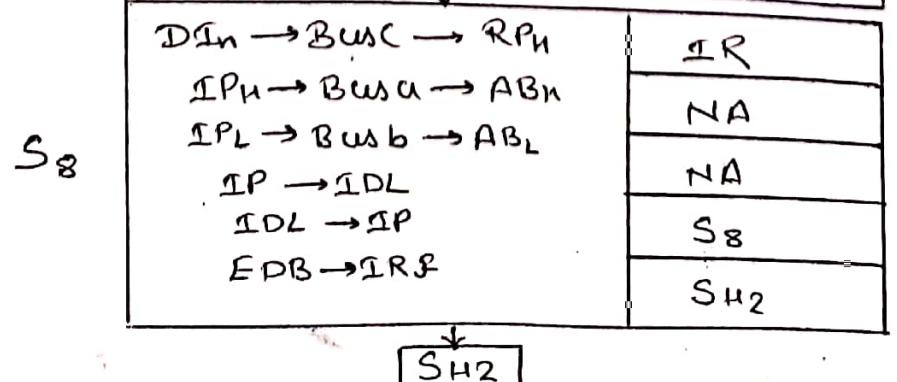
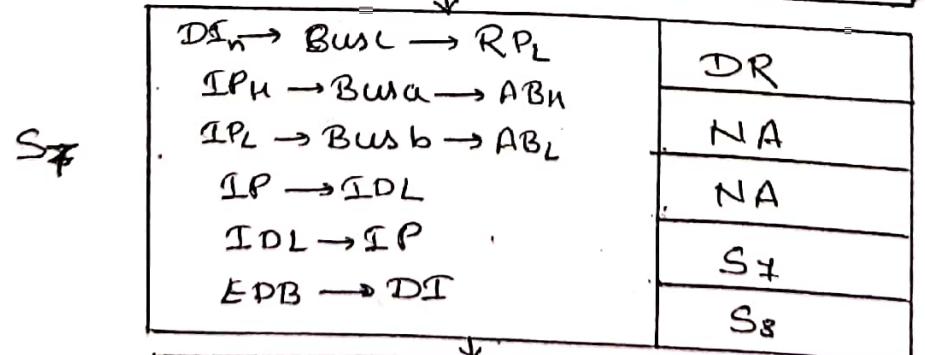
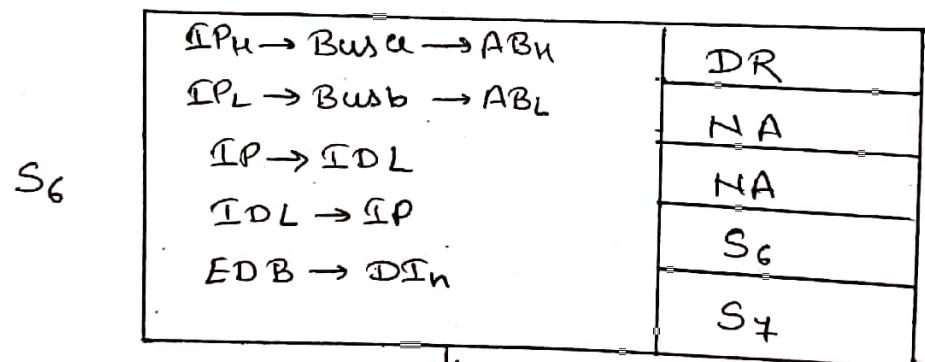
SH<sub>2</sub>

4) MVI R<sub>9</sub>, D<sub>08</sub> → 8 bit immediate Data

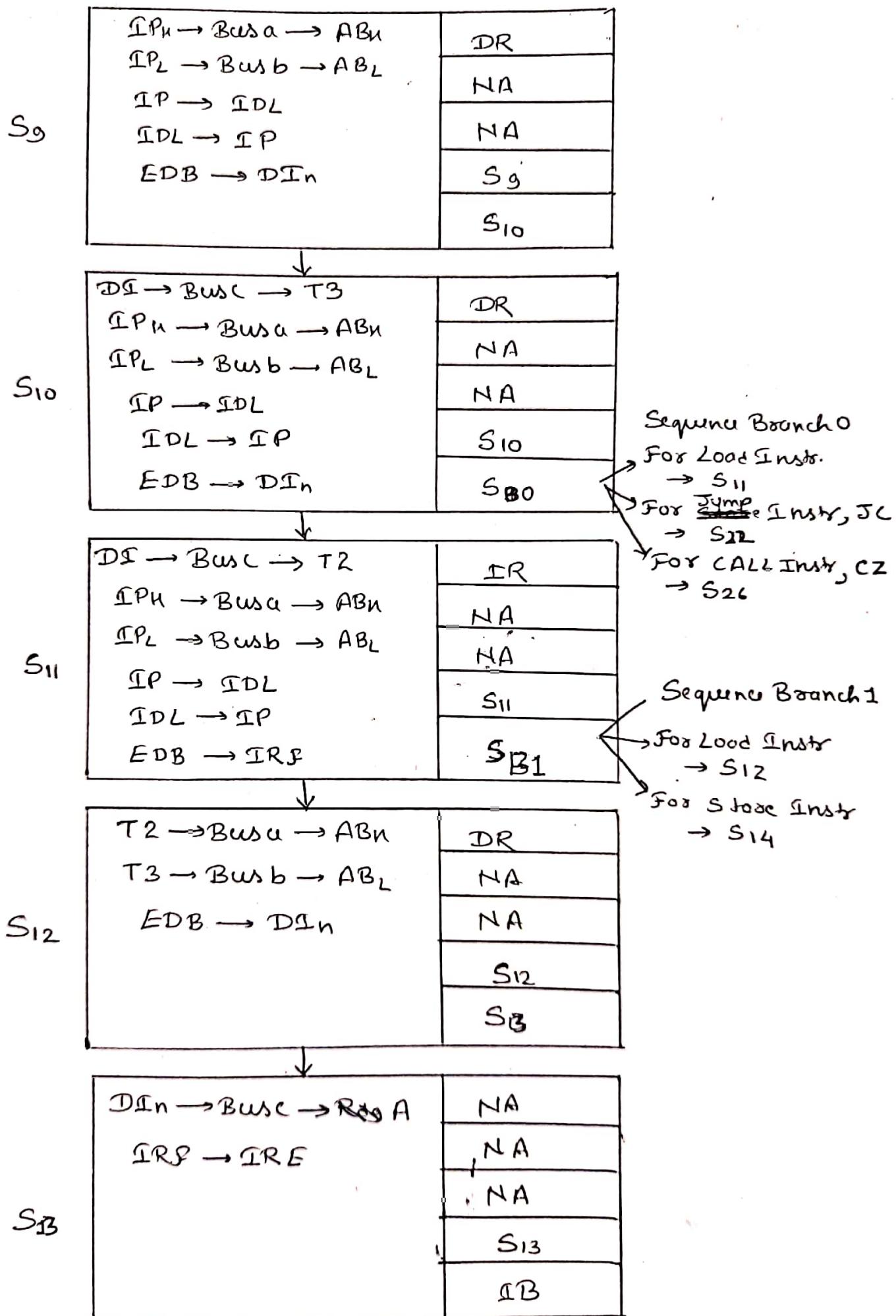
Assuming that first instruction is brought up by during booting process  
So the PC is pointing to next memory location where D<sub>08</sub> is stored.



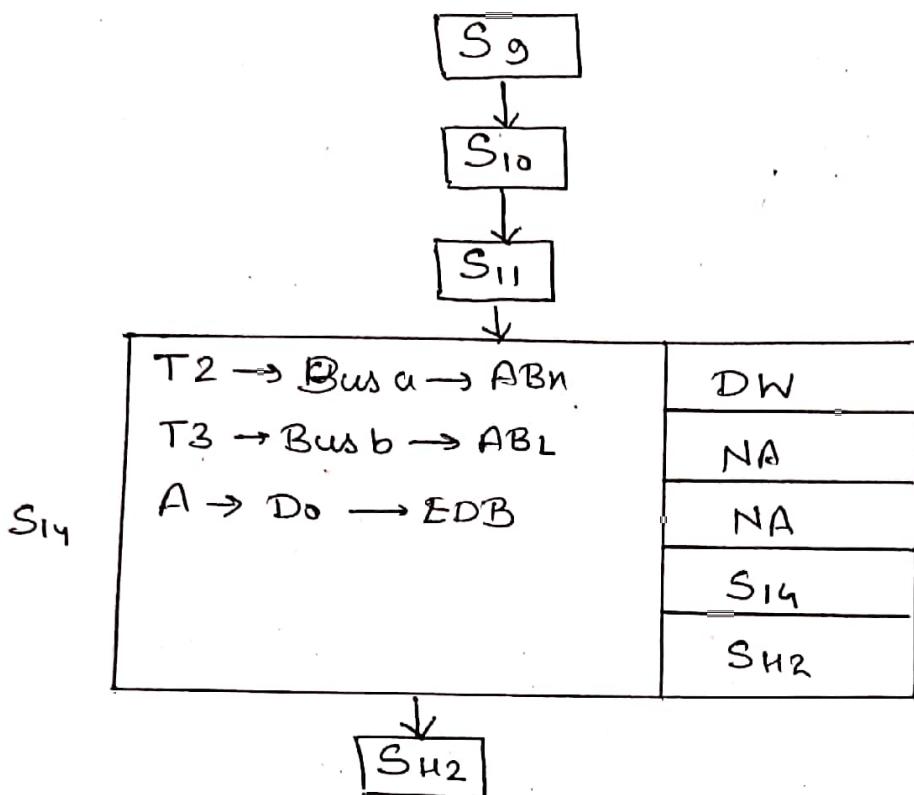
5) LXI Rp/Sp, DI<sub>16</sub>



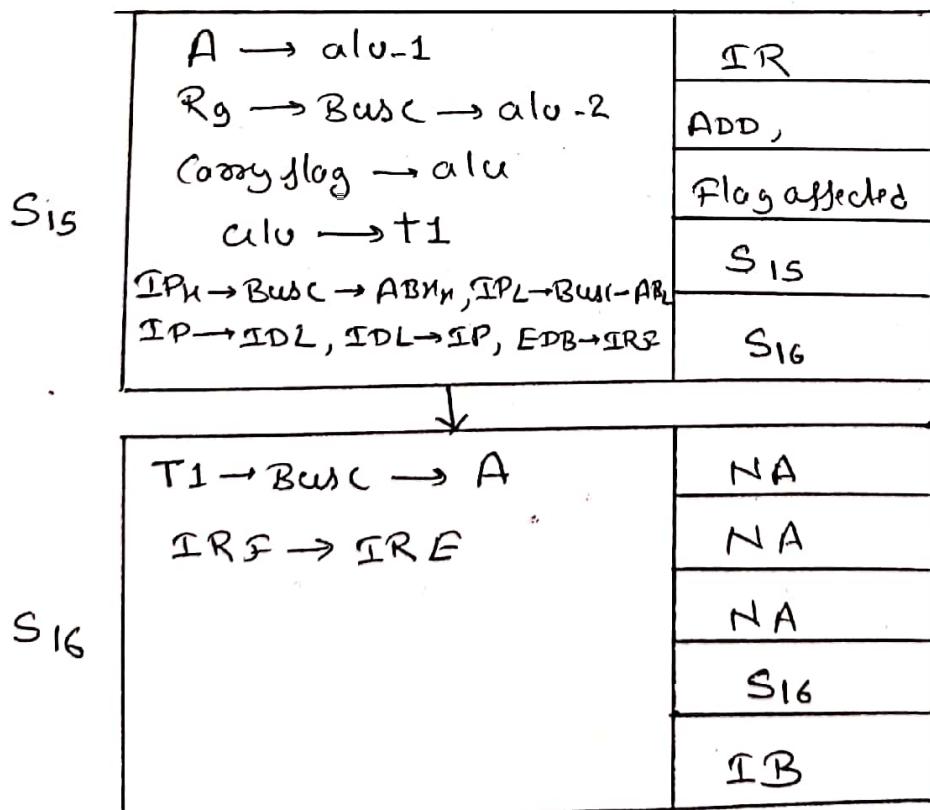
# 6) LDA D16



## 7) STA D16



## 8) ADC R9



## 9) ACI DO8

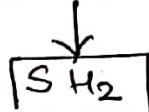
$S_{14}$

$IP_H \rightarrow Bus_A \rightarrow AB_H$	DR
$IP_L \rightarrow Bus_B \rightarrow AB_L$	NA
$IP \rightarrow IDL$	NA
$IDL \rightarrow IP$	
$EDB \rightarrow DI$	$S_{14}$
	$S_{18}$



$S_{18}$

$A \rightarrow alu-1$	IR
$DI \rightarrow Bus_C \rightarrow alu-2$	ADD
$Carry, flag \rightarrow alu$	Flag affected
$alu \rightarrow +1$	
$IP_H \rightarrow Bus_A \rightarrow AB_H$	$S_{18}$
$IP_L \rightarrow Bus_B \rightarrow AB_L$	
$IP \rightarrow IDL_B, IDL \rightarrow IP$	
$EDB \rightarrow IRR$	$S_{H2}$



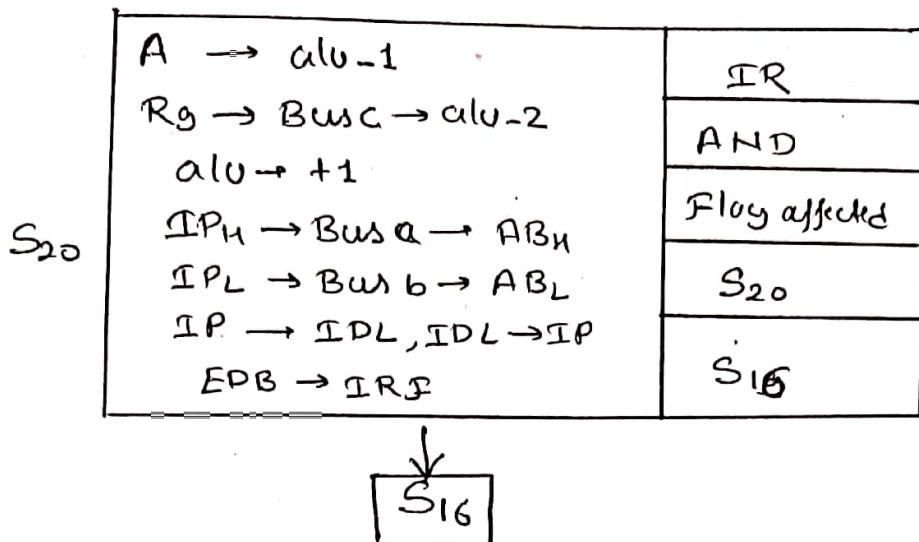
## 10) SBB R9

$S_{19}$

$A \rightarrow alu-1$	IR
<del><math>R_9 \rightarrow Bus_C \rightarrow alu-2</math></del>	SBB
$Carry, flag \rightarrow alu$	Flag affected
$alu \rightarrow +1$	
$IP_H \rightarrow Bus_A \rightarrow AB_H$	$S_{19}$
$IP_L \rightarrow Bus_B \rightarrow AB_L$	
$IP \rightarrow IDL, IDL \rightarrow IP$	
$EDB \rightarrow IRR$	$S_{16}$

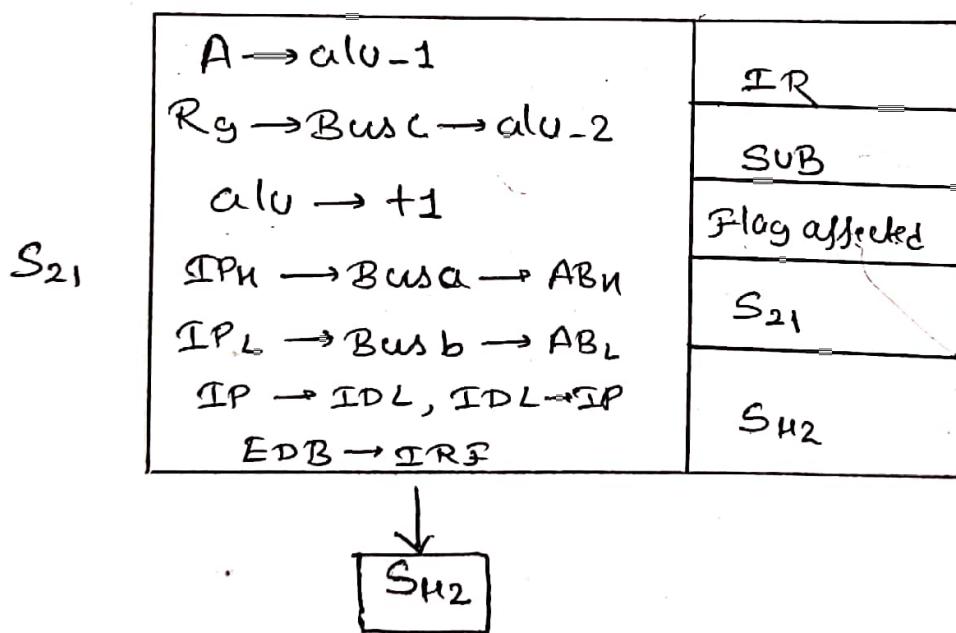


11) ANA Rg

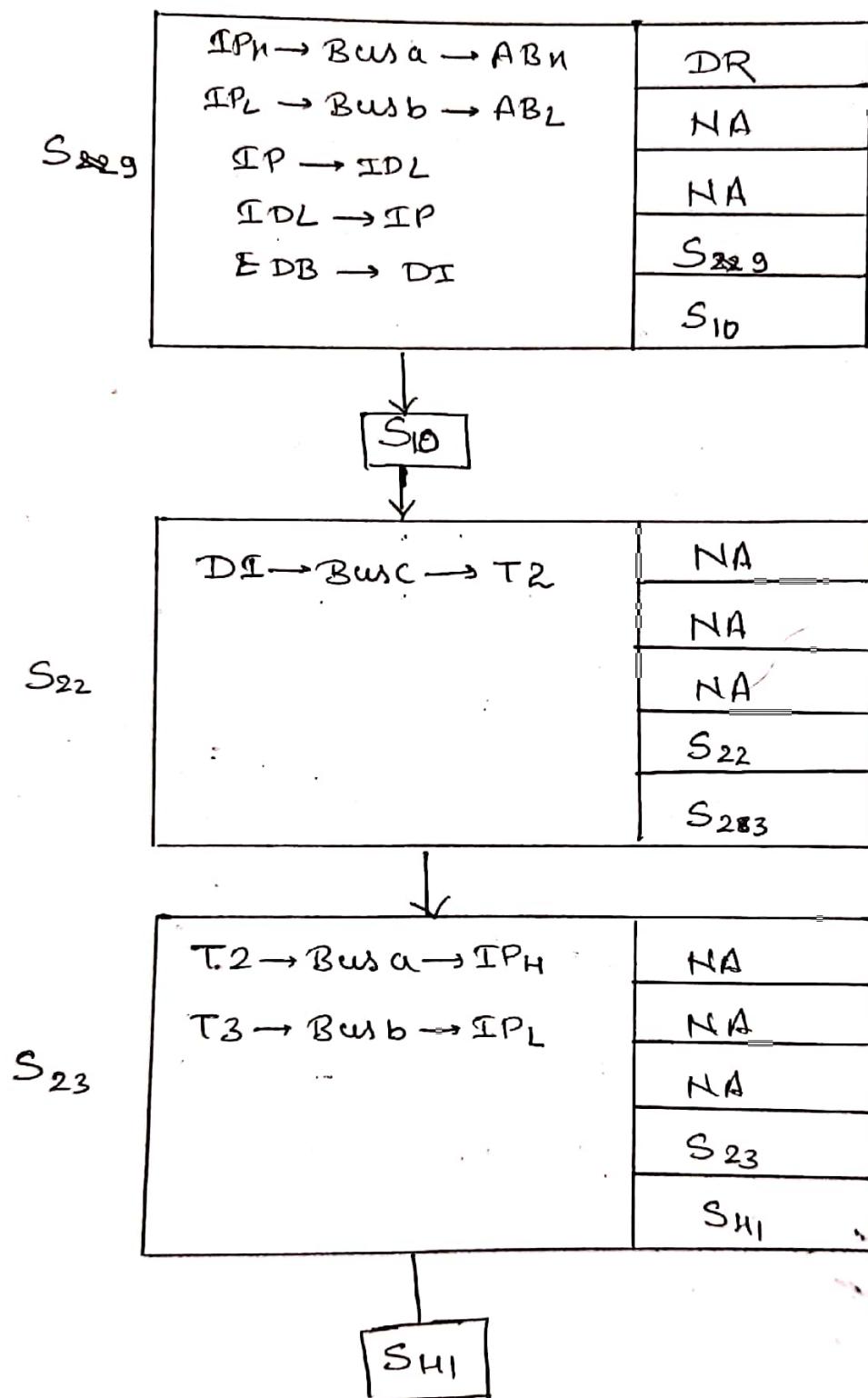


12)

CMP Rg

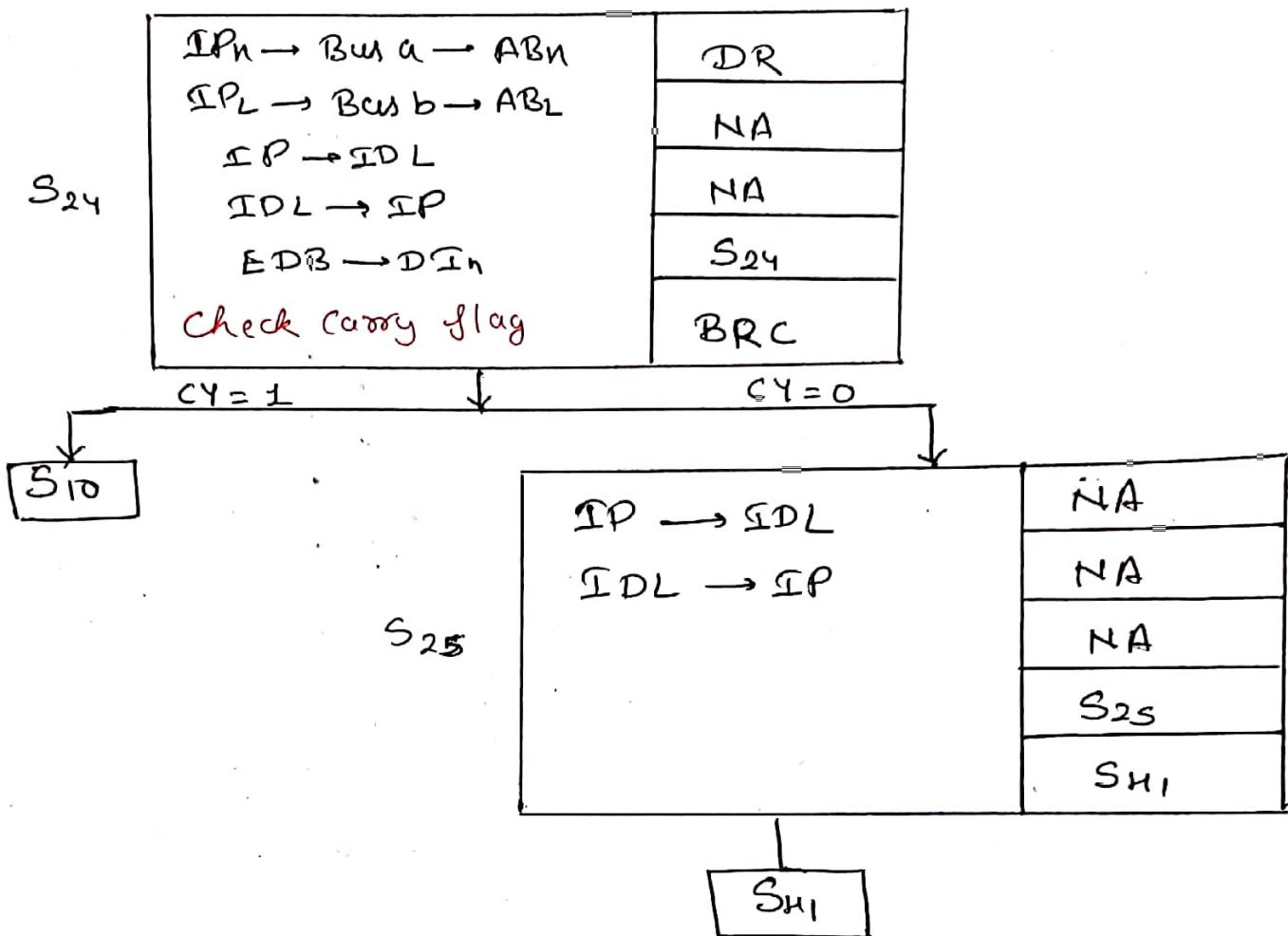


13) JMP D16 → Unconditional jump

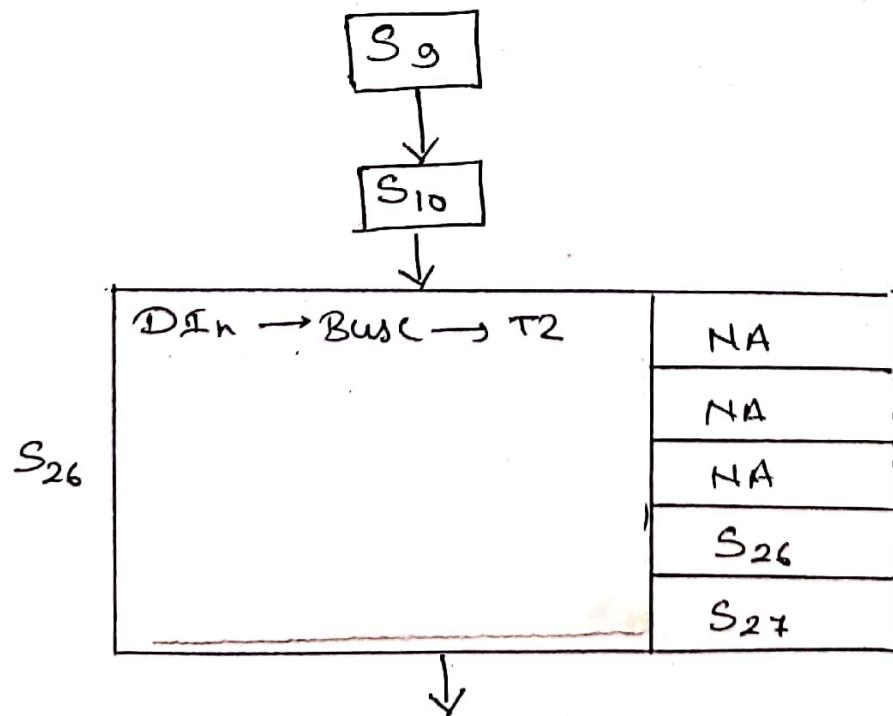


This state is same as first state of LDA i.e.  $S_9$

14) J C D16 → Conditional Jump



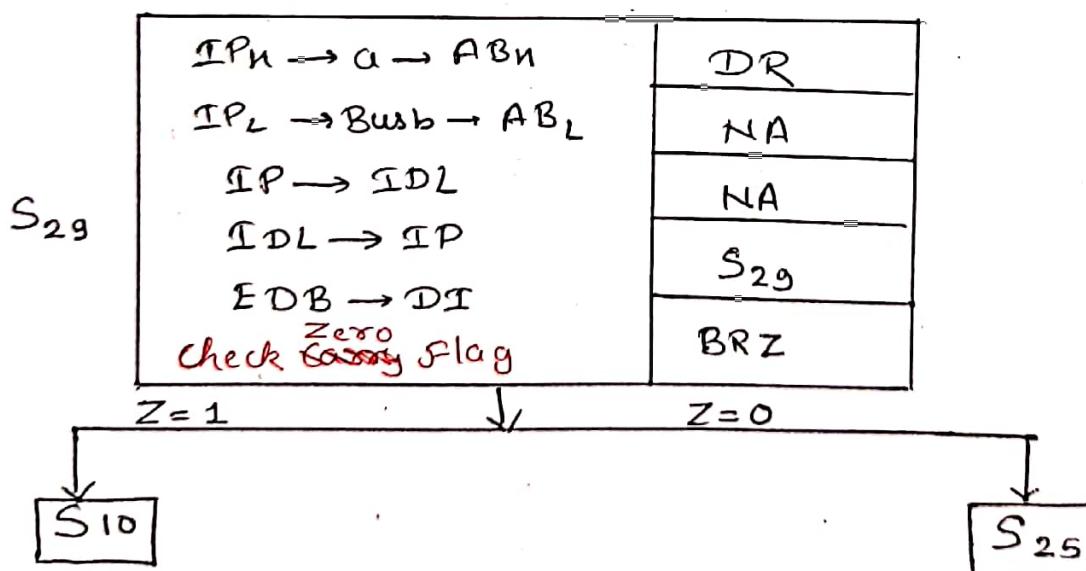
15) CAL2 D16



$IP_H \rightarrow Bus_C \rightarrow D_0$	DW
$SP_H \rightarrow Bus_A \rightarrow AB_H$	NA
$SP_L \rightarrow Bus_B \rightarrow AB_L$	NA
$SP \rightarrow IDL$ (Decrement)	
$IDL \rightarrow SP$	$S_{27}$
	$S_{28}$

$IP_L \rightarrow Bus_C \rightarrow D_0$	DW
$SP_H \rightarrow Bus_A \rightarrow AB_H$	NA
$SP_L \rightarrow Bus_B \rightarrow AB_L$	NA
$SP \rightarrow IDL$ (Decrement)	
$IDL \rightarrow SP$	$S_{28}$
	$S_{203}$

16) C Z DI6



17> RET

$S_{30}$

$SP \rightarrow IDL$	NA
$IDL \rightarrow SP$	NA
	NA
	$S_{30}$
	$S_{31}$



$S_{31}$

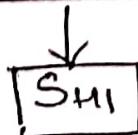
$SP_H \rightarrow Bus_a \rightarrow AB_H$	DR
$SP_L \rightarrow Bus_b \rightarrow AB_L$	NA
$SP \rightarrow IDL$	NA
$IDL \rightarrow SP$	
$EDB \rightarrow DI$	$S_{31}$
	$S_{32}$

$S_{32}$

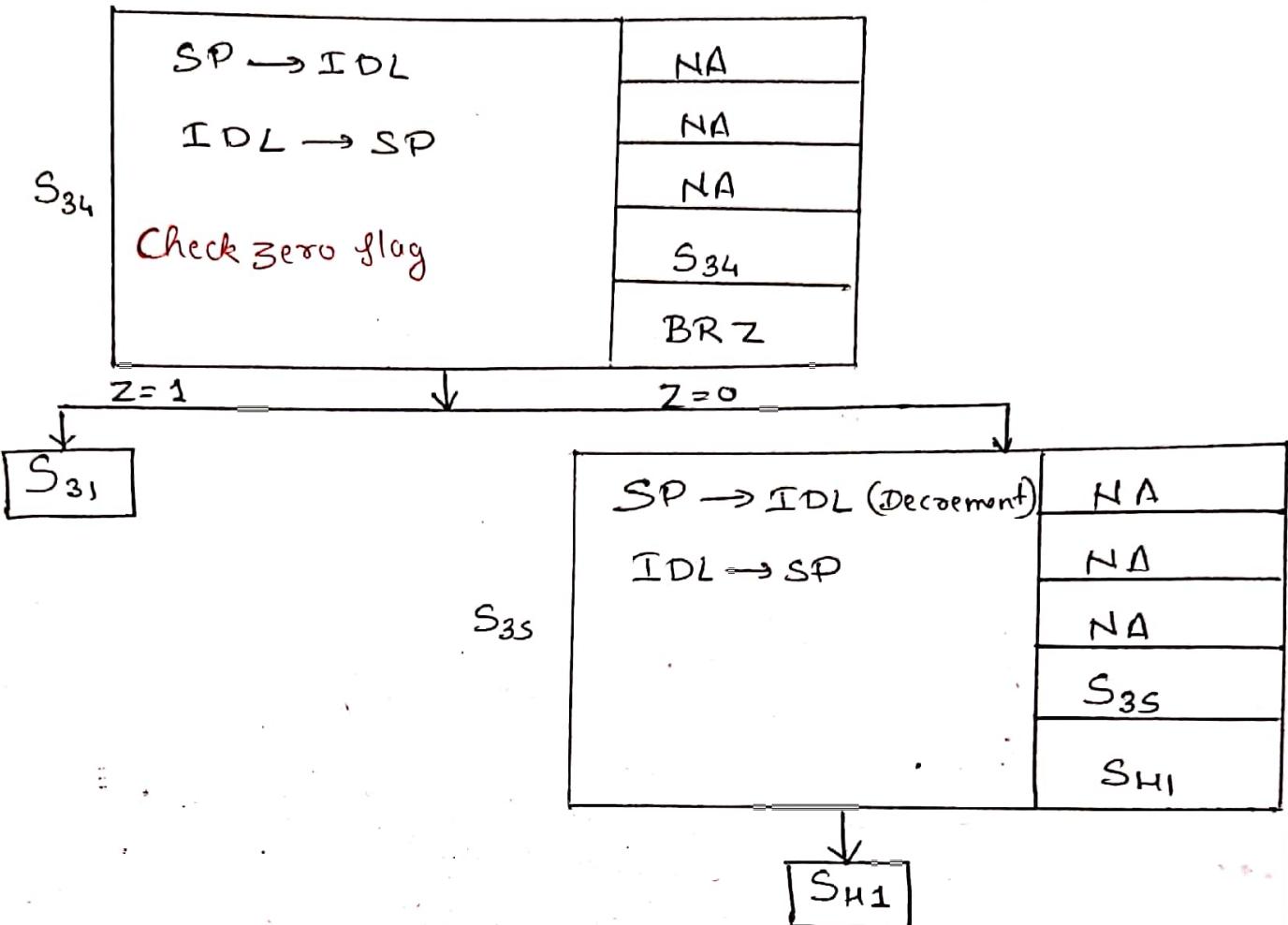
$DI \rightarrow Bus_C \rightarrow IP_L$	DR
$SP_H \rightarrow Bus_a \rightarrow AB_H$	NA
$SP_L \rightarrow Bus_b \rightarrow AB_L$	NA
$SP \rightarrow IDL$	
$IDL \rightarrow SP$	$S_{32}$
$EDB \rightarrow DI$	$S_{33}$

$S_{33}$

$DI \rightarrow Bus_C \rightarrow IP_H$	NA
$SP_H$	NA
	NA
	$S_{33}$
	$S_{H1}$



# 18) RZ

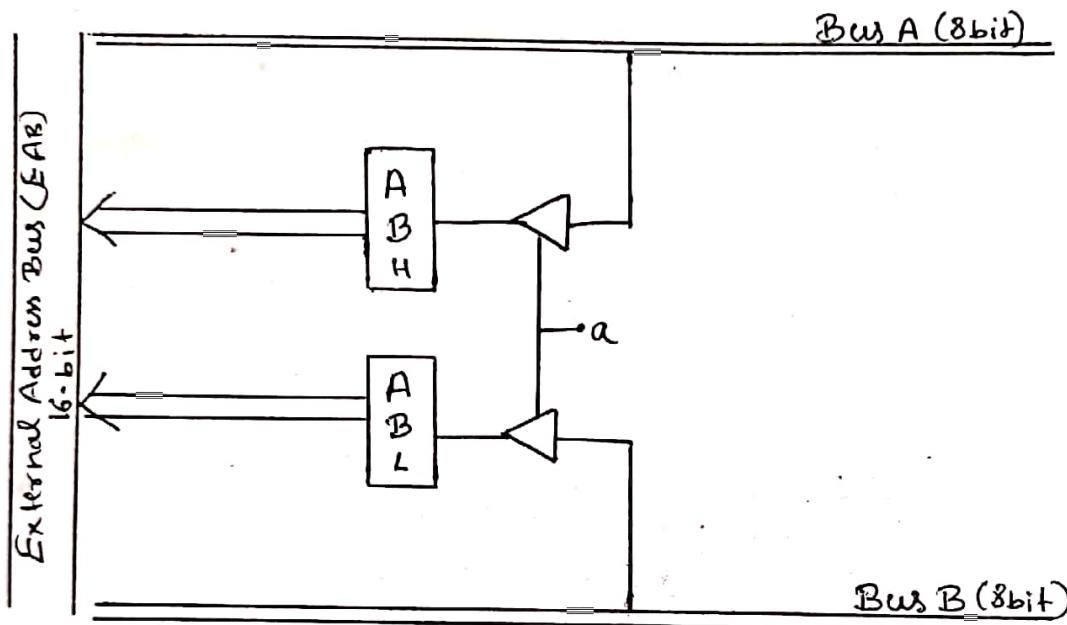


# Components Used in Datapath and their Encoding

## ADDRESS BUFFER (AB)

### Activity of Address Buffer

- 1) None
- 2) Bus A  $\rightarrow$  AB<sub>H</sub>  
Bus B  $\rightarrow$  AB<sub>L</sub>



Since only one control point for AB, so only 1 bit is required in control word corresponding to AB.

$$\alpha = x_1$$

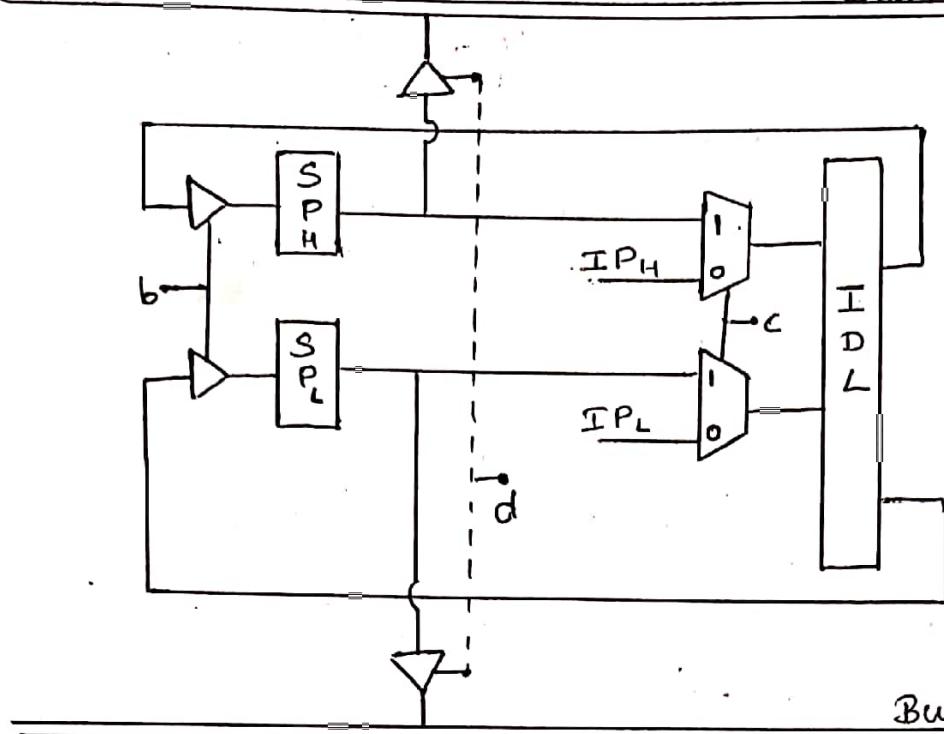
## STACK POINTER (SP)

### Activities of Stack Pointer

- 1) None
- 2) SP  $\rightarrow$  IDL, IDL  $\rightarrow$  SP
- 3) SP<sub>H</sub>  $\rightarrow$  Bus A  
SP<sub>L</sub>  $\rightarrow$  Bus B

Here we see that whenever SP is sent to IDL, at some time IDL is also sent to SP. So switch for both must be active at some time i.e.  $b=c$ .

Bus A (8 bit)



Bus B (8 bit)

Since  $b=c$ , Therefore we have two control point and also no of activities is '3' so we need two bit.

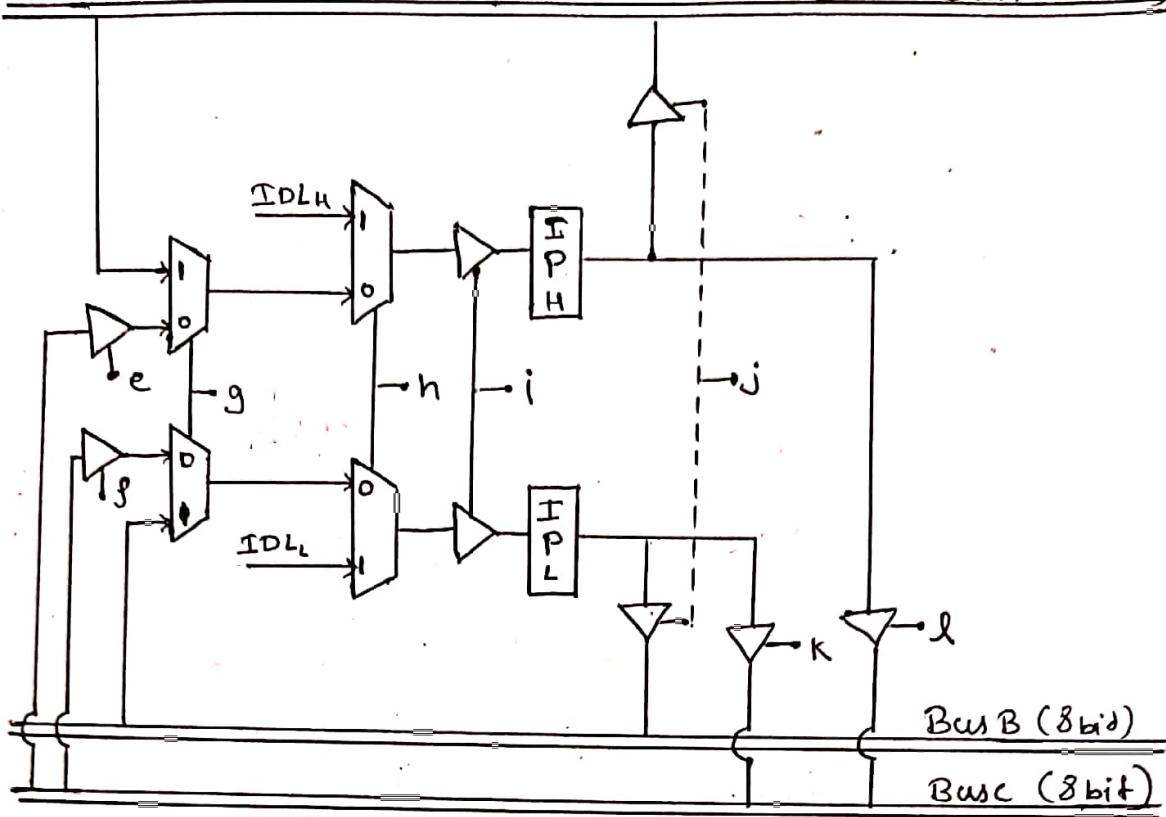
$$\therefore \boxed{b = c = x_2} \quad \boxed{d = x_3}$$

## INSTRUCTION POINTER (IP)

### Activities of Instruction Pointers

- 1) None
- 2)  $\text{IP}_H \rightarrow \text{Bus A}$   
 $\text{IP}_L \rightarrow \text{Bus B}$
- 3)  $\text{IDL} \rightarrow \text{IP}$
- 4)  $\text{IP}_H \rightarrow \text{Bus C}$
- 5)  $\text{IP}_L \rightarrow \text{Bus C}$
- 6)  $\text{Bus C} \rightarrow \text{IP}_H$
- 7)  $\text{Bus C} \rightarrow \text{IP}_L$
- 8)  $\text{Bus A} \rightarrow \text{IP}_H$   
 $\text{Bus B} \rightarrow \text{IP}_L$

Since there are 8 activities, we will need 3-bit in control word for  $\text{IP}_D$



The diagram illustrates the relationship between bus states (Bus A, Bus B, Bus C) and IP addresses (IP\_H, IP\_L, IP\_M) over time. The horizontal axis represents time, divided into four segments labeled 00, 01, 11, and 10. The vertical axis represents the bus states.

- Segment 00:** Labeled "Nope". It shows transitions from  $\text{IP}_H \rightarrow \text{Bus } a$ ,  $\text{IP}_L \rightarrow \text{Bus } b$ , and  $\text{IDL} \rightarrow \text{IP}_M$ .
- Segment 01:** Shows a transition from  $\text{Bus } a \rightarrow \text{IP}_H$ .
- Segment 11:** Shows a transition from  $\text{Bus } c \rightarrow \text{IP}_L$ .
- Segment 10:** Shows a transition from  $\text{Bus } b \rightarrow \text{IP}_L$ .

$$C = x_4 \bar{x}_5 \bar{x}_6$$

$$j = \overline{x_4} \overline{x_5} x_6$$

$$f = x_4 x_5 x_6$$

$$k = \overline{x_4} x_5 \overline{x_6}$$

$$g = x_4 x_5 \bar{x}_6$$

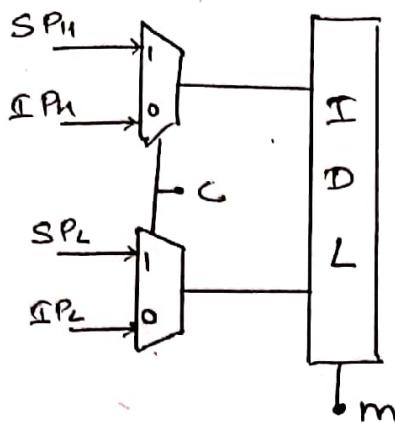
$$l = \overline{x_4 x_5 x_6}$$

$$h = \overline{x_5} x_6$$

$$l = \overline{x_4 x_5 x_6}$$

$$i = x_3 + x_4 + x_6$$

## INCREMENT/DECREMENT LATCH (IDL)



Control point 'c' has already taken care of in SP.

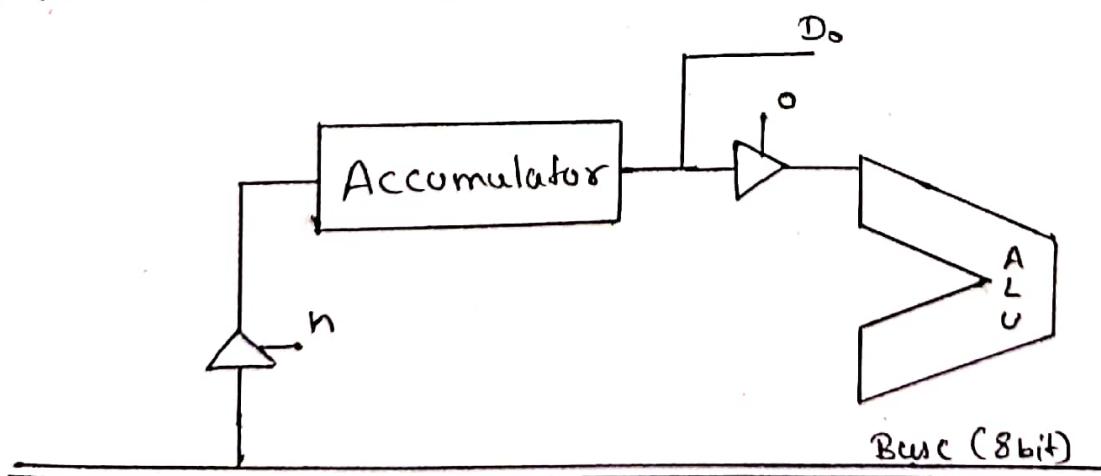
Here we need only one bit in control word for IDL to control Increment or Decrement operation.

$$m = x_7$$

## ACCUMULATOR (A)

### Activities of Accumulator

- 1) None
- 2) Bus  $C \rightarrow A$
- 3)  $A \rightarrow ALU-1$
- 4)  $A \rightarrow D_o$  (Data Out)



Since we have 4 activity related to Accumulator, so we need 2 bits and also we have two control points for 'A'.

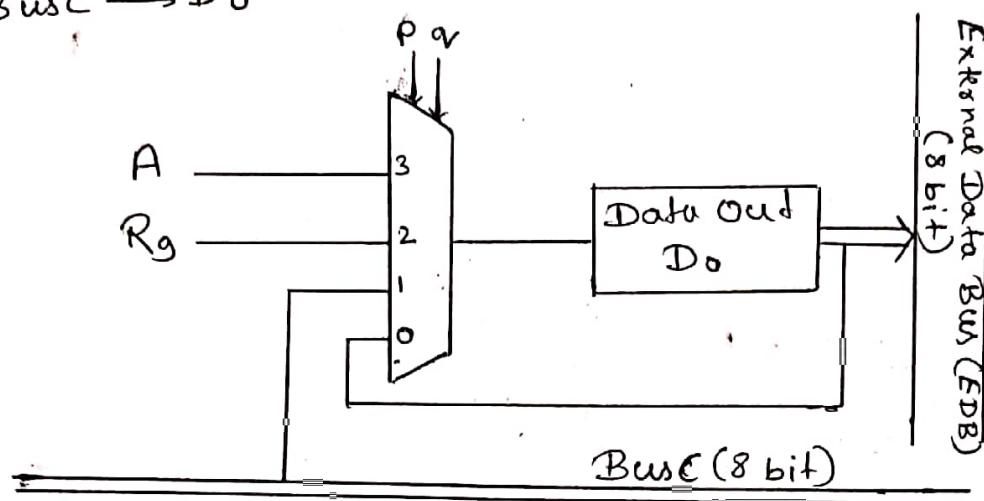
Therefore,  $n = x_8$

$$O = x_9$$

## Data Out (Do)

### Activities of Data Out

- 1) None (Here output of Do will be feed as i/p)
- 2)  $A \rightarrow Do$
- 3)  $R_g \rightarrow Do$
- 4)  $BusC \rightarrow Do$



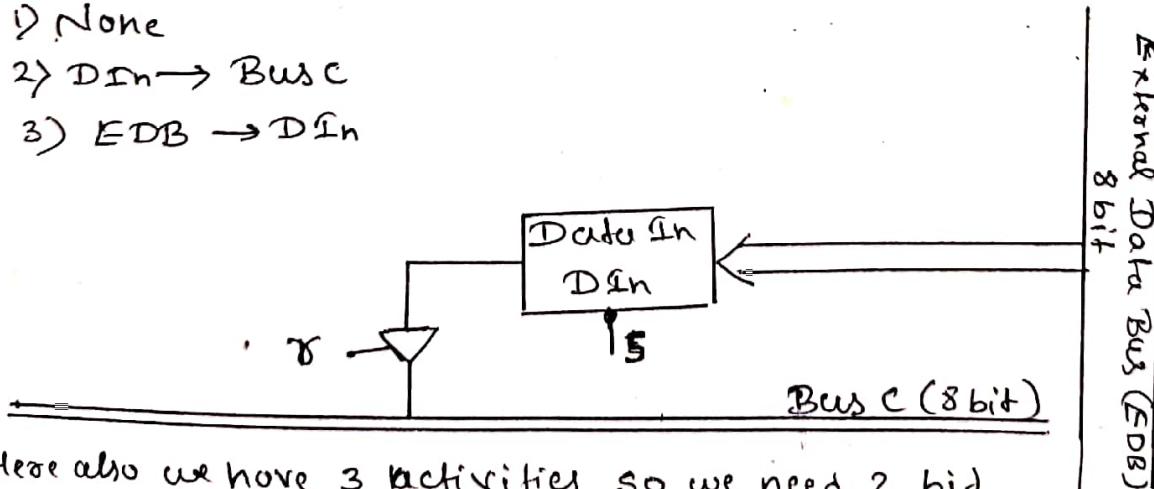
Here we have 4 activities and two control points for  $Do$ . so we need 2-bit in control word.

$$\boxed{P = x_{10}} \quad \boxed{q = x_{11}}$$

## DATA IN (DIn)

### Activities of Datain

- 1) None
- 2)  $DIn \rightarrow BusC$
- 3) EDB  $\rightarrow DIn$



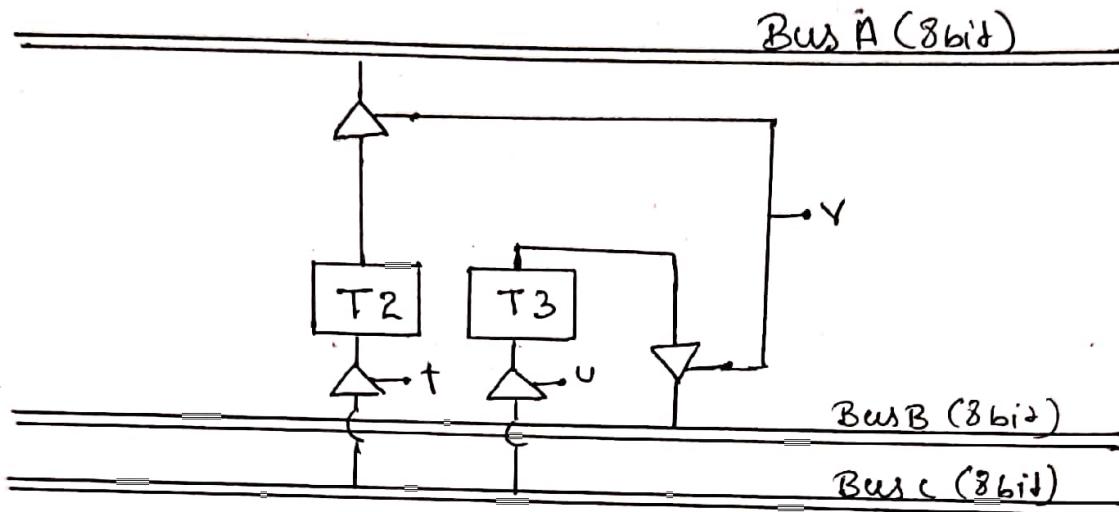
Here also we have 3 activities so we need 2 bit in control word. Since we have two control points for  $DIn$

Therefore,  $\boxed{\bar{r} = x_{12}} \quad \boxed{s = x_{13}}$

## T2 and T3 Register

### Activities of T2 and T3 register

- 1) None
- 2) BusC  $\rightarrow$  T2
- 3) BusC  $\rightarrow$  T3
- 4) T3  $\rightarrow$  BusB
- T2  $\rightarrow$  BusA



We have 4 activities. So we need 2 bit in control word for T2 & T3 register

$x_{15}$	$x_{14}$	0	1
	0	None	$BusC \rightarrow T2$
	1	$BusC \rightarrow T3$	$T3 \rightarrow BusB$ $T2 \rightarrow BusA$

$t = \bar{x}_{14} x_{15}$

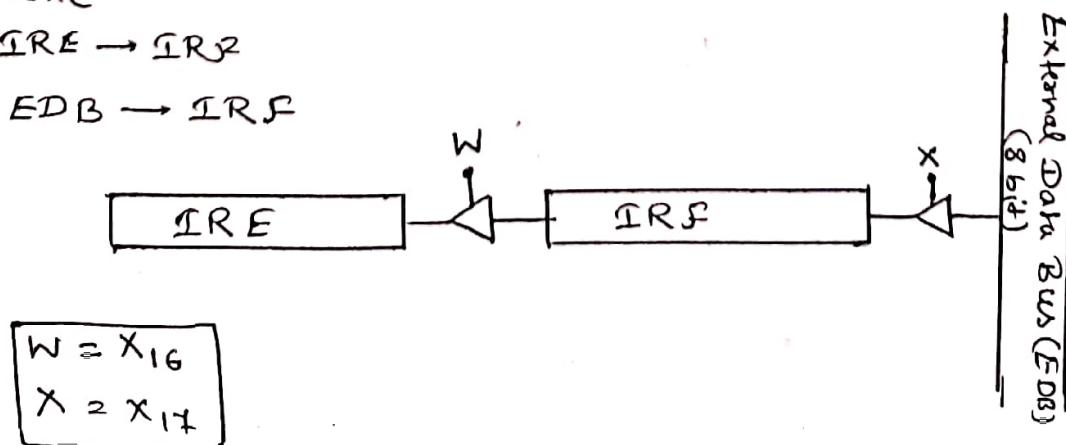
$u = x_{14} \bar{x}_{15}$

$v = x_{14} x_{15}$

## IRE and IRS

### Activities of IRE and IRS

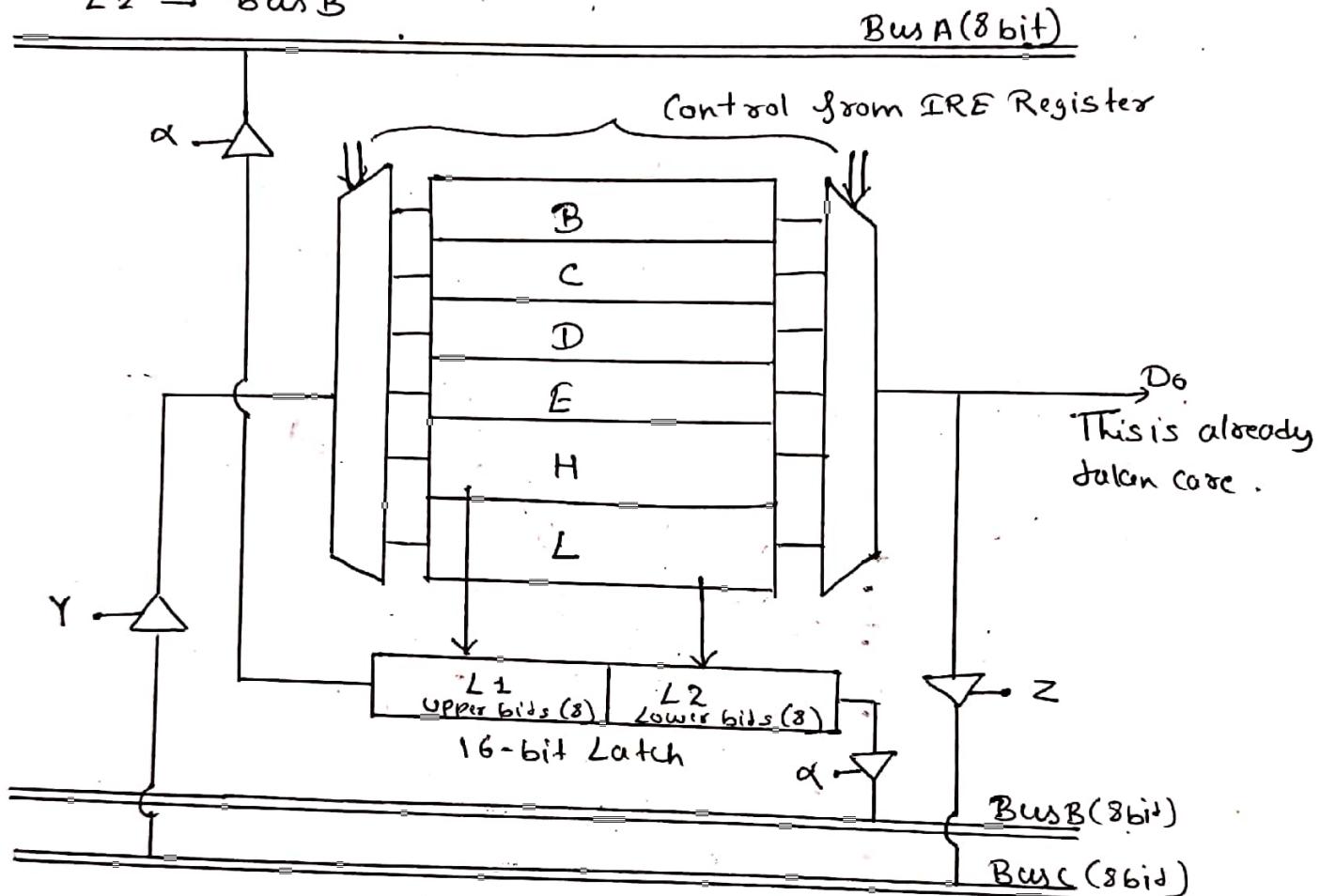
- 1) None
- 2) IRE  $\rightarrow$  IRS
- 3) EDB  $\rightarrow$  IRS



## Register File (RF)

### Activities of Register File

- 1) None
- 2)  $R_g \rightarrow Bus_c$ ,  $Bus_c \xrightarrow{ctrl} R_g$
- 3)  $Bus_c \rightarrow R_g$
- 4)  $L_1 \rightarrow Bus_A$   
 $L_2 \rightarrow Bus_B$



Hence we need only 2-bit in control word using we have 4 activities for RF.

$X_{19}$	$X_{18}$	0	1
0	0	None $Bus_c \rightarrow R_g$ $R_g \rightarrow Bus_c$	
1	1	$H \rightarrow Bus_a$ $L \rightarrow Bus_b$ $Bus_c \rightarrow R_g$	

$$\begin{aligned}
 Y &= X_{19} \\
 Z &= \overline{X_{18}} X_{19} \\
 \alpha &= X_{18} \overline{X_{19}}
 \end{aligned}$$

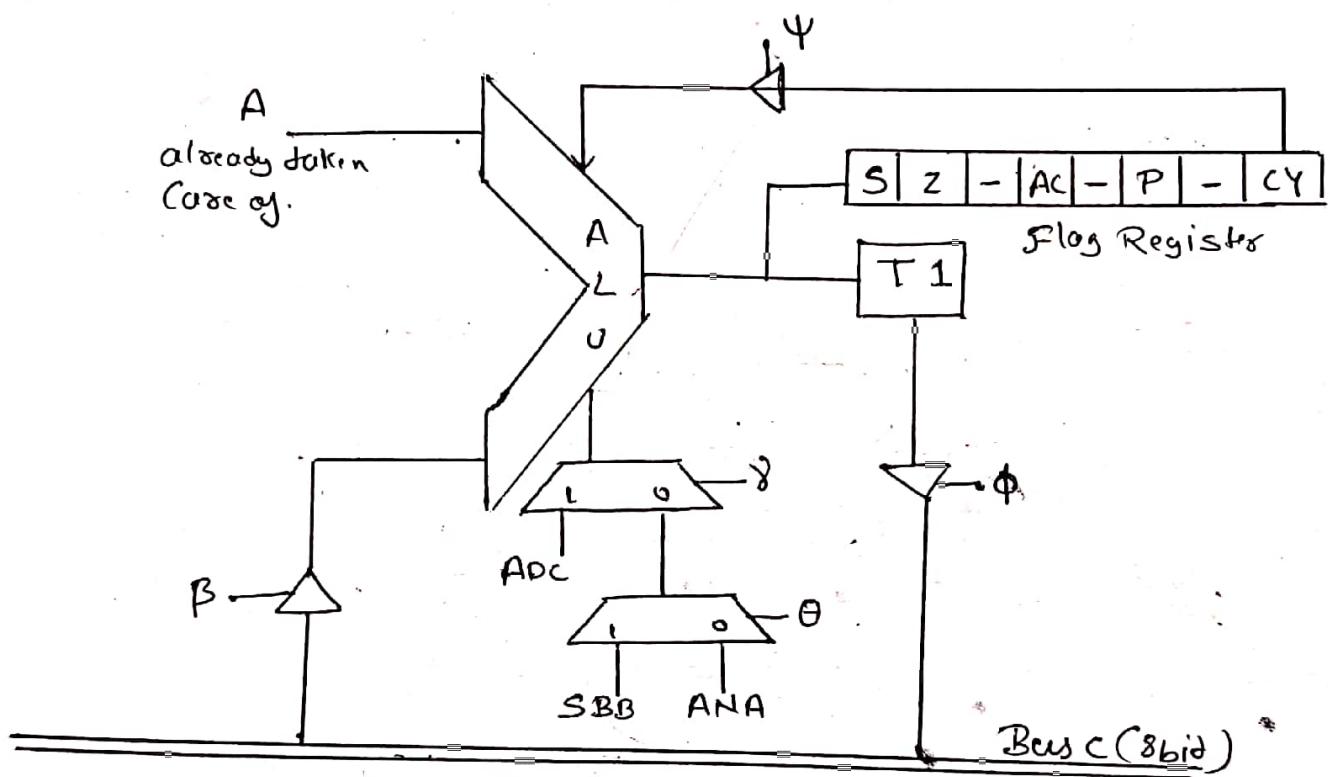
# ARITHMETIC AND LOGIC UNIT (ALU)

## Activities of ALU

- 1) None
- 2)  $A \rightarrow alu\_1$  ADC (ADD with carry)  
 $Busc \rightarrow alu\_2$  Flag modified.

- 3)  $A \rightarrow alu\_1$  SBB (Subtract with Borrow)  
 $Busc \rightarrow alu\_2$  Flag modified

- 4)  $A \rightarrow alu\_1$  ANA (AND operation)  
 $Busc \rightarrow alu\_2$  Flag modified



$x_{21}$	$x_{20}$	None	$Busc \rightarrow alu\_2$ $A \rightarrow alu\_1$ ADC Flag modified
1	1	$Busc \rightarrow alu\_2$ $A \rightarrow alu\_1$ ANA Flag mod	$Busc \rightarrow alu\_2$ $A \rightarrow alu\_1$ SBB Flag mod

$$\begin{aligned} \beta &= x_{20} + x_{21} \\ \gamma &= \overline{x_{20}} x_{21} \\ \theta &= x_{20} x_{21} \end{aligned}$$

For T1 and Carry flag (CY)

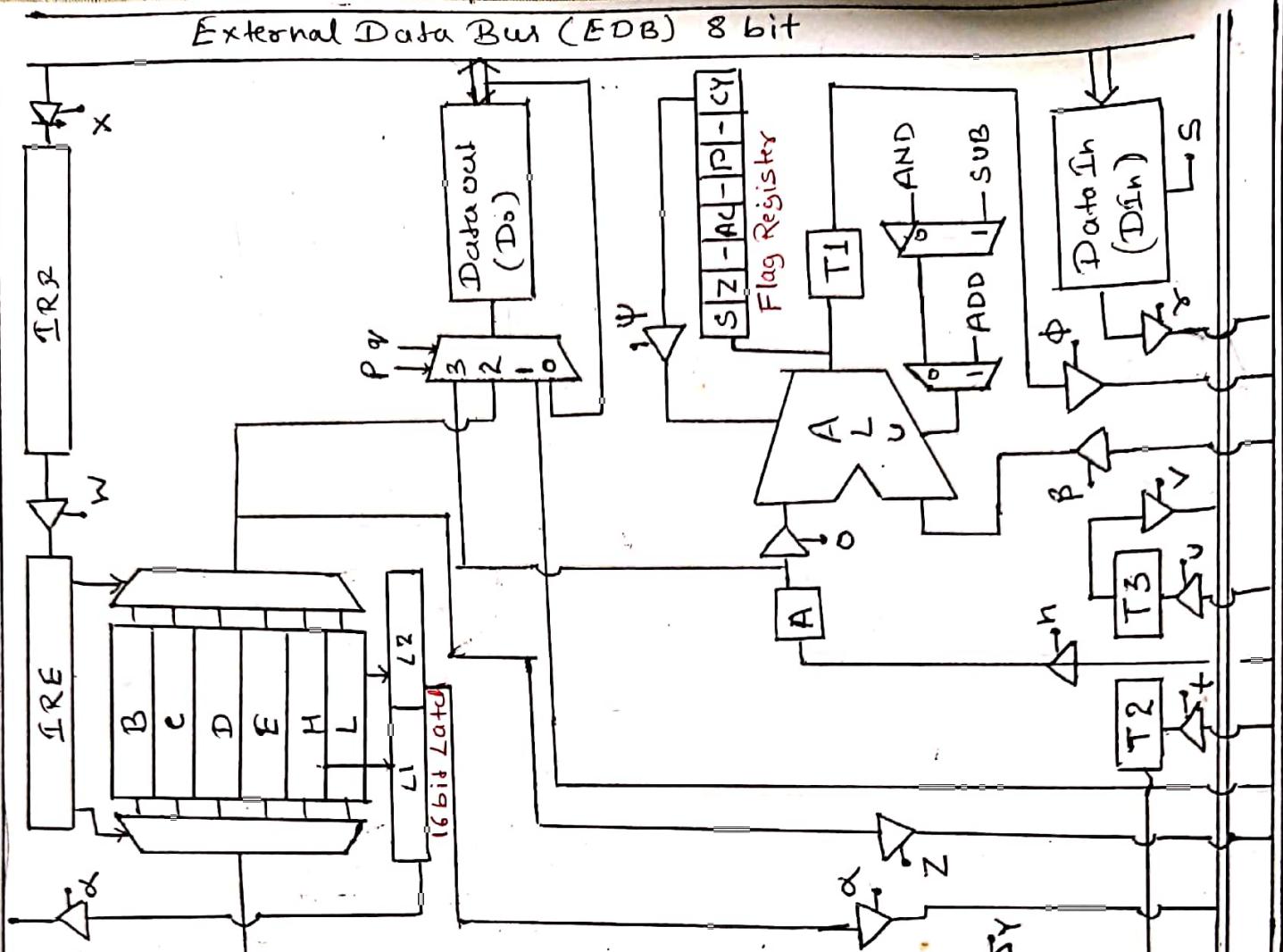
We need only one bit to control it in control word

$$\Phi = x_{22}$$

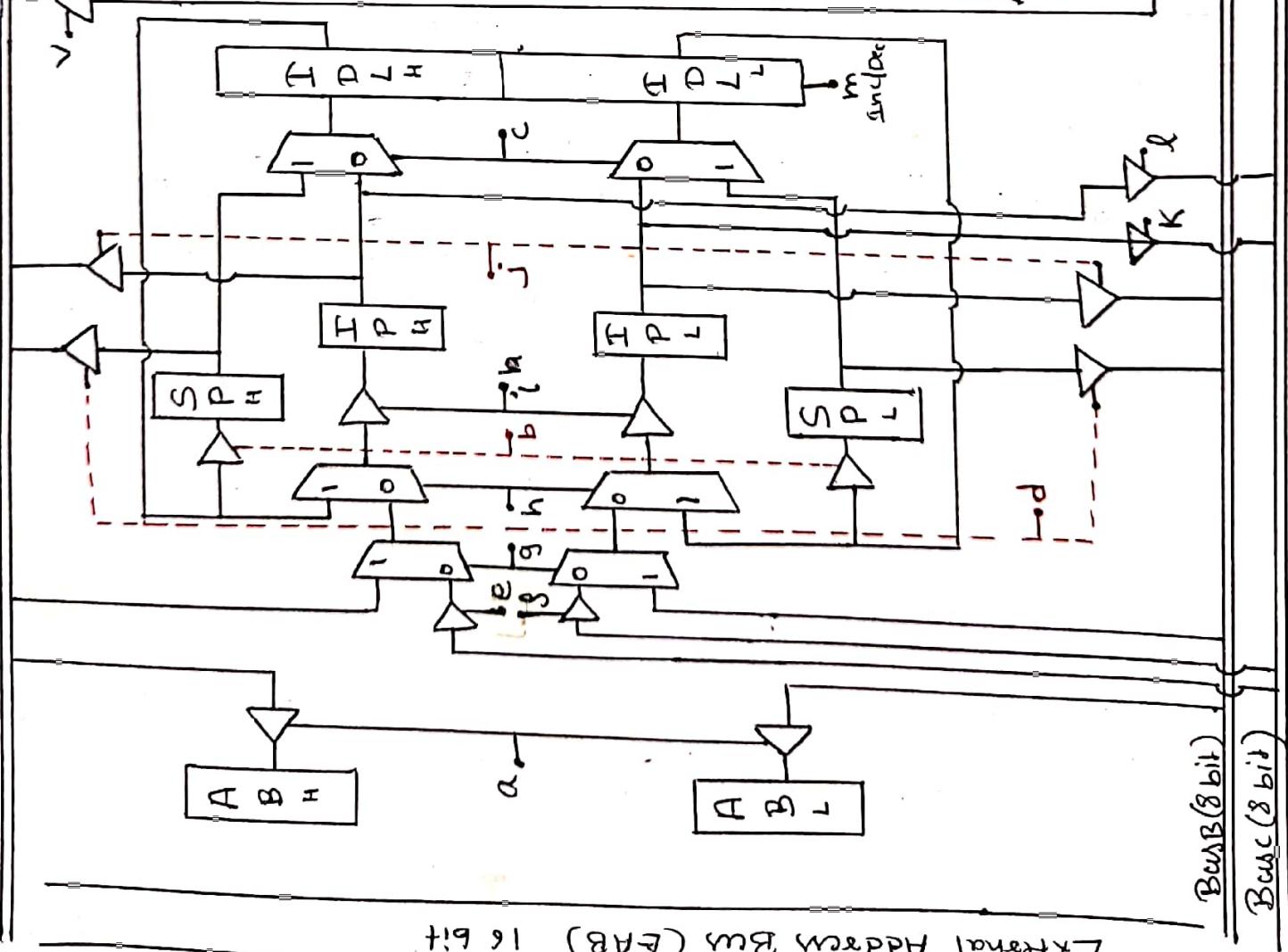
$$\Psi = x_{23}$$

# External Data Bus (EDB) 8 bit

Bus A (8 bit)



Data Path



External Address Bus (EAB) 16 bit

Scanned with CamScanner

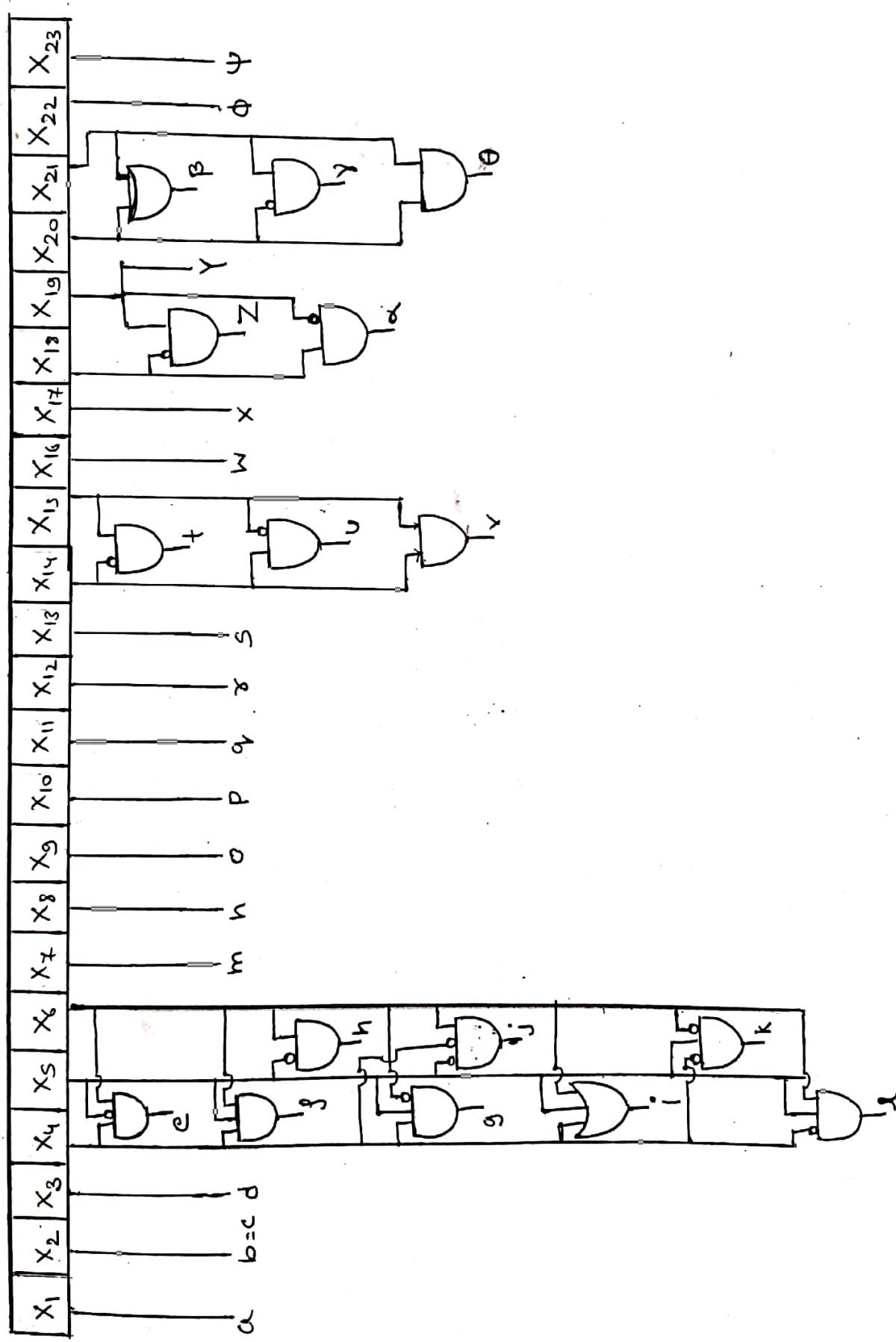
Bus B (8 bit)  
Bus C (8 bit)

## Control Word Structure

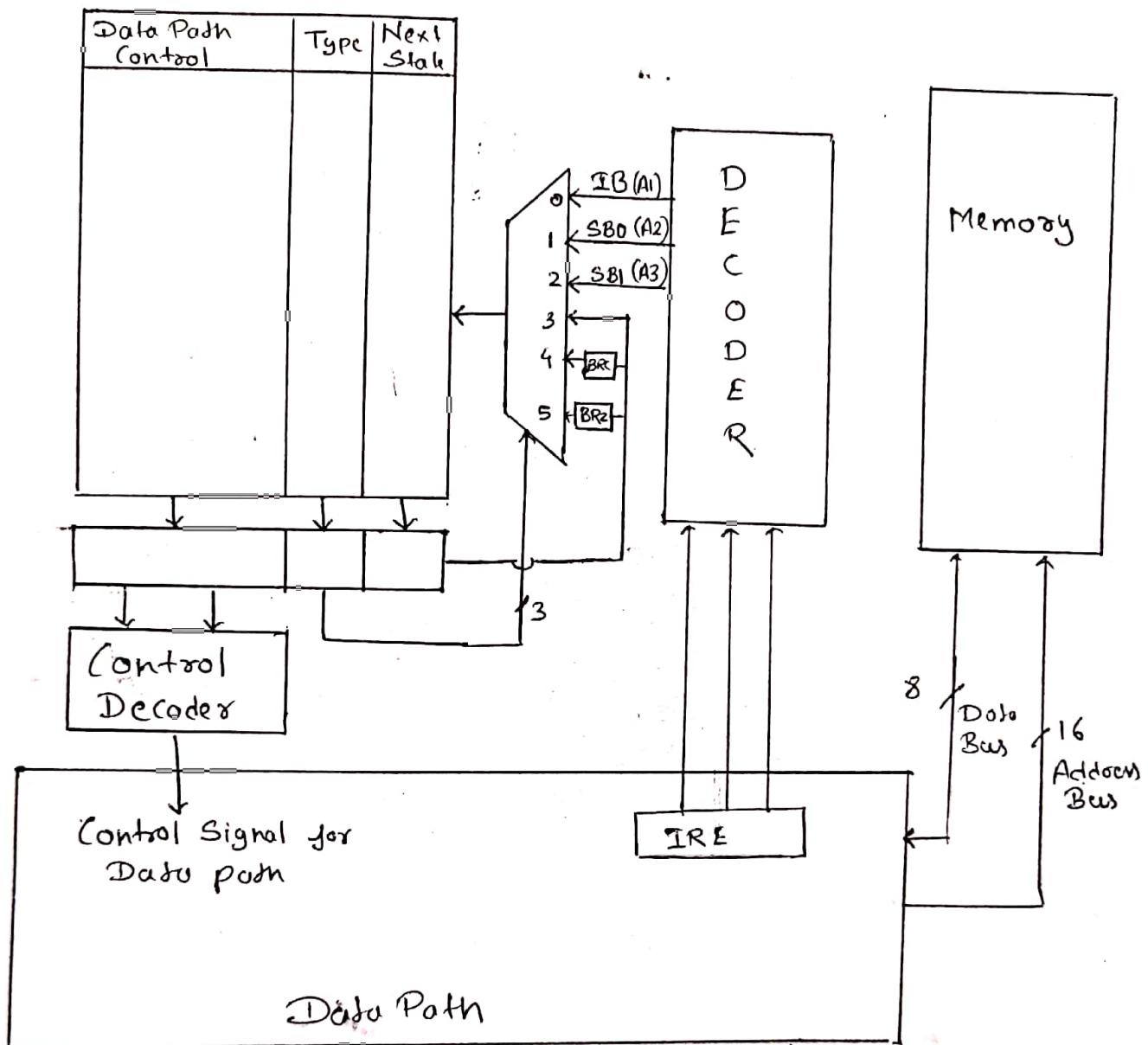
Current State	AB SP	IP	IDL	Accu	Do	DIn	T2 & T3 RE/IRF	RF	ALU	T1 CY	Next State																														
State	S <sub>5</sub>	S <sub>4</sub>	S <sub>3</sub>	S <sub>2</sub>	S <sub>1</sub>	S <sub>0</sub>	X <sub>1</sub>	X <sub>2</sub>	X <sub>3</sub>	X <sub>4</sub>	X <sub>5</sub>	X <sub>6</sub>	X <sub>7</sub>	X <sub>8</sub>	X <sub>9</sub>	X <sub>10</sub>	X <sub>11</sub>	X <sub>12</sub>	X <sub>13</sub>	X <sub>14</sub>	X <sub>15</sub>	X <sub>16</sub>	X <sub>17</sub>	X <sub>18</sub>	X <sub>19</sub>	X <sub>20</sub>	X <sub>21</sub>	X <sub>22</sub>	X <sub>23</sub>	T <sub>1</sub>	T <sub>0</sub>	S <sub>5</sub>	S <sub>4</sub>	S <sub>3</sub>	S <sub>2</sub>	S <sub>1</sub>	S <sub>0</sub>				
T <sub>D</sub>	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S <sub>H1</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S <sub>H2</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S <sub>1</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S <sub>2</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S <sub>3</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S <sub>4</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S <sub>5</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S <sub>6</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S <sub>7</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S <sub>8</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S <sub>9</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S <sub>10</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S <sub>11</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S <sub>12</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S <sub>13</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S <sub>14</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S <sub>15</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S <sub>16</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S <sub>17</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S <sub>18</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	Current State																Next State															
State	S <sub>0</sub>	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>	S <sub>7</sub>	S <sub>8</sub>	S <sub>9</sub>	S <sub>10</sub>	S <sub>11</sub>	S <sub>12</sub>	S <sub>13</sub>	S <sub>14</sub>	S <sub>15</sub>	S <sub>16</sub>	D <sub>0</sub>	IPL	Accu	DIN	T <sub>2</sub> &T <sub>3</sub> IRE/IRE	R <sub>F</sub>	ALU	T <sub>1</sub> CY							
T <sub>D</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
S <sub>H1</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0			
S <sub>H2</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
S <sub>1</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
S <sub>2</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
S <sub>3</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
S <sub>4</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
S <sub>5</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
S <sub>6</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
S <sub>7</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
S <sub>8</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
S <sub>9</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
S <sub>10</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
S <sub>11</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
S <sub>12</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
S <sub>13</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
S <sub>14</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
S <sub>15</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
S <sub>16</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
S <sub>17</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
S <sub>18</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			

## Control Word along with Decode logic or Data Path Control



## ROM



Complete Microprocessor, Mini-8085