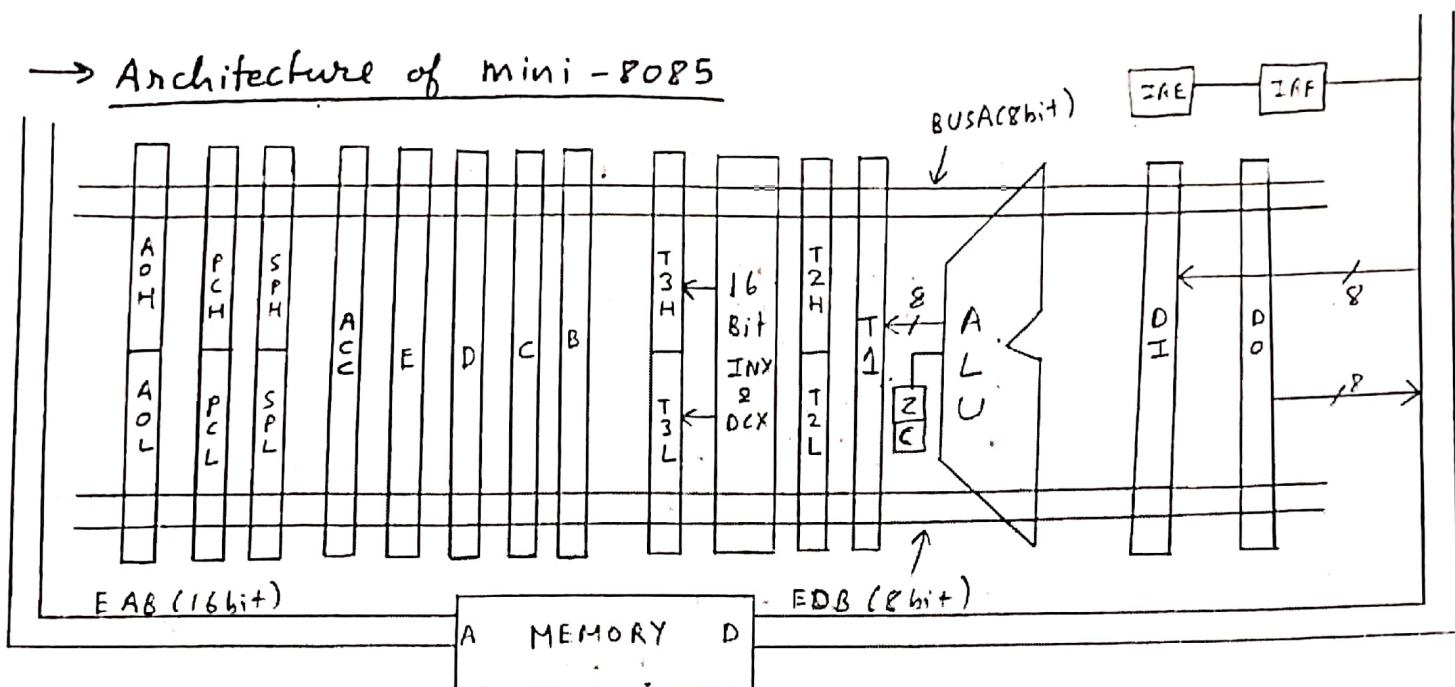


→ Architecture of mini - 8085

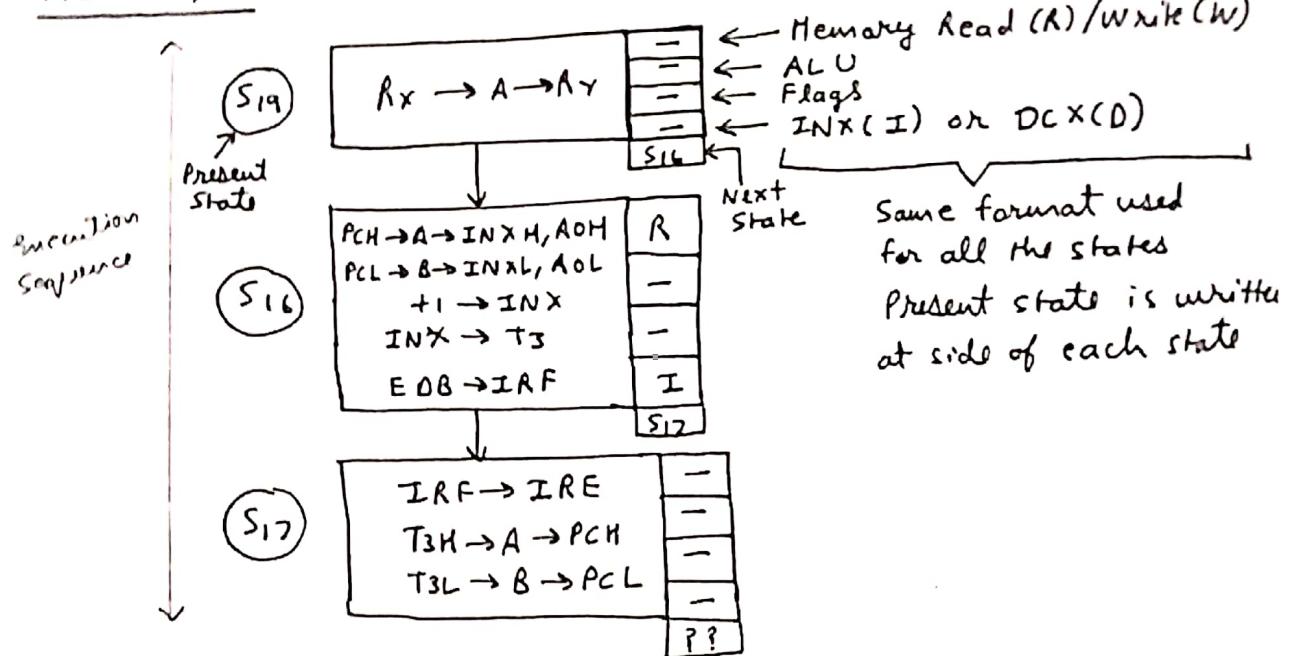


A brief description about the architecture

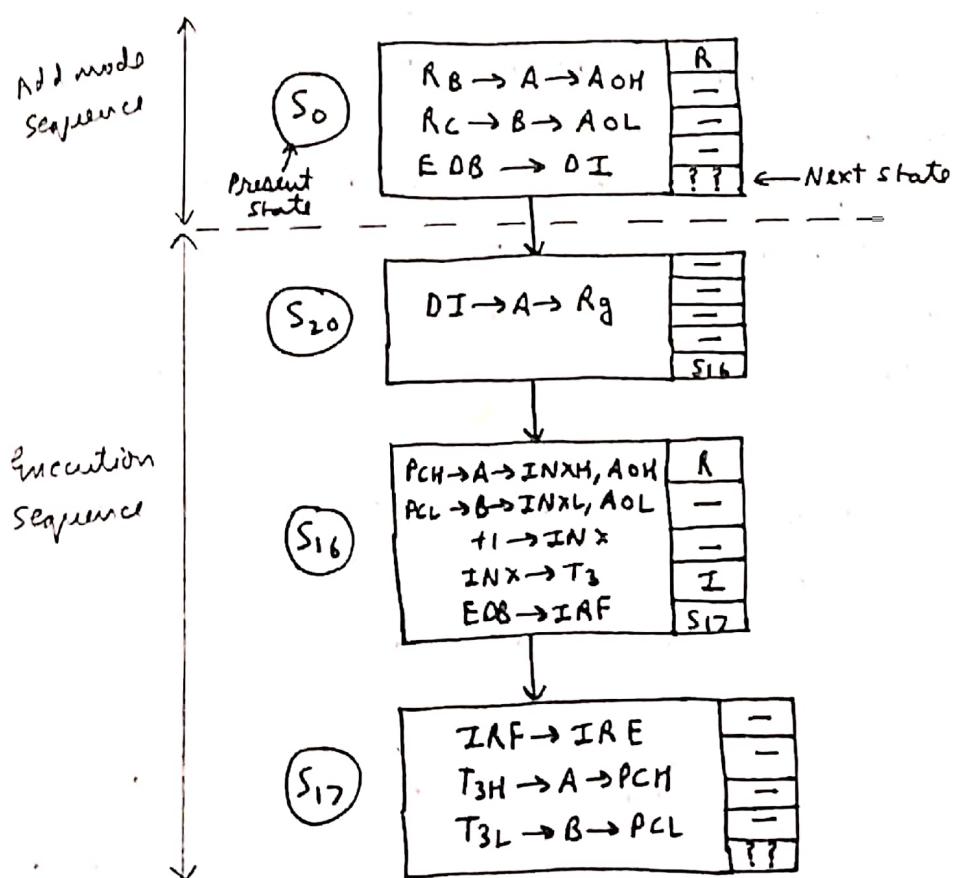
- i) ALU is 8 bit. Its output directly goes to T₁. There are two flags Z (zero flag) and C (carry flag).
- ii) There are 5 16 bit registers A₀, PC, SP, T₃ & T₂ (temp. reg.). Rest all registers are 8 bit.
- iii) There are 4 buses. External Address Bus is 16 bit. BUS A, BUS B & Ext. data bus is 8 bit
- iv) Apart from ALU, there is a separate 16 bit Adder/Sub. unit called INY/DCX whose output directly goes to the 16 bit register T₃.
- v) In the 16 bit register, the higher 8 bits are only connected to Bus A, the lower 8 bits are only connected to Bus B. All the 8 bit registers (except IRE & IFF) are connected to both the buses.
- vi) The purpose of reg T₂ is mainly to store the fetched 16 bit data. If the fetched data is 8 bit, it's stored in DI itself.
- vii) Register Pairs = BC & DE
- viii) M (mem pointer) = BC

→ Hardware Flow Charts

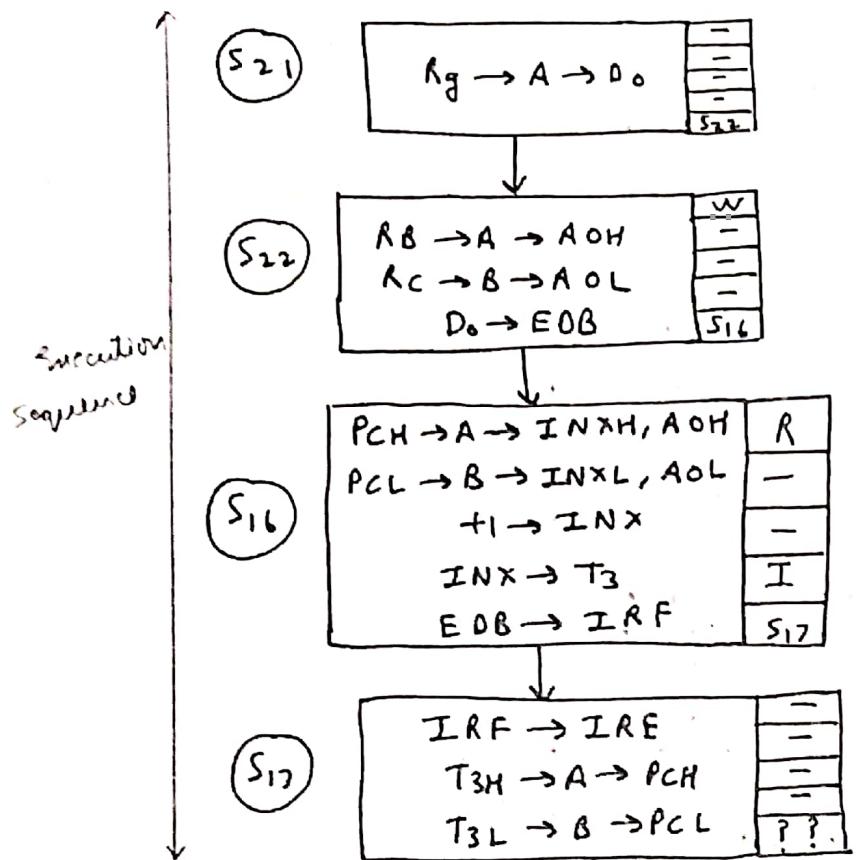
1) MOV Rx, Ry



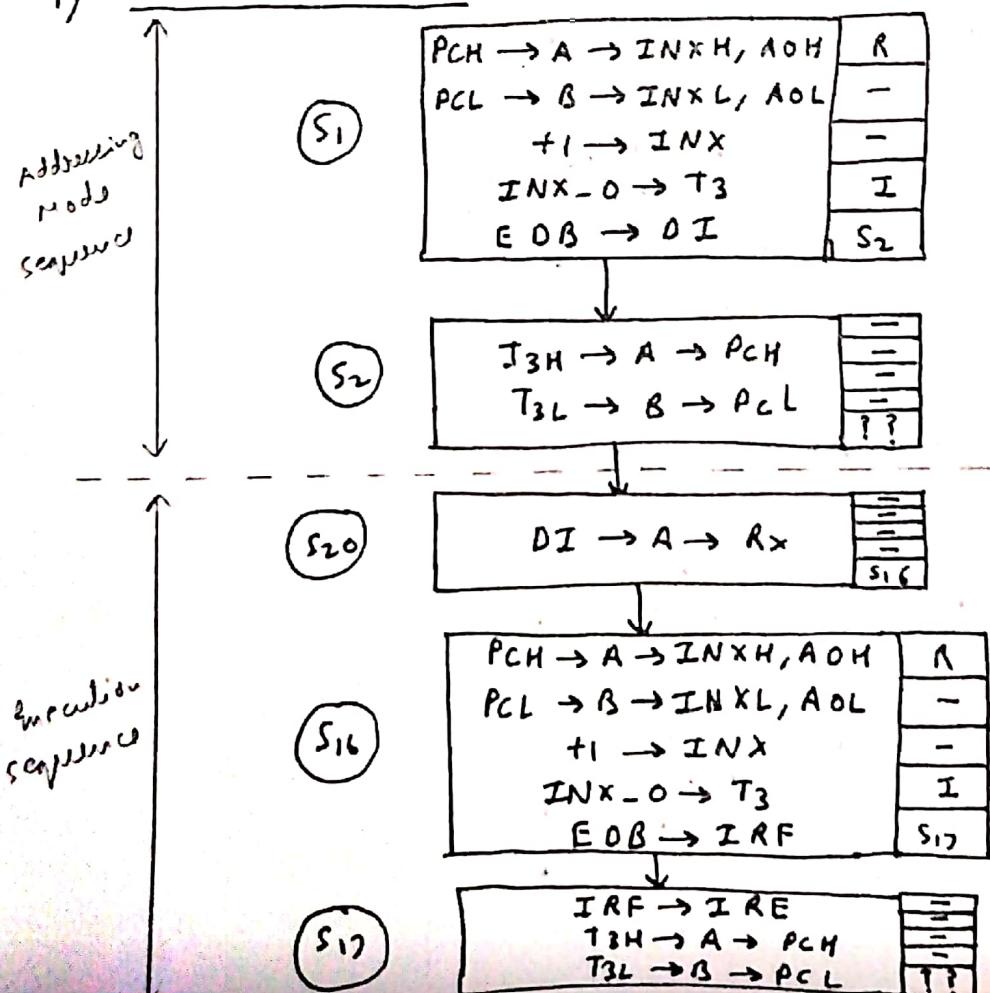
2) MOV Rg, M



3) MOV M, Rg

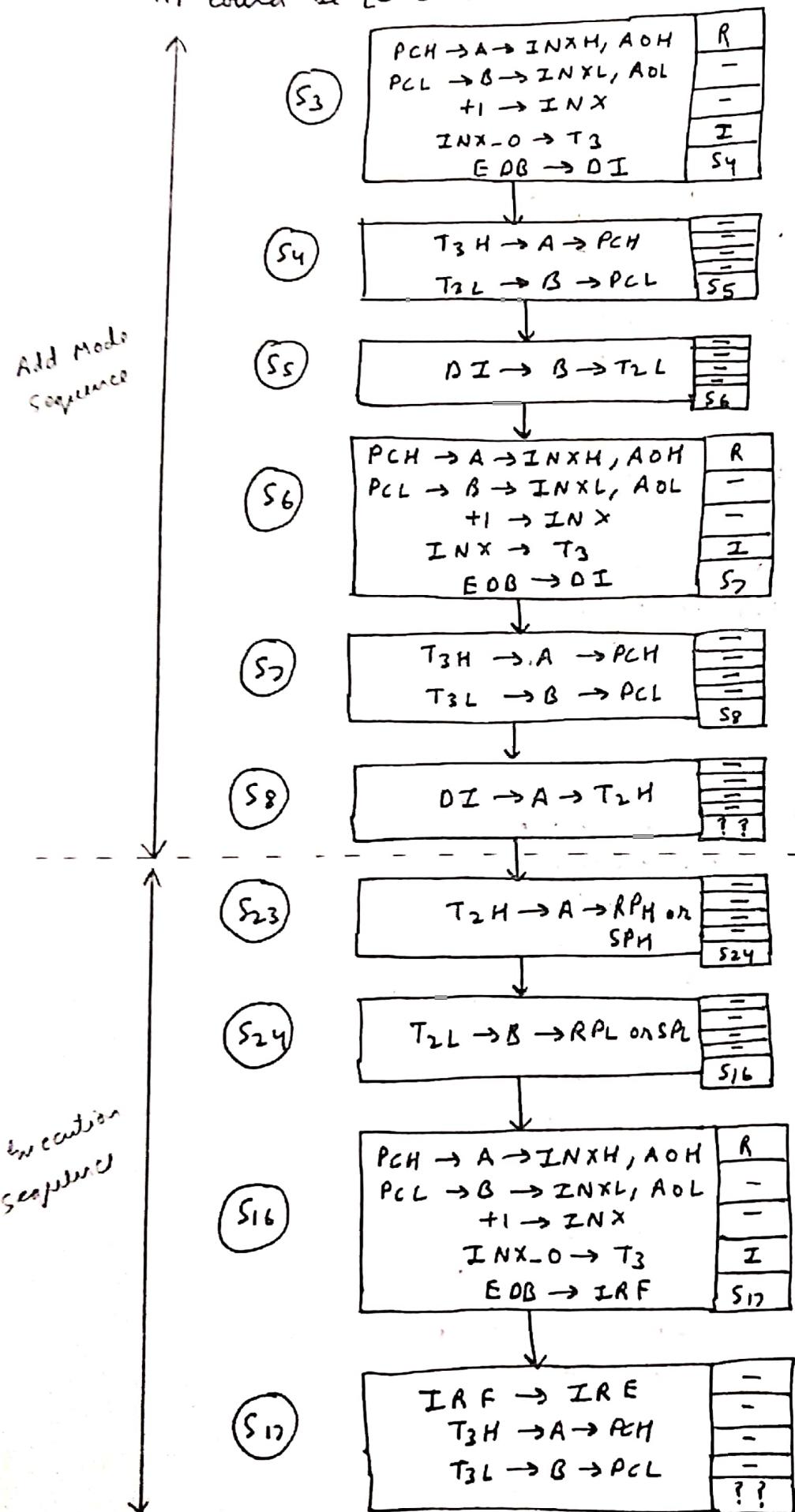


4) MVI Rx, Data 08



5) LXI RP/SP, 016

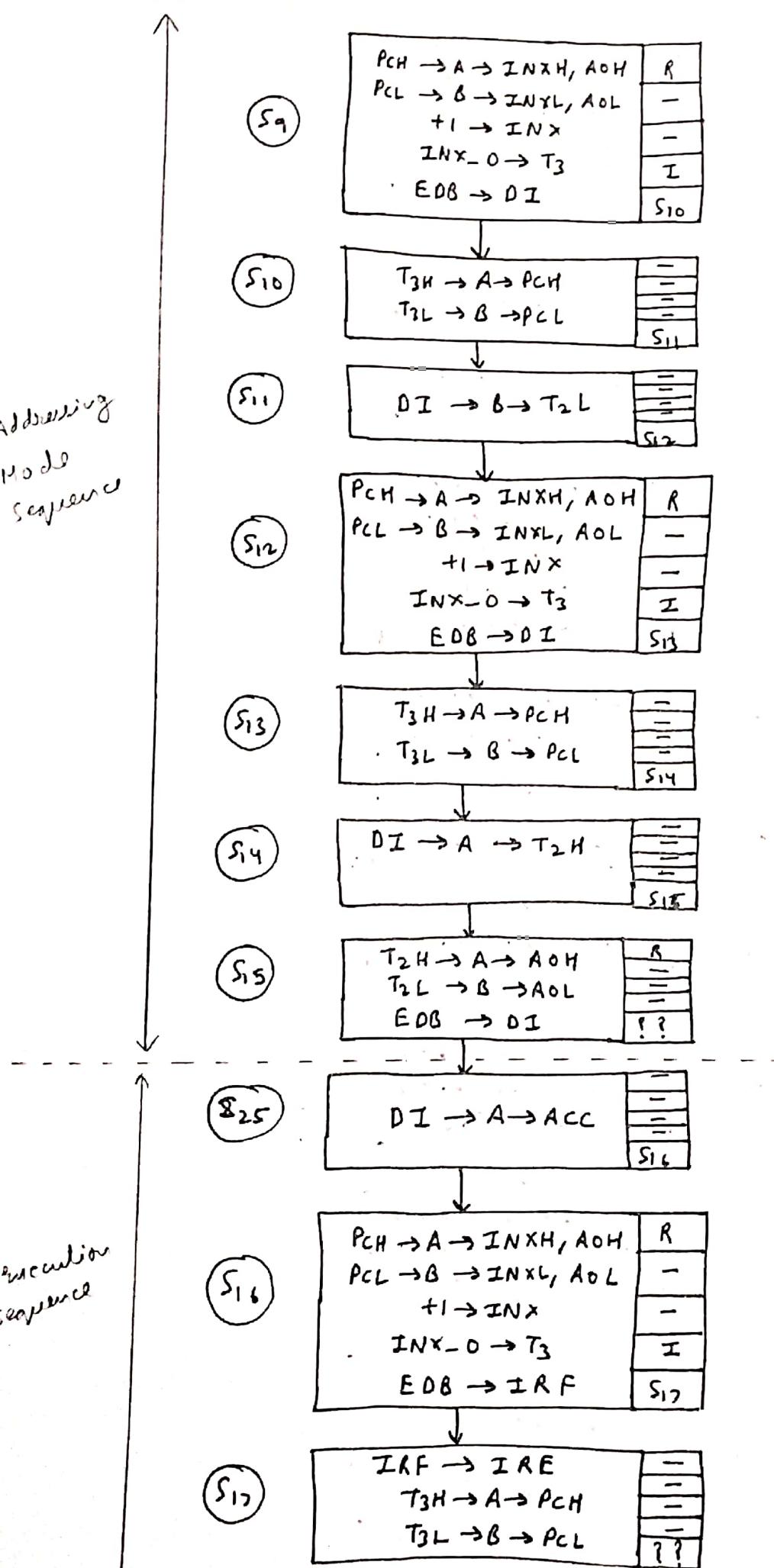
RP could be [BC] or [DE]



6) LOA 016

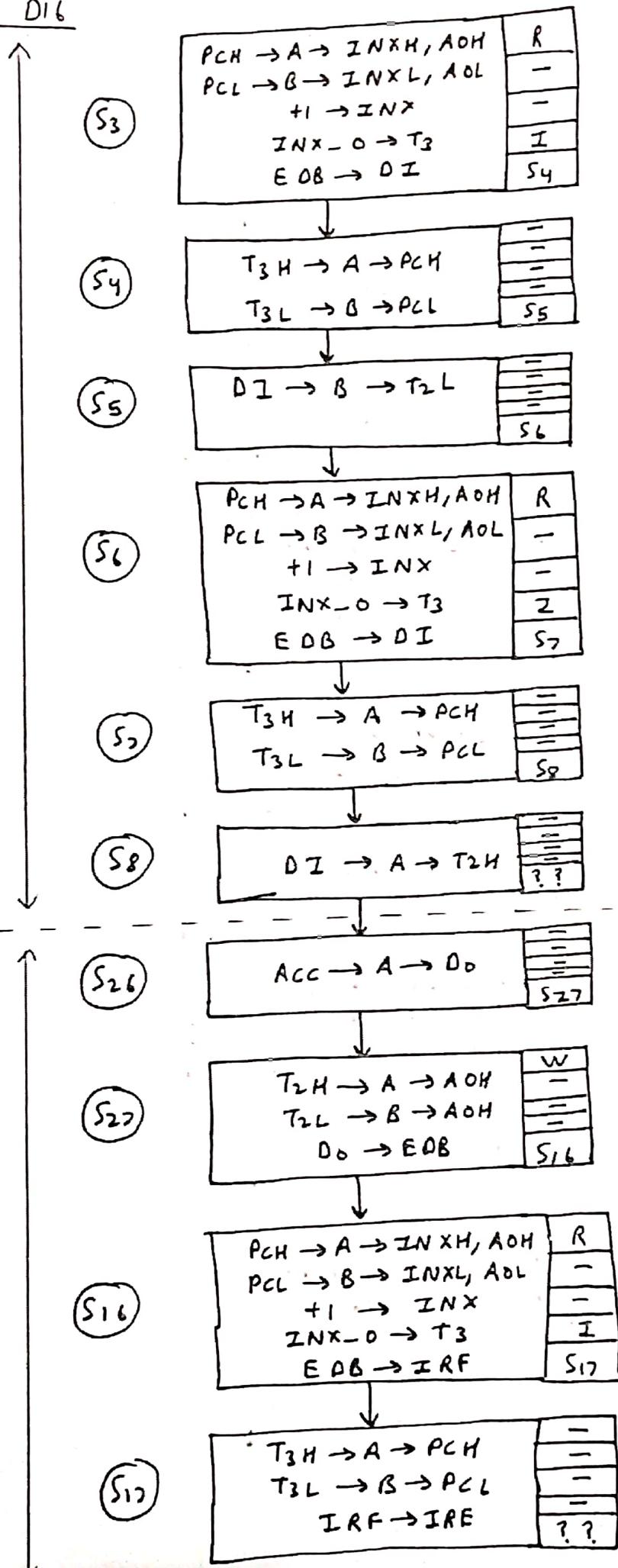
Addressing
Mode
Sequence

Execution
Sequence



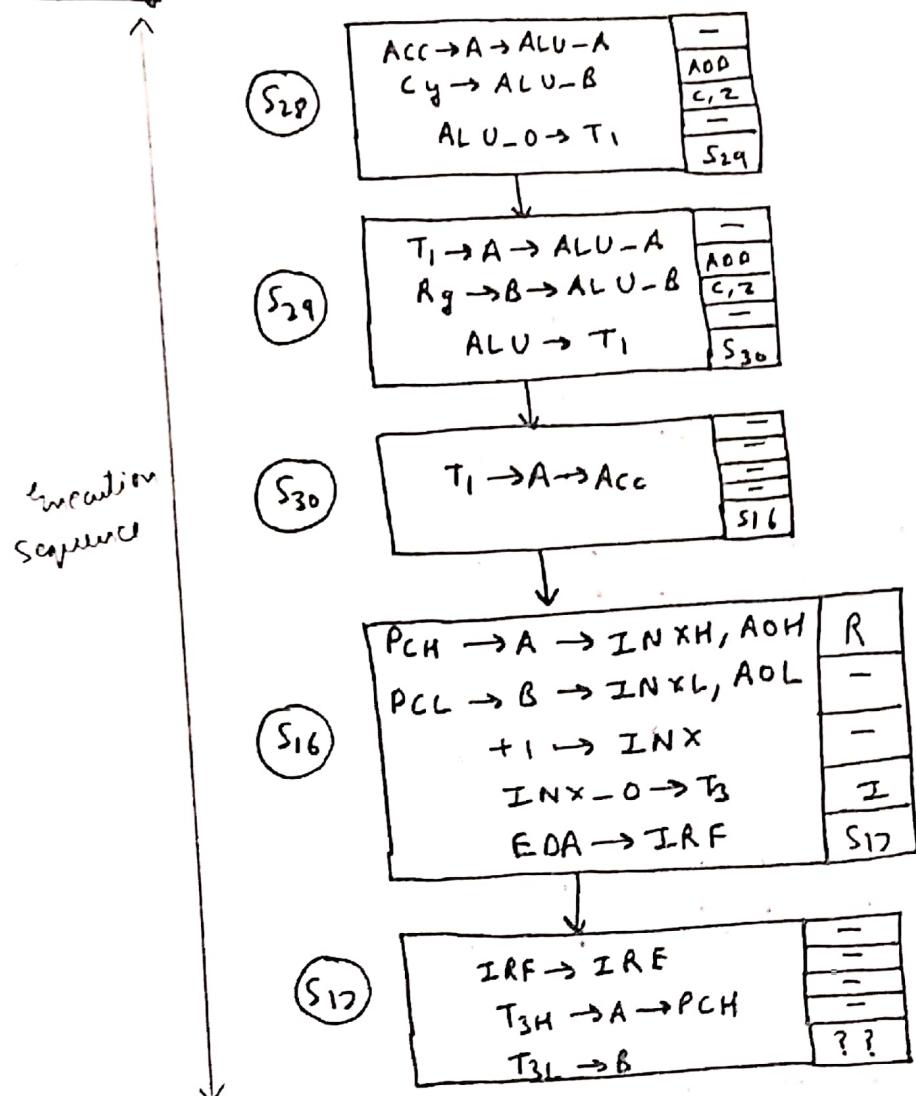
7) STA D16

Addressing
mode
sequence

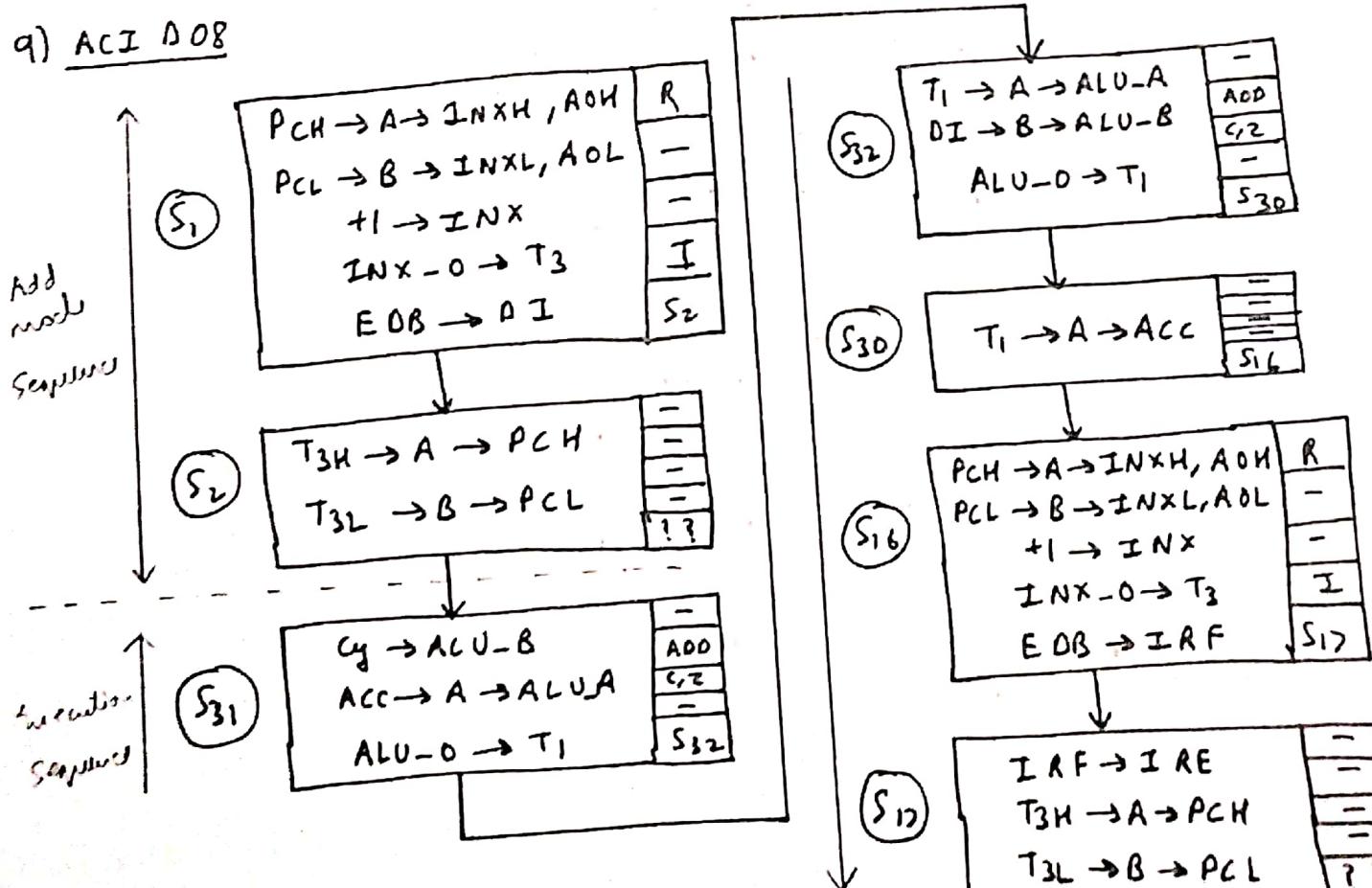


Execution
sequence

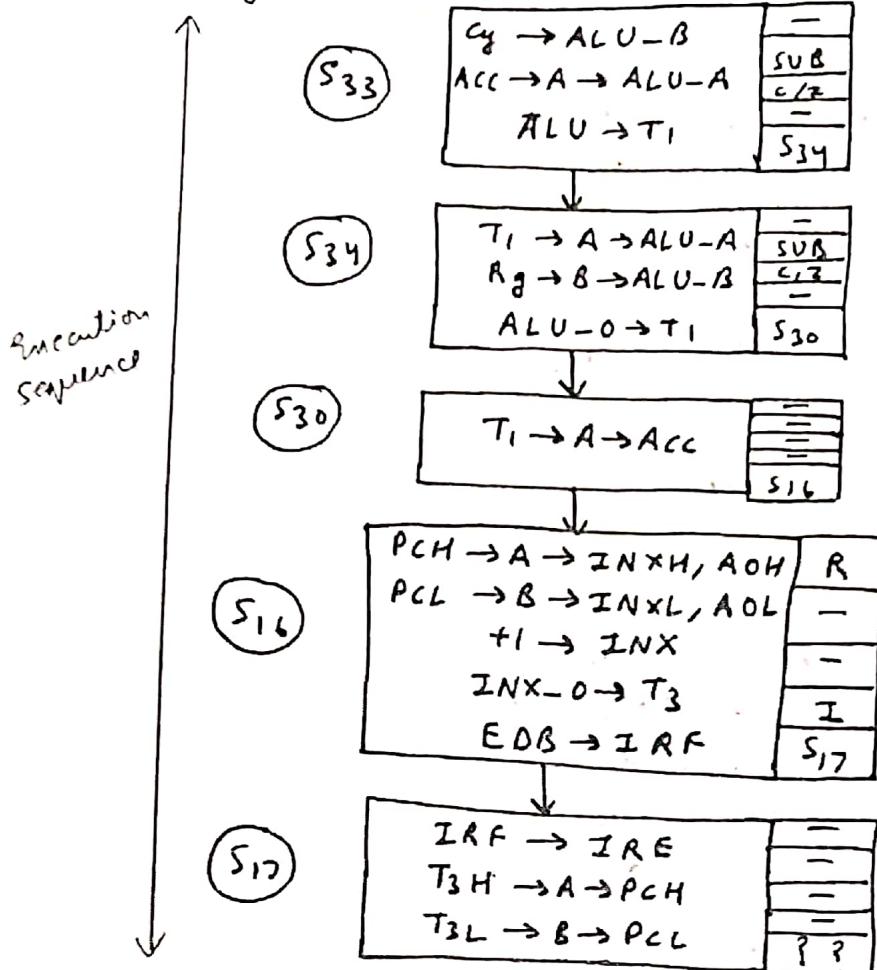
8) ADC Rg



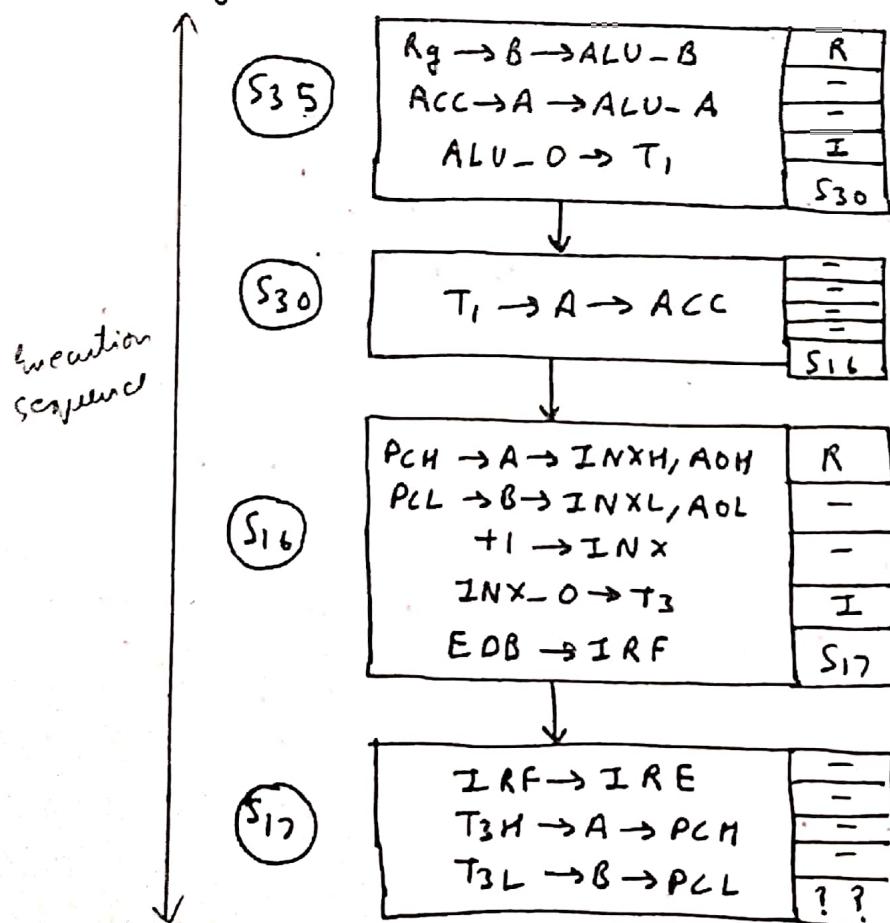
9) ACI D08



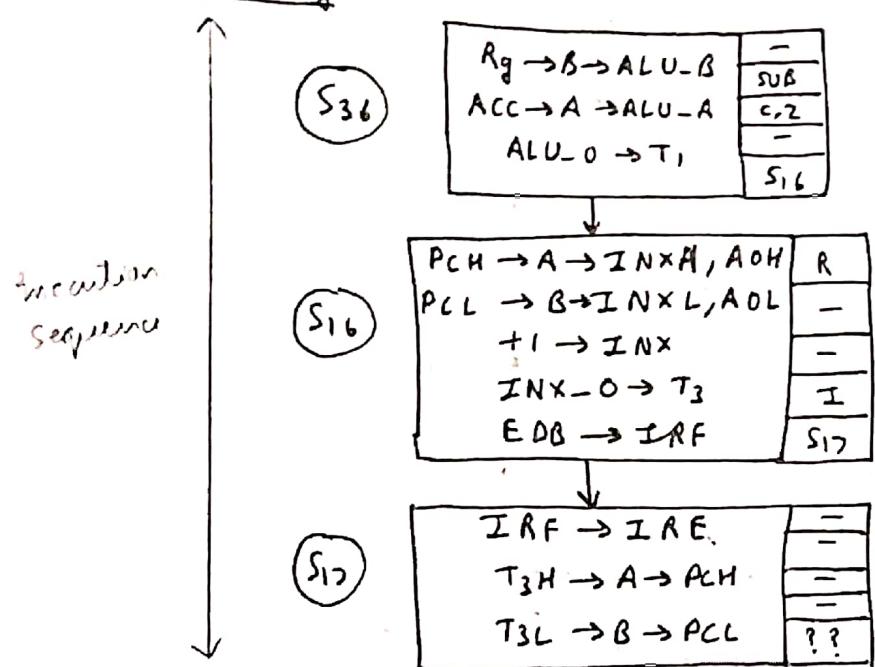
10)

SBB Rg

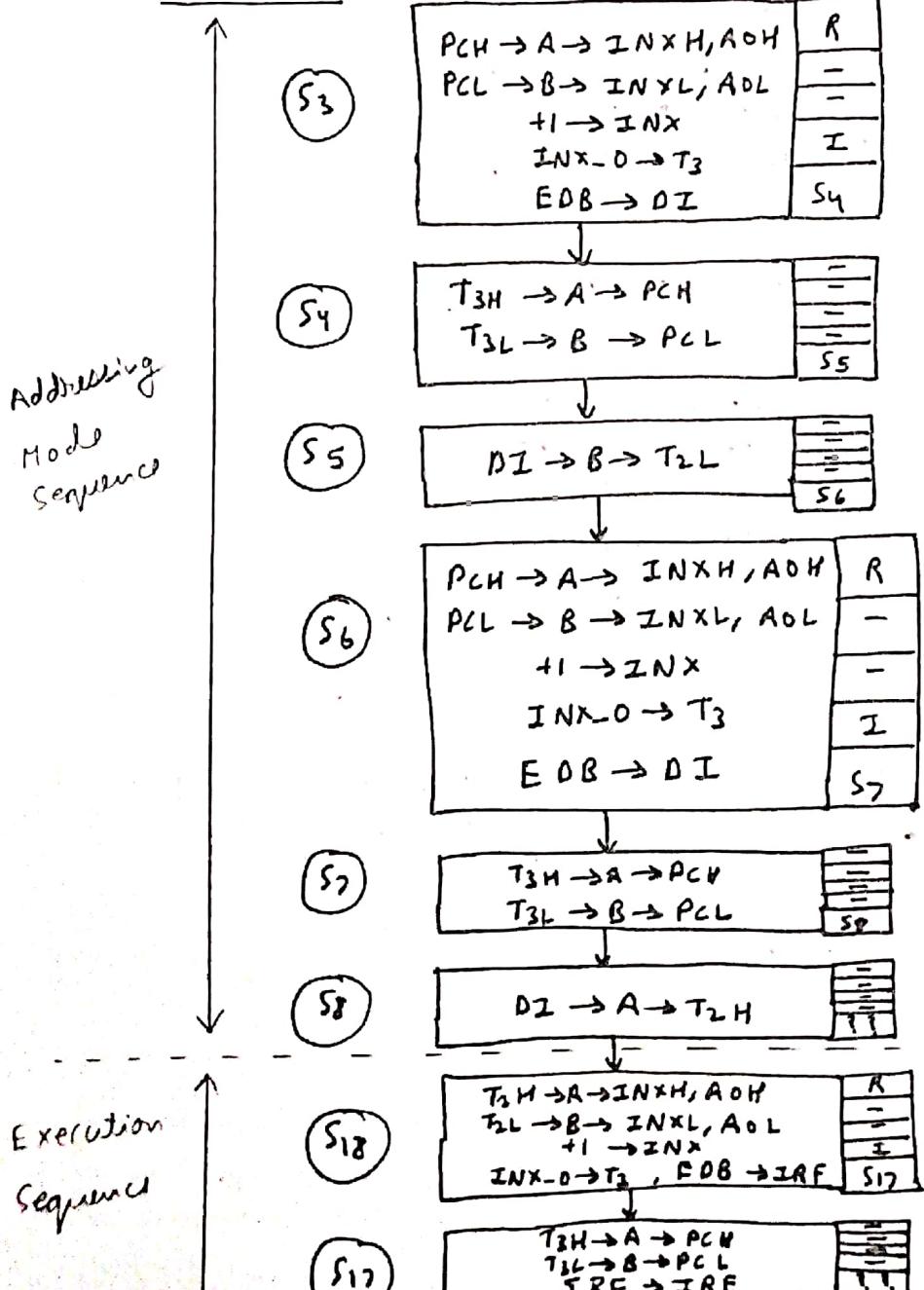
11)

ANA Rg

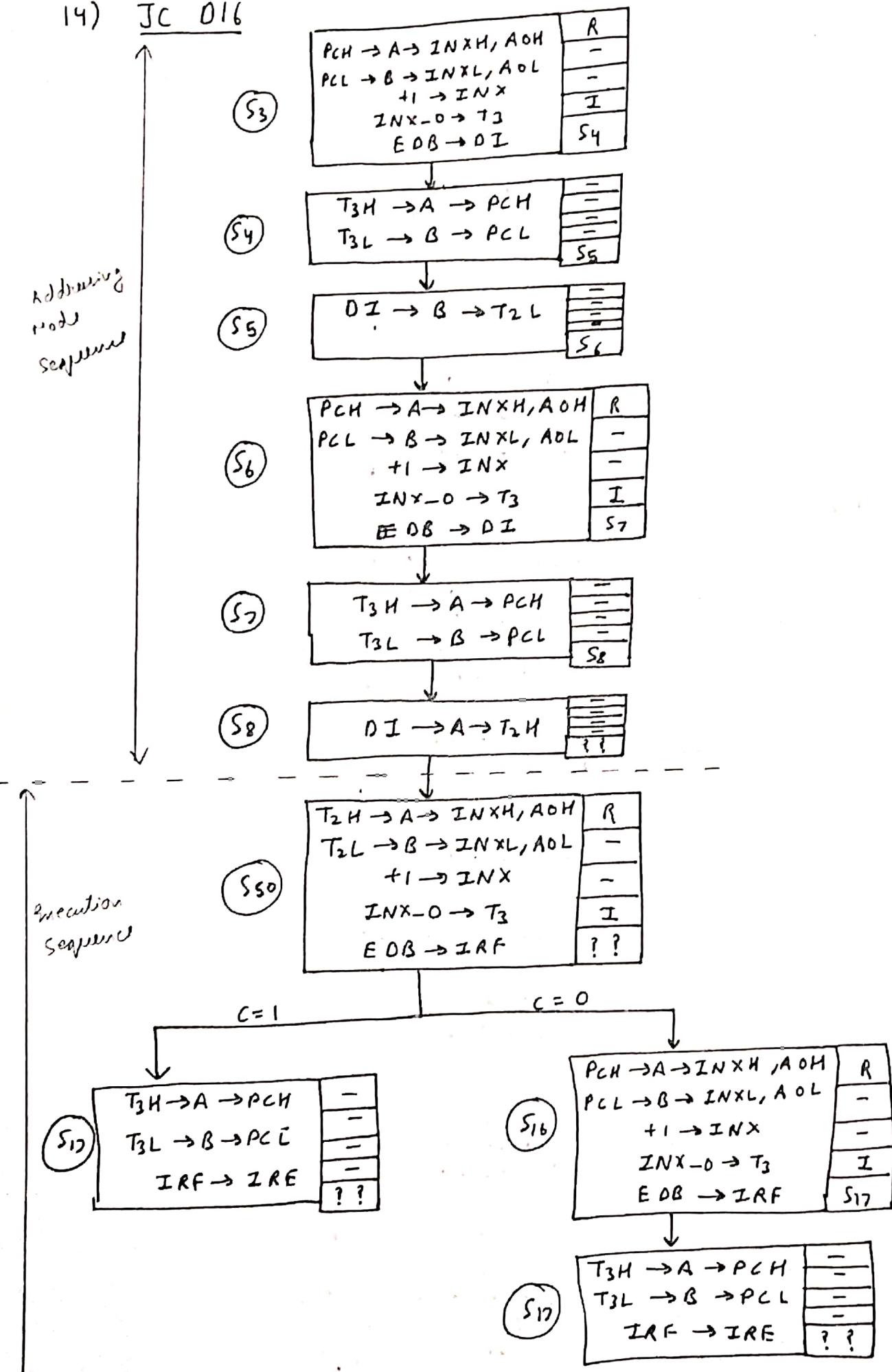
12) CMP Rg



13) JMP DI6

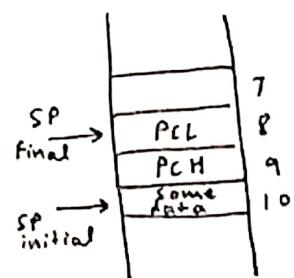
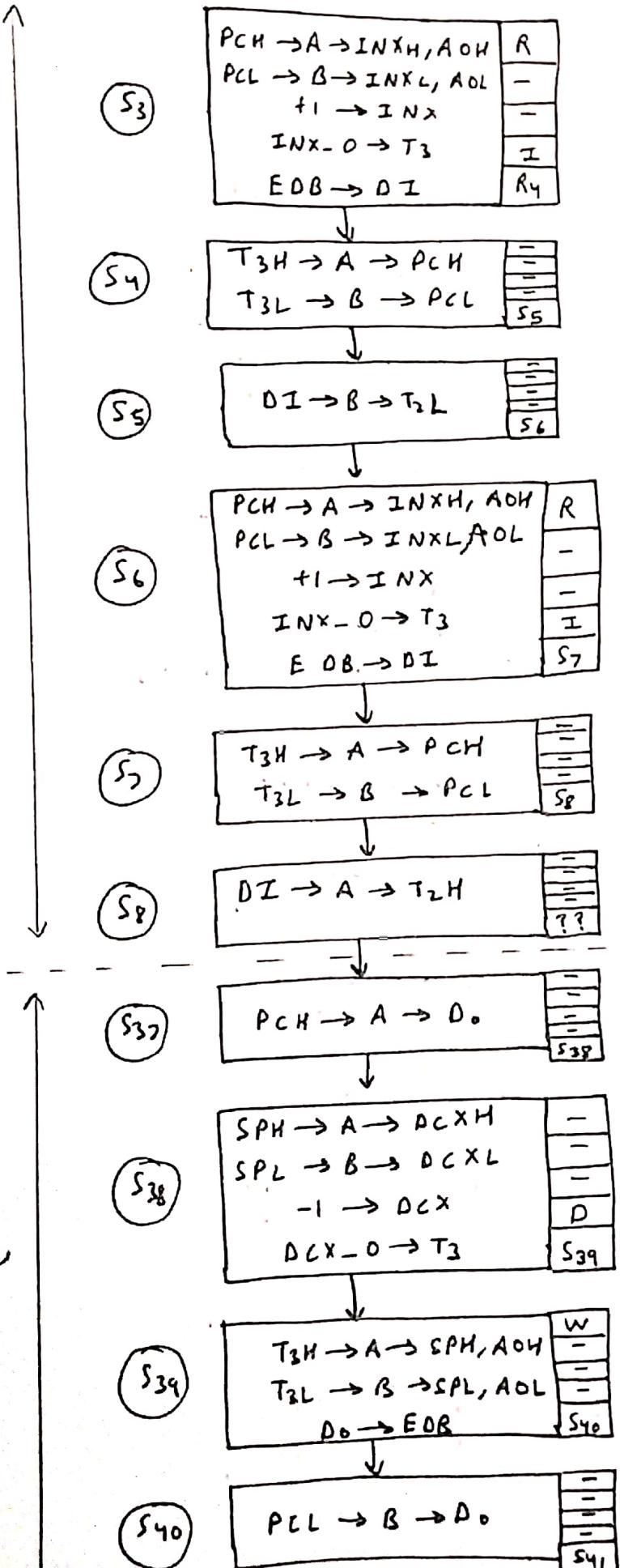


14) JC 016

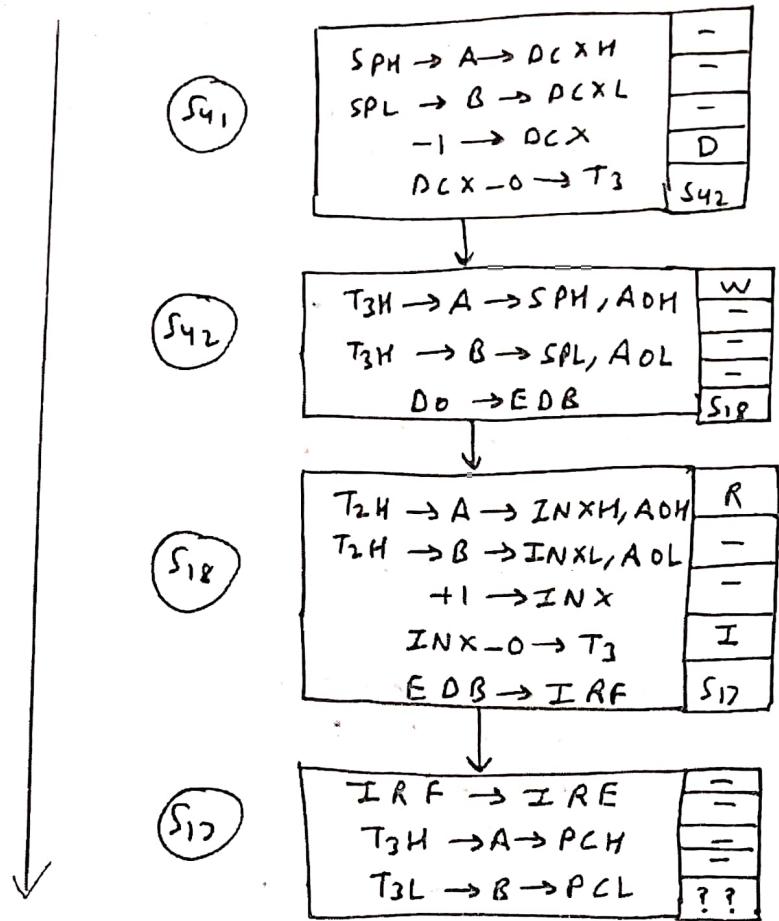


15) Call 016

Addressing
Mode
Sequence

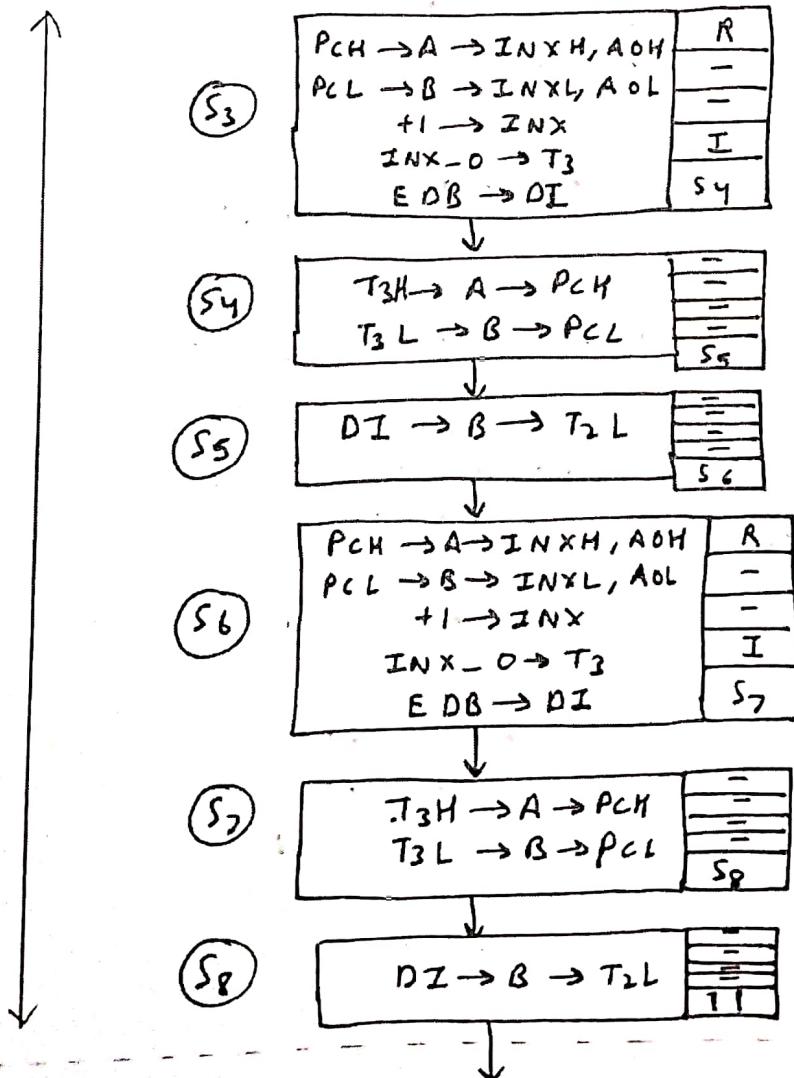


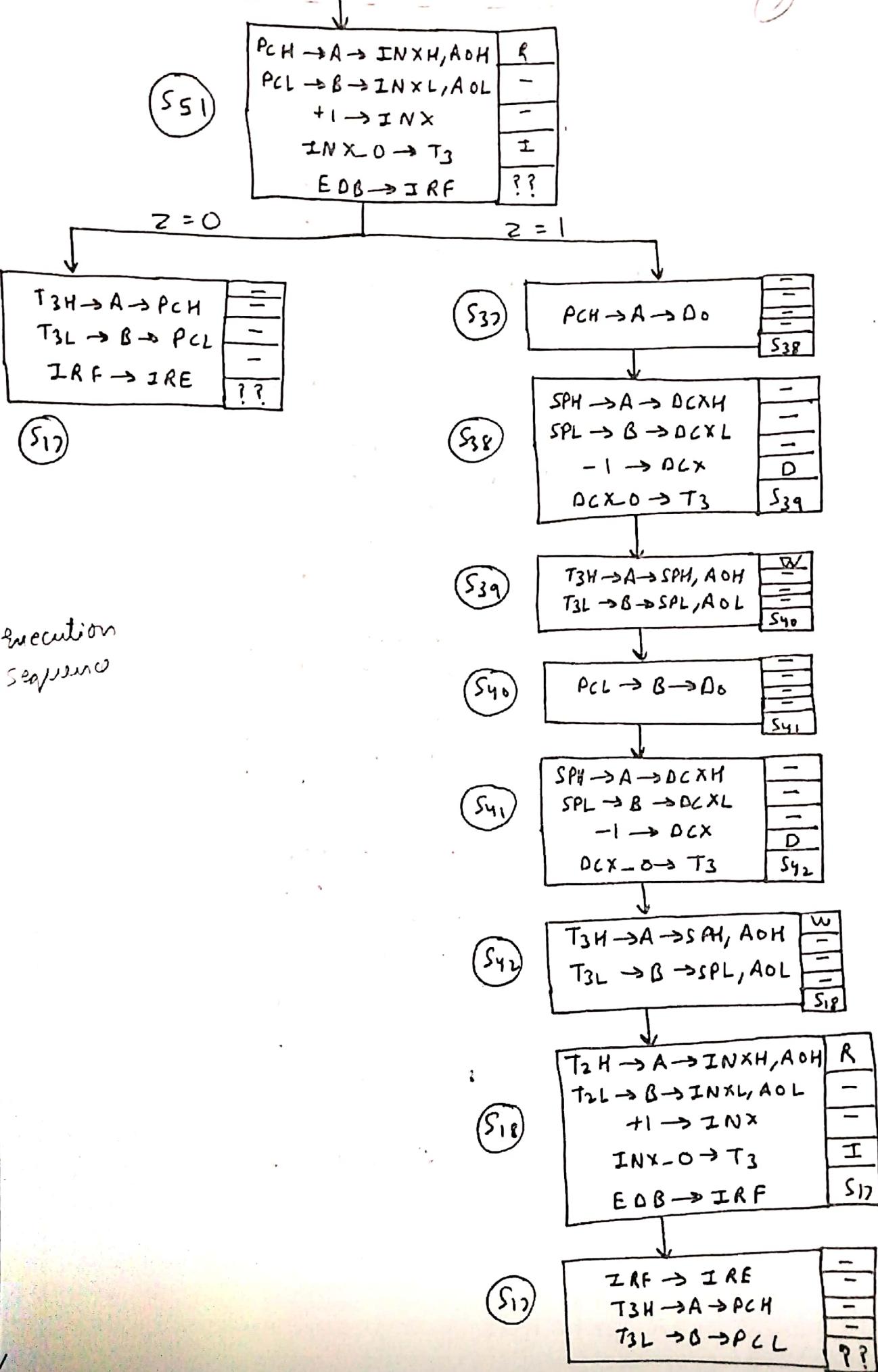
Execution
Sequence



(b) C2 D16

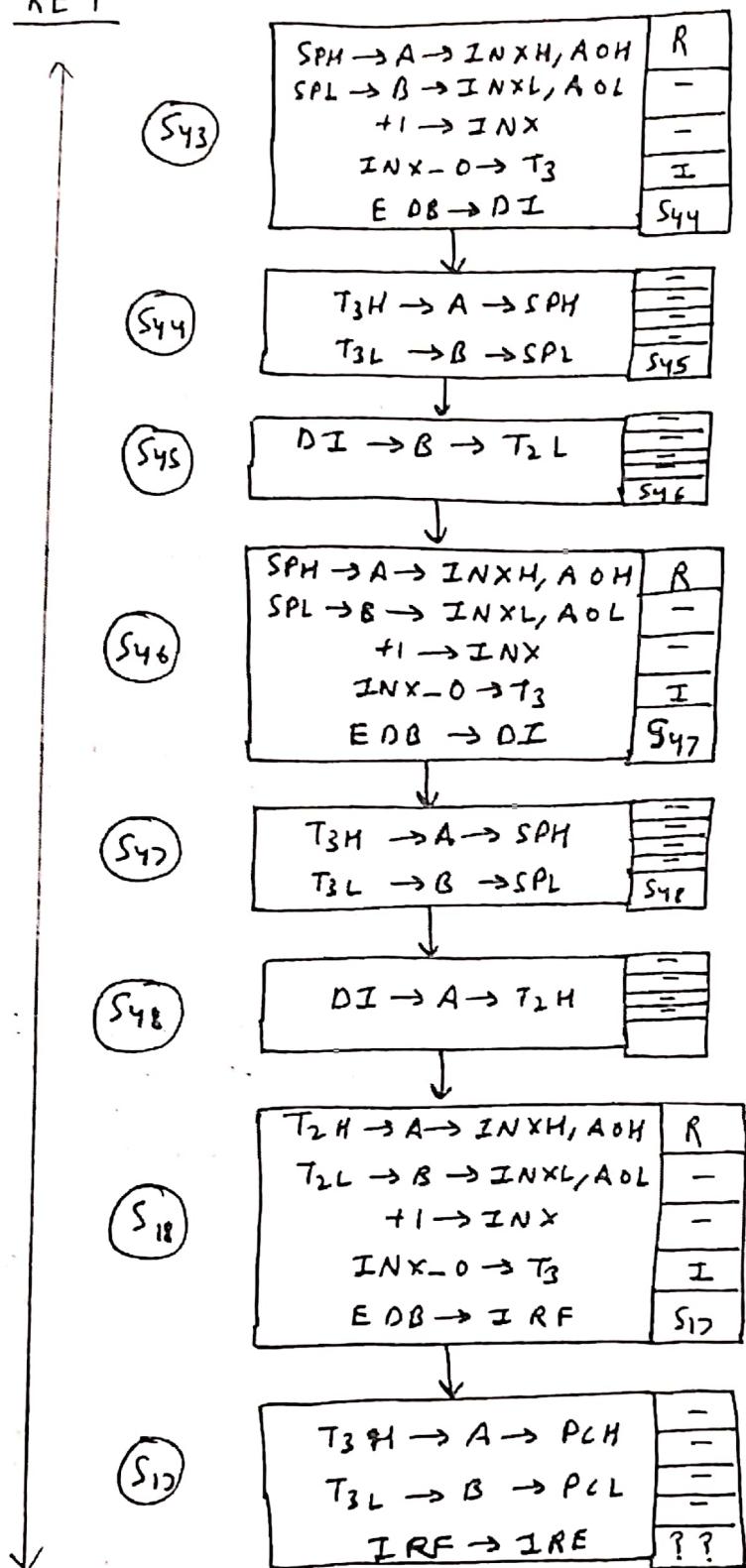
Addressing
Mode
Sequence



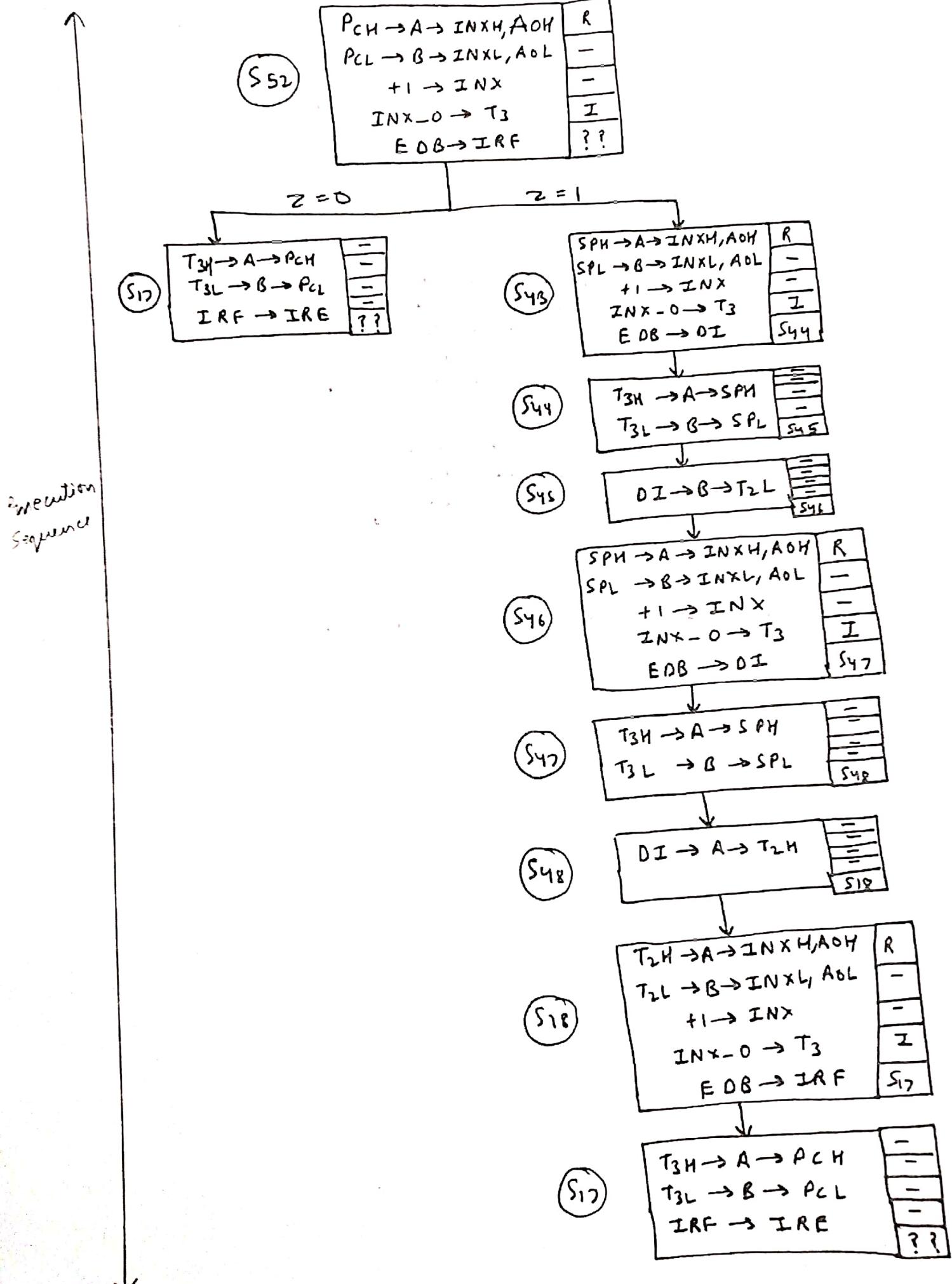


17) RET

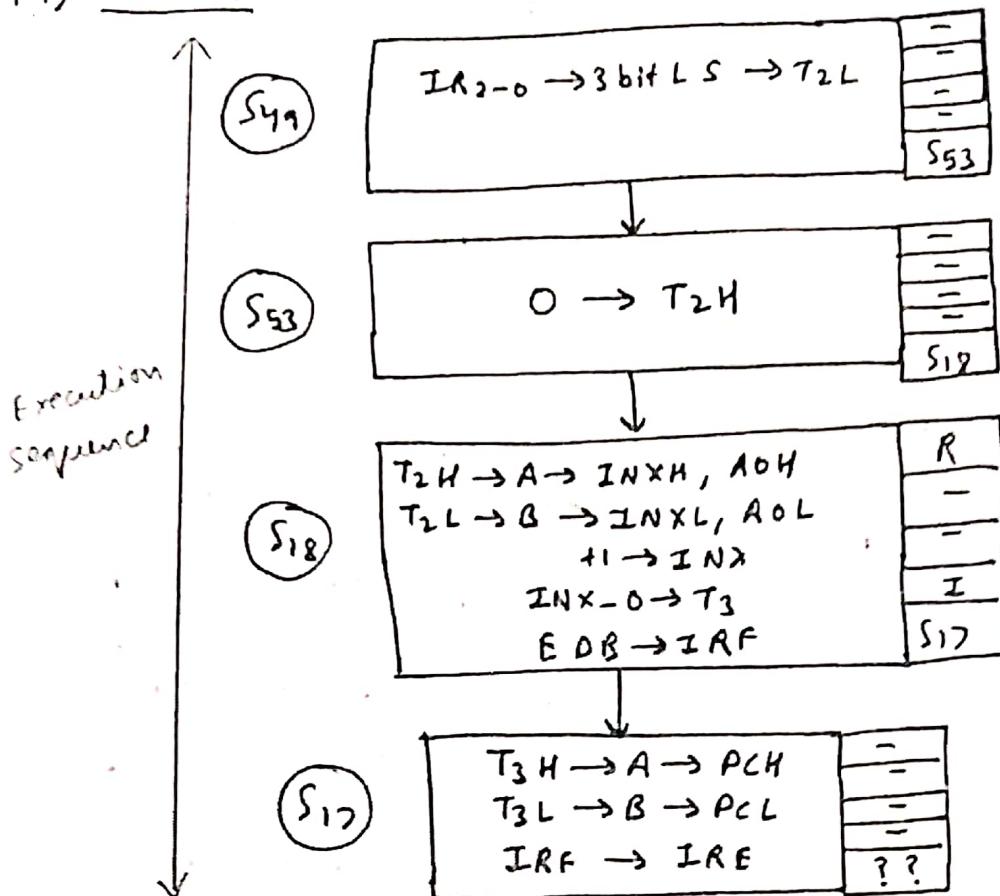
Execution sequence



18) R2



19) RSTn



Note

There are 4 different types of addressing mode sequences.

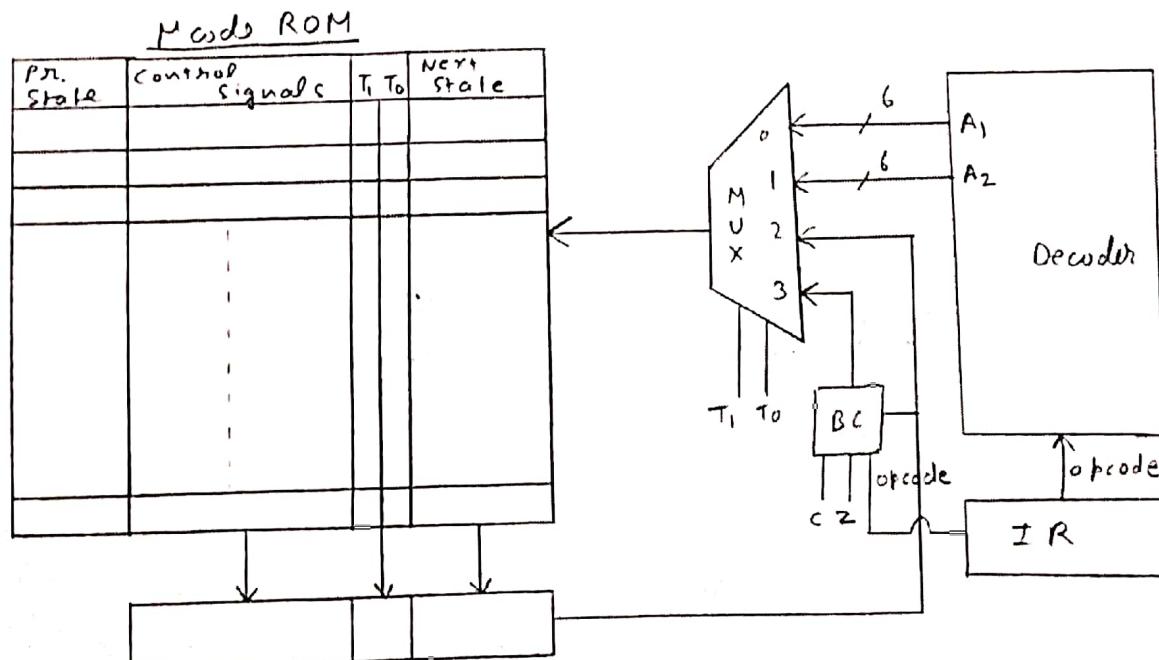
→ Micro ROM & Decoder (for selecting Next state)
The below table shows the output of ROM.

The below table lists the values of all opcodes for the 19 signals and also the value of A_1 & A_2 .

Instruction	opcode	A1 (inst. branch)	A2 (sequence branch)
MOV Rg,Rg	00XXX	010011	010011
MOV Rg,M	01000	000000	010100
MOV M,Rg	01001	010101	010101
MVI Rg,008	01010	000001	010100
LXI RP/SP, D16	01011	000011	010111
LOD D16	01100	001001	011001
STA D16	01101	000011	011010
AOC Rg	01110	011100	011100
ACI 008	01111	000001	011111
SBB Rg	10000	100001	100001
ANA Rg	10001	100011	100011
CMP Rg	10010	100100	100100
JMP D16	10011	000011	010010
JC D16	10100	000011	110010
CALL D16	10101	000011	100101
CZ D16	10110	000011	110011
RET	10111	101011	101011
RZ	11000	110100	110100
RSTn	11001	110001	110001

Note

Note: Instructions which doesn't have any Addressing mode such as
MOV Rg, Rg, ADC Rg, will have same A₁ & A₀



Note

Note: The branch control block (BC) will require opode as input so that it can decide whether to use C or Z as branch control.

Decoder logic calculation

Consider the opcode 5 bit as

A	B	C	D	E
---	---	---	---	---

$$A_{20} = \bar{A}\bar{B} + \bar{B}\bar{C}\bar{D} + \bar{B}\bar{D}\bar{E} + \bar{C}\bar{D}\bar{E} + \bar{A}\bar{D}\bar{E} + \bar{B}\bar{C}\bar{D} + \bar{A}\bar{C}\bar{D}\bar{E}$$

$$A_{21} = \bar{A}\bar{B} + \bar{D}\bar{E} + \bar{B}\bar{C}\bar{E} + \bar{B}\bar{C}\bar{E} + \bar{A}\bar{C}\bar{E}$$

$$A_{22} = \bar{B}\bar{D} + \bar{A}\bar{B}\bar{C} + \bar{B}\bar{C}\bar{E} + \bar{A}\bar{C}\bar{D}\bar{E} + \bar{A}\bar{C}\bar{D}\bar{E}$$

$$A_{23} = \bar{B}\bar{C} + \bar{A}\bar{C}\bar{D}\bar{E}$$

$$A_{24} = \bar{A} + \bar{B} + \bar{C}\bar{E} + \bar{C}\bar{D}\bar{E}$$

$$A_{25} = A\bar{D} + A\bar{E} + AC$$

$$A_{10} = E + \bar{B}\bar{D} + \bar{B}\bar{C} + C\bar{D} + \bar{A}\bar{C}\bar{D}$$

$$A_{11} = \bar{A}\bar{B} + \bar{B}\bar{E} + \bar{B}\bar{C} + \bar{C}\bar{D}\bar{E} + C\bar{D}\bar{E}$$

$$A_{12} = A\bar{B}\bar{E} + B\bar{C}\bar{D}\bar{E} + A\bar{C}\bar{D}\bar{E} + A\bar{B}\bar{C}\bar{D}\bar{E}$$

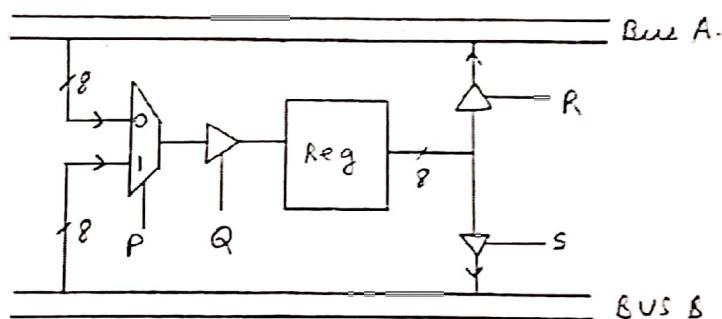
$$A_{13} = \bar{B}\bar{C}\bar{E} + A\bar{C}\bar{D}\bar{E}$$

$$A_{14} = \bar{A}\bar{B} + A\bar{B} + \bar{A}\bar{C}\bar{D}\bar{E} + \bar{A}\bar{C}\bar{D}\bar{E}$$

$$A_{15} = A\bar{C}\bar{D} + A\bar{C}\bar{E} + A\bar{C}\bar{D}\bar{E}$$

→ Internal Register Structures and Control signals for μ ROM

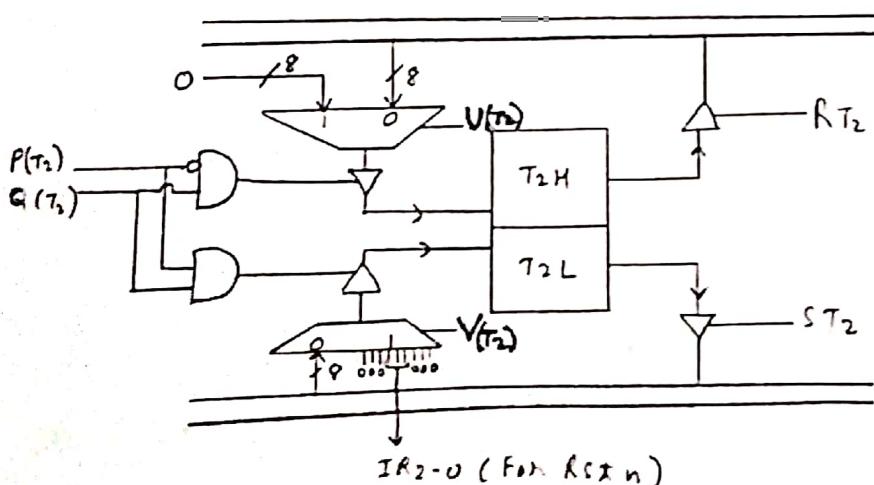
I) Registers B, C, D & E (8 bit)



Here P is P_B, P_C, P_D & P_E R is R_B, R_C, R_D & R_E

Q is Q_B, Q_C, Q_D & Q_E S is S_B, S_C, S_D & S_E

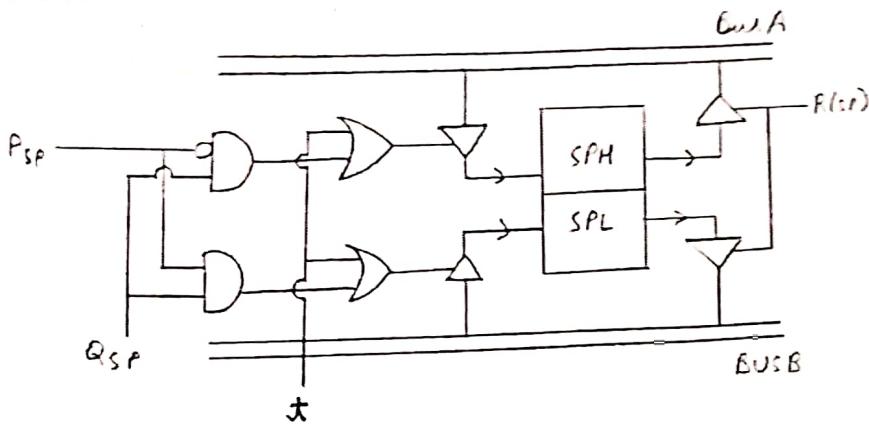
II) Register T2 (16 bit)



U = 1 during RSTn int. we load upper byte with all 0s and lower byte with vector address

V = 1 during RSTn. It used to load the vector address

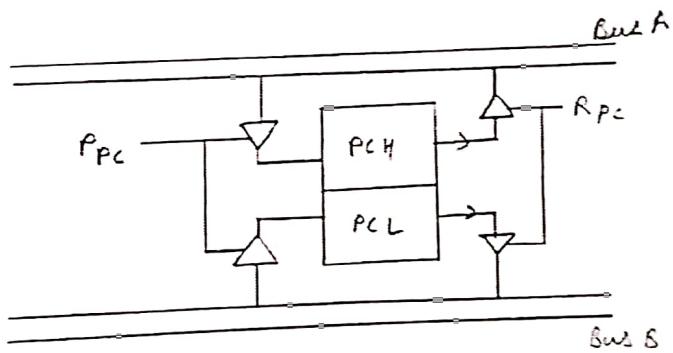
III) SP



In LXI RP/SP, D_16 , the upper and lower bytes of SP is written in different states. At that state we use P_{SP} & Q_{SP} and put $t=0$

In remaining instructions, both the bytes of the SP is written in the same state. At that state, we put $t=1$ and $P_{SP} = X$, $Q_{SP} = X$

IV) PC

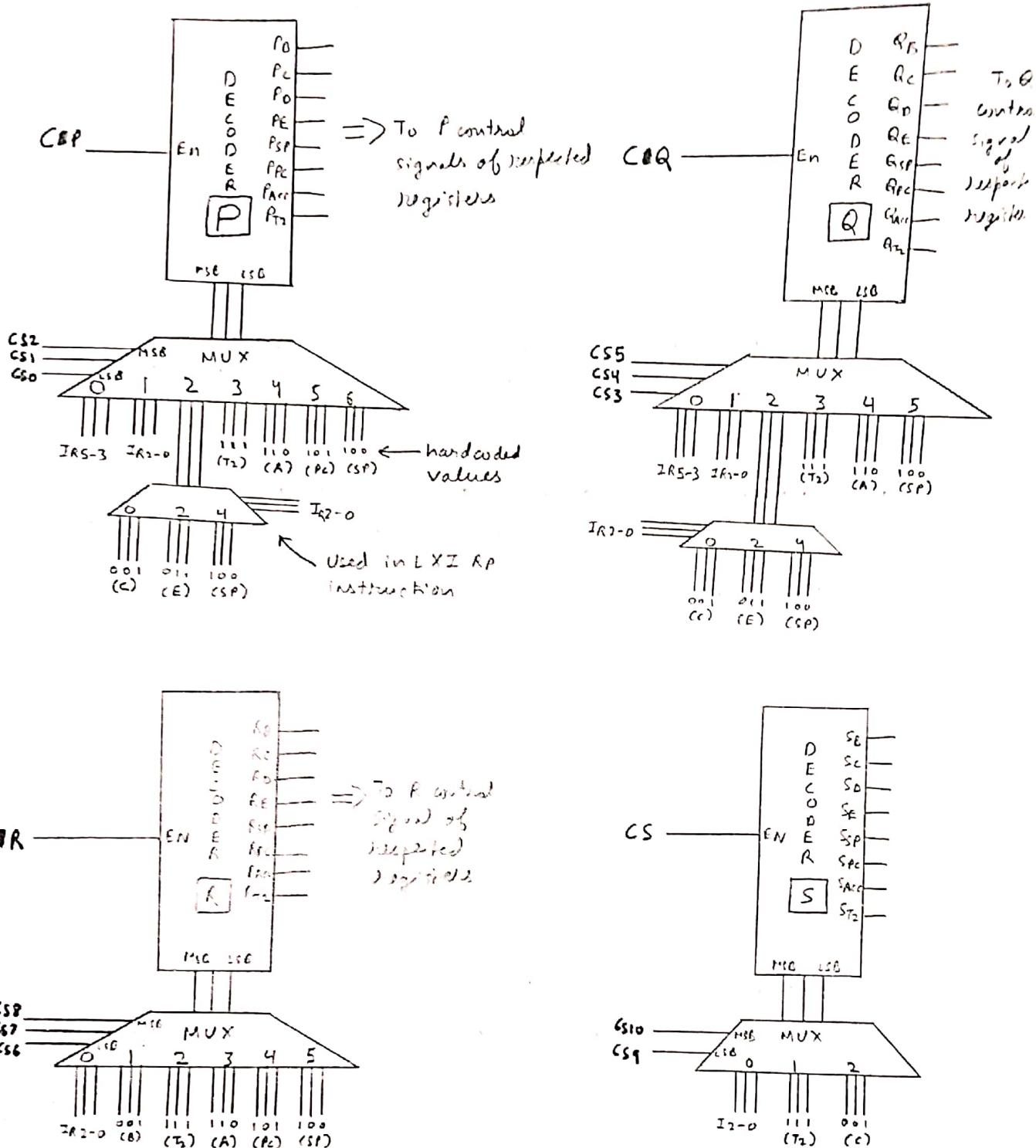


~~All~~ Both the bytes of PC are written in a single cycle state. Hence there is no need for Q_{PC} & R_{SPC} .

Note

Upto now the only control signals are t (from SP) & U & V (from T_2). The P, Q, R & S signals of all reg will be generated by register decoder circuit discussed next.

→ Register Decoder Circuit



NOTE

In decoder Q, Decoder R & Decoder S, some outputs will never occur.
Decoder Truth Table (For reference)

000 → B
000 → C
010 → D
011 → E
100 → SP
101 → PC
110 → Acc
111 → T ₂

So from the register decoder circuit, the control signals that are obtained are:-

CP, CS₂, CS₁, CS₀, CA, CS₅, CS₄, CS₃, CR, CS₈, CS₇, CS₆, CS, CS₁₀, CS₉

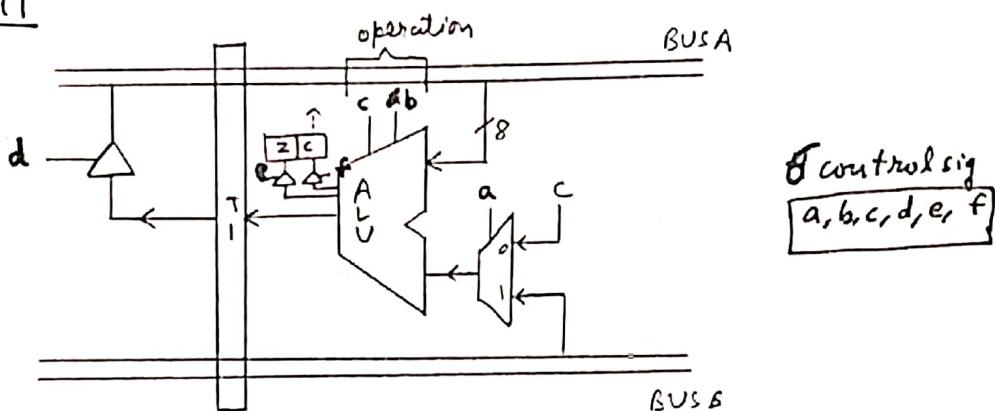
And the control sig from the earlier reg circuits are:-

t, u, v

So that makes 18 control signals up till now for controlling registers T₂, SP, PC, B, C, D, E & Acc.

→ Control signals for ALU, INX/DCX, DI, Do, IRF & IRE

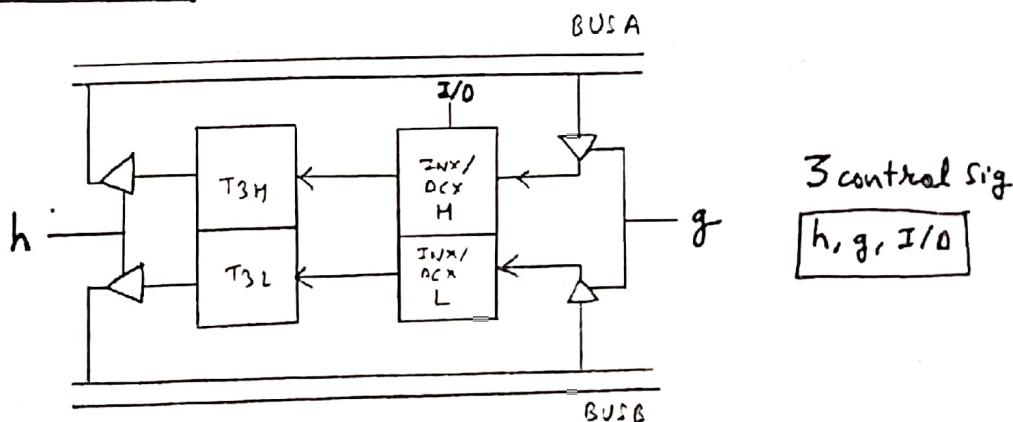
I) ALU & T₁



The operations are:-

c	b	operation
0	0	add
0	1	sub
1	0	ANA
1	1	-

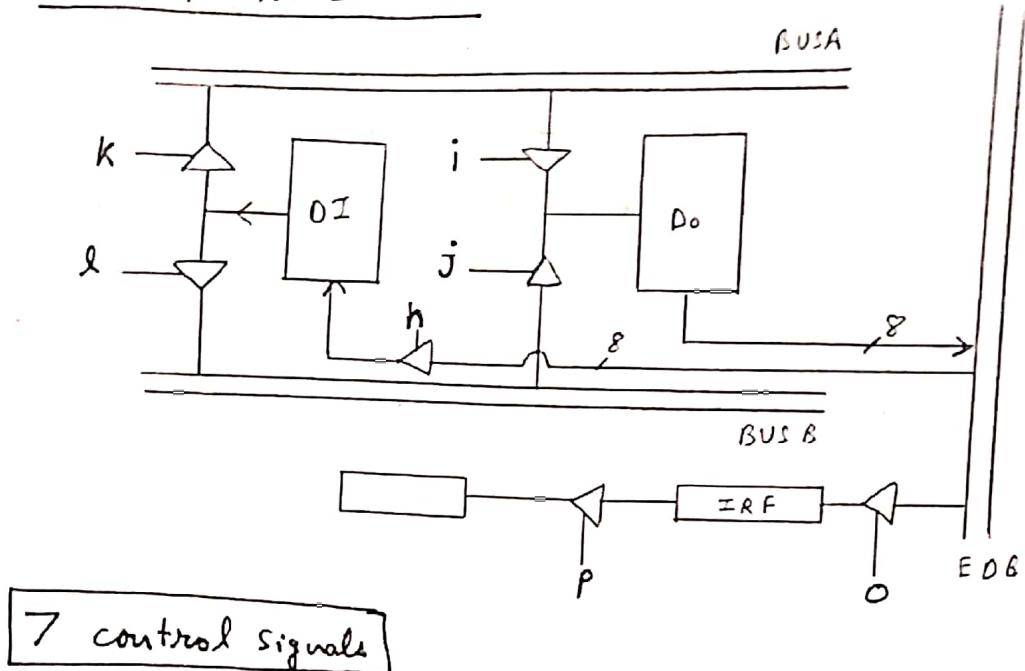
II) IND/DCX & T₃



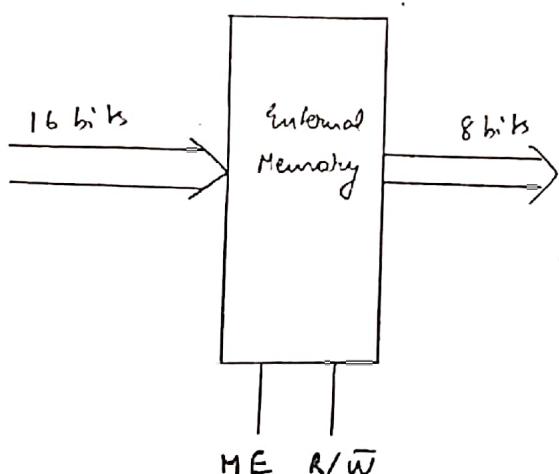
I/O = 1 → increment

I/O = 0 → decrement

III) DI, DO, I_{RE} & I_{RF}



IV) Internal Memory



2 control signals.

So in total there are $18 + 6 + 3 + 7 + 2 = 36$ control signals without encoding of control signals.

All the 36 control signals are shown for different states.

→ Encoding of control signals to reduce the no. of control signals

I) Reducing Control signals in ALU

a x	b x	c x	d 0	e 0	f 0	ALU1	ALU2	ALU3	ALU4
0	0	0	0	1	1	0	0	0	0
1	0	0	1	1	1	0	0	1	0
x	x	x	1	0	0	0	0	1	1
1	0	0	0	1	1	0	1	0	0
0	1	0	0	1	1	0	1	0	1
1	1	0	1	1	1	0	1	1	0
1	0	1	0	1	0	0	1	1	1
1	1	0	0	1	1	1	0	0	0

$$a = \overline{ALU4} + ALU3$$

$$b = ALU1 + (ALU2 \cdot \overline{ALU3} + ALU4) + (ALU2 \cdot ALU3 \cdot \overline{ALU4})$$

$$c = ALU3 \cdot ALU4$$

$$d = \overline{ALU2} \cdot ALU3 + ALU3 \cdot \overline{ALU4}$$

$$e = ALU2 + ALU1 + \overline{ALU3} \cdot ALU4 + ALU3 \cdot \overline{ALU4}$$

$$f = ALU1 + \overline{ALU3} \cdot ALU4 + ALU3 \cdot \overline{ALU4} + ALU2 \cdot \overline{ALU3}$$

8 control signals $\xrightarrow{\text{Reduced to}}$ 4 control signals

II) INX/DCX + T3

I/D	g	h	IOT1	IOT2
x	0	0	0	0
0	1	0	0	1
x	0	1	1	0
1	1	0	1	1

$$I/D = IOT1$$

$$g = IOT2$$

$$h = IOT1 \cdot \overline{IOT2}$$

3 control signals $\xrightarrow{\text{Reduced to}}$ 2 control signals

III) D₀ / D₁

i	j	k	l	n	D ₀₁	D ₀₂	D ₀₃
0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	1	0	0	0	0	1	1
1	0	0	0	0	1	0	0
0	0	1	0	0	1	0	1

$$i = D_{01} \cdot \overline{D_{03}}$$

$$j = D_{02} \cdot D_{03}$$

$$k = D_{01} \cdot D_{03}$$

$$l = D_{02} \cdot \overline{D_{03}}$$

$$n = \overline{D_{01}} \cdot \overline{D_{02}} \cdot \overline{D_{03}}$$

so 5 control signals $\xrightarrow[\text{to}]{\text{Reduced}}$ 3 control signals

II) Decoder [P] related control signals

CP	CS ₂	CS ₁	CS ₀	PP ₁	PP ₂
0	x	x	x	0	0
1	1	0	1	0	1
1	0	1	1	1	0
1	0	1	0	1	1

$$CP = PP_1 + PP_2$$

$$CS_2 = \overline{PP_1}$$

$$CS_3 = PP_1$$

$$CS_4 = \overline{PP_1} + \overline{PP_2}$$

so 4 control signals $\xrightarrow[\text{to}]{\text{reduced}}$ 2 control signals

IV) Decoder [Q] related control signals

CQ	CS ₅	CS ₄	CS ₃	QQ ₁	QQ ₂	QQ ₃
0	x	x	x	0	0	0
1	0	1	1	0	0	1
1	0	0	0	0	1	0
1	0	0	1	0	1	1
1	0	1	0	1	0	0
1	1	0	0	1	0	1

$$CQ = Q Q_1 + Q Q_2 + Q Q_3$$

$$CS_5 = Q Q_1 \cdot Q Q_3$$

$$CS_4 = \overline{Q} Q_1 \cdot \overline{Q} Q_2 + \overline{Q} Q_2 \cdot \overline{Q} Q_3$$

$$CS_3 = \overline{Q} Q_1 \cdot Q Q_3$$

So 4 control signals $\xrightarrow[\text{to}]{\text{reduced}}$ 3 control signals.

Conclusion

Total no. of control signals reduced from 36 $\rightarrow \underline{\underline{28}}$