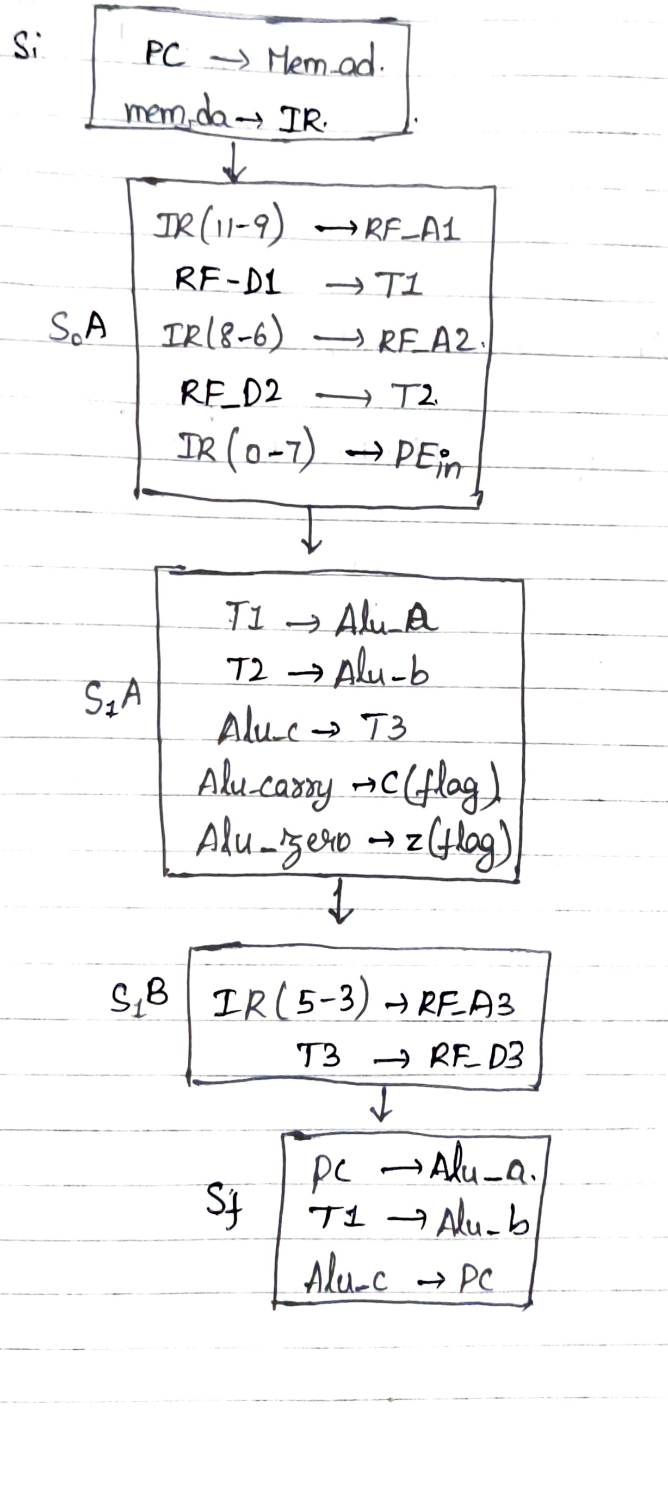
CS 230 Project

Datapaths

1) ADD: add r_c, r_a, r_b

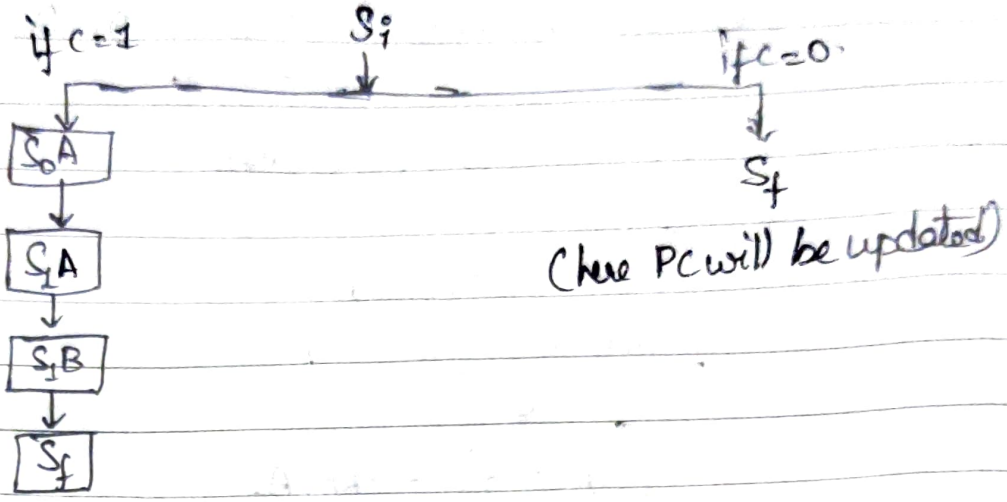
00_01	RA	RB	RC	0	0 0

$r_c = r_a + r_b$ & Cy, Z modifies.



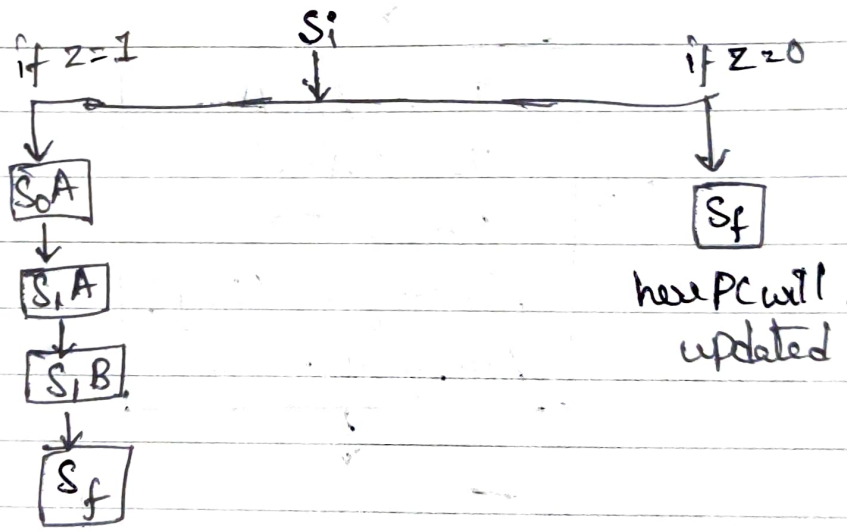
② ADC : adc r_c, r_a, r_b.

00-01	RA	RB	RC	0	10

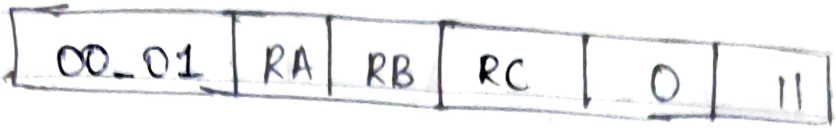


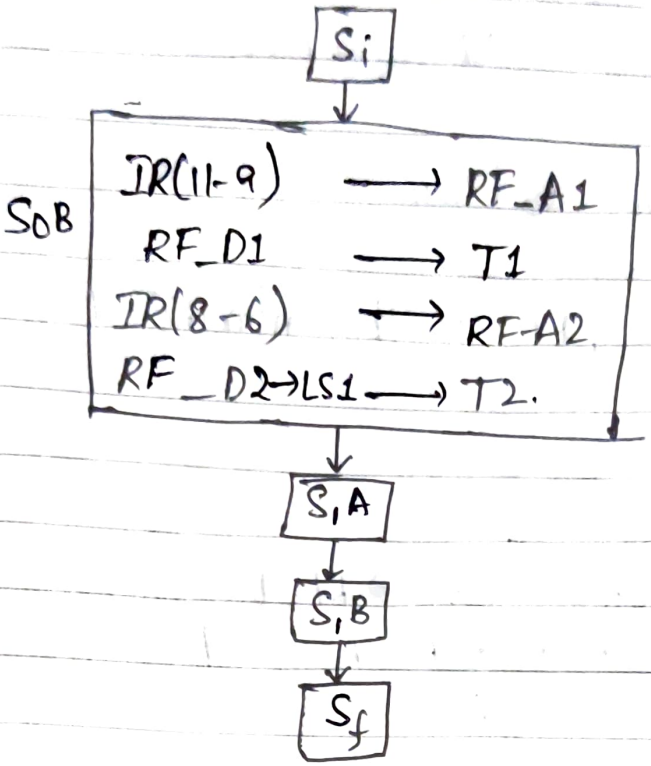
③ ADZ : adz r_c, r_a, r_b.

00_01	RA	RB	RC	0	0 1

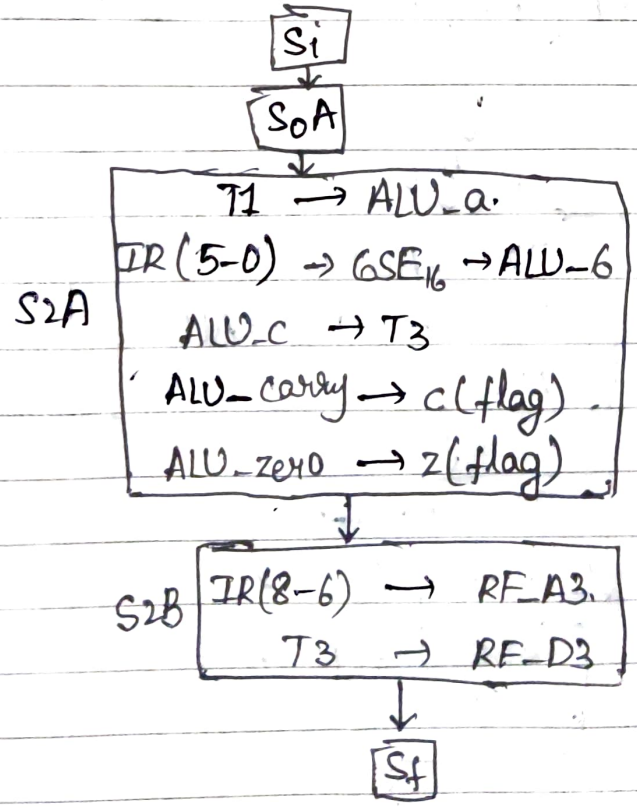


④ ADL : Adl r_c, r_a, r_b.





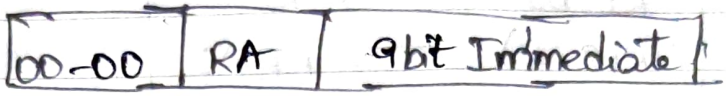
⑤ ADI : adi r_b, r_a, imm6

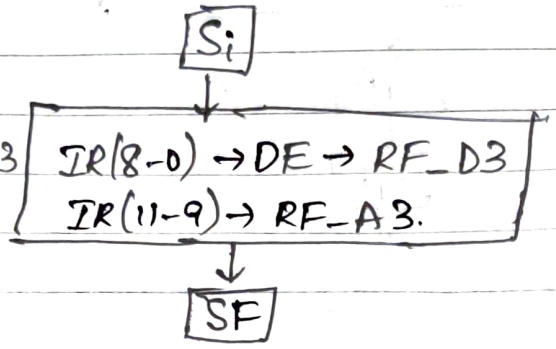


6) NDU : ndu r_c, r_a, r_b.

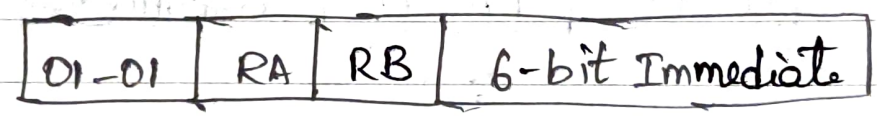
→ similiar as add instruction but the operation in ALU is nand which is decided by alu_control. similarly for NDC and NDZ

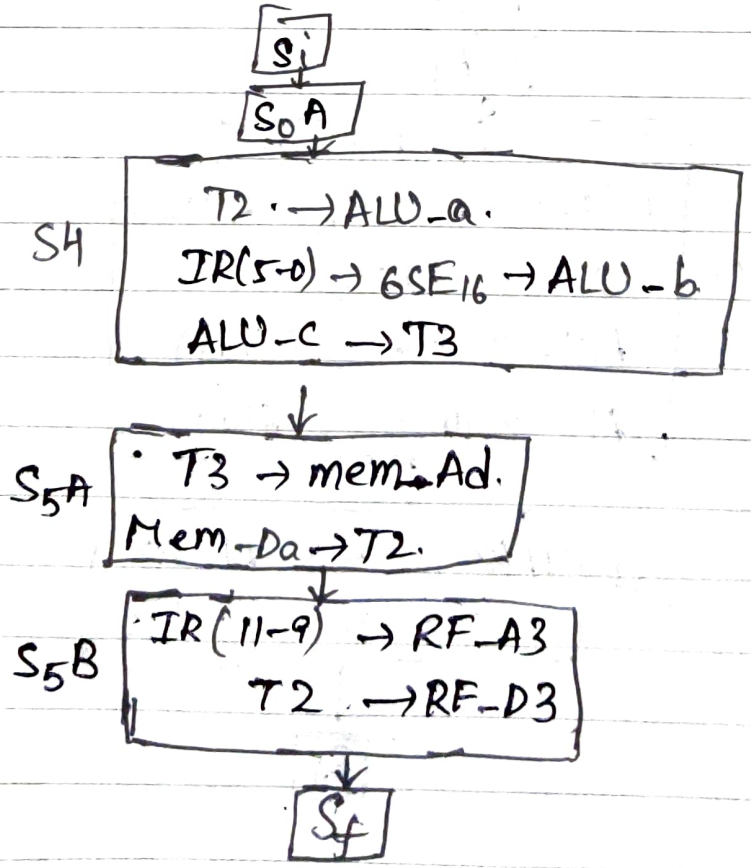
9) LHI : lhi r_a, Imm



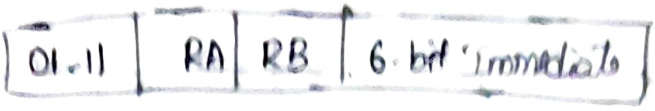


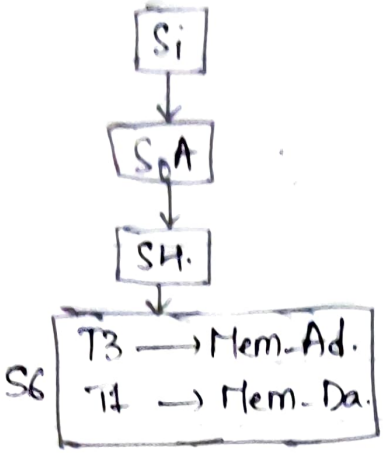
10) LW : Lw r_a, r_b, Imm



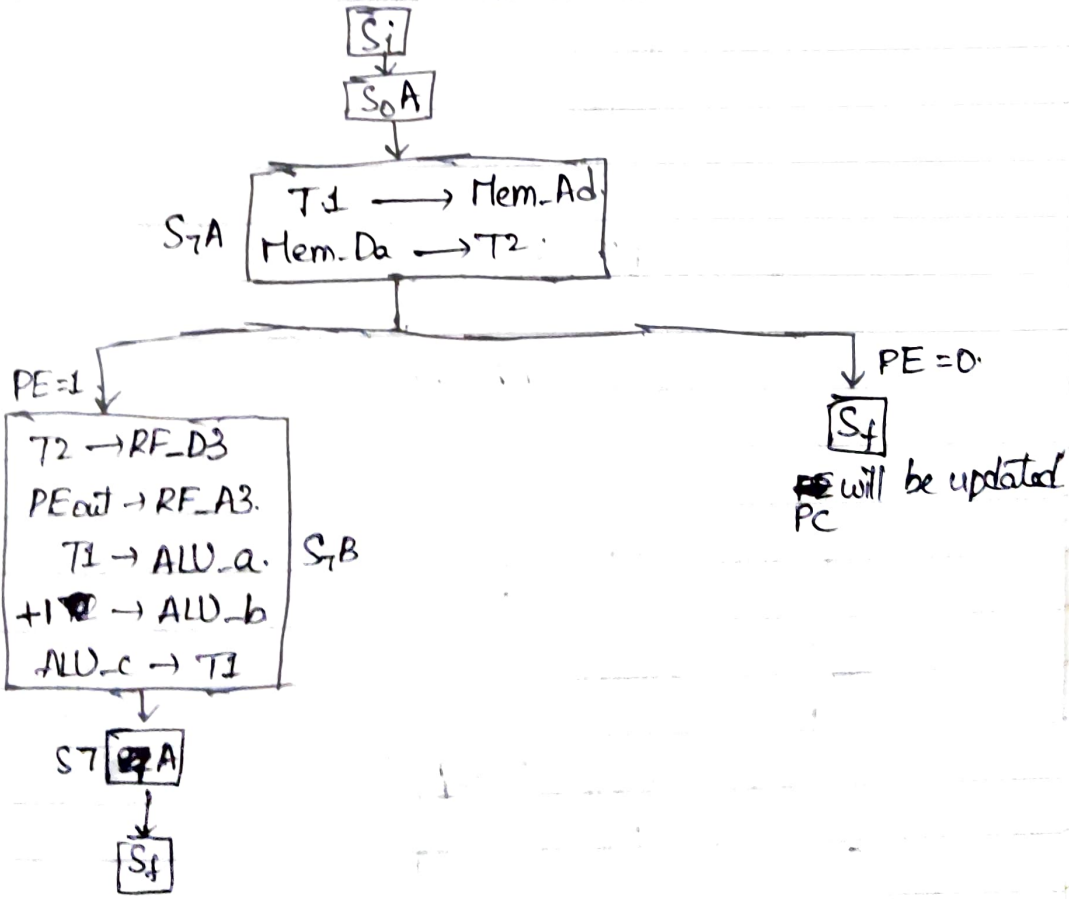


11) SW : Sw Ra, Rb, Imm.

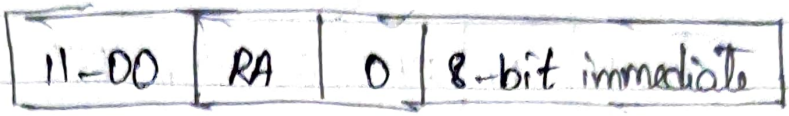


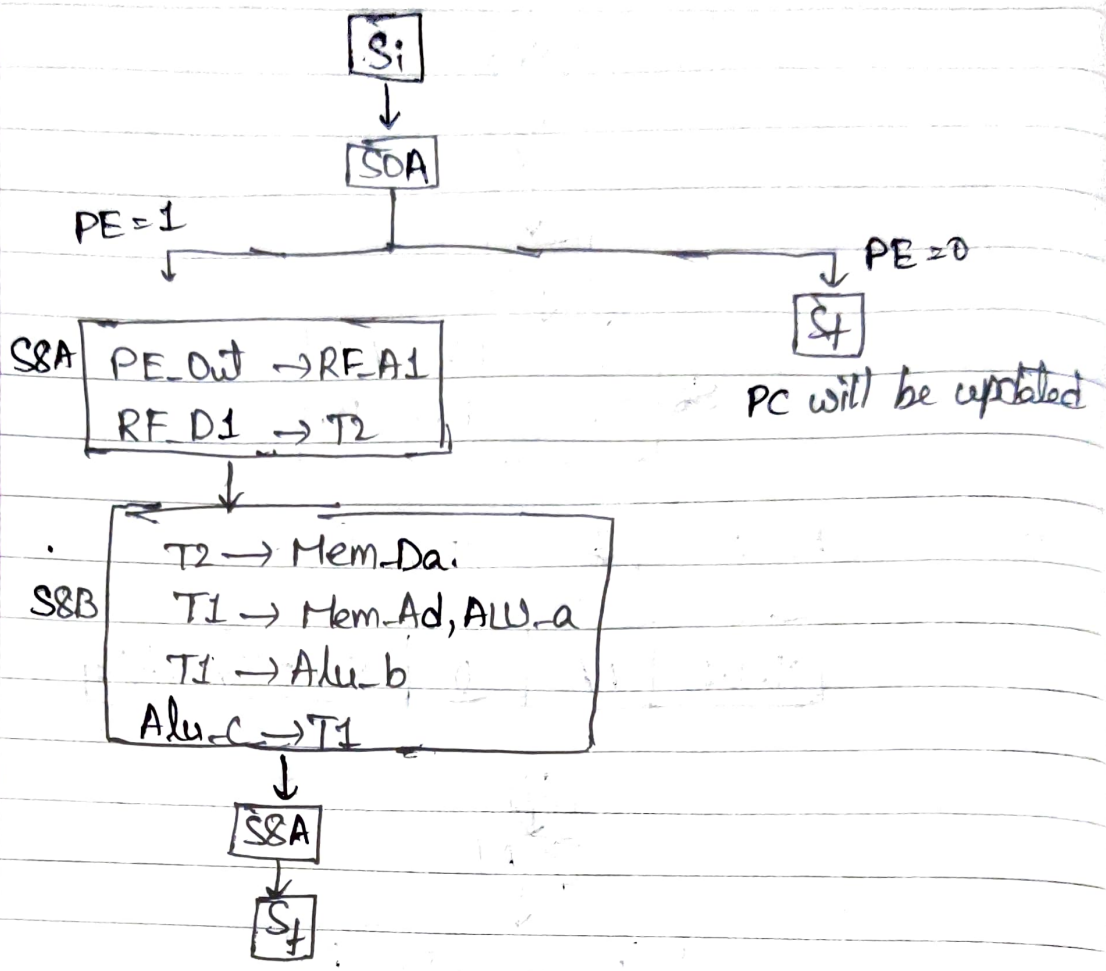


12) LM : L[illegible] Ra, Imm

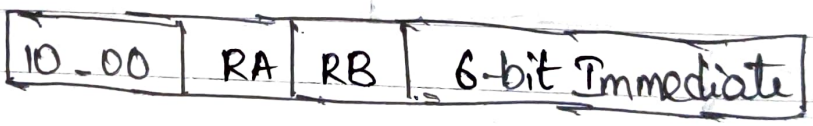


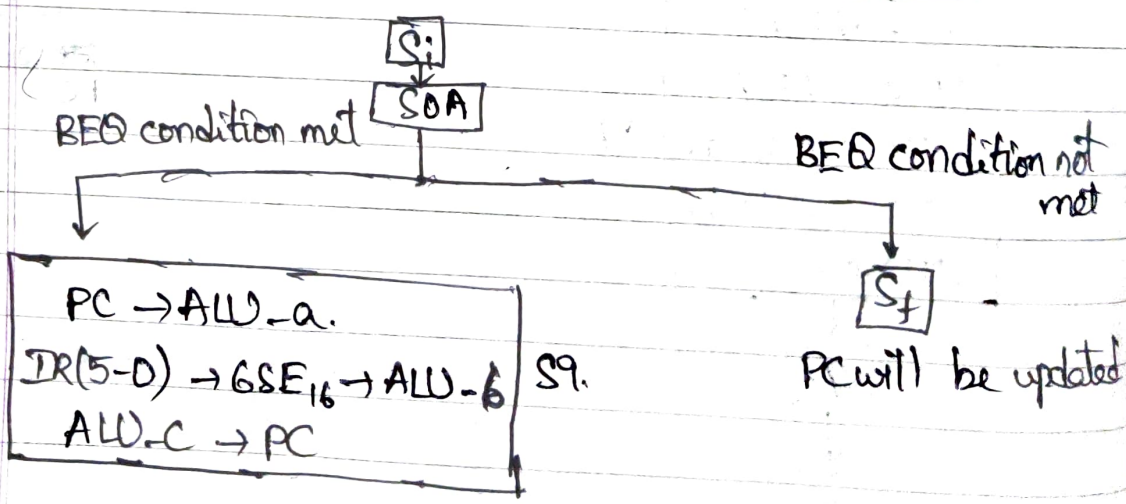
(13) SM : Sm @Ra, Imm



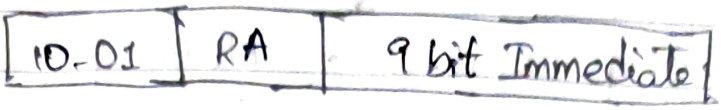


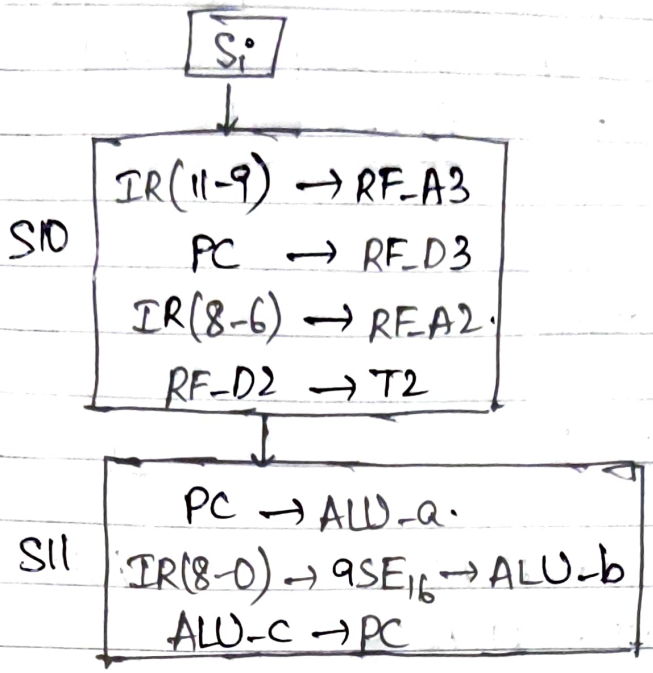
(14) BEQ : Beq Ra, Rb, Imm.



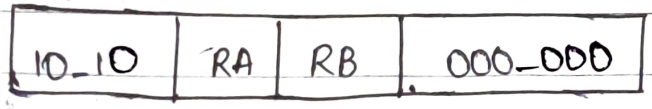


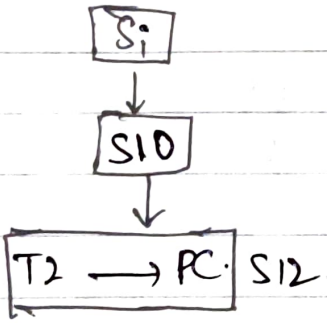
(15) JAL : jalr ra, Imm



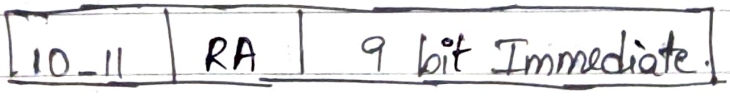


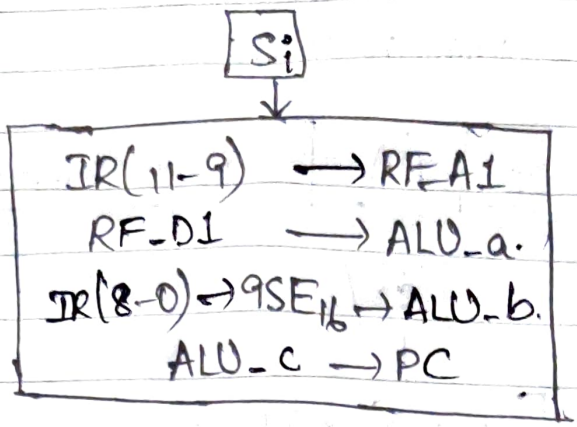
(16) JLR: jalr ra, rb





(17) JRI : jri ra, Imm





Connections :-

RF_A1 : (M1)

IR(11-9) → RF_A1 0
PE_Out → RF_A1 1

RF_A2 :

IR(8-6) → RF_A2.

RF_A3 : (M2)

IR(5-3) → RF_A3 00
IR(8-6) → RF_A3 01
IR(11-9) → RF_A3 10
PE_out → RF_A3 11

RF_D3 : (M3)

T3 → RF_D3 00
IR(8-0) → DE → RF_D3 01
T2 → RF_D3 10
PC → RF_D3 11

PC : (M4)

ALU_C → PC 0
T2 → PC 1

T1 : (M5)

RF_D1 → T1 0
ALU_C → T1 1

T2 : (M6)

RF_D2 → T2 00
RF_D2 → LS1 → T2 01
RF_D1 → T2 10
Mem_Da → T2 11

T3 : (M7)

Mem_Da → T3 0
ALU_C → T3 1

ALU_a : (M8)

T1 → ALU_a 00
PC → ALU_a 01
T2 → ALU_a 10
RF_D1 → ALU_a 11

ALU_b: (M9)

T2 → ALU_b 00
T1 → ALU_b 01
IR(5-0) → 6SE16 → ALU_b 10
IR(8-0) → 9SE16 → ALU_b 11

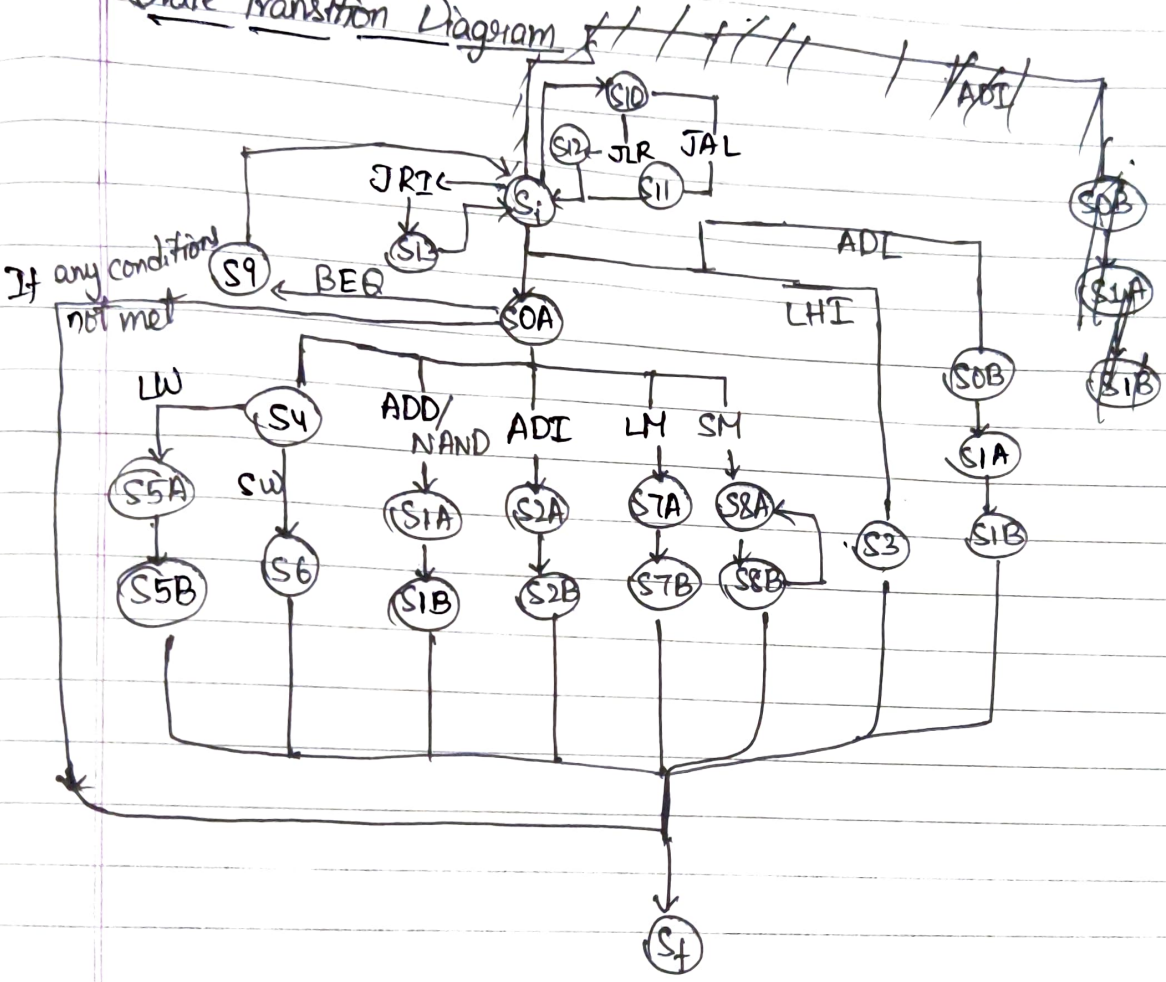
Mem_Da (M10)

T1 → Mem_Da 0
T2 → Mem_Da 1

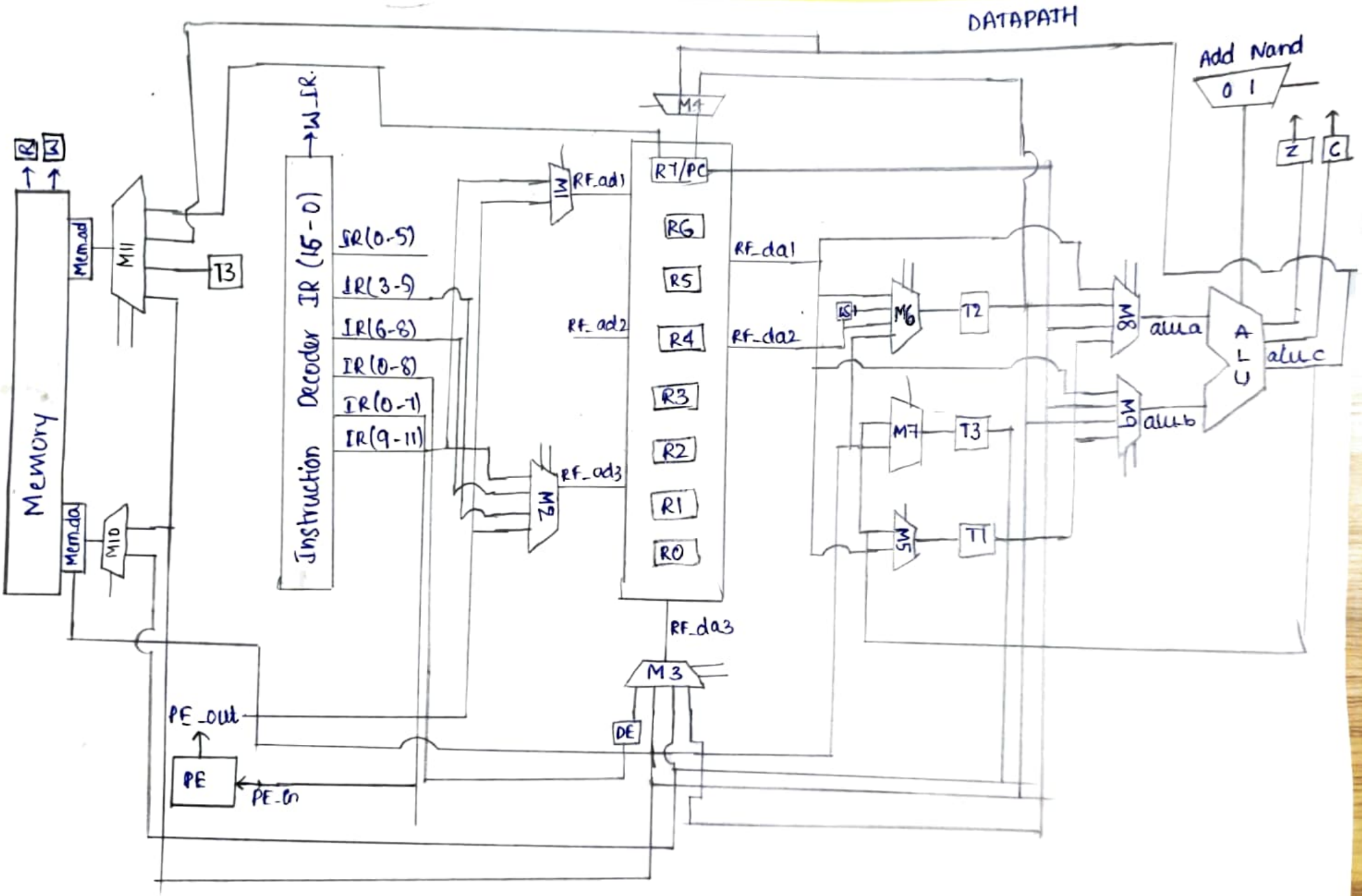
Mem_Ad:

PC → Mem_Ad 00
T3 → Mem_Ad 01
ALU_C → Mem_Ad 10
T1 → Mem_Ad 11

State Transition Diagram



	W_PC	W_IR	W_Mem	W_RF	M1 (RF_A1)	M2 (RF_A3)	M3 (RF_D3)	M4 PC	M5 T1	M6 T2	M7 T3	M8 ALU_a	M9 ALU_b	M10 Mem_Da	M11 Mem_Ad
Si	0	1	0	0	x	00	xx	xx	x	xx	x	xx	xx	x	xx
S0A	0	0	0	0	0	xx	xx	xx	0	00	x	xx	xx	x	xx
S0B	0	0	0	0	0	xx	xx	xx	0	01	x	xx	xx	x	xx
S1A	0	0	0	0	x	xx	xx	xx	x	xx	1	00	00	x	xx
S1B	0	0	0	1	x	00	00	xx	x	xx	x	xx	xx	x	xx
S2A	0	0	0	0	x	xx	xx	xx	x	xx	x	00	10	x	xx
S2B	0	0	0	1	x	01	00	xx	x	xx	x	xx	xx	x	xx
S3	0	0	0	1	x	10	01	xx	x	xx	x	xx	xx	x	xx
S4	0	0	0	0	x	xx	xx	xx	x	xx	1	10	10	x	xx
S5A	0	0	0	0	x	xx	xx	xx	x	11	x	xx	xx	x	10
S5B	0	0	0	1	x	10	10	xx	x	xx	x	xx	xx	x	xx
S6	0	0	1	0	x	xx	xx	xx	x	xx	x	xx	xx	0	01
S7A	0	0	0	0	x	xx	xx	xx	x	11	x	xx	xx	0	xx
S7B	0	0	0	1	x	11	10	xx	1	xx	x	00	01	x	xx
S8A	0	0	0	0	1	xx	xx	xx	x	10	x	xx	xx	x	xx
S8B	0	0	1	0	x	xx	xx	xx	1	xx	x	00	01	x	11
S9	1	0	0	0	0	xx	xx	0	x	xx	x	01	10	x	xx
S10	0	0	0	1	x	10	11	xx	x	00	x	xx	xx	x	xx
S11	1	0	0	0	x	xx	xx	0	x	xx	x	01	11	x	xx
S12	1	0	0	0	x	xx	xx	1	x	xx	x	xx	xx	x	xx
S13	1	0	0	0	0	xx	xx	0	x	xx	x	11	11	x	xx
Sf	1	0	0	0	0	xx	xx	0	x	xx	x	01	01	x	xx


DATAPATH
Add Nand
0 1
Z
C
M4
R7/PC
RG
R5
R4
R3
R2
R1
R0
RF_da1
RF_da2
RF_da3
RF_ad1
RF_ad2
RF_ad3
M1
M2
M3
M6
M7
M5
M8
M9
T2
T3
T1
alua
alub
alu
ALU
W_IR
Instruction Decoder IR (15-0)
IR(0-5)
IR(3-5)
IR(6-8)
IR(0-8)
IR(0-7)
IR(9-11)
Memory
Mem_ad
Mem_da
M11
M10
T3
R
W
PE_out
PE
PE_in
DE