<u>States:</u>

Current State	Next State	Level 2 Info MemAcess, Operation, Flag	Control Word (ALU(2)_T1(3)_IR(2)_RF(8)_memAcc(3)_INC(2))
S0: PC -> b16 -> a0 Edb -> IR	IB	IR, NA, NA	00 100 00 00000100 001 01
S1: PC -> INC -> PC Ra -> a16 -> ALU Rb -> b16 -> ALU ALU -> t1	S2	NA, OP, CZ	01 010 00 11011100 000 01
S2: T1 -> a16 -> Rc PC -> b16 -> a0 edb->ir	IB	IR, NA, NA	00 110 00 01000101 001 01
S3: IR[5:0] -> se16-> a16->ALU Rb -> b16- > ALU ALU -> T1 PC -> INC -> PC	SB	NA, OP, Z	10 010 01 11010100 000 01
S4: T1 -> b16 ->a0 Edb -> DI	S5	DR, NA, NA	00 100 00 00000000 001 01
S5: DI -> a16 -> Ra PC -> b16 -> a0 Edb -> IR	IB	IR, NA, NA	00 100 00 01000100 001 01
S6: T1 -> b16 -> a0 Ra -> a16 -> DO	S0	DW, NA, NA	00 101 00 00001000 100 01
S7: IR[8:0] -> shift7 -> a16 -> Ra	S0	NA, NA, NA	00 100 10 01000000 000 01
S8: Ra -> a16 -> ALU Rb -> b16 -> ALU ALU -> T1	BCZ (S11;S9) // (Z = 0, Z = 1)	NA, SUB, NA	11(SUB) 010(ALU -> T1) 00(IR) 00(wr) 01(as1as2) 11(read en) 00(D) 000(mem acc) 01
S9: IR[5:0] -> se16 -> a16 -> ALU PC -> b16 -> ALU	S10	IR, OP, NA	00 010 11 00 10 01 00 000 01

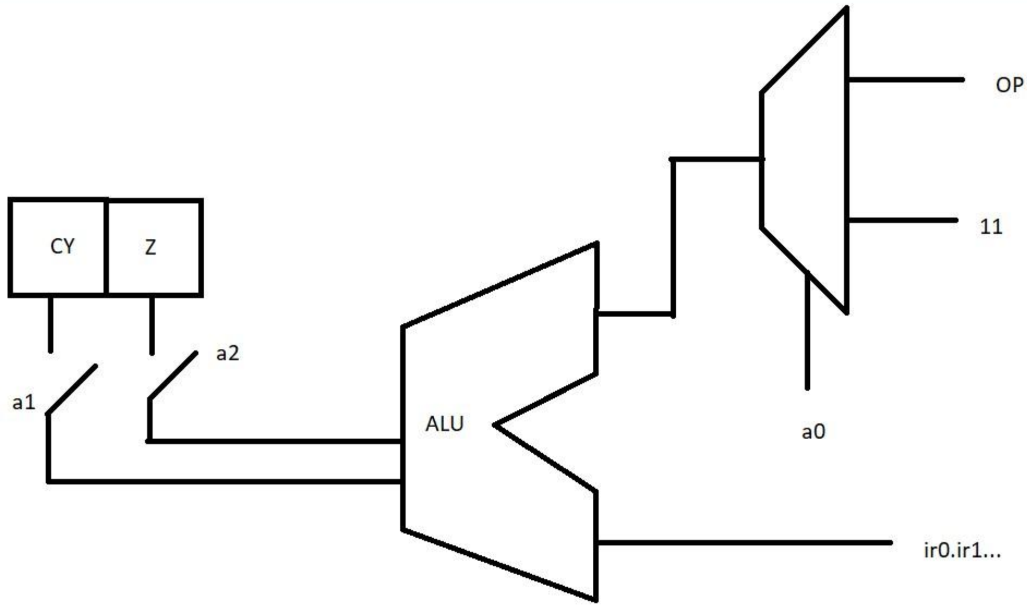
ALU -> T1			
S10: T1 -> a16 -> PC	S0	NA, NA, NA	00 110 00 01 00 00 11 000 01
S11: PC -> INC -> PC	SB	NA, NA, NA	00 100 00 11 00 00 00 000 01
S12: IR[5:0] -> a16 -> ALU Ra -> b16 -> ALU ALU -> T1 PC -> INC -> PC	SB	IR, OP, NA	00 010 01 11 10 01 00 000 01
S13: IR[5:0] -> se16 -> a16 -> ALU PC -> b16 -> ALU ALU -> T1 PC -> INC -> PC	S14	IR, OP, NA	00 010 11 11 10 01 00 000 01
S14: PC -> a16 -> ra T1 -> b16 -> PC	S0	NA, NA, NA	00 101 00 01 00 10 00 000 01
S15: Ra -> b16 -> t1 PC -> INC -> PC	SB	NA, NA, NA	00 001 00 00 11 01 00 000 01
S16: PC -> a16 -> ra Rb -> b16 -> PC	IB	NA, NA, NA	00 100 00 01 00 10 00 000 01
S17: DI -> a16 -> RF T1 -> INC -> t1	S18	NA, NA, NA	00 011 00 00 00 00 00 010 10
S18: SR -> INC -> SR	BCBSR(S0;S18;SB) // (SR = 0, SR != 0 && B = 0, SR !=0 && B = 1)	NA, NA, NA	00 100 00 00 00 00 00 000 00
S19: T1 -> b16 ->a0 Edb -> DI	S17	DR, NA, NA	00 101 00 00 00 00 00 000 01
S20: T1 -> INC -> t1 t1 -> b16 -> a0 SR -> a16 -> do	S18	DW, NA, NA	00 011 00 00 01 10 10 000 10

Decoder:

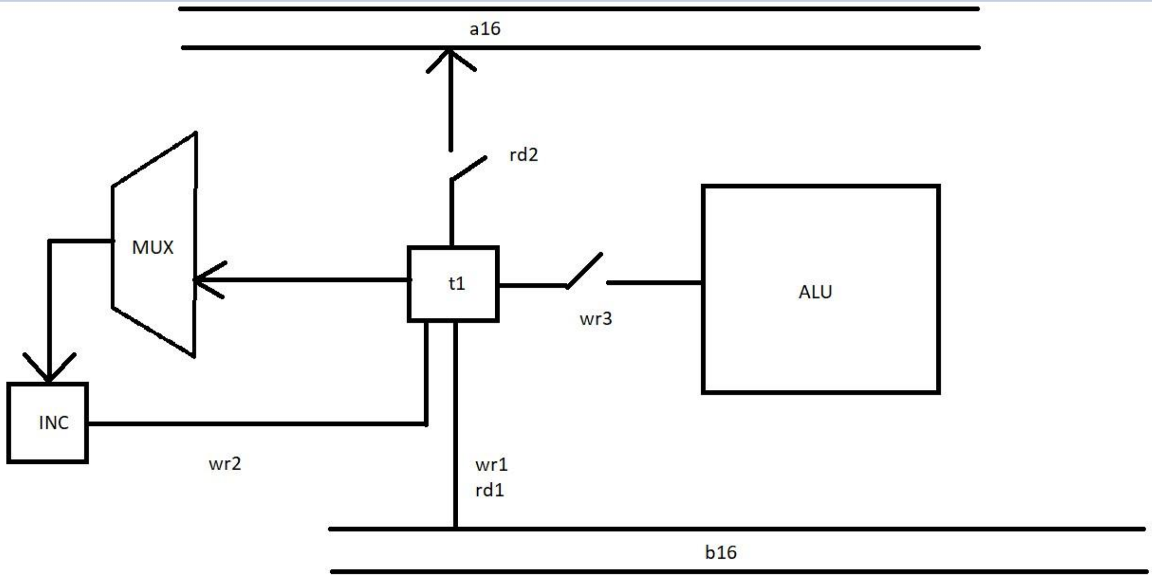
Instruction	IB	SB
0001	S1	-
0000	S12	S2
0010	S1	-
1111(LHI)	S11	S7
0111	S3	S4
0101	S3	S6
1100	S15	S19
1101	S15	S20
1000	S8	S0
1001	S13	-
1010	S11	S16
1011	S12	S10

Control diagrams:

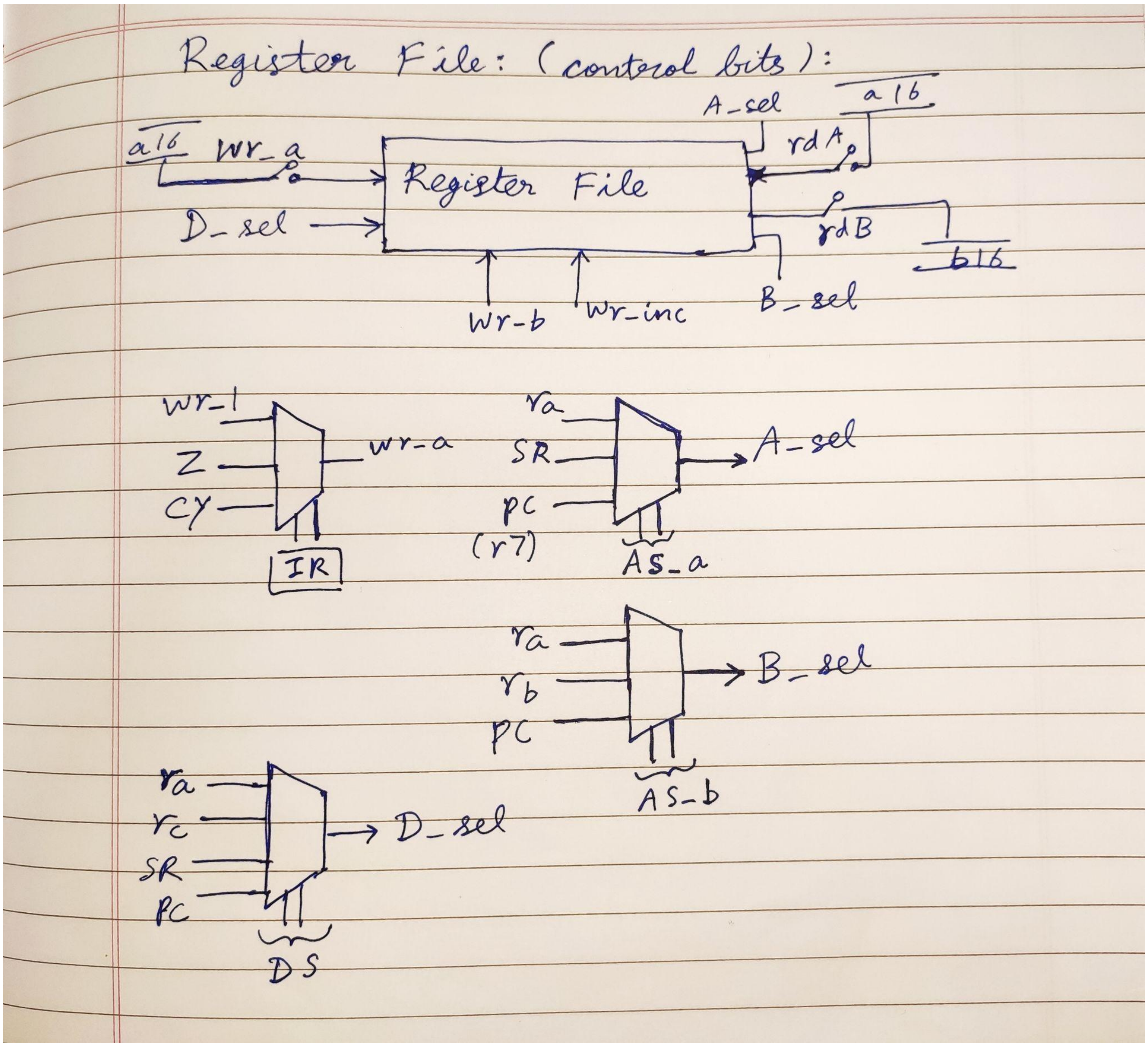
ALU control:



T1 control:

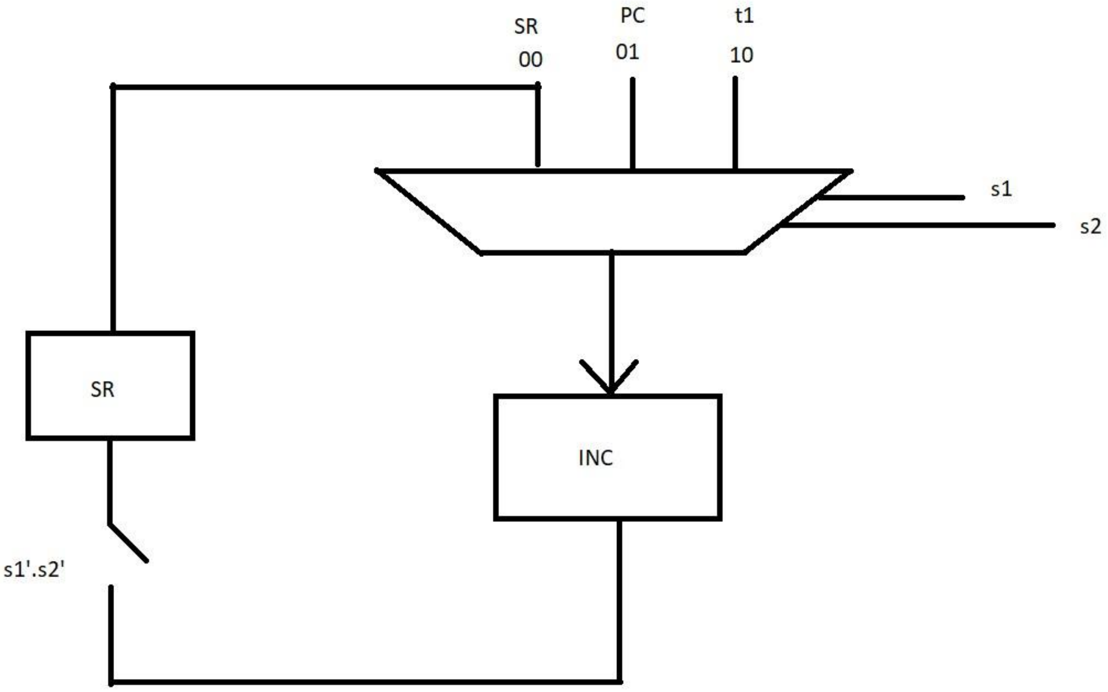


Register File control:



Register File: (control bits):

Incrementer control:



Control bits:

- ALU

alu_c1c2	a0	a1	a2
00	0	0	0
01	0	1	1
10	0	0	1
11	1	0	0

A0 = c1.c2
A1 = c1’.c2
A2 = c1’.c2 + c1.c2’

- T1

t1_c1c2c3	wr1	wr2	wr3	rd1	rd2
000	0	1	0	0	0
001	1	0	0	0	0
010	0	0	1	0	0

011	0	1	0	1	0
100	0	0	0	0	0
101	0	0	0	1	0
110	0	0	0	0	1

Wr1 = c1'c2'c3
Wr2 = c2c3 + c1'c2'c3'
Wr3 = c1'c2c3'
Rd1 = c2c3 + c1c3
Rd2 = c1c2

- IR

ir_c1c2	sel	en	rd
00	0	0	0
01	0	0	1
10	1	1	1
11	1	0	1

Sel = c1
En = c1c2’
Rd = c1 + c2

- Register file
  - Write bits

wr_c1c2	wr1	wr_b	wr_inc
00	0	0	0
01	1	0	0
10	1	1	0
11	0	0	1

Wr1 = c1'c2 + c1c2'
Wr_b = c1c2’
Wr_inc = c1c2

- Out addresses

Out_addr	AS1	AS2
00	00	10
01	00	01
10	10	01
11	01	00

As1_1 = c1c2’
As1_2 = c1’c2
As2_1 = c1’c2’
As2_2 = c1'c2 + c1c2'

- Read bits

rd_c1c2	rd1	rd2
00	0	0
01	0	1
10	1	0
11	1	1

Rd1 = rd_c1   rd2 = rd_c2

- D

DS	D
00	ra
01	rc
10	SR
11	PC(111)

Implementation using MUX

- INC

inc_s1s2	SR_write	INC_input	
00	1	SR	
01	0	PC	

10	0	T1	

External Data:

We will have two EDBs, one for reading and one for writing. DO → EDBw will have a control, this will also be used as write enable . AO will always be connected to the EAB. EDBr → DI will always be connected. There will be one control bit each between a16 → DO and DI → a16 each. B16 → AO will have a control, EDBr -> IR will have one control

DO->EDBw	DI -> a16	EDBr -> IR
0	0	0
1	1	1

Control word format:

ALU(2)	t1(3)	IR(2)	RF(8)	memAcc(3)	INC(2)

State transition for Instructions:

1) Arithmetic and logical instructions (ADD, ADC, ADZ, ADL, NDU, NDC, NDL)

   S1 → S2 → IB

2) LW:

   S3 → S4 → S5 → IB

3) SW:

   S3 → S6 → S0 → IB

4) LHI:

   S7 → S0 → IB

5) BEQ:

   If z == 0: S8 → S11 → S0 → IB
   If z==1; S8 → S10 → S0 → IB

6) ADI:

   S12 → S2 → IB

7) JAL:

   S13 → S14 → S0 → IB

8) JLR:

   S11 → S16 → IB

9) JRI:

   S12 → S10 → S0 → IB

10) LM:

   S15 → S19 → S17 → S18
   if(SR == 0): → S0 → IB

if(SR != 0 and B == 0): → S18
if(SR != 0 and B != 0): → S19

11) SM:

S15 → S20 → S18
if(SR == 0): → S0 → IB
if(SR != 0 and B == 0): → S18
if(SR != 0 and B != 0): → S20