EE224

Course Project

Arin Weling: 22B1230

Chiransh Somani: 22B1202

Sathvik Reddy: 22B3946

Tanish Raghute: 22B3974

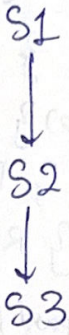
EE224 Project Design Document

FSM

ADD (R)

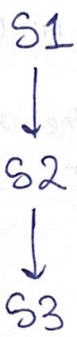
SUB (R)

MUL (R)

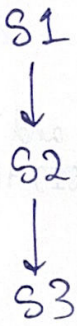


ADI (I)

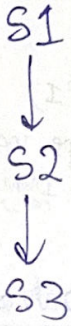
AND (R)



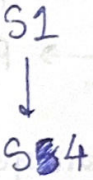
ORA (R)



IMP (R)

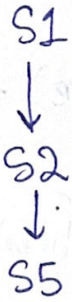


LHI (J)

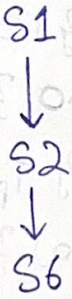


LLI (J)

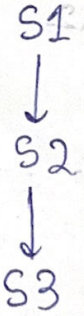
LW (I)



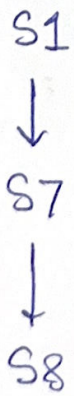
SW (I)



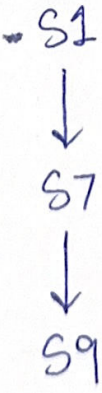
BEQ (I)



JAL (I)



JLR (I)



Flowcharts & Controls

S1: Fetch instruction, update Program Counter, store updated program counter value in temporary register Tpc as well

Flowchart:

'111' → RF_A1
RF_D1 → MEM_Add
MEM_DataOut → IR
RF_D1 → ALU_A
+2 → ALU_B
ALU_C → Tpc
ALU_C → RF_D3
'111' → RF_A3

Controls: (the ~~enables~~ mentioned here are set to 1, others are unless otherwise mentioned)

MEM_R } MEMORY
ADD (0000) } ALU (select lines)
Tpc_WE } Tpc
RF_WE } RF

↳ (apart from this, registers inside the RF are selected/enabled by a combinational function of the bits in RF_A3 ⟹ Enable for R_5 (example)
= (RF_WE) AND ((A3(2)) AND (NOT A2) AND (A0))

S2: Store operands for (R)-type instructions in T1 and T2, update the value of T_{pc}, which will then hold PC+2 & $IMM_{(0-5)}$ * 2 if it is an (I) type instruction and (I)

Flowchart:

IR_{9-11} → RF_A1
IR_{6-8} → RF_A2
RF_D1 → T1
RF_D2 → T2
.....
IR_{0-5} → SE_in
SE_out → LS_in
LS_out → ALU_B
Tpc → ALU_A
ALU_C → Tpc

Controls:

T1_WE } T1
T2_WE } T2
SE_sel = 0 } SE (select line) → left padding
Tpc_WE } Tpc
ADD (0000) } ALU (select lines)

S3: The execution phase, for all (R) type instructions, BEQ (I) and ADI (I), the updating of registers and operands in the ALU are controlled by ADI, BEQ, and Z bits, as select lines of MUXes or combinational functions fed to RF_WE (write enable)

Flowchart:

IR_{0-5} → SE_in
T1 → ALU_A
if (ADI == '1') then
SE_out → ALU_B
else
T2 → ALU_B
if (Z == '1') then
T_{pc} → RF_D3
else
ALU_C → RF_D3
if (Z == '0' and ADI == '0') then
IR_{3-5} → RF_A3
elsif (Z == '0' and ADI == '1') then
IR_{6-8} → RF_A3
elsif (Z == '1' and ADI == '0') then
'111' → RF_A3

Controls:

~~SE_Sela (2) } SE (select lines)~~

RF_WE = $\overline{BEQ}$ + Z.BEQ } RF
SE_Sel = '0' } SE (select lines)
IR_{12-15} (opcode) } ALU (select lines)

S4: For LHI, RLI instructions, this state involves left extending or right extending immediates in (I) type instructions, and updating RegA with the value.

Flowchart

IR_{0-8} → SE_in
SE_out → RF_D3
IR_{9-11} → RF_A3

Controls:

SE_Sel = $\overline{IR(12)}$ } SE
RF_WE } RF

S5: This state involves computation of the Address in load instruction and accessing the memory to update the value in the RF.

Flowchart:

IR_{0-5} → SE_in
SE_out → ALU_B
T2 → ALU_A
ALU_C → MEM_Add
MEM_DataOut → RF_D3
IR_{9-11} → RF_A3

Controls:

SE_Sel = '0' } SE
ADD ('0000') } ALU (select lines)
MEM_R } MEMORY
RF_WE } RF

S6: This state, in the store instruction computes the address using RegB value stored in T2 and stores the RegA value in the memory.

Flowchart:

IR_{0-5} → SE_in
SE_out → ALU_B
T2 → ALU_A
ALU_C → MEM_Add
T1 → MEM_DataIn

Controls:

SE_Sel = '0' } SE
ADD (0000) } ALU (select lines)
MEM_W } MEMORY

S7: Used in JAL & JLR instructions, this state stores the ~~instru~~ Program counter value in another register in RF.

Flowchart:

'111' → RF_A1
RF_D1 → RF_D3
IR_{9-11} → RF_A3

Controls:

RF_WE } RF

S8: For the JAL instruction, we use the value PC + 2 + IMM*2 stored using T_{pc} (CS2 is skipped for these instructions) and immediate. This value is stored in the RF.

Flowchart:

IR_{0-5} → SE_in
SE_out → LS_in
LS_out → ALU_B
T_{pc} → ALU_A
ALU_C → RF_D3
'111' → RF_A3

Controls:

SE_Sel = '0' } SE
ADD (0000) } ALU (select lines)
RF_WE } RF

S9: For the JLR instruction, we directly update PC with the value stored in Reg B.

Flowchart:

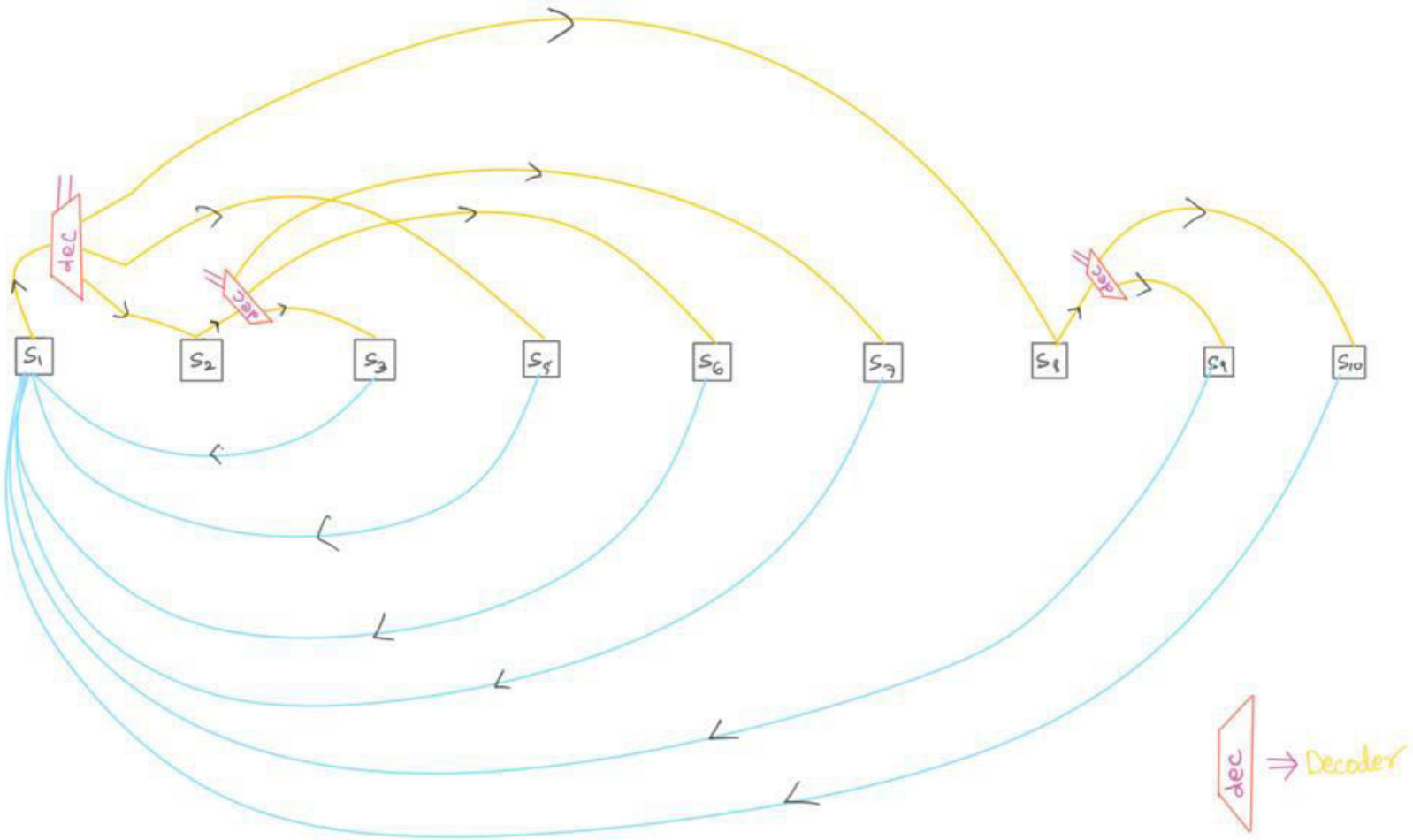
IR_{6-8} → RF_A2
RF_D2 → RF_D3
'111' → RF_A3

Controls:

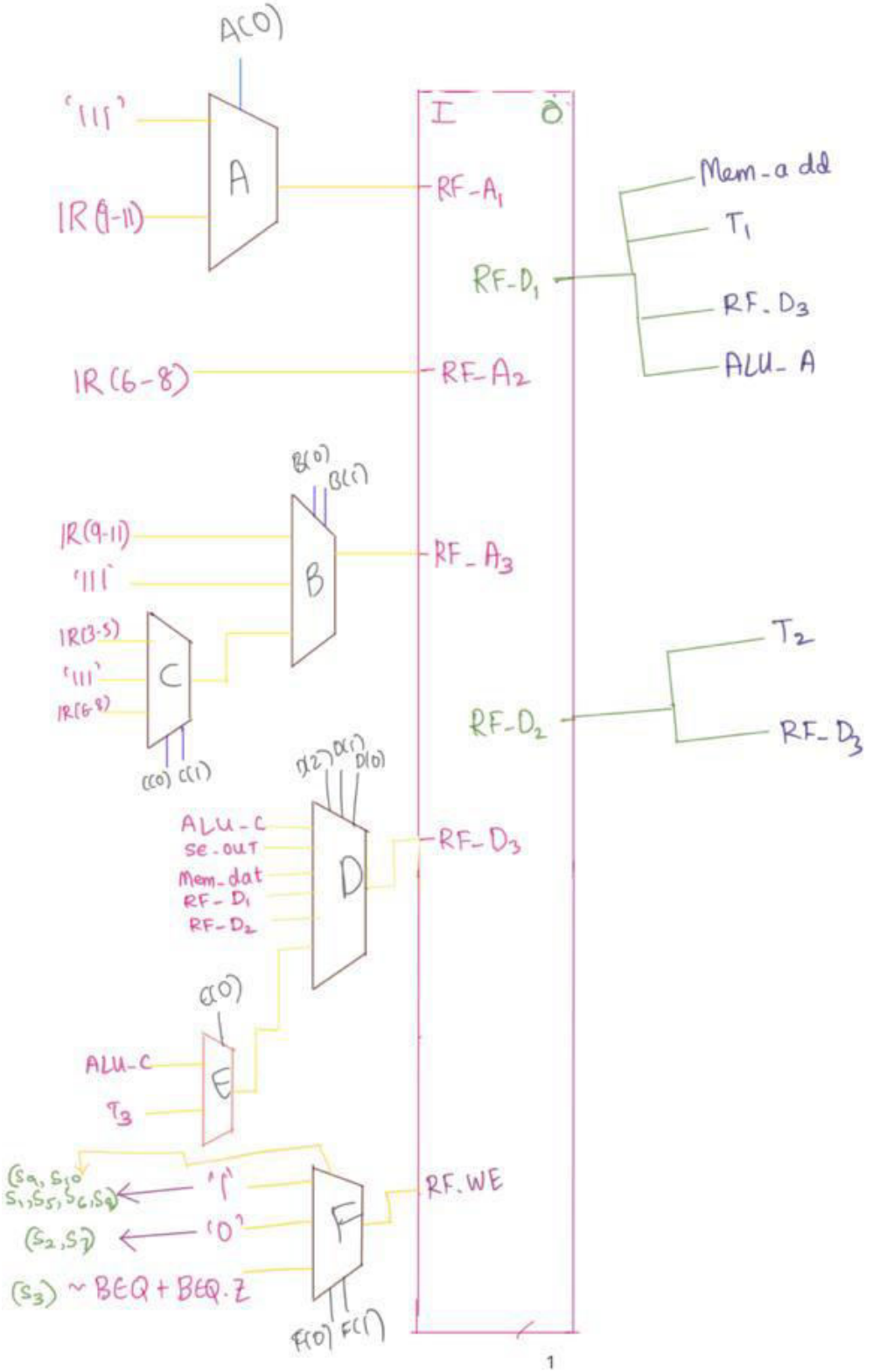
RF_WE } RF

* Control bits corresponding to the muxes at each input of the components are made clear from the component diagrams, and the data flowcharts indicating which input is taken.

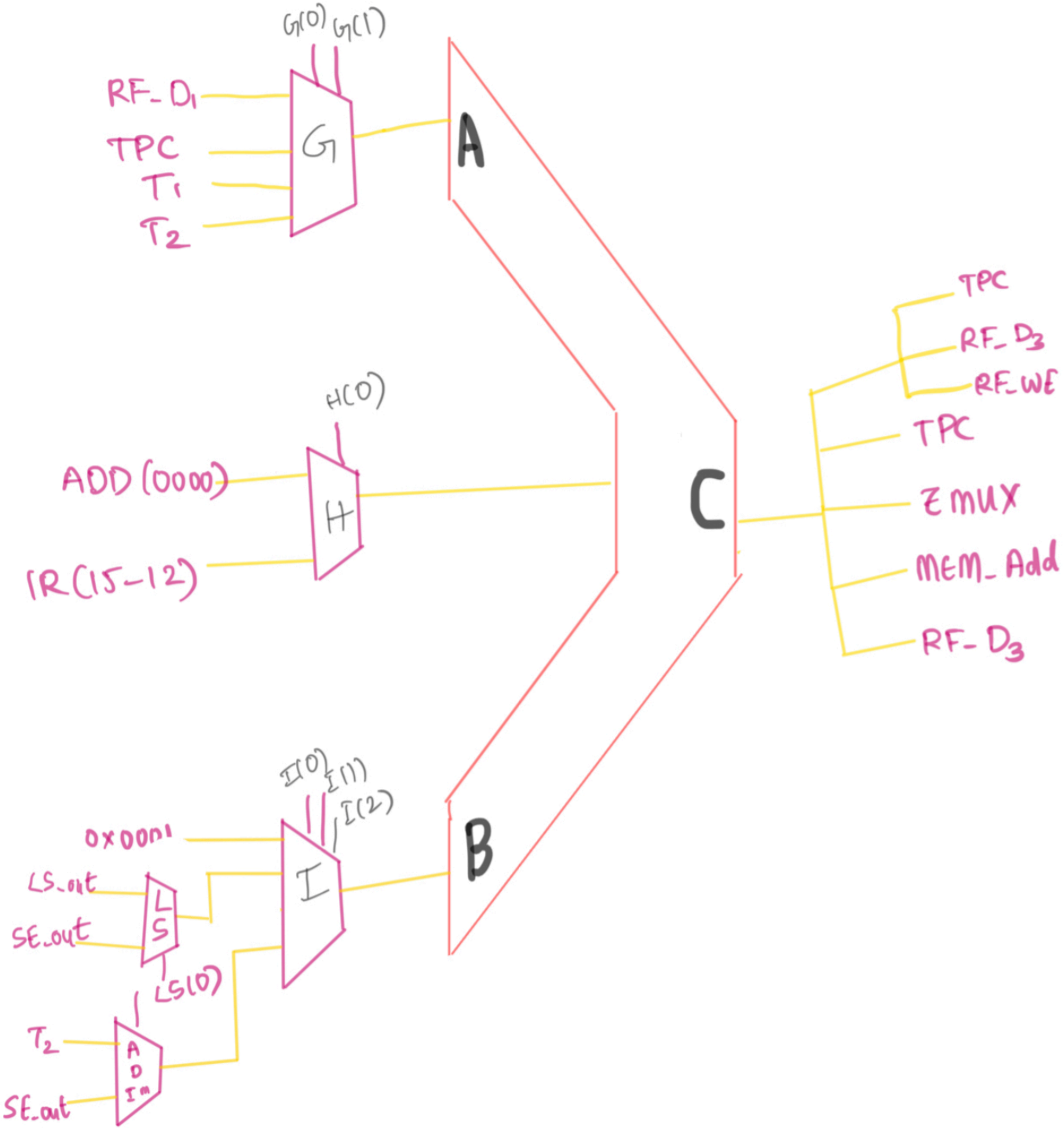
FINAL FSM



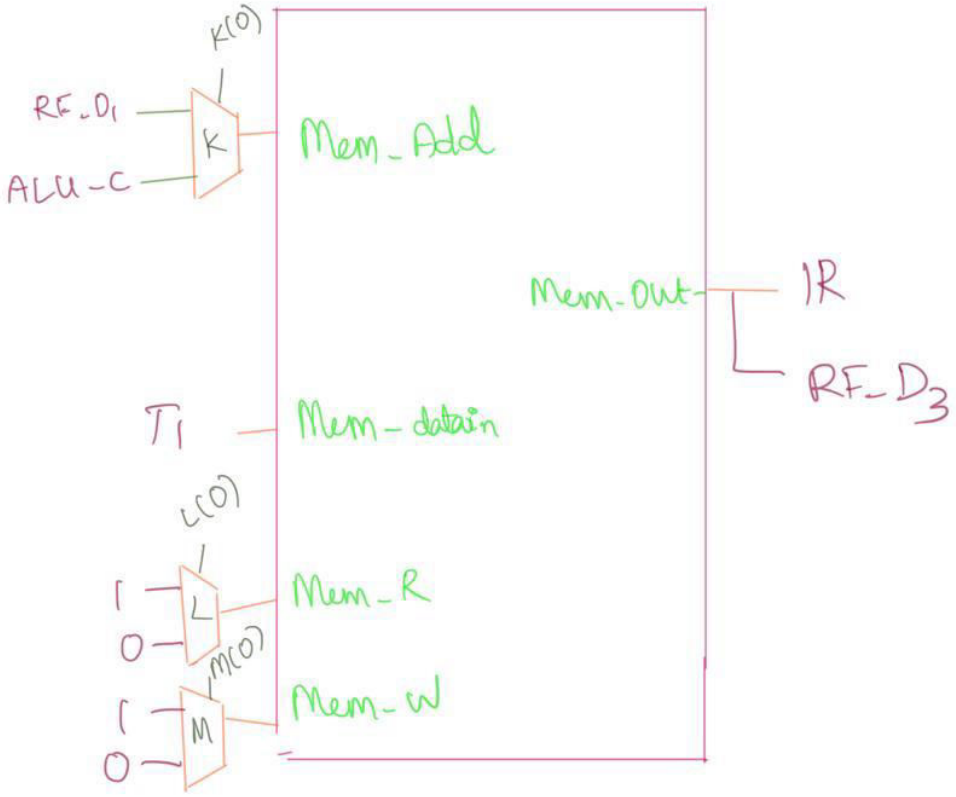
Register file



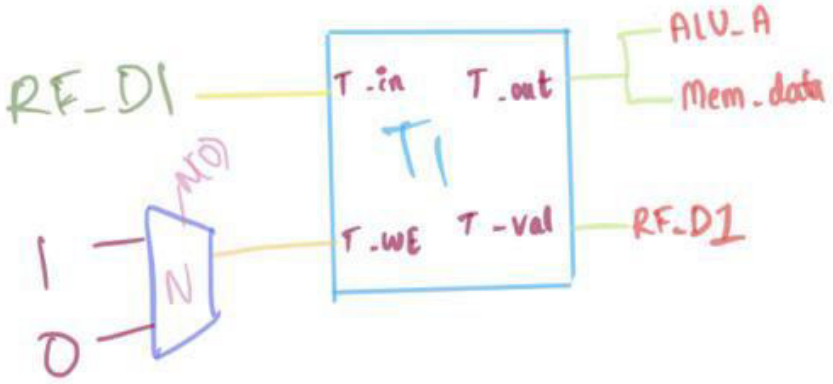
ALU

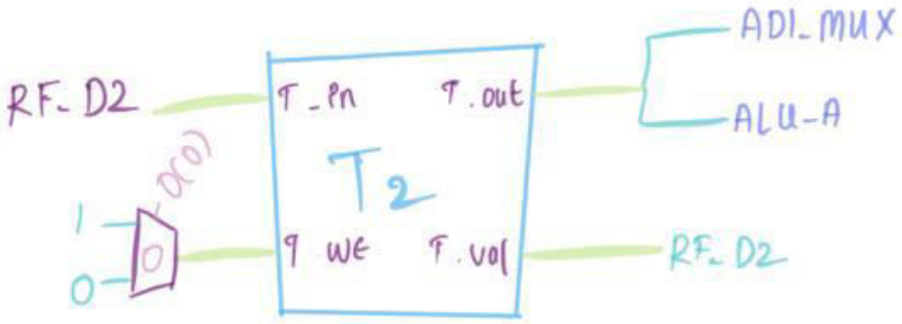

G(0)
G(1)
RF_D1
TPC
T1
T2
G
A
H(0)
ADD (0000)
IR(15-12)
H
C
TPC
RF_D3
RF_WE
TPC
2 MUX
MEM_Add
RF_D3
I(0)
I(1)
I(2)
0x0001
LS_out
SE_out
LS
LS(0)
T2
ADI
SE_out
I
B

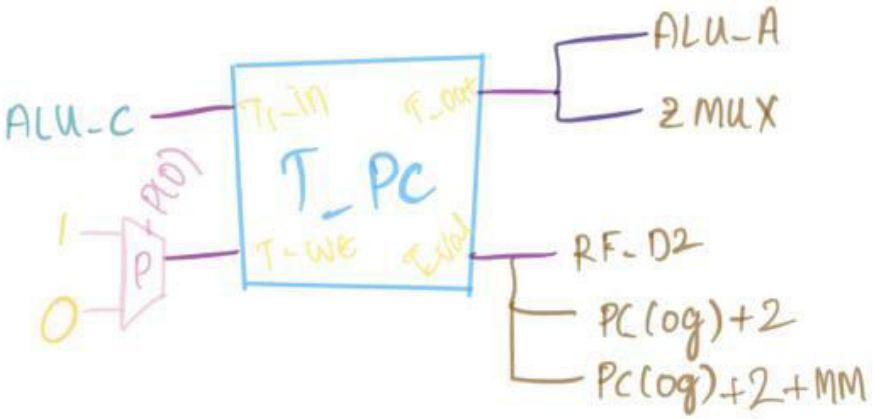
Memory


K(0)
RF_O1
K
ALU_C
Mem_Add
Mem_Out
IR
RF_D3
T1
Mem_datain
L(0)
1
L
0
Mem_R
M(0)
1
M
0
Mem_W

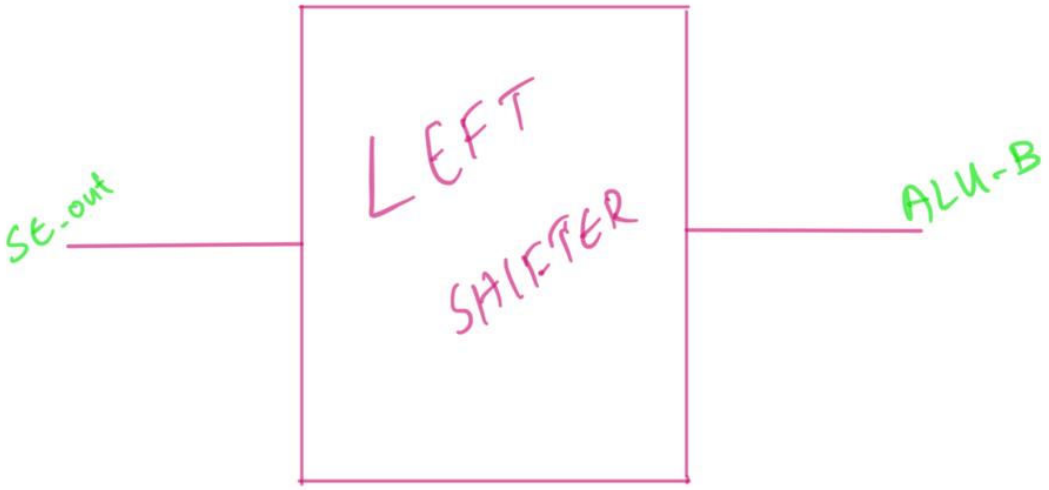
Temporary Registers



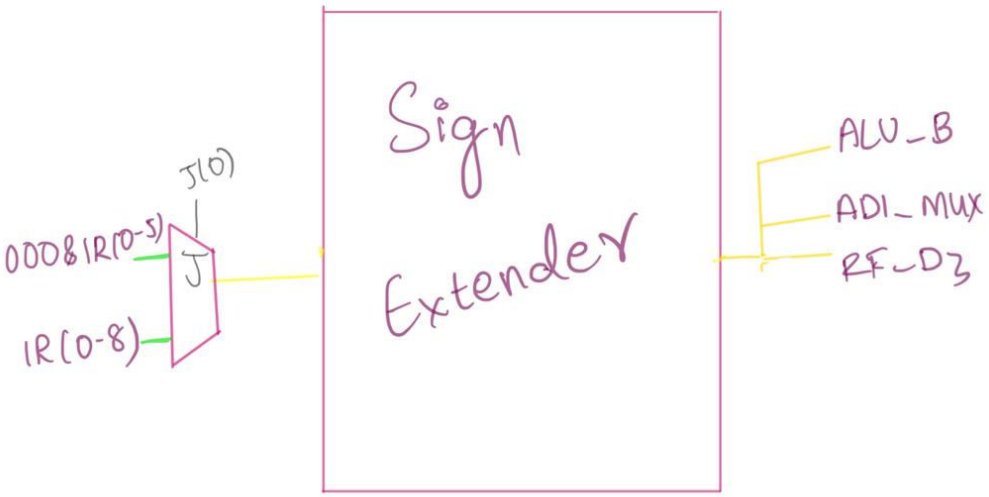




Left Shifter

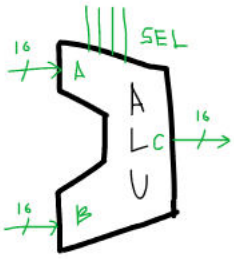


Sign_Extender (S.E)

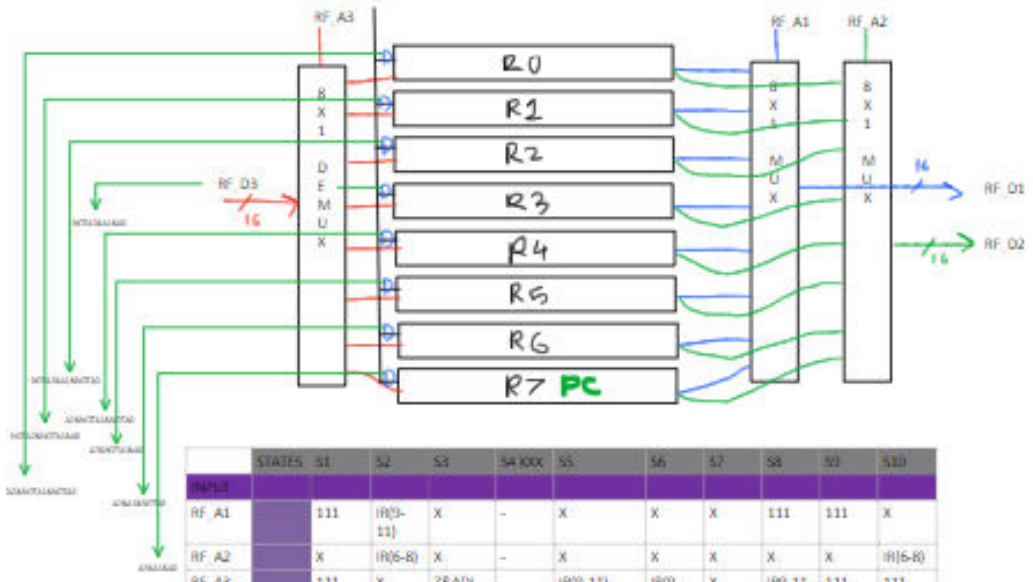


Component Diagrams

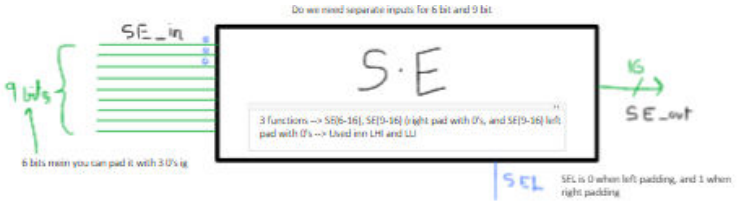
Tuesday, 14 November 2023 9:52 PM



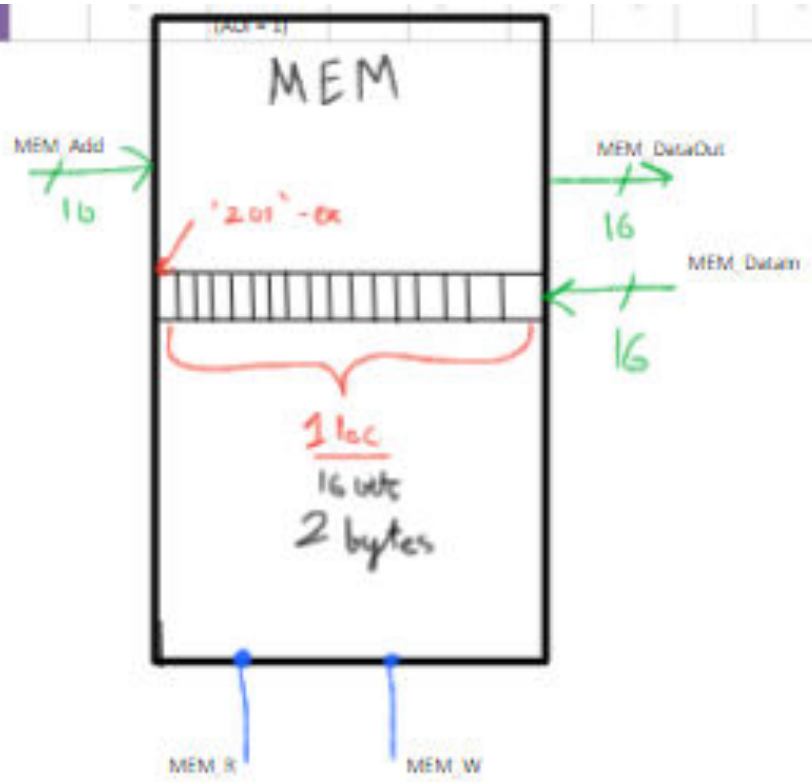
(Only connections shown here)	STATES	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
INPUT											
A		RF_D1	TPC	T1	-	X	T2	T2	X	RF_D1	X
B		0000000000000001	SE(6-16)	ADI MUX	-	X	SE(6-16)	SE(6-16)	X	SE(6-16)	X
SEL		ADD (0000)	ADD (0000)	IR(15 down to 12)	-	X	ADD (0000)	ADD (0000)	X	ADD (0000)	X
OUTPUTS											
C		TPC, RF_D3 RF_WE	TPC	Z MUX (0)	-	X	Mem_Add	Mem_Add	X	RF_D3	X



	STATES	S1	S2	S3	S4 XXX	S5	S6	S7	S8	S9	S10
INPUT											
RF_A1		111	IR(9-11)	X	-	X	X	X	111	111	X
RF_A2		X	IR(6-8)	X	-	X	X	X	X	X	IR(6-8)
RF_A3		111	X	Z&ADI MUX || IR3-5 (Z=0, ADI=0)/ '111'(Z=1 ,ADI=0)/ IR(6-8) (Z=0,ADI =1)	-	IR(9-11)	IR(9-11)	X	IR(9-11	111	111
RF_D3		ALU_C	X	Z MUX || ALU_C (Z=0) / T3	-	SE_out	Mem_data	X	RF_D1	ALU_C	RF_D2
RF_WE		1	0	~BEQ+ BEQ.Z (USE DECODER)	-	1	1	0	1	1	1
OUTPUTS											
RF_D1		Mem_add, ALU_A	T1	X	-	X	X	X	RF_D3	ALU_A	X
RF_D2		X	T2	X	-	X	X	X	X	X	RF_D3



(Only connections shown here)	STATES	S1	S2	S3	S4 XXX	S5	S6	S7	S8	S9	S10
INPUT											
SE_in		X	000 & IR[0-5]	000 & IR[0-5]	-	IR[0-8]	000 & IR[0-5]	000 & IR[0-5]	X	000 & IR[0-5]	X
SEL		X	0	0	-	NOT IR[12]	0	0	X	0	X
OUTPUT											
SE_out		X	ALU_B	ADI MUX (ADI = 1)	-	RF_D3	ALU_B	ALU_B	X	ALU_B	X

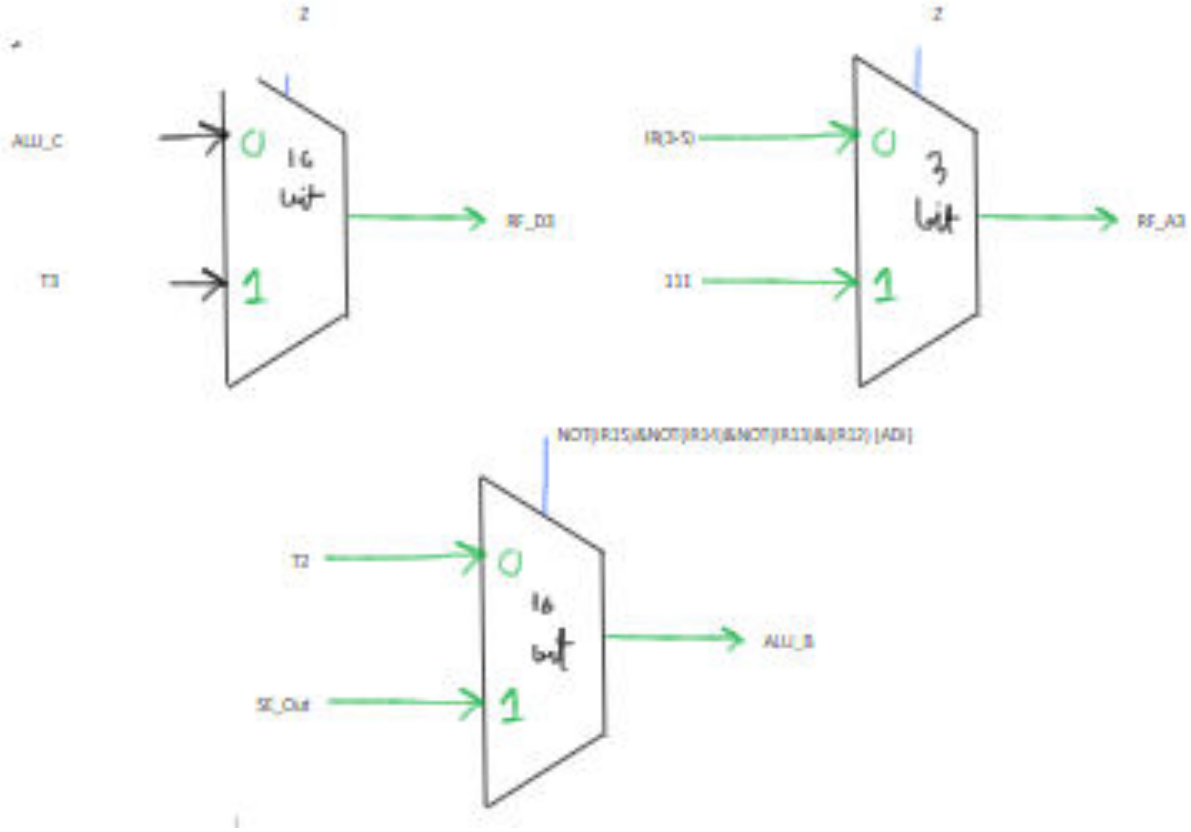


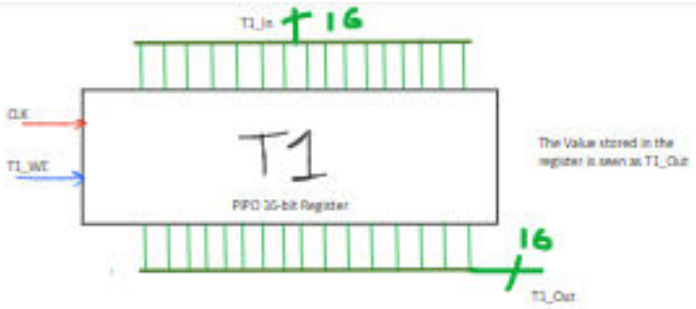
	STATES	S1	S2	S3	S4 XXX	S5	S6	S7	S8	S9	S10
INPUT											
MEM_Add		RF_D1	X	X	-	X	ALU_C	ALU_C	X	X	X
MEM_Datain		X	X	X	-	X	X	T1	X	X	X
MEM_R		1	0	0	-	0	1	0	0	0	0
MEM_W		0	0	0	-	0	0	1	0	0	0
OUTPUTS											
MEM_DataOut		IR	X	X	-	X	RF_D3	X	X	X	X

Z

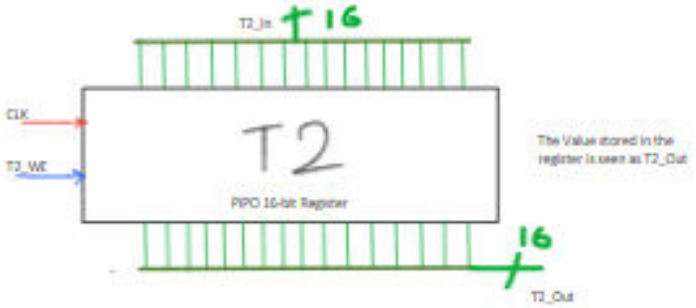
Z

S


2
ALU_C
0
16
bit
T3
1
RF_D3
2
IR[3-5]
0
3
bit
111
1
RF_A3
NOT(IR15)&NOT(IR14)&NOT(IR13)&(IR12) |ADI
T2
0
16
bit
SE_Out
1
ALU_B



(Only connections shown here)	STATES	S1	S2	S3	S4 XXX	S5	S6	S7	S8	S9	S10
INPUT											
T1_In		X	RF_D1	X	-	X	X	X	X	X	X
T1_WE		0	1	0	-	0	0	0	0	0	0
OUTPUT											
T1_Out		X	X	ALU_A	-	X	X	MEM_Data in	X	X	X
VALUE											
T1_Value		X	X	RF_D1	-	X	RF_D1	RF_D1	X	X	X



(Only connections shown here)	STATES	S1	S2	S3	S4 XXX	S5	S6	S7	S8	S9	S10
INPUT											
T2_In		X	RF_D2	X	-	X	X	X	X	X	X
T2_WE		0	1	0	-	0	0	0	0	0	0
OUTPUT											
T2_Out		X	X	ADI MUX (ADI=0)	-	X	ALU_A	ALU_A	X	X	X
VALUE											
T2_Value		X	X	RF_D2	-	X	RF_D2	RF_D2	X	X	X

(Only connections shown here)	STATES	S1	S2	S3	S4 XXX	S5	S6	S7	S8	S9	S10
INPUT											
Tpc_In		ALU_C	ALU_C	X	-	X	X	X	X	X	X
Tpc_WE		1	1	0	-	0	0	0	0	0	0
OUTPUTS											
Tpc_Out		X	ALU_A	Z MUX (Z=1)	-	X	ALU_A	ALU_A	X	X	X
VALUE											
Tpc_Value		X	PC(op) +1	PC(op)+1=I MM	-	X	RF_D2	RF_D2	X	X	X