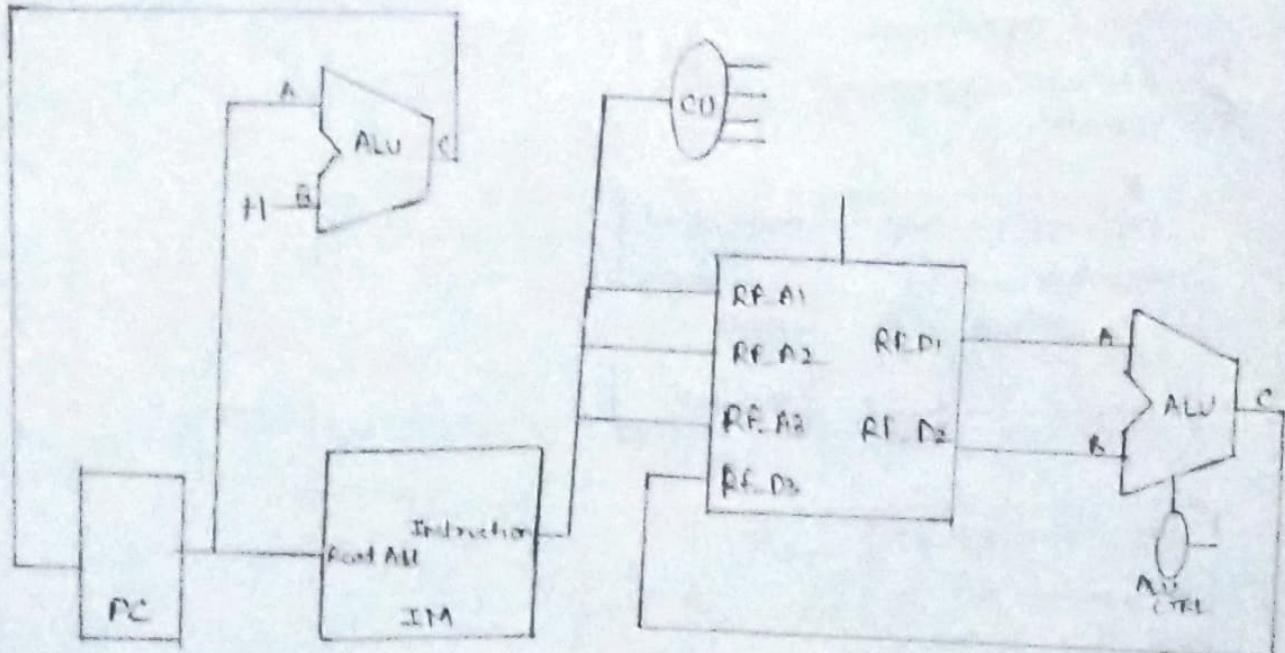


EE224: Digital Design  
Course Project  
IITB - CPU Design

Group Members : 3

1. Ashwin Nagarwal 210070013
2. Kudupudi Puja Naga Prasanna 210070046
3. Putcha Ushashvi 210070065.

# 1. ADD & NAND Instruction (R-type)



# ① ADD & NAND

15-12	11-9	8-6	5-3	2	1-0	
Opcode	RA	RB	RC	O	CZ	
00..00 & 0010	(3bit)	(3bit)	Unused bit	(2bit)		16 bits

- ① Read Instruction
- ② Read operands
- ③ Execute operation
- ④ Update

S-1

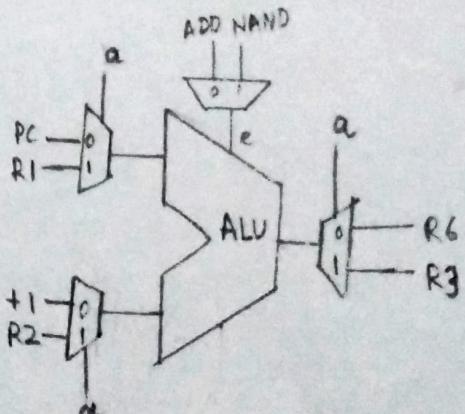
PC → Mem-add	mem read
memdata → R7	R7-WR
PC → ALU-A	ADD
+1 → ALU-B	
ALU-C → R6-1	R6-WR

S2

R7 <sub>11-9</sub> → RF-A1	
R7 <sub>8-6</sub> → RF-A2	
R7 <sub>15-12</sub> → CU	
RF-D1 → R1	R1-WR
RF-D2 → R2	R2-WR

S3

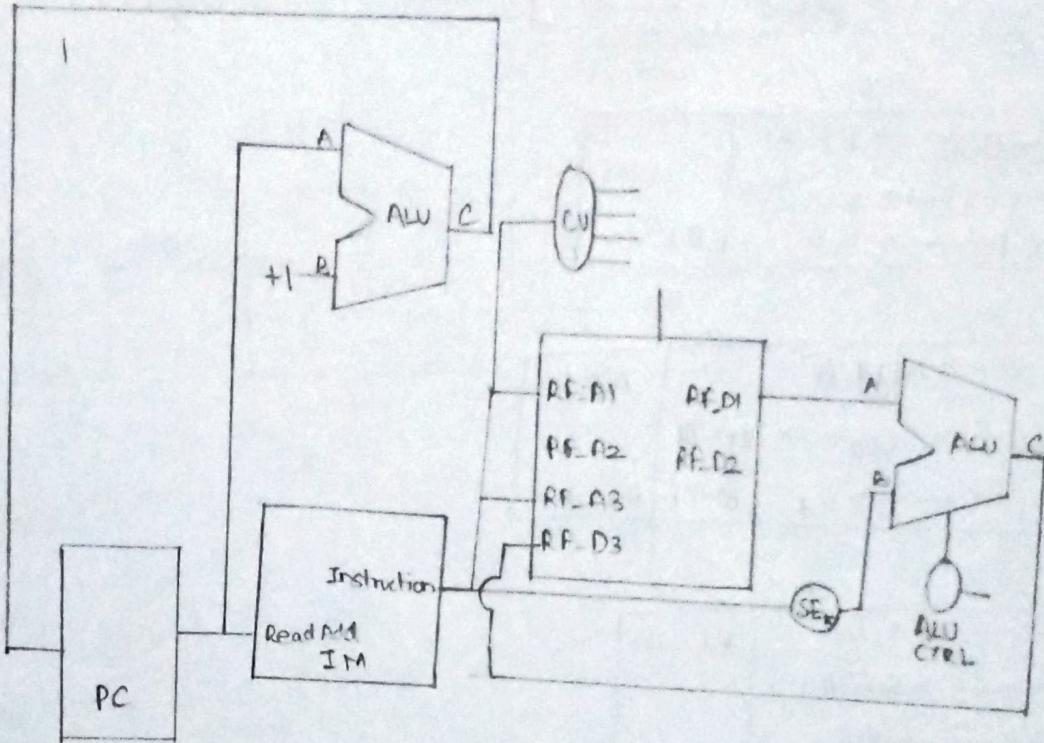
R1 → ALU-A	ADD
R2 → ALU-B	
if CZ <sub>0</sub> = 1	
ALU-C → R3	R1-WR



S4

R3 → RF-D3	RF-WR
R7 <sub>5-3</sub> → RF-A3	
R6-1 → PC	

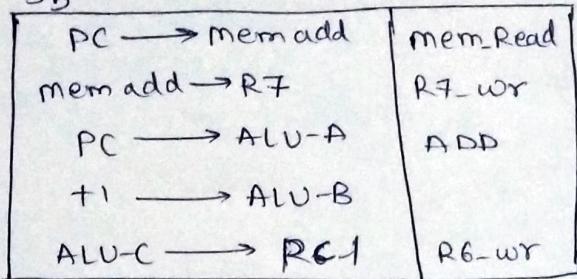
## 2. ADD Immediate Instruction (I-type)



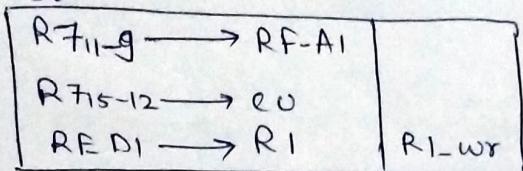
## ② ADD

OP code	R A	R B	Immediate	16 bits
00-01	(3bit)	(3bit)	(6bit)	

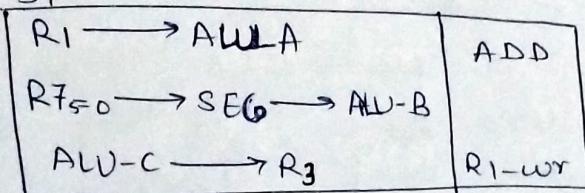
S5



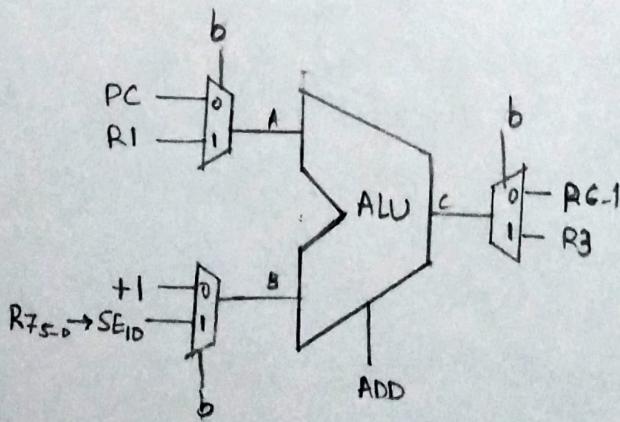
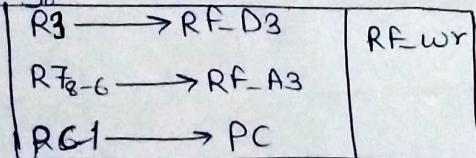
S6



S7

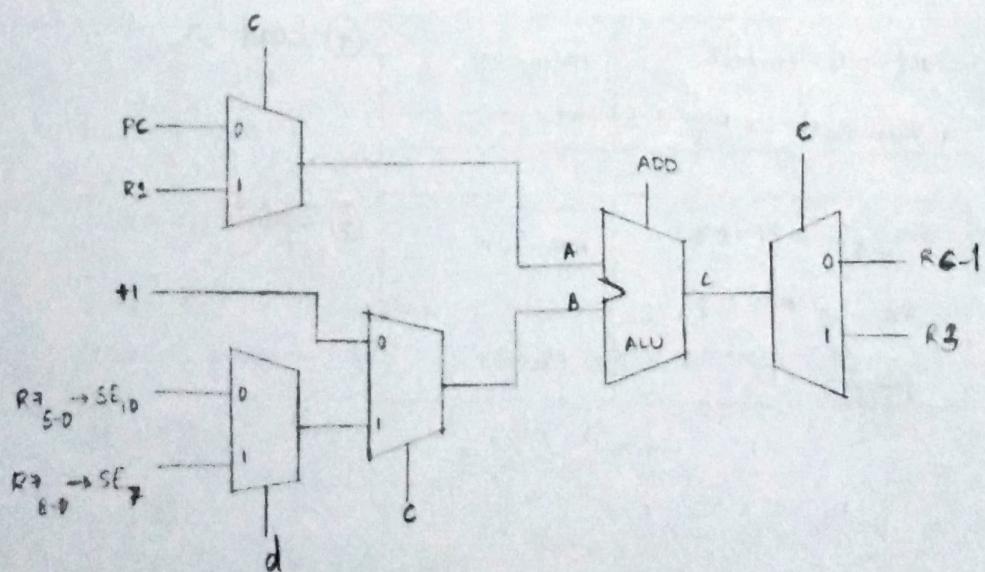
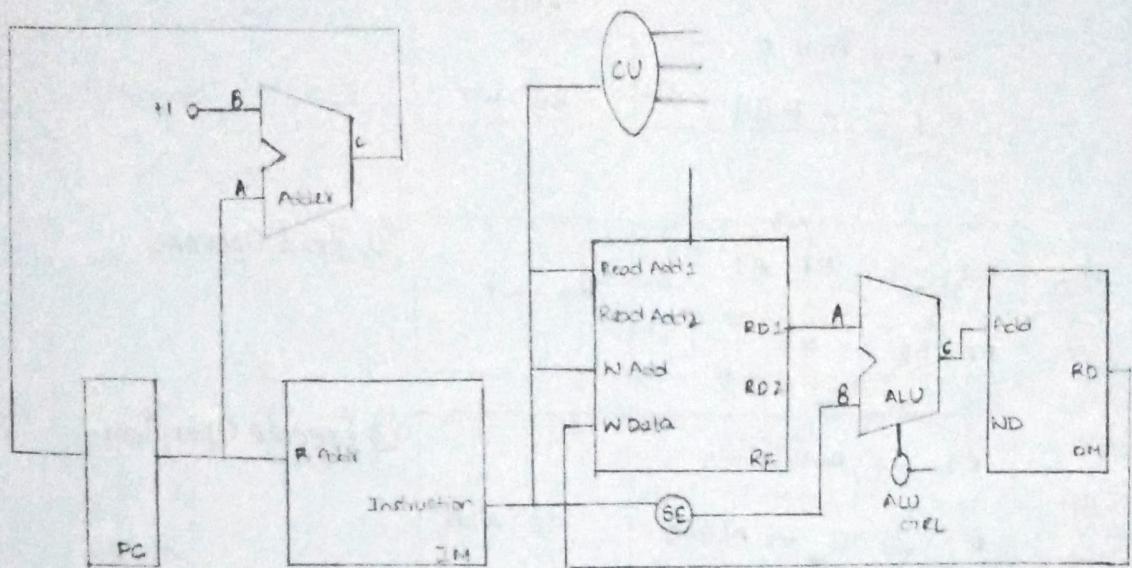


S8



### 3. LHI - Load Higher Immediate (J)

J-Type Instruction



FOR LW, SW, LHI INSTRUCTIONS

Opcode	Reg A	Immediate
15-12	11-9	8-0
4-bit	3-bit	9-bit

Sq	$PC \rightarrow \text{Mem-Add}$ $\text{Mem-Data} \rightarrow R7$ $PC \rightarrow \text{ALU-A}$ $+1 \rightarrow \text{ALU-B}$ $\text{ALU-C} \rightarrow PG1$	$\text{Mem Read}$ $RF-WR$ $ADD$ $R6-WR$	① Read Instruction
----	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------	--------------------

S10	$R7_{11-9} \rightarrow RF-A1$ $RF-D1 \rightarrow R1$	$R1-WR$	② Read Operand
-----	---------------------------------------------------------	---------	----------------

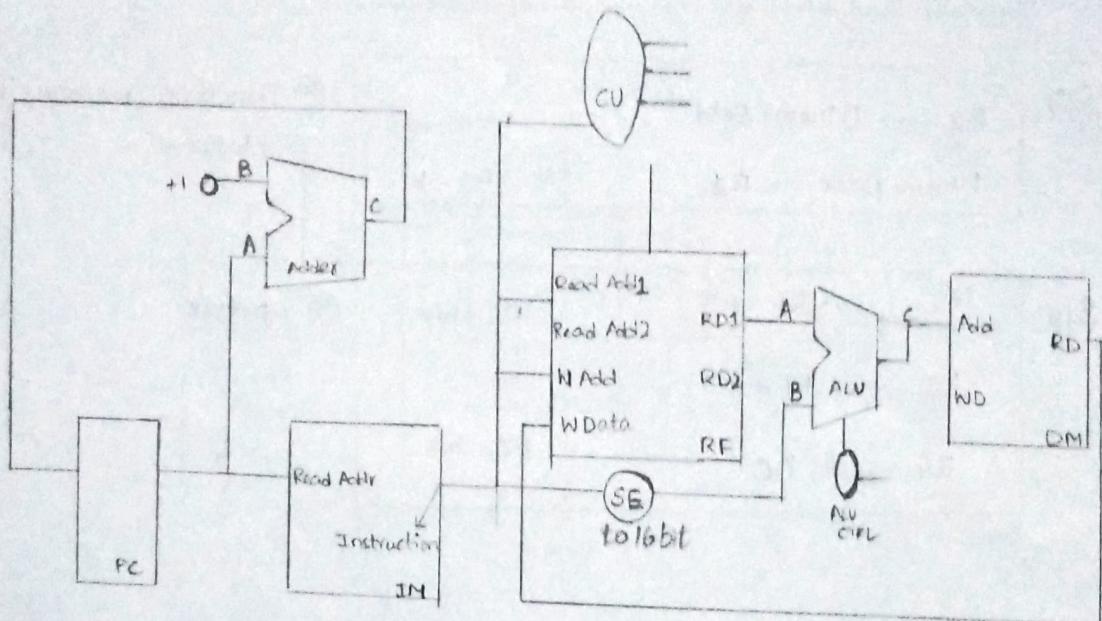
S11	$R1 \rightarrow \text{ALU-A}$ $R7_{8-0} \rightarrow SF_q \rightarrow \text{ALU-B}$ $\text{ALU-C} \rightarrow R3$	$SE, ADD$ $R3-WR$	③ Execute Operation
-----	------------------------------------------------------------------------------------------------------------------------	----------------------	---------------------

S12	$R3 \rightarrow D\text{MemAdd}$ $D\text{MemData} \rightarrow R4$	$\text{Mem-WR}$ $R4-WR$	④ Load
-----	---------------------------------------------------------------------	----------------------------	--------

S13	$R7_{11-9} \rightarrow RF-A3$ $R4 \rightarrow RF-D3$ $PG1 \rightarrow PC$	$RF-WR$ $PC-WR$	⑤ Update
-----	---------------------------------------------------------------------------------	--------------------	----------

#### 4. Load (LW) (I-type)

Opcode: 01\_00.



#### I-type Instruction

Opcode	Reg A	Reg B	Immediate
4 bit	2 bit	3 bit	6-bit
15-12	11-9	8-6	5-0

S14	$PC \rightarrow \text{Mem-Add}$ $\text{Mem Data} \rightarrow R7$ $PC \rightarrow \text{ALU-A}$ $+1 \rightarrow \text{ALU-B}$ $\text{ALU-C} \rightarrow R6$	$\text{Mem Read}$ $R7 \text{ WR}$ $\text{ADD}$ $R6 \text{-WR}$	① Read Instructions
-----	------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------	---------------------

S15	$R7_{8-6} \rightarrow RF-A1$ $RF D1 \rightarrow R1$	$R1 \text{-WR}$	② Read Operands
-----	--------------------------------------------------------	-----------------	-----------------

S16

$R1 \rightarrow ALU-A$	
$R7_{5-0} \rightarrow SE_6 \rightarrow ALU-B$	SE, ADD
$ALU-C \rightarrow R3$	R1 WR

(3) Execute Operation

S17

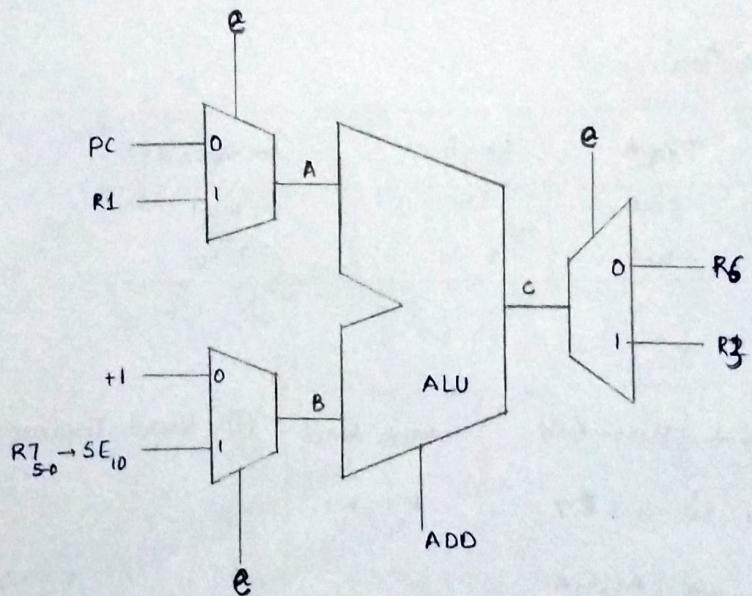
$R3 \rightarrow DMem-Add$	Mem-WR
$DMem-Data \rightarrow R4$	R1 - WR

(4) Transfer memory to data

S18

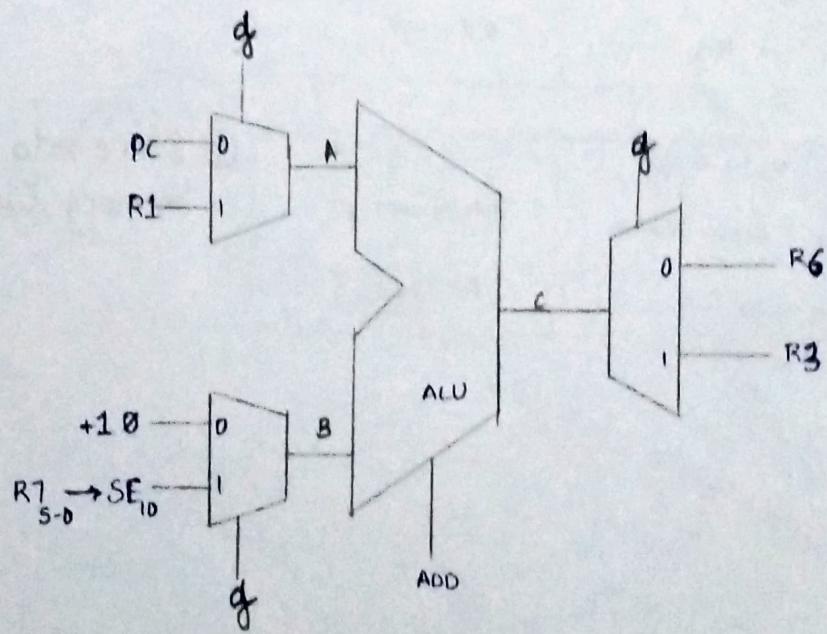
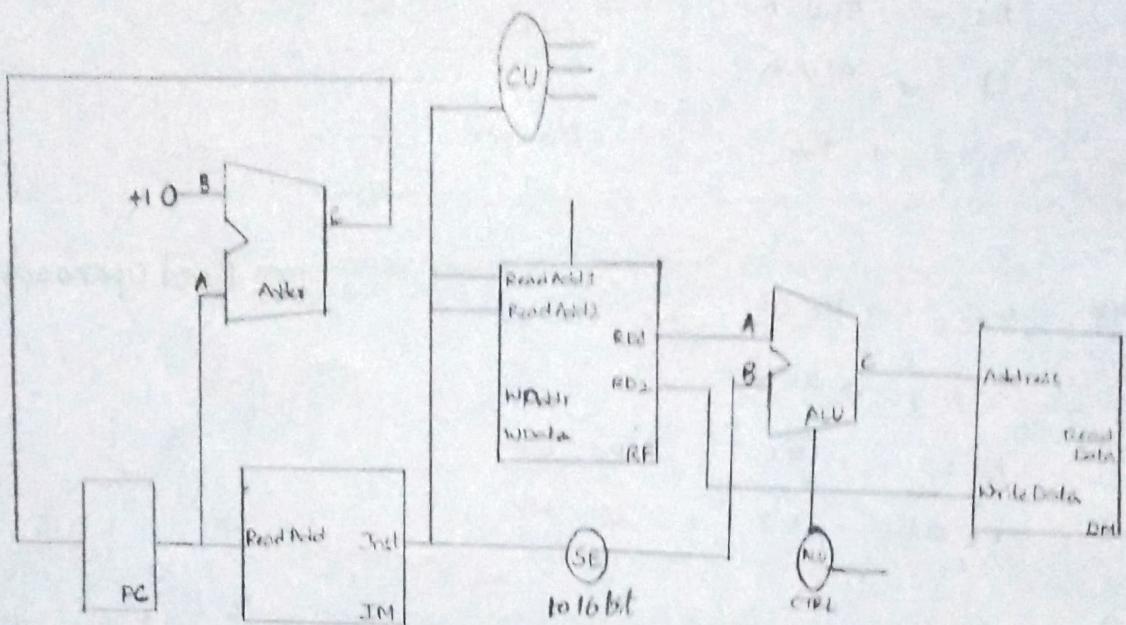
$R7_{11-9} \rightarrow RF-A3$	RF-WR
$R4 \rightarrow RF-D3$	
$R6 \rightarrow PC$	PC-WR

(5) update



## 5. STORE (SW)

I-type Instruction



Opcode	Reg A	Reg B	Immediate
4 bit 15-12	3 bit 11-9	3 bit 8-6	6 bit 5-0

S19 PC → Mem Add MemData → R7 PC → ALU-A $+1 \rightarrow$ ALU-B ALU-C → R6	Mem-Read R7 - WR ADD R6 - WR	① Read Instruction
-------------------------------------------------------------------------------------------	---------------------------------------	--------------------

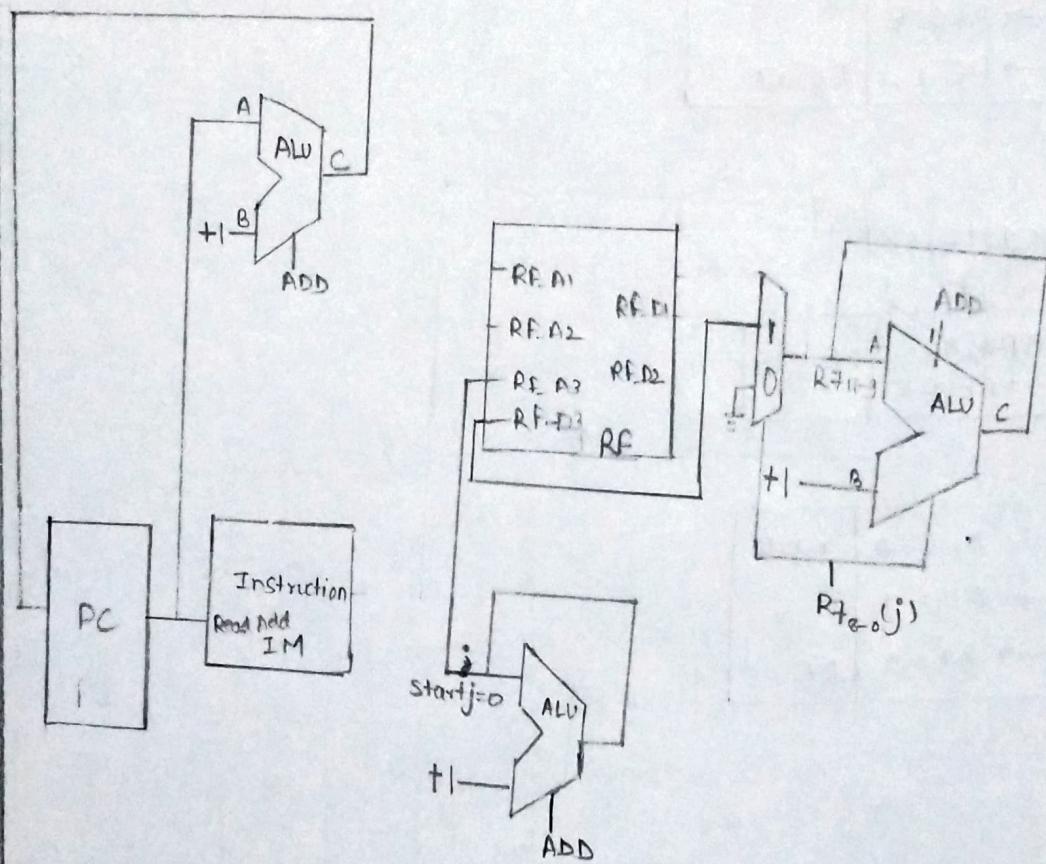
S20 $R7_{8-6} \rightarrow RF\ A1$ $R7_{11-9} \rightarrow RF\ A2$ $RF\ D1 \rightarrow R1$ $RF\ D2 \rightarrow R2$	R1 - WR R2 - WR	② Read Operands
------------------------------------------------------------------------------------------------------------------------------	--------------------	-----------------

S21 $R1 \rightarrow ALU\text{-}A$ $R7_{5-0} \rightarrow SE\ 6 \rightarrow ALU\text{-}B$ $ALU\text{-}C \rightarrow R3$	SE, ADD R1 - WR	③ Execute operation
--------------------------------------------------------------------------------------------------------------------------------	--------------------	---------------------

S22 $R1 \rightarrow Mem\ Add$ $R2 \rightarrow Mem\ Data$ $R6 \rightarrow PC$	Mem-WR PC-WR	④ Store into memory & update
---------------------------------------------------------------------------------------	-----------------	------------------------------

## 6. LOAD MULTIPLE (LM)

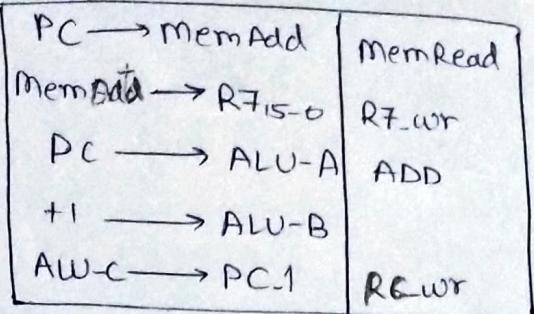
LM:	Opcode	RegisterA	0+8bit Immediate
	01_10	(3bit)	(9bit)



## 6. LOAD MULTIPLE (J-type)

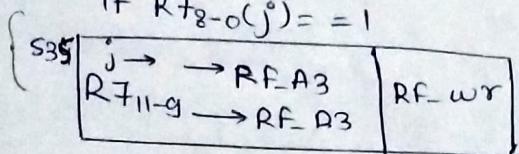
15-12	11-9	8-0
Op code	RA	Immediate

S34

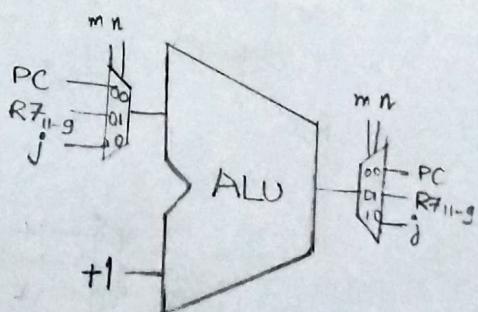
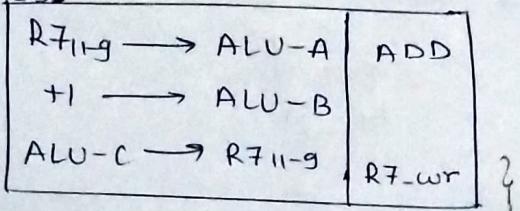


for (j=0, j++, j<8)

if R7<sub>8-0</sub>(j) == 1

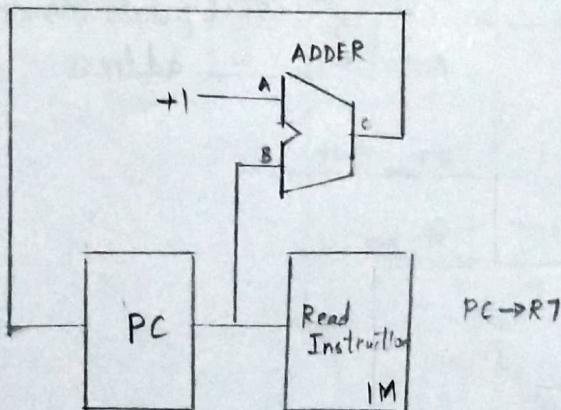
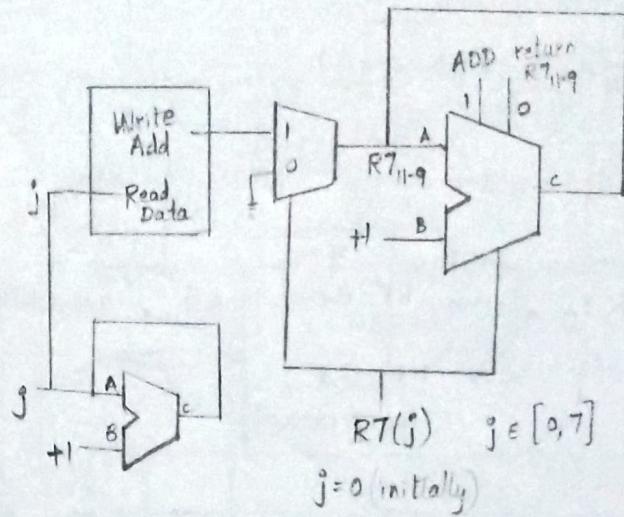
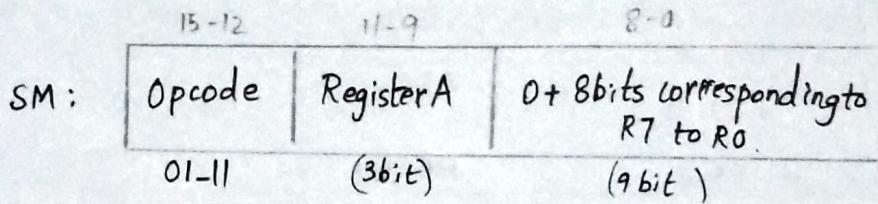


S36



## 7. STORE MULTIPLE (SM)

J-Type Instruction



S 37

$PC \rightarrow \text{Mem Add}$	$\text{Mem - Read}$
$\text{Mem Data} \rightarrow R7$	$R7 - \text{WR}$
$PC \rightarrow \text{ALU-A}$	$\text{ADD}$
$+1 \rightarrow \text{ALU-B}$	
$\text{ALU-C} \rightarrow PC-1$	$PC - \text{WR}$

(1) Read Instruction

for ( $j = 0$ ;  $j++$ ;  $j < 8$ )if  $R7(j) == 1$ ,

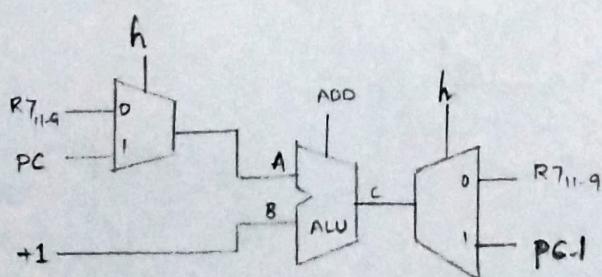
S 38

$R7_{11-9} \rightarrow RF-A3$	$R7_{11-9} - \text{Read}$
$j \rightarrow RF-D3$ (wr-data)	

(2) Store register  
to memory address

S 39

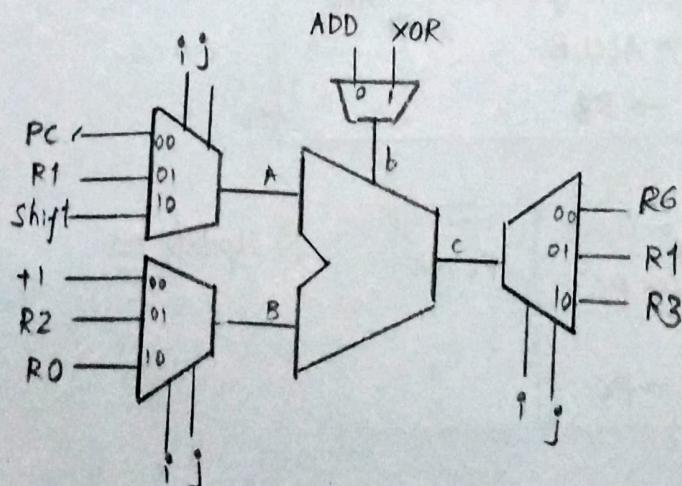
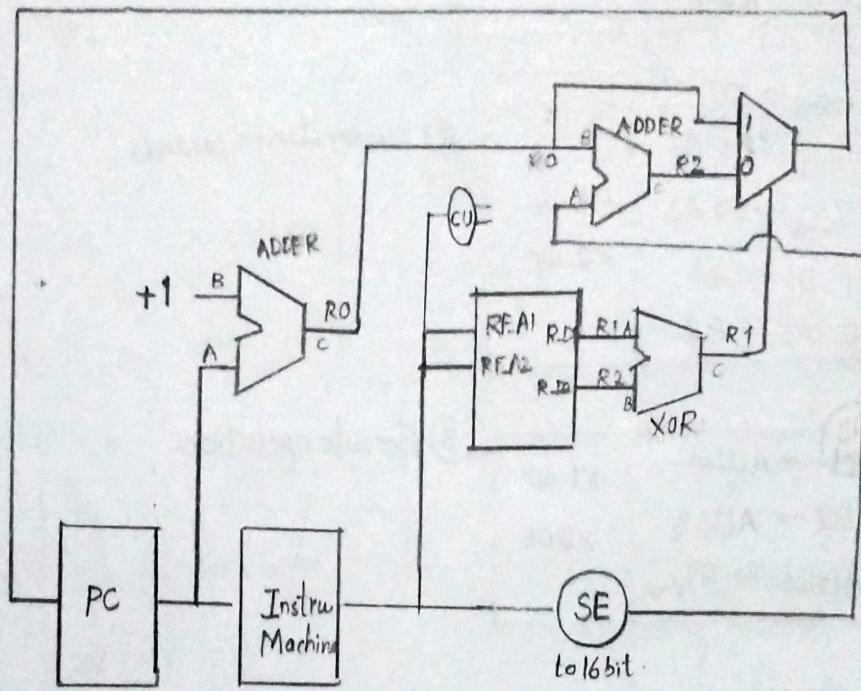
$R7_{11-9} \rightarrow \text{ALU-A}$	
$+1 \rightarrow \text{ALU-B}$	$\text{ADD}$
$\text{ALU-C} \rightarrow R7_{11-9}$	$R7_{11-9} - \text{WR}$
$PC-1 \rightarrow PC$	$PC - \text{WR}$

(3) Update Memory  
address

Opcode	Reg A	Immediate
18-12 (4-bit)	11-9 (3-bit)	8-0 (9-bit)

### 8. Branch on Equality (I type).

BEQ :	15-12	11-9	8-6	5-0
	Opcode	RegisterA	RegisterB	Immediate
	11_00	(8bit)	(3bit)	(6bit)



S23

$PC \rightarrow \text{Mem Add}$	
$\text{Mem Data} \rightarrow R7_{15-0}$	$\text{Mem Read}$
$PC \rightarrow \text{ALU-A}$	$R7\_wr$
$+1 \rightarrow \text{ALU-B}$	$\text{ADD}$
$\text{ALU-C} \rightarrow R6\_{1}$	$R6\_wr$

① Read Instruction

S24

$R7_{15-12} \rightarrow CU$	
$R7_{11-9} \rightarrow RF.A1$	
$R7_{8-6} \rightarrow RF.A2$	$R1\_wr$
$RF\_D1 \rightarrow R1$	$R2\_wr$
$RF\_D2 \rightarrow R2$	

② Understand Operands

S25

$R1 \rightarrow \text{ALU-A}$	$R1\_wr$
$R2 \rightarrow \text{ALU-B}$	$XOR$
$\text{ALU-C} \rightarrow R1_{0-0}$	

③ Execute operation

S26

$R7_{50} \rightarrow SE_6 \rightarrow \text{Shift left by } 0$	$\rightarrow \text{ALU-A}$
$R6 \rightarrow \text{ALU-B}$	$\text{ADD}$
$\text{ALU-C} \rightarrow R3$	

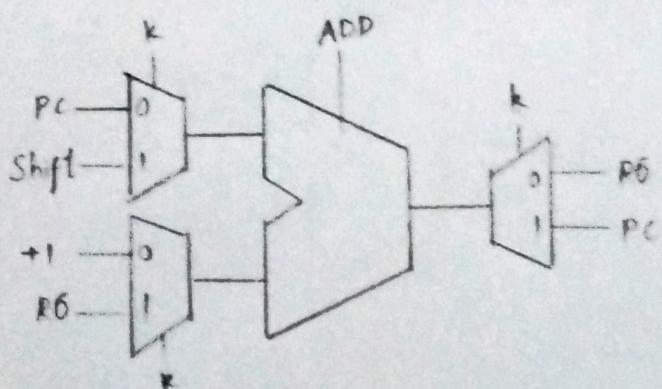
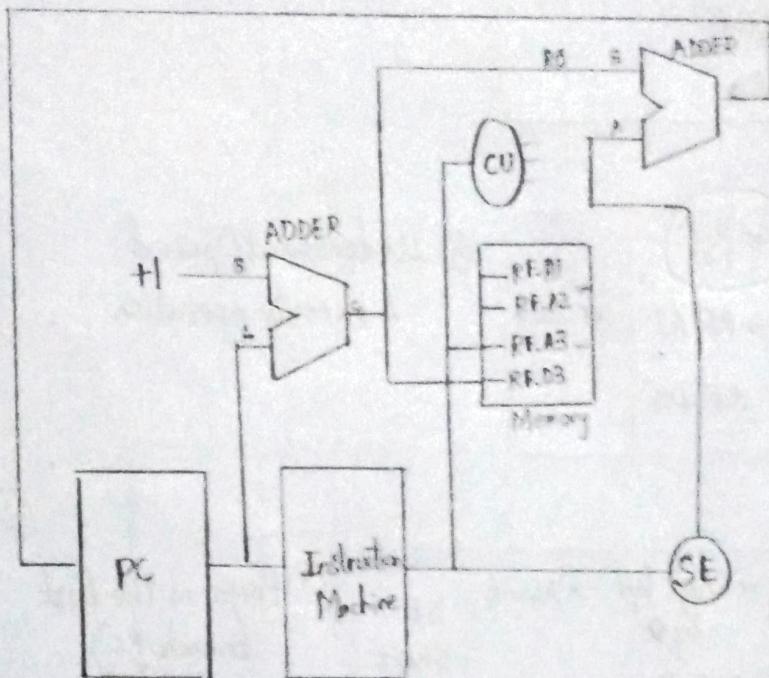
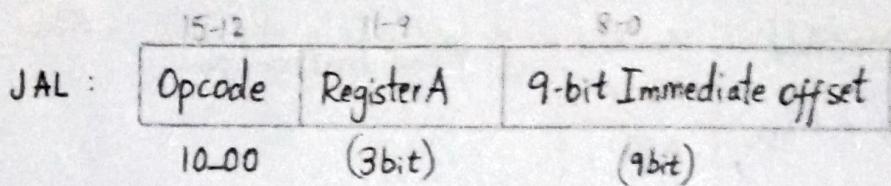
④ Branch PC + Imm

S27

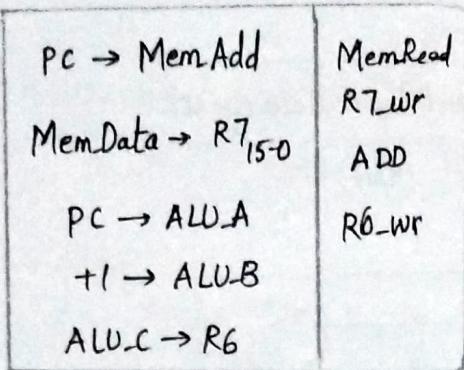
$\text{if } R1_{0-0} == 0$	$\text{PC\_wr}$
$R2 \rightarrow \text{PC}$	
$\text{else}$ $R6 \rightarrow \text{PC}$	

⑤ Update PC.

## 9. Jump & Link (J-type)

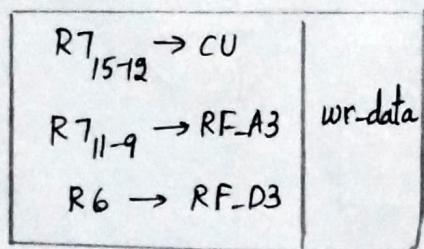


S28



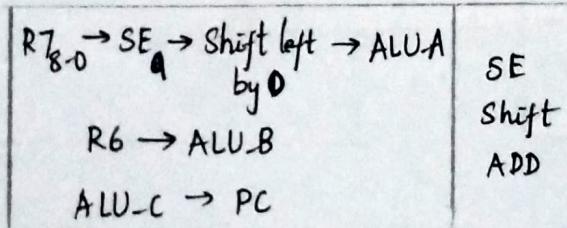
① Read Instruction

S29



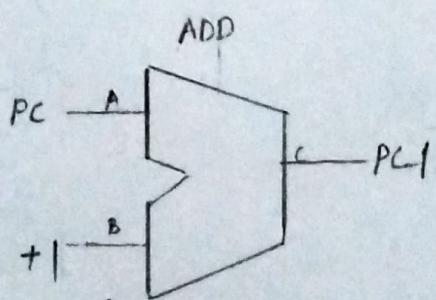
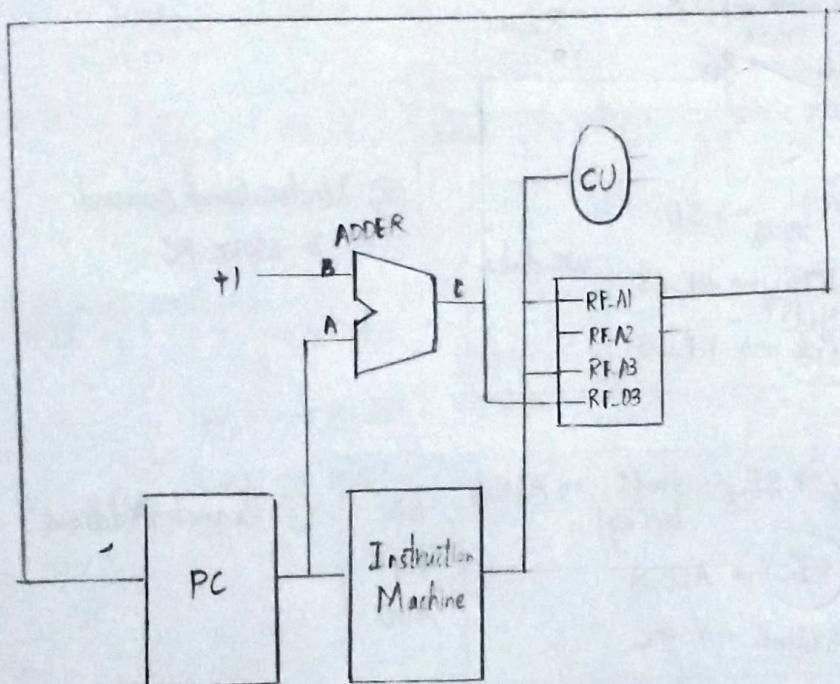
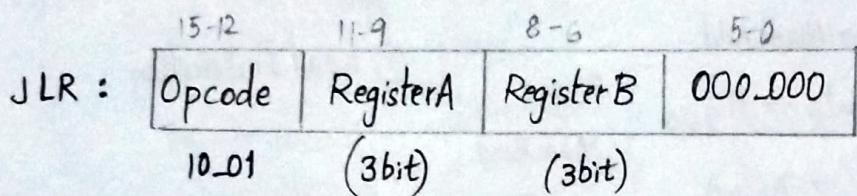
② Understand Operand  
& execute operation

S30



③ Perform the task  
(Branch PC  
+ Imm)

## 10. Jump & Link Register (I-type)



S31

$PC \rightarrow Mem\_Add$	
$Mem\_Data \rightarrow R7_{15-50}$	R7\_WR
$PC \rightarrow ALU\_A$	Mem\_Read
$+1 \rightarrow ALU\_B$	ADD
$ALU\_C \rightarrow R6$	R6\_WR

① Read Instruction

S32

$R7_{15-12} \rightarrow CU$	
$R7_{11-9} \rightarrow RF\_A3$	wr\_data
$R6 \rightarrow RF\_D3$	

② Understand Operand  
→ store PC

S33

$R7_{8-6} \rightarrow SE_{13} \rightarrow Shift$ <small>left by 1</small>	$\rightarrow ALU\_A$
$R6 \rightarrow ALU\_B$	SE Shift
$ALU\_C \rightarrow PC$	ADD

③ Branch Address

S31

PC → Mem\_Add

Mem\_Data → R7<sub>15:0</sub>

PC → ALU\_A

+1 → ALU\_B

ALU\_C → PC-1

R7WR

Mem\_Read

ADD

PC1WR

S32

R7<sub>8-6</sub> → REA1

read

RF\_D1 → R1

S33.

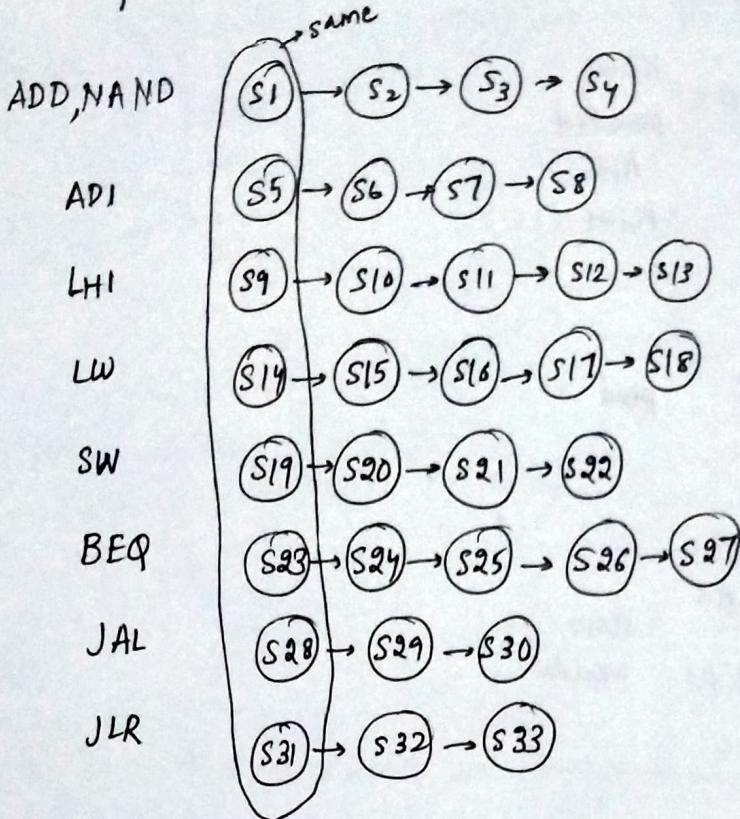
R7<sub>14:9</sub> → RF\_A3

store/  
wr.data

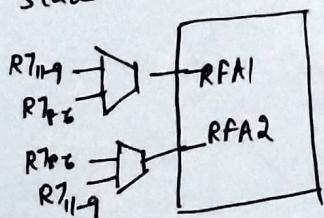
PC-1 → RF\_D3

R1 → PC

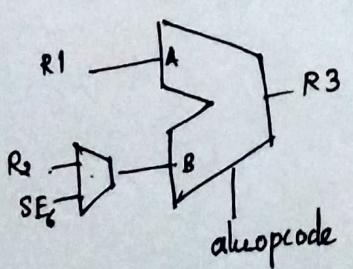
States for the instructions.



for 2<sup>nd</sup> state

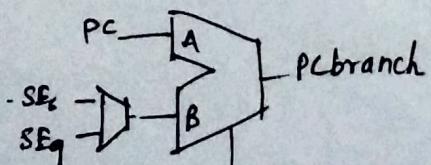


RF-ALA2sel = '1' load, store  
'0' others.



aluB.sel = '0' ADD, NAND, BEQ  
'1' ADI, LOAD, STORE  
'0' others.

aluopcode - "00" ADD  
"01" NAND  
"11" XOR  
"00" others.



pc-branchsel. '0' BEQ, others.  
'1' JAL

S31

PC → Mem\_Add

Mem\_Data → R7<sub>15-0</sub>

PC → ALU\_A

+1 → ALU\_B

ALU\_C → PC-1

R7WR

Mem\_Req

ADD

PCWR

S32

R7<sub>8-6</sub> → REA1

read

RF\_D1 → R1

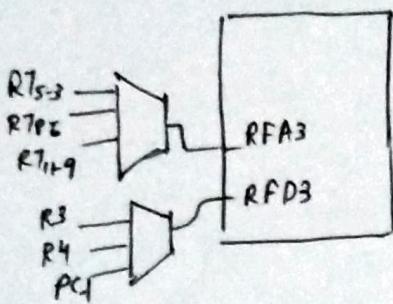
S33.

R7<sub>14-9</sub> → RF\_A3

store/  
wr\_data

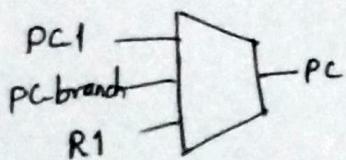
PC-1 → RF\_D3

R1 → PC



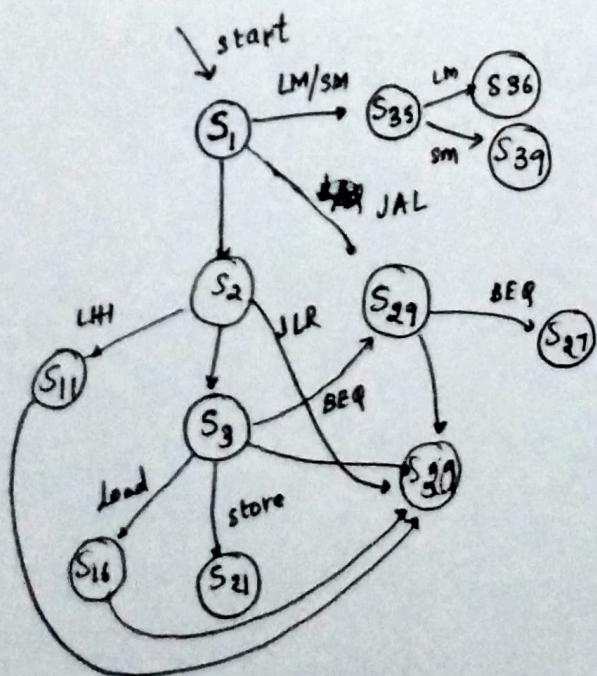
$RF_{A3\_sel}$  = "00" ADD, NAND  
"01" ADI  
"10" others.

$RF_{D3\_sel}$  "00" ADD, NAND, ADI, LH1  
"01" load  
"10" JAL, JLR



$PC\_select$  = "10" JLR  
"01" BEQ, JAL  
"00" others.

So using the select lines we can combine the 2<sup>nd</sup> states of the instruction other than S29, as they all are RF.



On reset  
next state =  $S_1$   
@ any stage of FSM.

The end stages of FSM will have their next state as  $S_1$  for the next instruction.

∴ FSM of CPU