CS 230 Project IITB RISC Report

Adish Shah, Akshay Padakanti, Swayam Chube and Vivek Veer

3rd April, 2022

Contents

1 Instructions

The basic structure of all the instructions is as follows :

R Type :	op-code (4 bits)	RegA (3)	Reg B (3)	Reg C (3)	Unused(1)	Condition (2)

I Type : Immediate (8)	op-code (4 bits)	RegA (3)	Reg B (3)

J Type : Immediate (9 bit signed)	op-code (4 bits)	RegA (3)

The implementation is as follows :

1. Sending the Program Counter to memory and fetching the instruction and incrementing PC
2. Decode the instructions and read the registers. In case of Jump and BEQ, we can also calculate the target address by adding immediate to PC.
3. Then, operations are performed on the read values using the ALU. Also immediate is added to the base address in this step.
4. In case of load/store, memory is read/written to in this step.
5. The result is stored back in the Register File.

1.1 Finite State Machine

1.1.1 Instructions : ADD,ADC,ADZ,NDU,NDC,NDZ

States :

PC → Mem_A Mem_D → IR	Mem-Read IR-Write

Table 1: S0 : Read Instruction

PC → ALU_A +1 → ALU_B ALU_C → PC	PC-Write ALU_ADD

Table 2: S00 : Increment PC

IR_{11-9} → RF-A1 IR_{8-6} → RF-A2 RF-D1 → t1 RF-D2 → t2	t1-Write t2-write

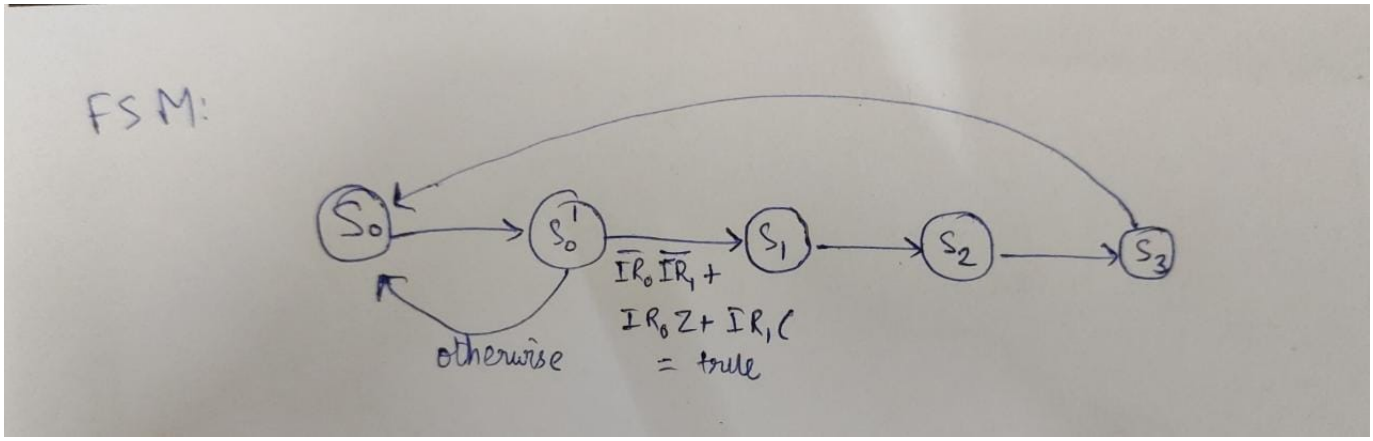
Table 3: S1 : Read Registers

t1 → ALU_A t2 → ALU_B ALU_Carry → C ALU_Zero → Z	t3-Write ALU-Operation ALU_C → t3

Table 4: S2 : Perform Operation

t3 → RF-D3 IR_{5-3} → RF-A3	t1-Write

Table 5: S3: Write to register File



The condition on S00 $->$ S1 is to ensure that the addition is done only if conditions for instruction are met. For example, ADC needs C to be set. So, if the command is ADC (identified by IR1 = 1), we need to make sure C = 1. This is done using the given formula. The condition is written in compact here, we can easily use a decoder to do the transition (use 4 bit decoder with inputs being ir0, ir1, c, z).

1.1.2 ADI

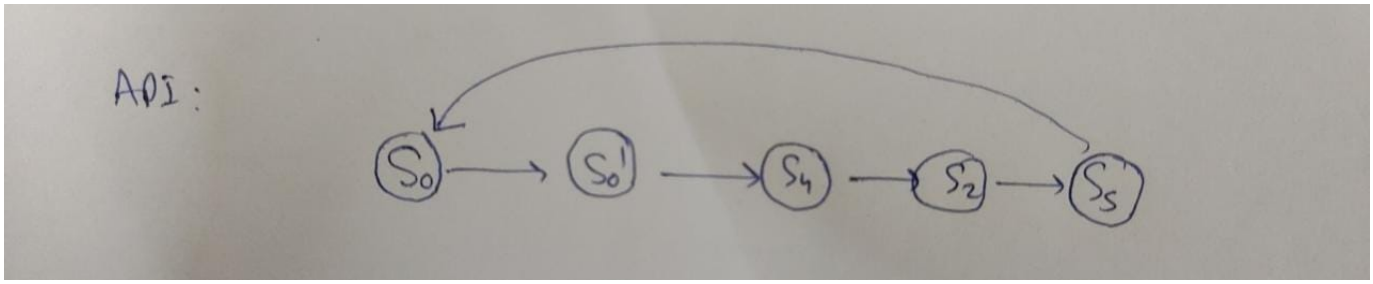
States :

$IR_{11-9} \rightarrow$ RF-A1	t1-Write
$IR_{5-0} \rightarrow SE_6$	t2-write
RF-D1 $\rightarrow$ t1	
$SE_6 \rightarrow$ t2	

Table 6: S4 : Read Register and immediate value

t3 $\rightarrow$ RF-D3	RF-Write
$IR_{8-6} \rightarrow$ RF-A3	

Table 7: S5: Write to register File



1.1.3 LHI

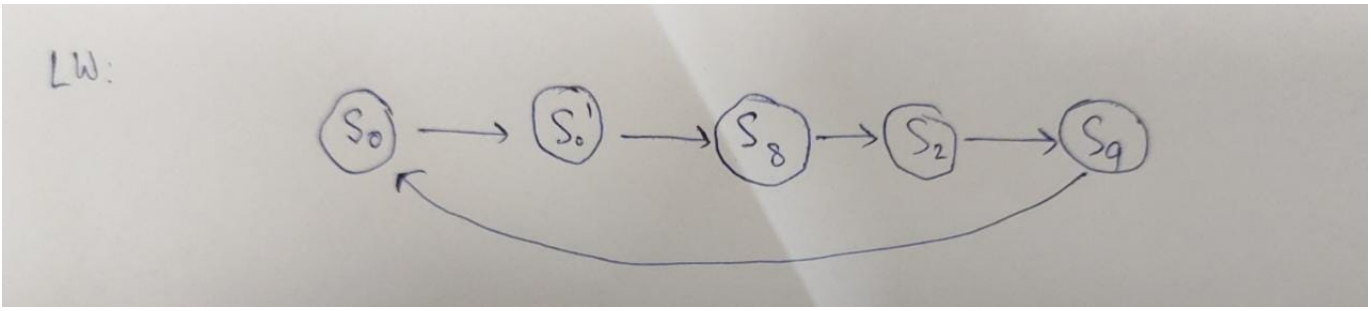
States :

$IR_{8-0} \rightarrow SE_9$	7 bit shifter
$SE_9 \rightarrow$ ALU_A	t3-Write
ALU_C $\rightarrow$ t3	

Table 8: S6

t3 $\rightarrow$ RF-D3	RF-Write
$IR_{11-9} \rightarrow$ RF-A3	

Table 9: S7: Write to RF



1.1.4 LW

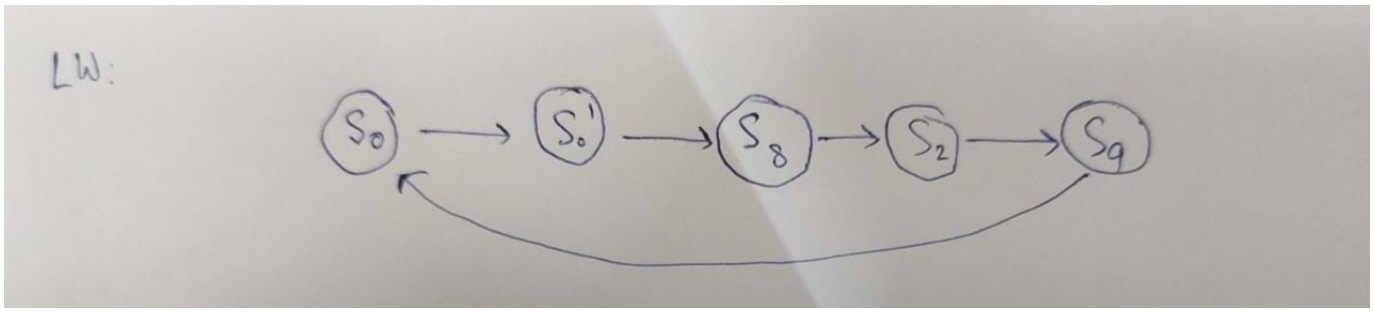
States :

$IR_{8-6} \rightarrow$ RF-A2	t1-Write
$IR_{5-0} \rightarrow SE_6$	t2-write
RF-D2 $\rightarrow$ t2	
$SE_6 \rightarrow$ t1	

Table 10: S8 : Calculating Address

t3 $\rightarrow$ Mem_A	Mem-Read
Mem_D $\rightarrow$ RF-D3	RF-Write
$IR_{11-9} \rightarrow$ RF-A3	

Table 11: S9: Load

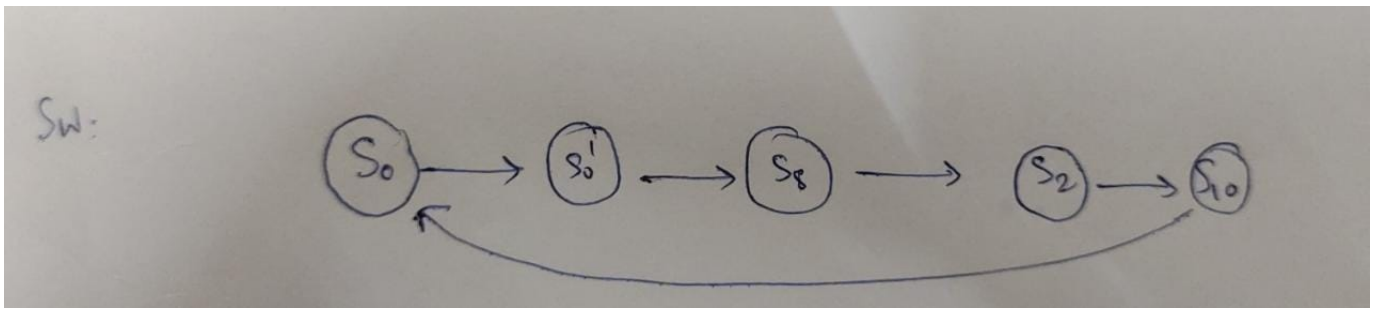


1.1.5 SW

States :

t3 $\rightarrow$ Mem_A	Mem-write
$IR_{11-9} \rightarrow$ RF-A RF-D $\rightarrow$ Mem_D	

Table 12: S10: writing to memory



1.1.6 LM

States :

$IR_{11-9} \rightarrow$ RF-A1	t1-Write
$IR_{8-0} \rightarrow SE_9$	t2-write
$SE_9 \rightarrow$ t3	t3-write
RF-D1 $\rightarrow$ t1	
000 *rightarrow* t2	

Table 13: S11 : Read memory address, immediate and initialise counter

if (immediate bit of register is set)	
t1 $\rightarrow$ Mem_A	Mem-Read
Mem_D $\rightarrow$ RF-D3	RF-Write
t2 $\rightarrow$ RF-A3	

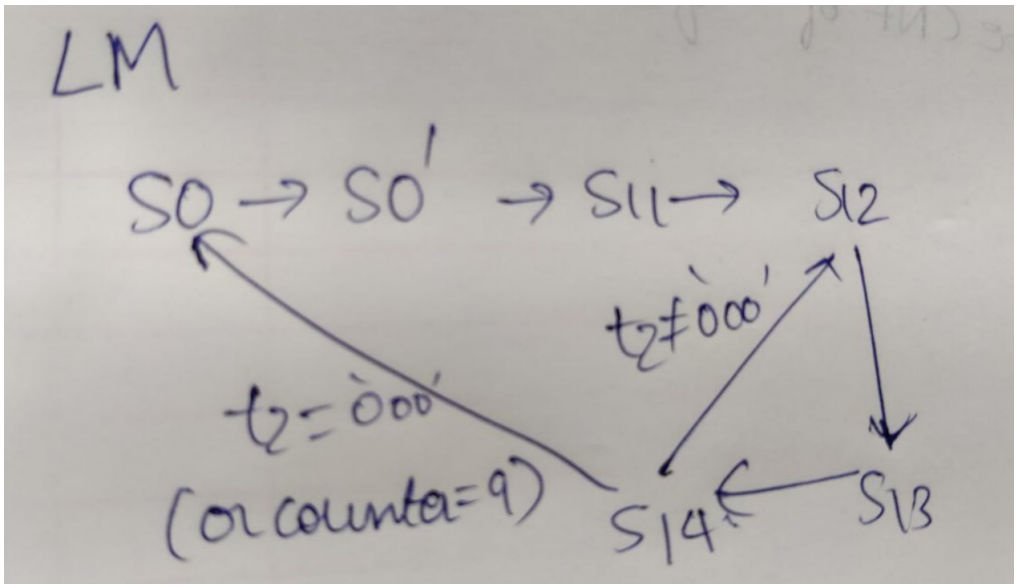
Table 14: S12 : read memory and write to RF

t1 $\rightarrow$ ALU_A	t1-Write
+1 $\rightarrow$ ALU_B	ALU_ADD
ALU_C $\rightarrow$ t1	

Table 15: S13 : Increase memory address

t2 $\rightarrow$ ALU_A	t2-Write
+1 $\rightarrow$ ALU_B	ALU_ADD
ALU_C $\rightarrow$ t2	

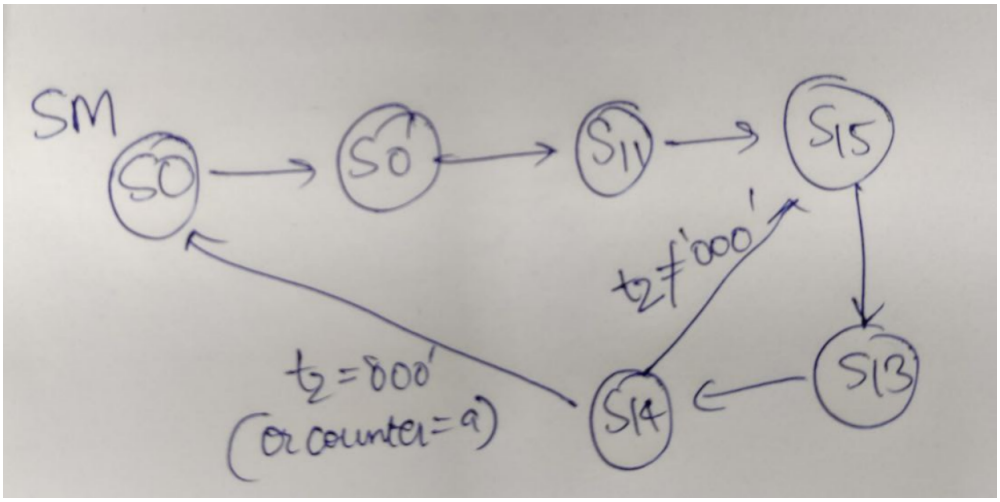
Table 16: S14 : Increase counter



1.1.7 SM

if (immediate bit of register is set) t2 → RF-A1 RF-D1 → Mem_D t1 → Mem_A	Mem-write

Table 17: S15: writing to memory



1.1.8 BEQ

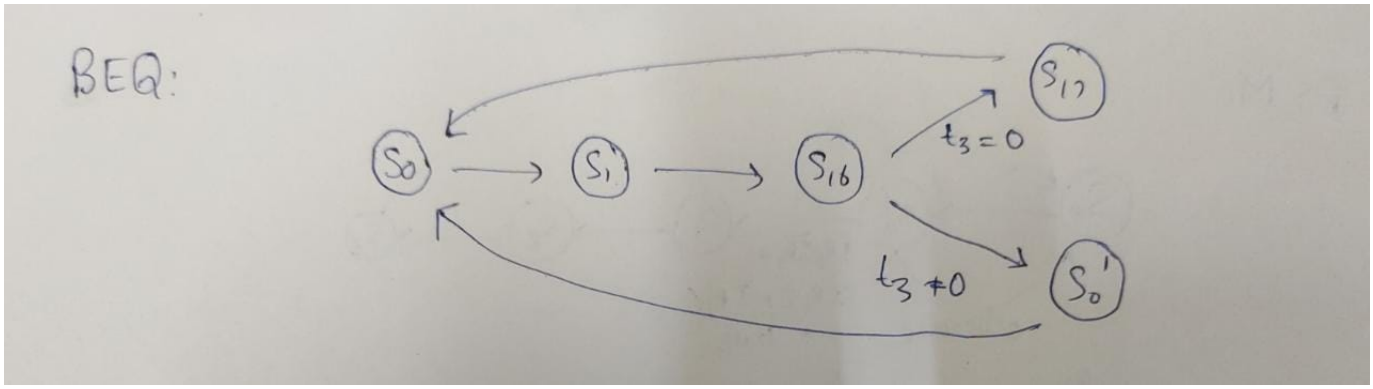
States :

t1 → ALU_A t2 → ALU_B ALU_C → t3	t3-write ALU_SUB

Table 18: S16 : Increment PC

PC → ALU_A $IR_{5-0} \rightarrow SE_6$ SE_6 → ALU_B ALU_C → PC	PC-Write ALU_ADD

Table 19: S17 : Increment PC



1.1.9 JAL

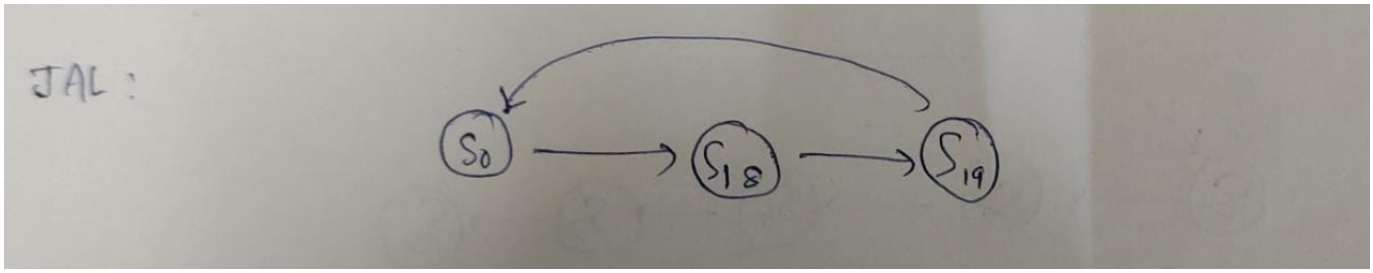
States :

$IR_{11-9} \rightarrow$ RF-A3 PC $\rightarrow$ RF-D3	RF-Write

Table 20: S18: Store PC

PC $\rightarrow$ ALU_A $IR_{8-0} \rightarrow SE_9$ $SE_9 \rightarrow$ ALU_B ALU_C $\rightarrow$ PC	PC-Write ALU_ADD

Table 21: S19 : Increment PC

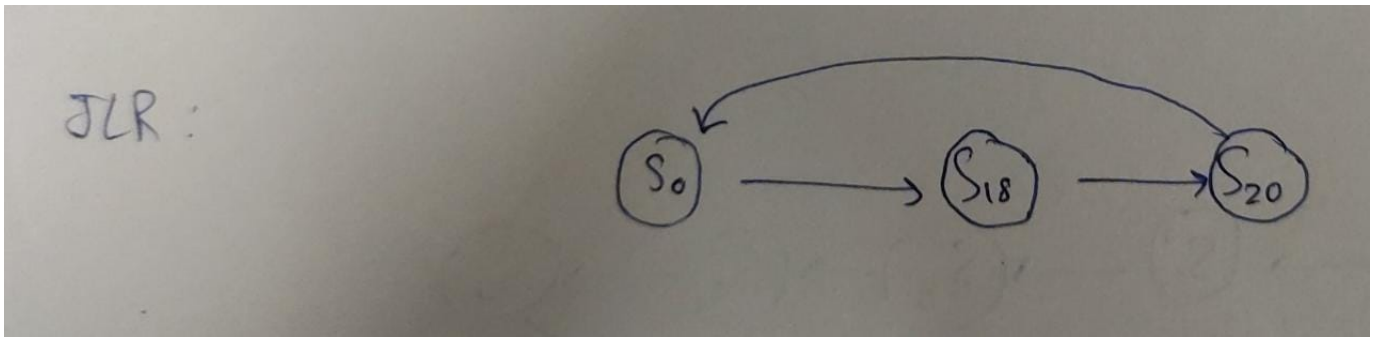


1.1.10 JLR

States :

$IR_{8-6} \rightarrow$ RF-A1 RF-D1 $\rightarrow$ PC	PC-Write

Table 22: S20: Store PC



1.1.11 JRI

States :

$IR_{11-9} \rightarrow$ RF-A1 RF-D1 $\rightarrow$ t3	RF-Write t3-Write

Table 23: S21: Store PC

t3 $\rightarrow$ ALU_A if $iR(8) == 0$ "000000" & $IR_{8-0} \rightarrow$ ALU_B else "111111" & $IR_{8-0} \rightarrow$ ALU_B ALU_C $\rightarrow$ PC	ALU-ADD PC-Write

Table 24: S22: Store PC