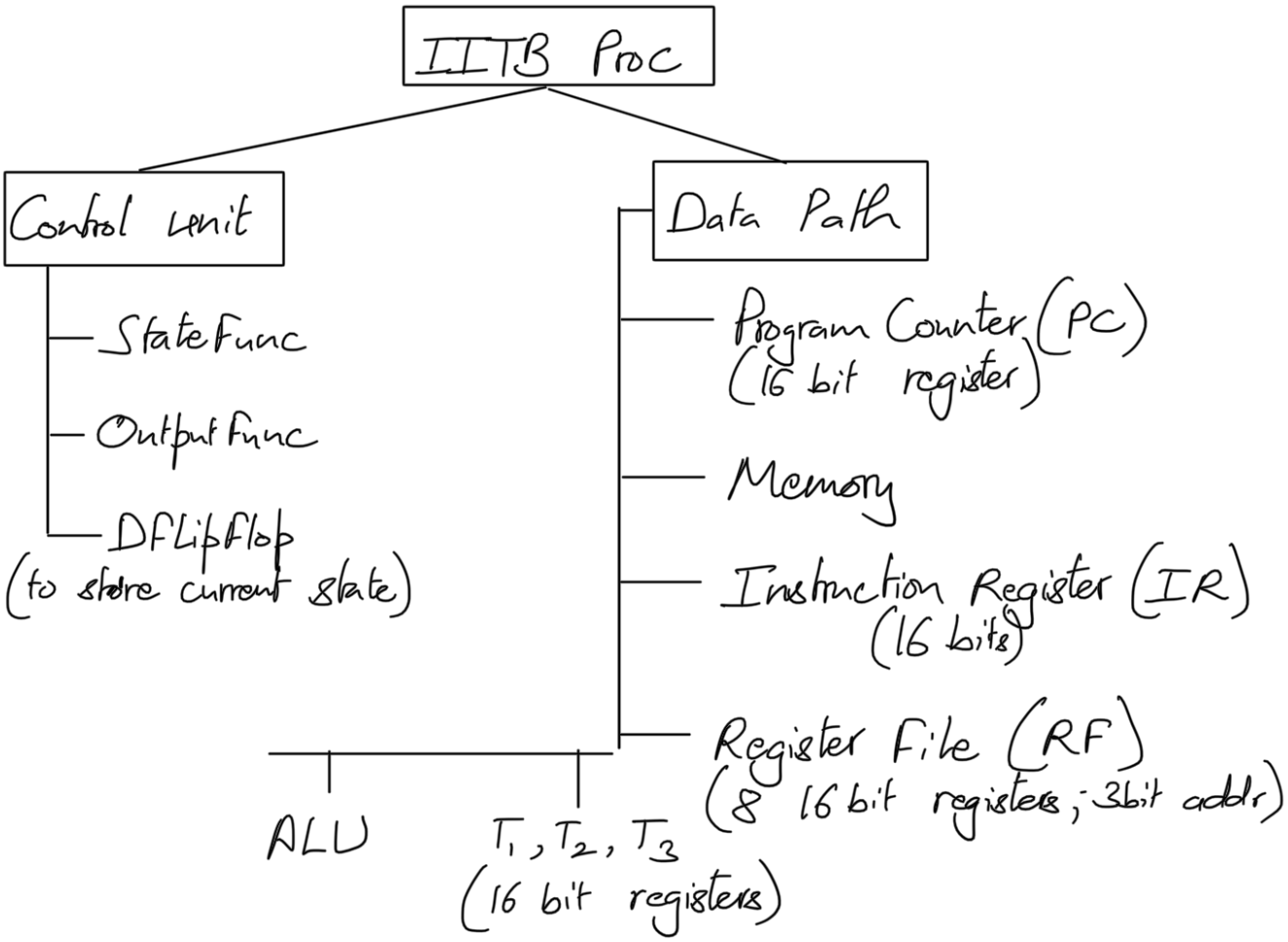
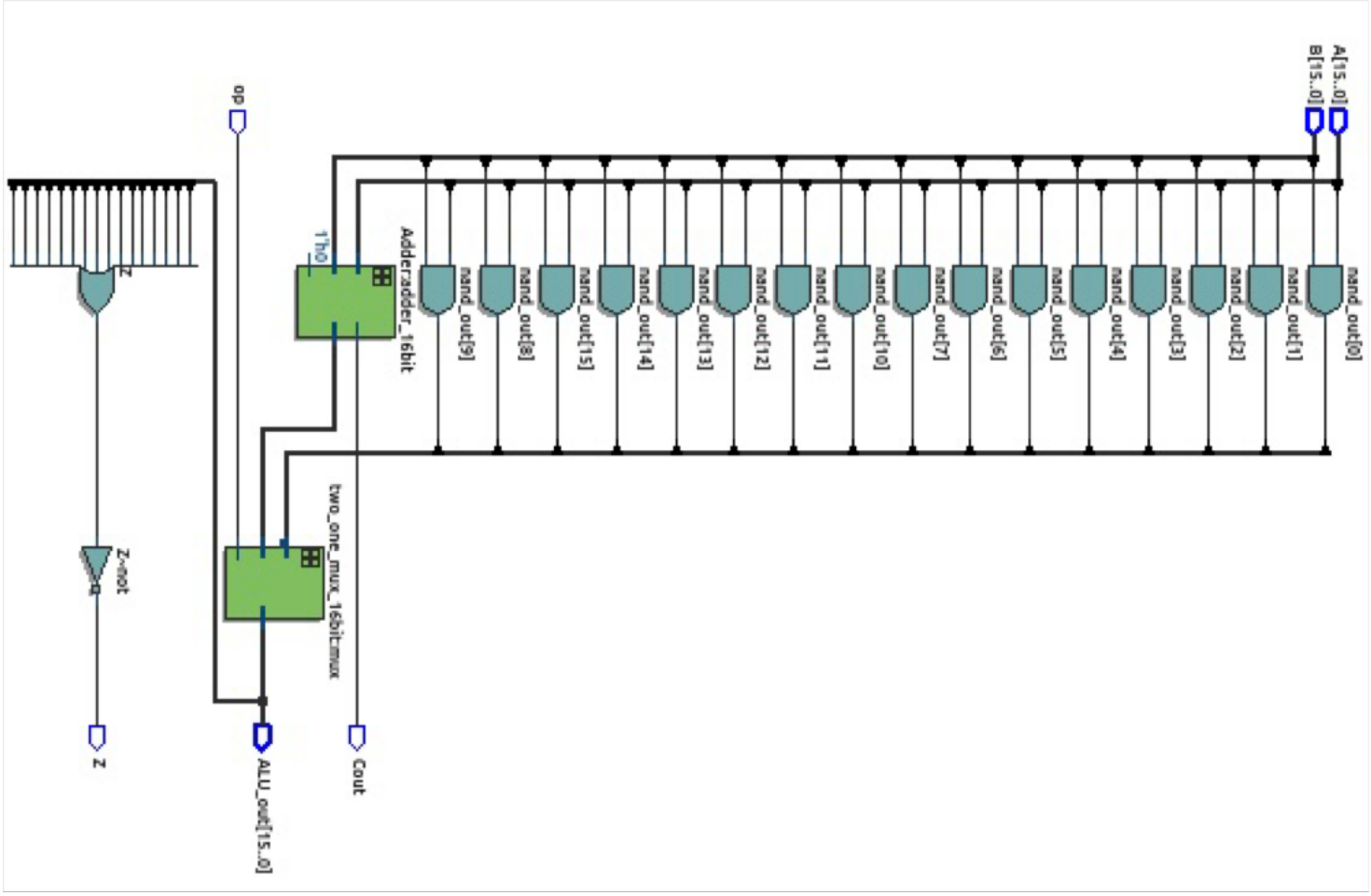
IITB-Proc
Course Project

Bhavnoor Singh Marok - 190050027
Jayesh Singla - 190050053
Shrey Singla - 190050114
Tanu Goyal - 190050123

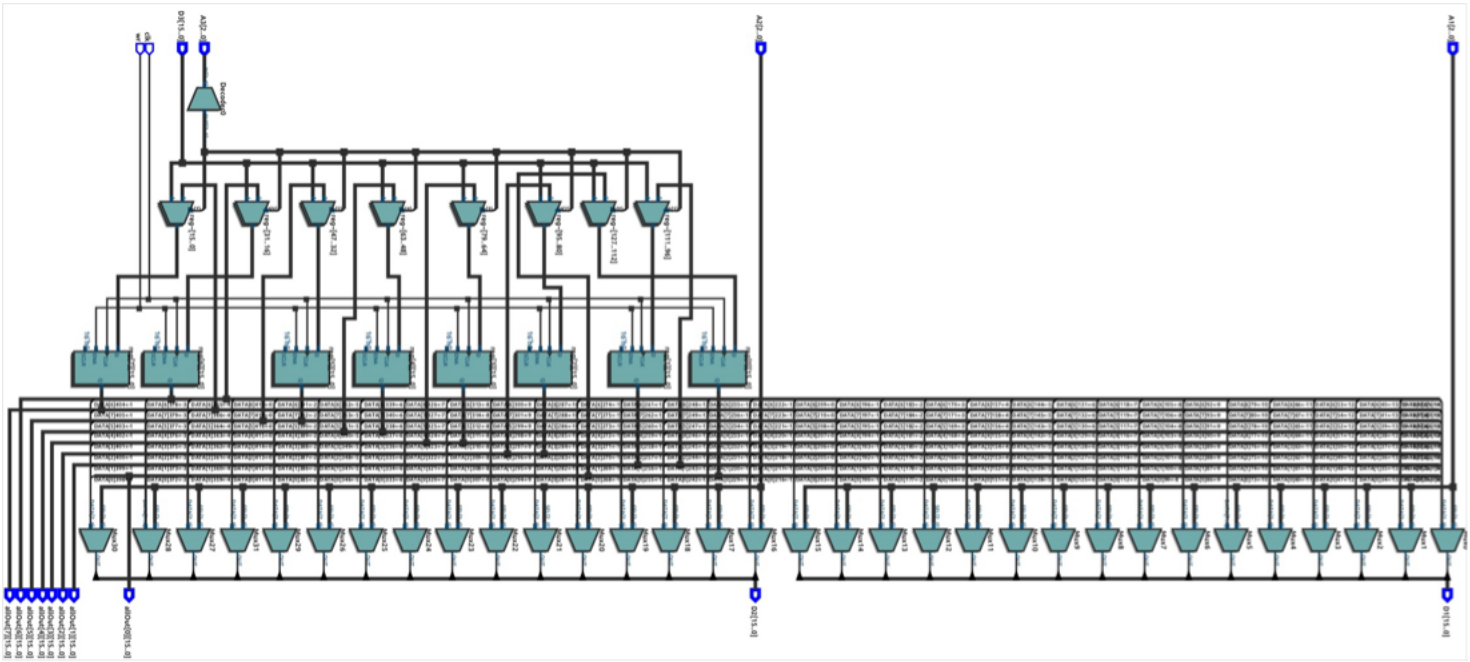
Components :-



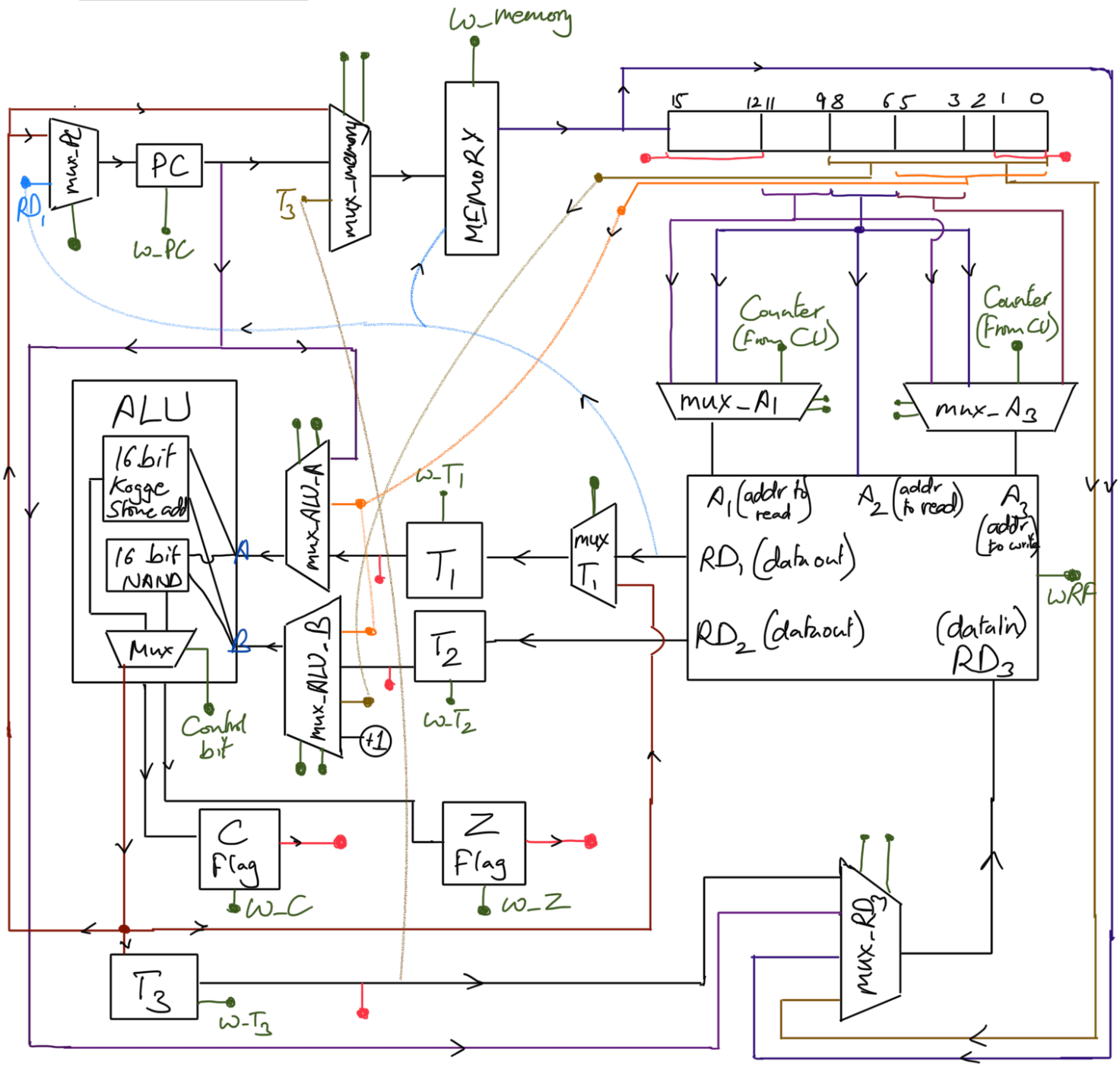
ALU :-



Register File :-



Data Path :-

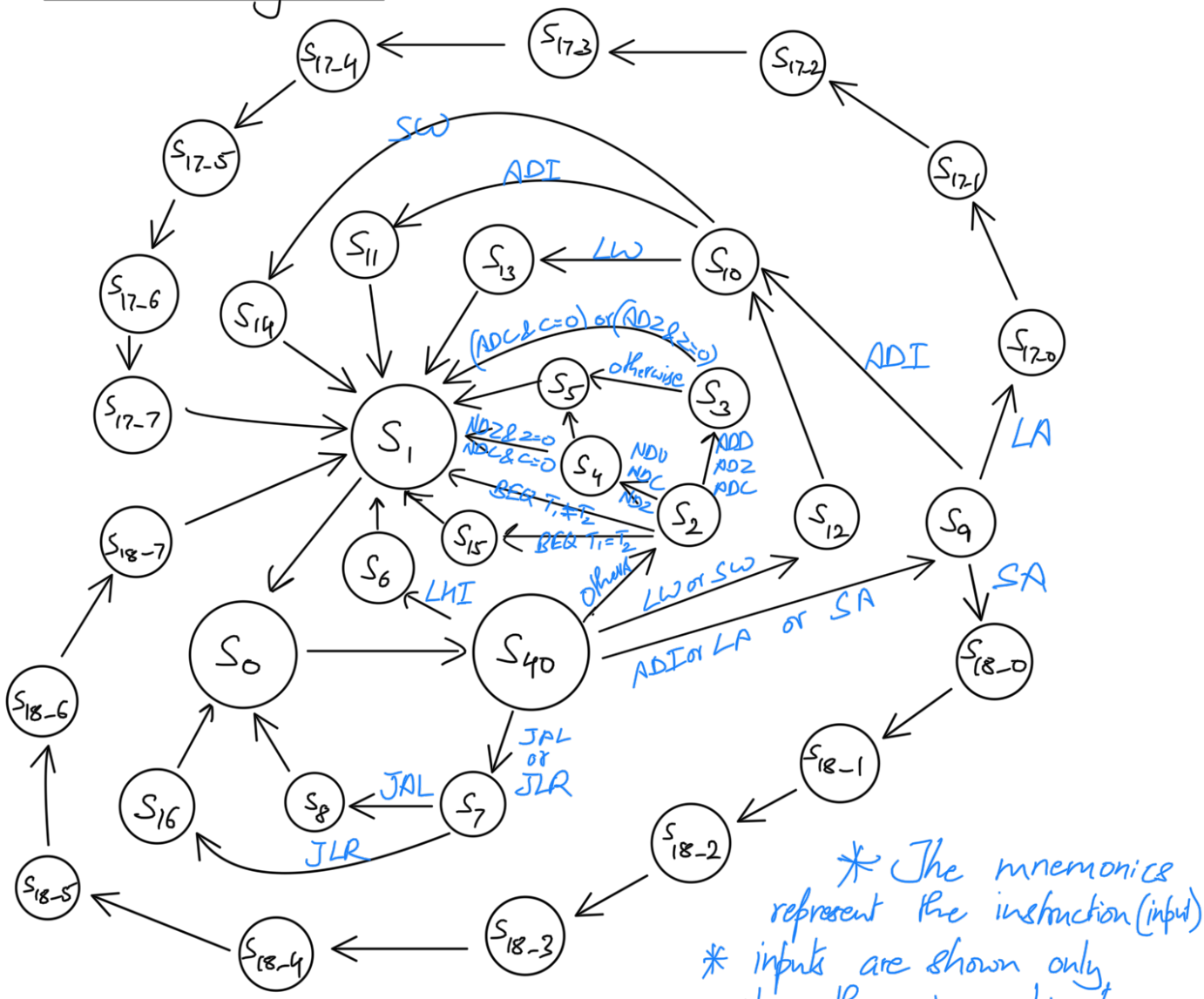


Indicate outputs to control unit

Instruction, C-flag, Z-flag, T_1, T_2, T_3 are given as Input to control unit.

Indicates inputs from control unit

State Diagram:-



* The mnemonics represent the instruction (input)
* inputs are shown only where there is ambiguity.

State Outputs:-

States	mux_PC	w_PC	mux_memory	w_memory	w_IR	mux_A1	mux_T1	w_T1	w_T2	mux_ALU_A	mux_ALU_B	Control bit ALU	w_T3	mux_A3	mux_RD3	w_RF	w_C	w_Z	Counter
s0	-	-	--	-	-	--	-	-	-	--	--	-	-	--	--	-	-	-	---
s1	0	1	--	0	0	--	-	0	0	10	00	1	0	--	--	0	0	0	---
s2	-	0	--	0	0	00	0	1	1	--	--	-	0	--	--	0	0	0	---
s3	-	0	--	0	0	--	-	0	0	01	01	1	1	--	--	0	1	1	---
s4	-	0	--	0	0	--	-	0	0	01	01	0	1	--	--	0	0	1	---
s5	-	0	--	0	0	--	-	0	0	--	--	-	0	11	01	1	0	0	---
s6	-	0	--	0	0	--	-	0	0	--	--	-	0	00	10	1	0	0	---
s7	-	0	--	0	0	--	-	0	0	--	--	-	0	00	00	1	0	0	---
s8	0	1	--	0	0	--	-	0	0	10	10	1	0	--	--	0	0	0	---
s9	-	0	--	0	0	00	0	1	0	--	--	-	0	--	--	0	0	0	---
s10	-	0	--	0	0	--	-	0	0	01	11	1	1	--	--	0	1	1	---
s11	-	0	--	0	0	--	-	0	0	--	--	-	0	01	01	1	0	0	---
s12	-	0	--	0	0	01	0	1	0	--	--	-	0	--	--	0	0	0	---
s13	-	0	10	0	0	--	-	0	0	--	--	-	0	00	11	1	0	0	---
s14	-	0	10	1	0	00	-	0	0	--	--	-	0	--	--	0	0	0	---
s15	0	1	--	0	0	--	-	0	0	10	11	1	0	--	--	0	0	0	---
s16	1	1	--	0	0	01	-	0	0	--	--	-	0	--	--	0	0	0	---
s17_0	-	0	11	0	0	--	1	1	0	01	00	1	0	10	11	1	0	0	000
s17_1	-	0	11	0	0	--	1	1	0	01	00	1	0	10	11	1	0	0	001
s17_2	-	0	11	0	0	--	1	1	0	01	00	1	0	10	11	1	0	0	010
s17_3	-	0	11	0	0	--	1	1	0	01	00	1	0	10	11	1	0	0	011
s17_4	-	0	11	0	0	--	1	1	0	01	00	1	0	10	11	1	0	0	100
s17_5	-	0	11	0	0	--	1	1	0	01	00	1	0	10	11	1	0	0	101
s17_6	-	0	11	0	0	--	1	1	0	01	00	1	0	10	11	1	0	0	110
s17_7	-	0	11	0	0	--	1	1	0	01	00	1	0	10	11	1	0	0	111
s18_0	-	0	11	1	0	10	1	1	0	01	00	1	0	--	--	0	0	0	000
s18_1	-	0	11	1	0	10	1	1	0	01	00	1	0	--	--	0	0	0	001
s18_2	-	0	11	1	0	10	1	1	0	01	00	1	0	--	--	0	0	0	010
s18_3	-	0	11	1	0	10	1	1	0	01	00	1	0	--	--	0	0	0	011
s18_4	-	0	11	1	0	10	1	1	0	01	00	1	0	--	--	0	0	0	100
s18_5	-	0	11	1	0	10	1	1	0	01	00	1	0	--	--	0	0	0	101
s18_6	-	0	11	1	0	10	1	1	0	01	00	1	0	--	--	0	0	0	110
s18_7	-	0	11	1	0	10	1	1	0	01	00	1	0	--	--	0	0	0	111
s40	-	0	01	0	1	--	-	0	0	--	--	-	0	--	--	0	0	0	---

"–" denotes dont care

State transitions for different mnemonics :-

Mnemonic	Transitions
ADD	$S_0 \rightarrow S_{40} \rightarrow S_2 \rightarrow S_3 \rightarrow S_5 \rightarrow S_1$
ADC	$S_0 \rightarrow S_{40} \rightarrow S_2 \rightarrow S_3 \rightarrow S_5 \rightarrow S_1$ if C flag set $S_0 \rightarrow S_{40} \rightarrow S_2 \rightarrow S_3 \rightarrow S_1$ if C flag not set
ADZ	$S_0 \rightarrow S_{40} \rightarrow S_2 \rightarrow S_3 \rightarrow S_5 \rightarrow S_1$ if Z flag set $S_0 \rightarrow S_{40} \rightarrow S_2 \rightarrow S_3 \rightarrow S_1$ if Z flag not set
ADI	$S_0 \rightarrow S_{40} \rightarrow S_9 \rightarrow S_{10} \rightarrow S_{11} \rightarrow S_1$
NDU	$S_0 \rightarrow S_{40} \rightarrow S_2 \rightarrow S_4 \rightarrow S_5 \rightarrow S_1$
NDC	$S_0 \rightarrow S_{40} \rightarrow S_2 \rightarrow S_4 \rightarrow S_5 \rightarrow S_1$ if C flag set $S_0 \rightarrow S_{40} \rightarrow S_2 \rightarrow S_4 \rightarrow S_1$ if C flag not set
NDZ	$S_0 \rightarrow S_{40} \rightarrow S_2 \rightarrow S_4 \rightarrow S_5 \rightarrow S_1$ if Z flag set $S_0 \rightarrow S_{40} \rightarrow S_2 \rightarrow S_4 \rightarrow S_1$ if Z flag not set
LHI	$S_0 \rightarrow S_{40} \rightarrow S_6 \rightarrow S_1$
LW	$S_0 \rightarrow S_{40} \rightarrow S_{12} \rightarrow S_{10} \rightarrow S_{13} \rightarrow S_1$
SW	$S_0 \rightarrow S_{40} \rightarrow S_{12} \rightarrow S_{10} \rightarrow S_{14} \rightarrow S_1$
LA	$S_0 \rightarrow S_{40} \rightarrow S_9 \rightarrow S_{17\text{-}0} \rightarrow S_{17\text{-}1} \rightarrow S_{17\text{-}2} \rightarrow S_{17\text{-}3}$ $\rightarrow S_{17\text{-}4} \rightarrow S_{17\text{-}5} \rightarrow S_{17\text{-}6} \rightarrow S_{17\text{-}7} \rightarrow S_1$
SA	$S_0 \rightarrow S_{40} \rightarrow S_9 \rightarrow S_{18\text{-}0} \rightarrow S_{18\text{-}1} \rightarrow S_{18\text{-}2} \rightarrow S_{18\text{-}3}$ $\rightarrow S_{18\text{-}4} \rightarrow S_{18\text{-}5} \rightarrow S_{18\text{-}6} \rightarrow S_{18\text{-}7} \rightarrow S_1$
BEQ	$S_0 \rightarrow S_{40} \rightarrow S_2 \rightarrow S_{15}$ if $T_1 = T_2$ $S_0 \rightarrow S_{40} \rightarrow S_2 \rightarrow S_1$ if $T_1 \neq T_2$
JAL	$S_0 \rightarrow S_{40} \rightarrow S_7 \rightarrow S_8$
JLR	$S_0 \rightarrow S_{40} \rightarrow S_7 \rightarrow S_{16}$

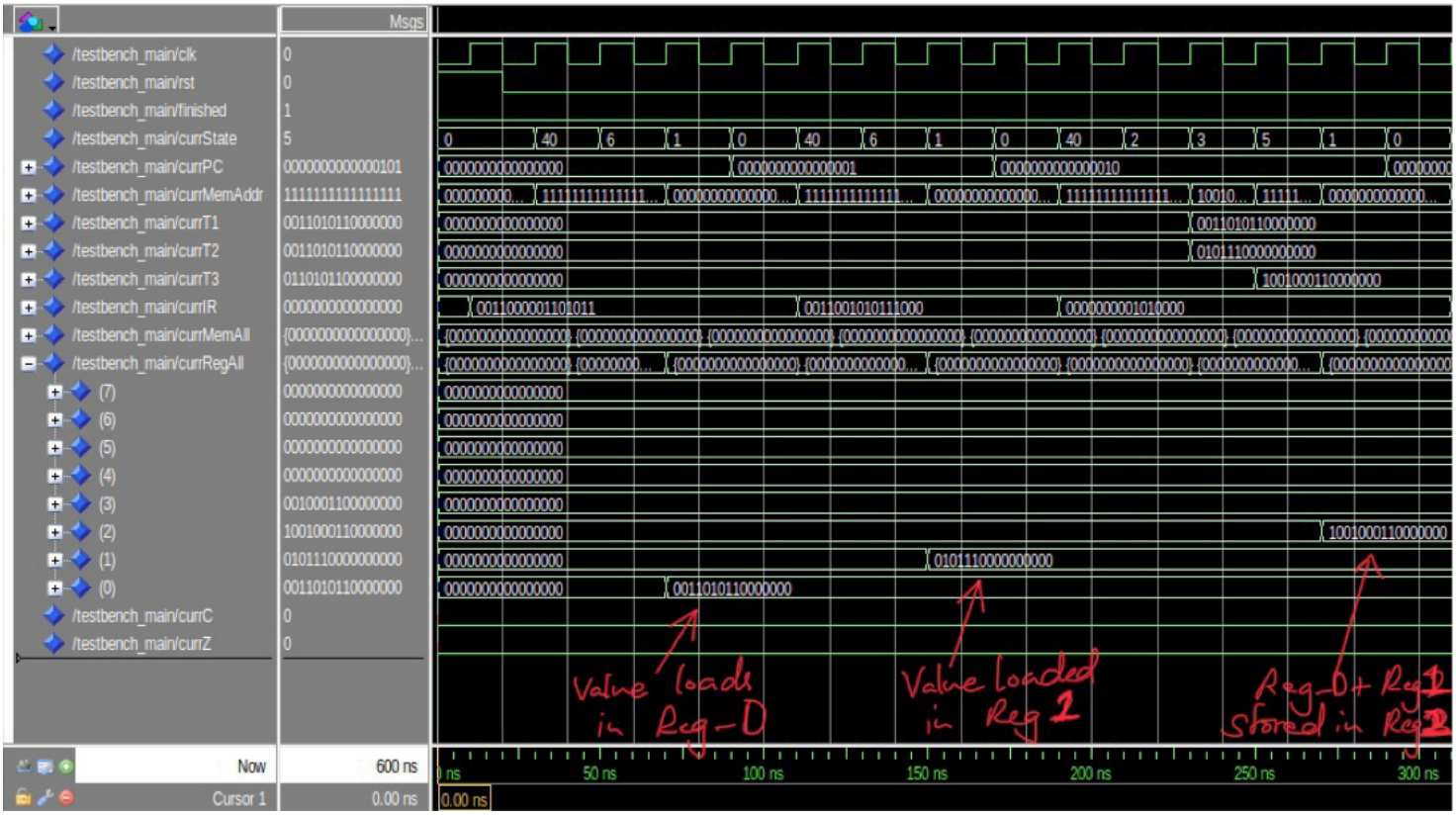
Waveforms:-

1) Instructions in memory:- x"306b", x"3268", x"0050"

Loads 0011010110000000 in Register_0 (LHI)

Loads 0101110000000000 in Register_1 (LHI)

Add Reg_0 & Reg_1 & store result in Reg_2 (ADD)



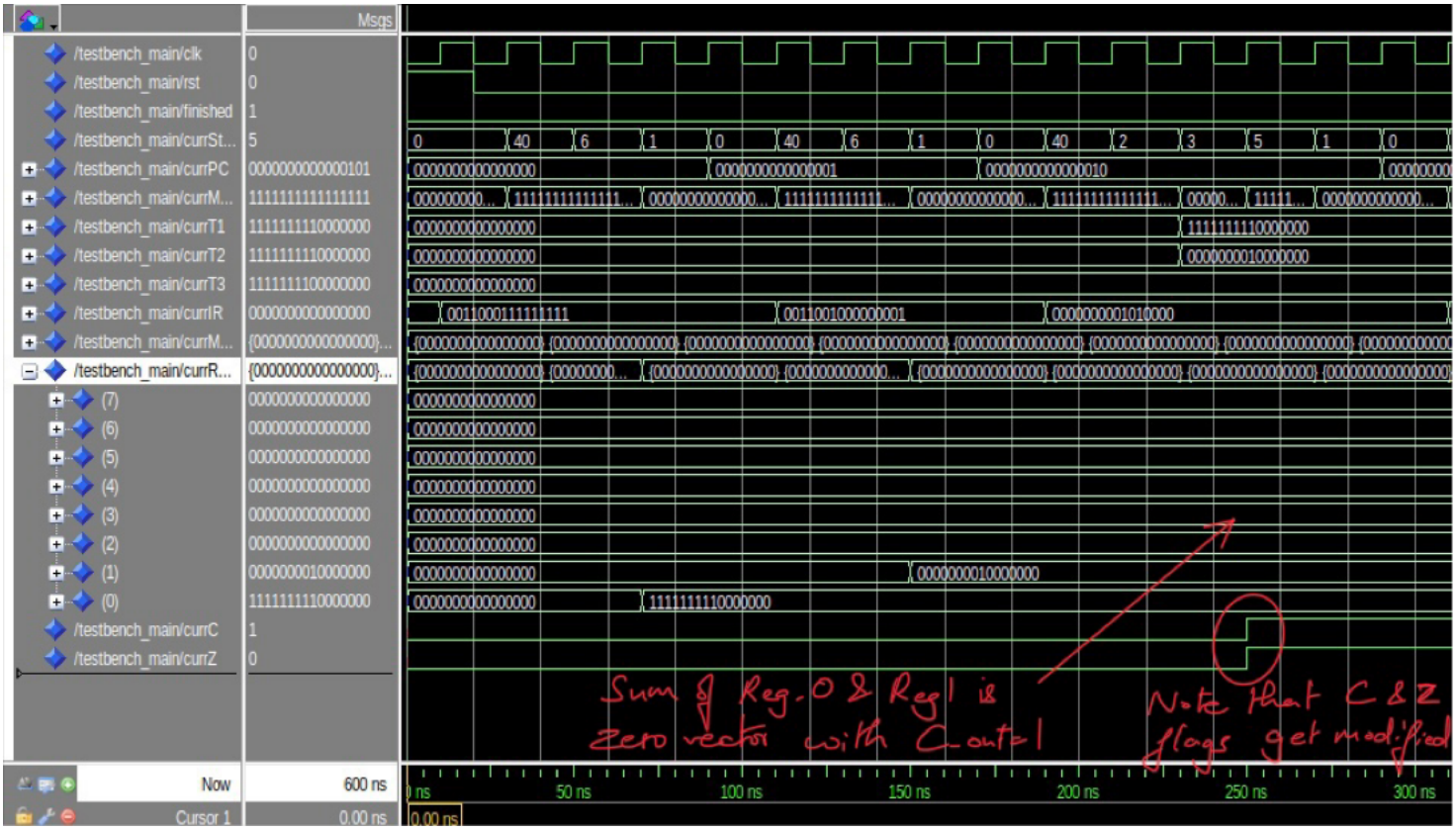
2) Instructions in memory:-

x"31ff", x"3201", x"0050"

loads 1111111110000000 in Register_0 (LHI)

loads 0000000010000000 in Register_1 LHI

Add Reg_1 & Reg_2 & store result in Reg_3 ADD



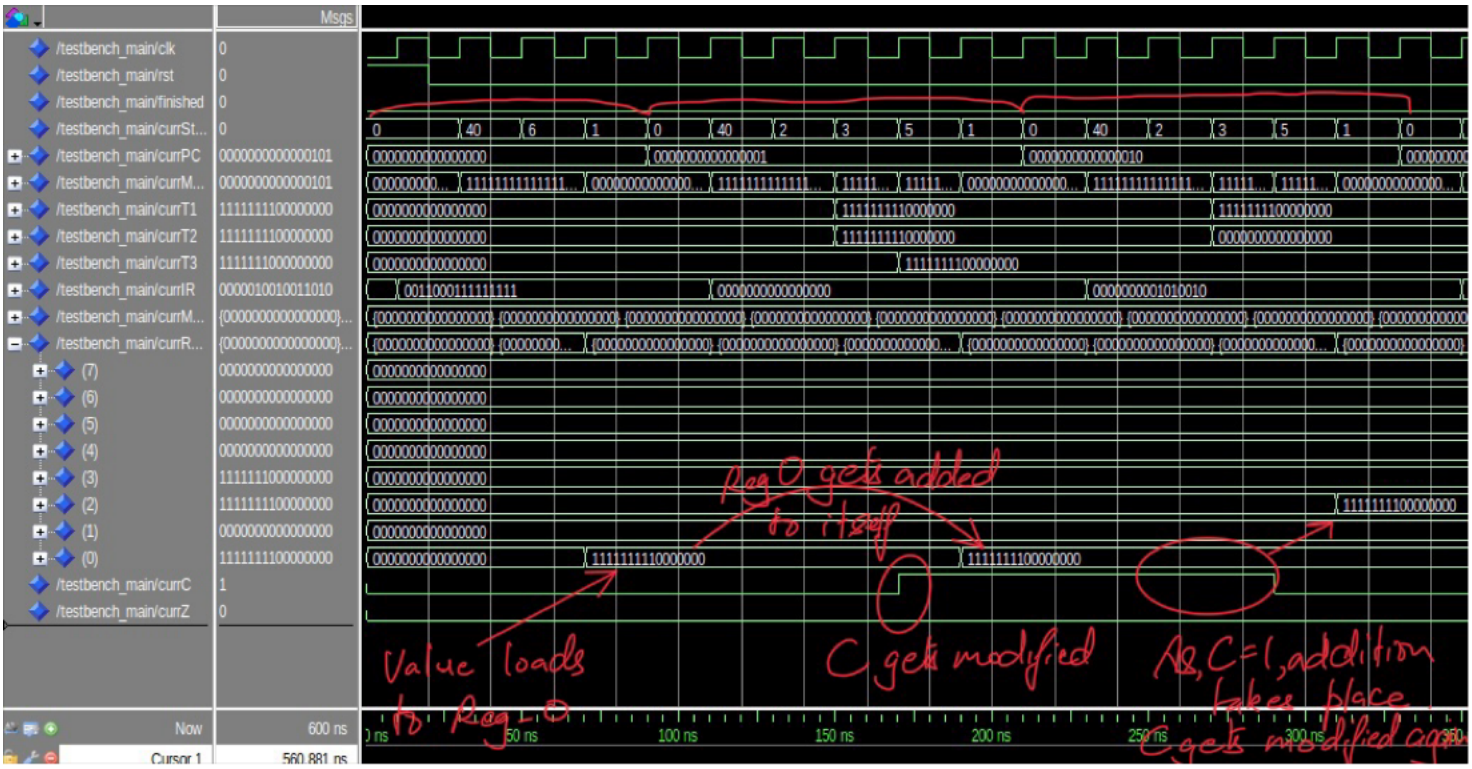
3) Instructions in memory:-

x"31ff", x"0000", x"0052"

loads 1111111110000000 in Register_0 (LHI)

Add Reg_0 to itself & store in Reg_0 (Note that C flag will become 1 here) (ADD)

Add Reg_0 & Reg_1 & store in Reg_2 iff carry is 1 (ADC)



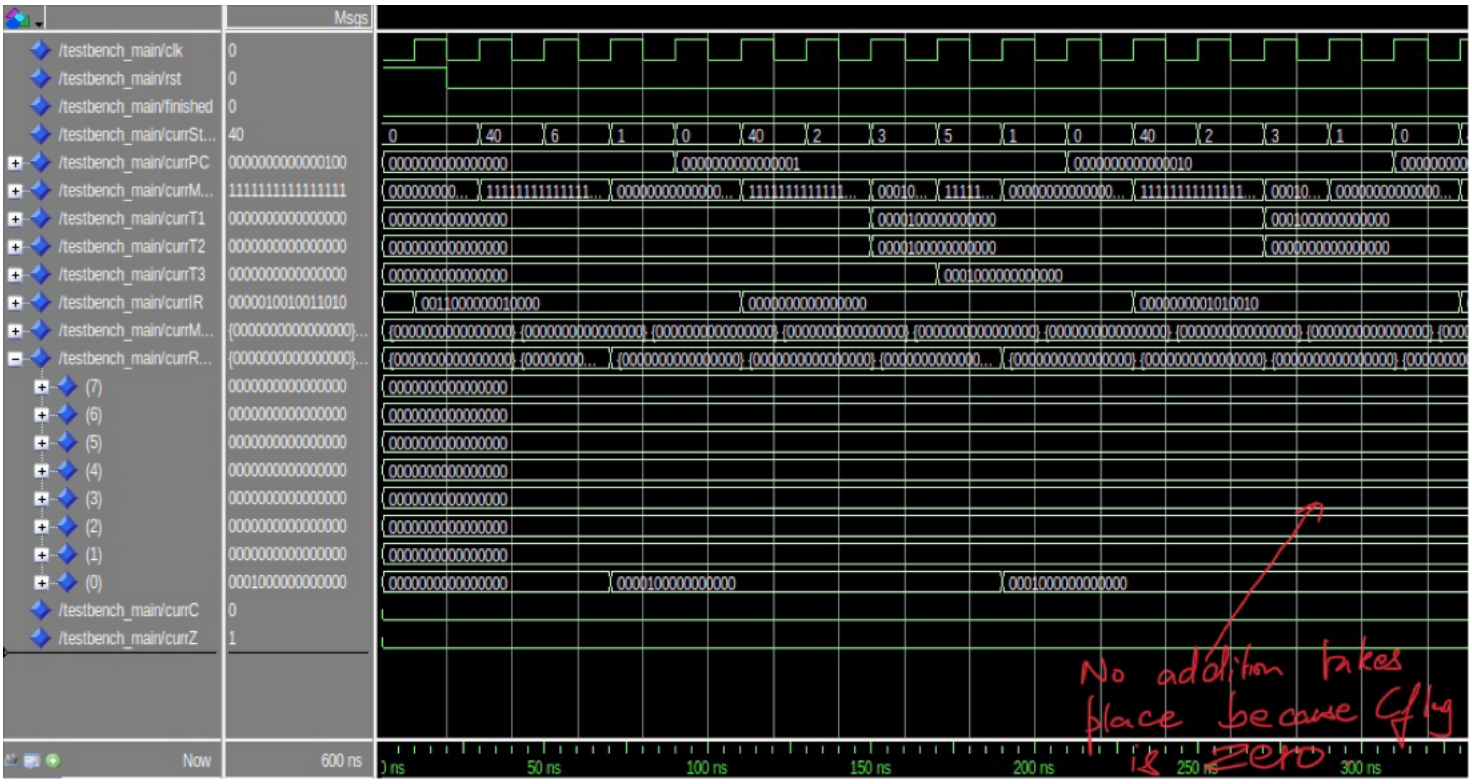
4) Instructions:-

x"3010", x"0000", x"0052"

Load 0001000000000000 in Reg_0 (LHI)

Add Reg_0 to itself & store in Reg_0 (C flag will be 0 in this case)

Add Reg_0 & Reg_1 & store in Reg_2 iff carry is 1 (ADC)



5) ADZ (zflag is set) :-

Instructions :- x"3201"; x"0000", x"0051"

Put some value in Reg-1

This will set zeroflag to 1

Add Reg-1 & Reg0 & store in Reg2 iff Zflag = 1

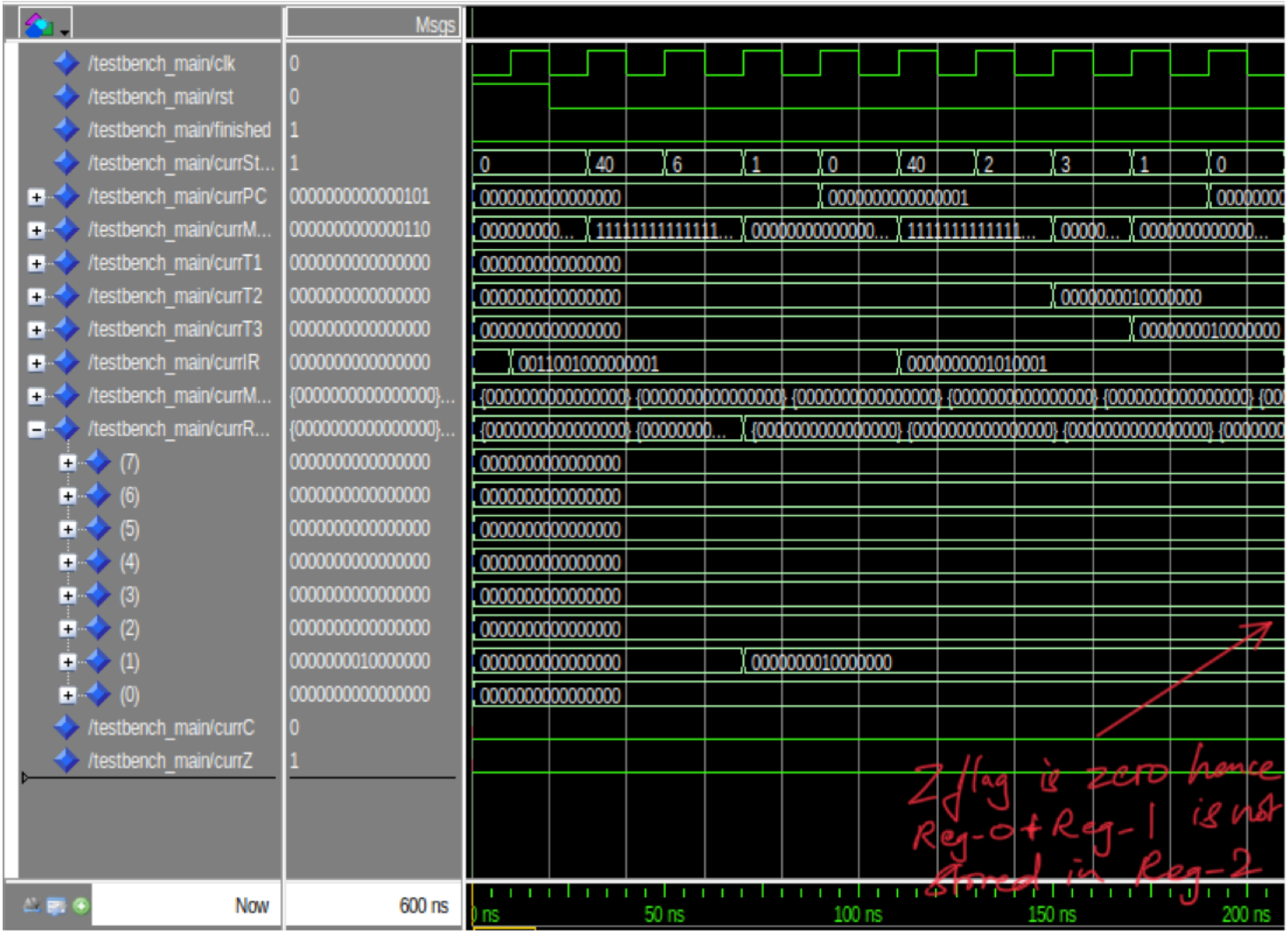
ADZ (Z.flag is zero) :-

Instruction :- x"3201", x"0051".

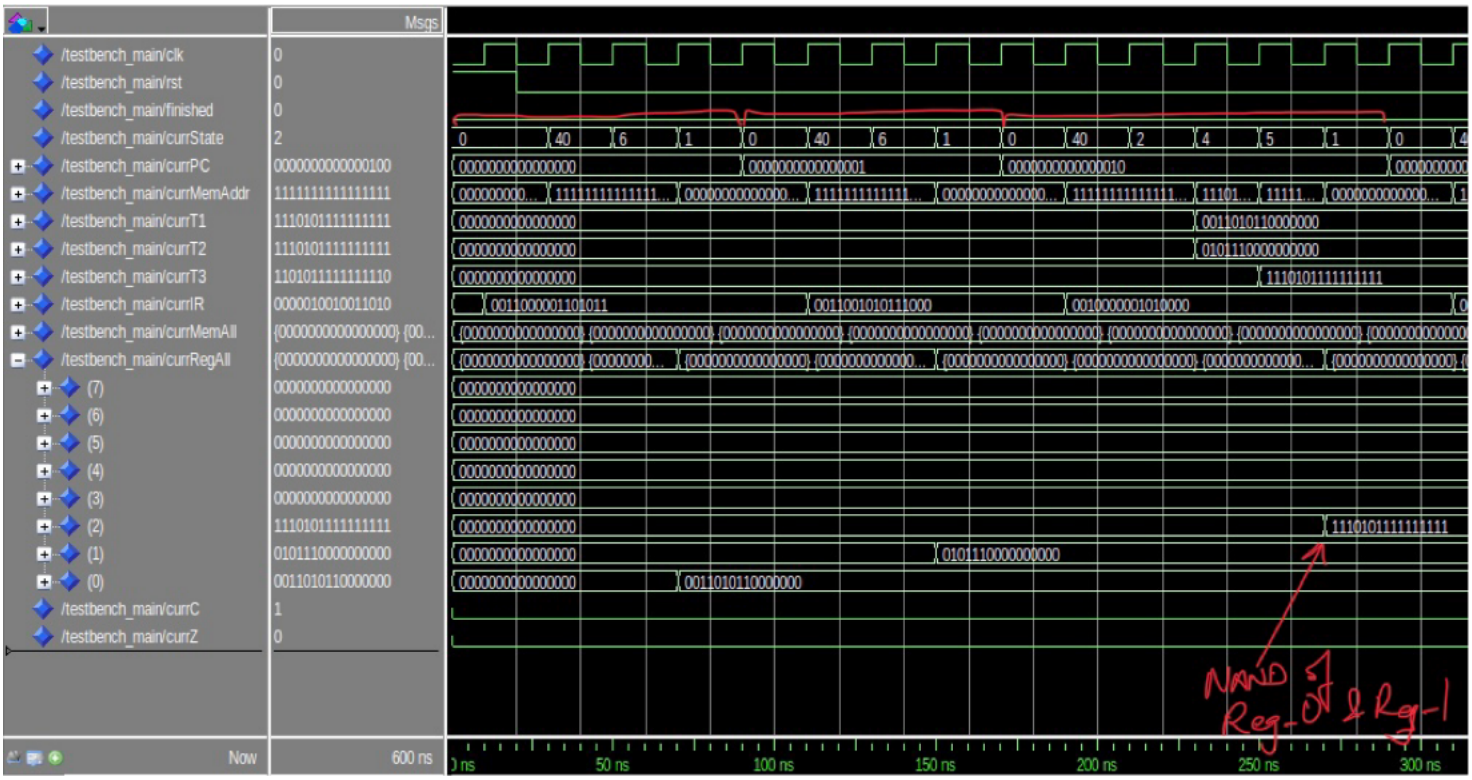
Put some value in Reg 1

Add Reg-0 & Reg-1 & store in Reg-2 iff Z-flag = 1

Msgs
/testbench_main/clk
/testbench_main/rst
/testbench_main/finished
/testbench_main/currSt...
/testbench_main/currPC
/testbench_main/currM...
/testbench_main/currT1
/testbench_main/currT2
/testbench_main/currT3
/testbench_main/currIR
/testbench_main/currM...
/testbench_main/currR...
(7)
(6)
(5)
(4)
(3)
(2)
(1)
(0)
/testbench_main/currC
/testbench_main/currZ
Now
600 ns
Value stored in Reg-1
Zero-flag sets
Addition takes place because Z-flag was set.


Msgs
/testbench_main/clk
/testbench_main/rst
/testbench_main/finished
/testbench_main/currSt...
/testbench_main/currPC
/testbench_main/currM...
/testbench_main/currT1
/testbench_main/currT2
/testbench_main/currT3
/testbench_main/currIR
/testbench_main/currM...
/testbench_main/currR...
(7)
(6)
(5)
(4)
(3)
(2)
(1)
(0)
/testbench_main/currC
/testbench_main/currZ
Now
600 ns
Zflag is zero hence Reg-0+Reg-1 is not stored in Reg-2

6) NDU :- Follows similar structure to ADD



7) NDC (with carry-flag = 0) :-

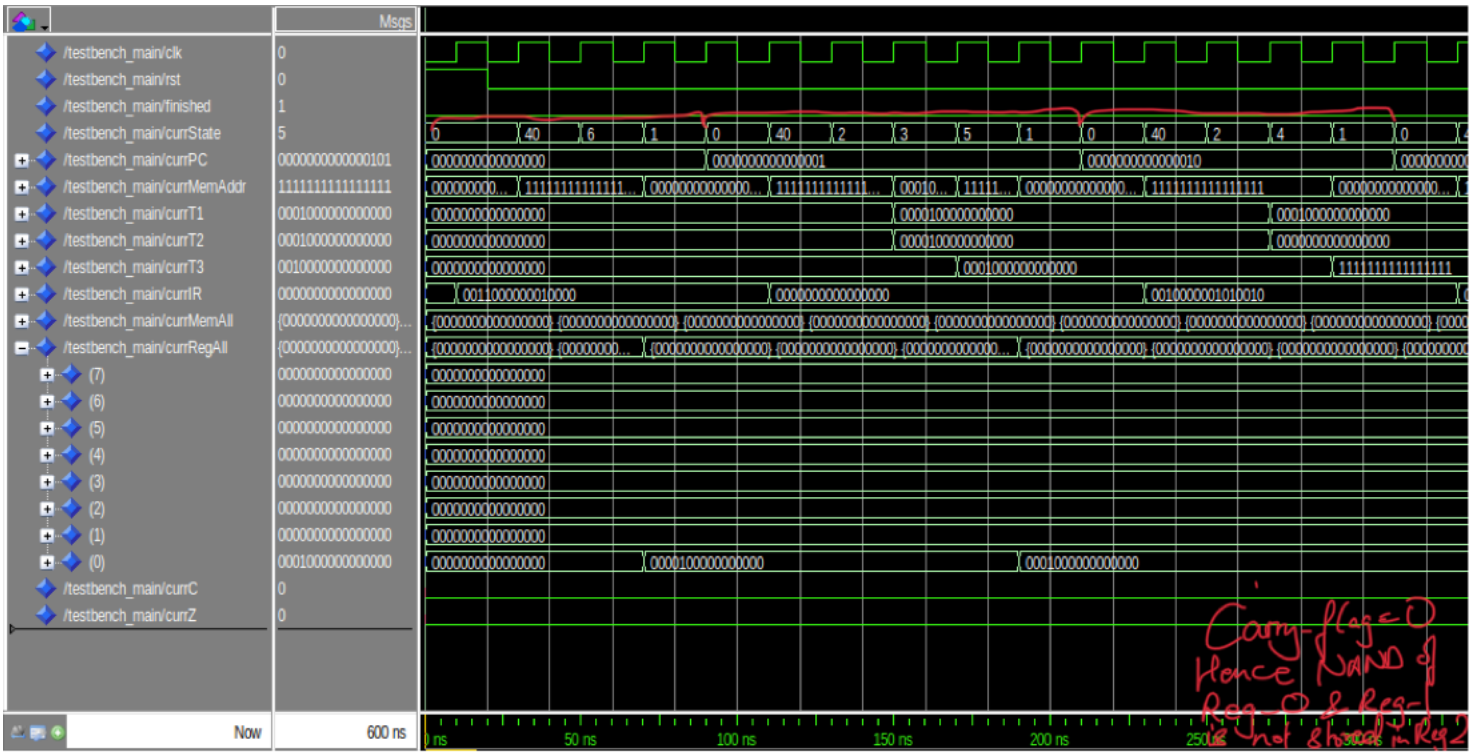
Instructions:-

x"3010", x"0000", x"2052"

Load 0001000000000000 in Reg_0 (LHI)

Add Reg_0 to itself & store in Reg 0 (C flag will be 0 in this case)

NAND Reg_0 & Reg_1 & store in Reg_2 iff carry is 1 (NDC)



8) NDC (with carry bit = 1)

Instructions in memory:-

x"31ff", x"0000", x"2052"

loads 1111111110000000 in Register_0

(LHI)

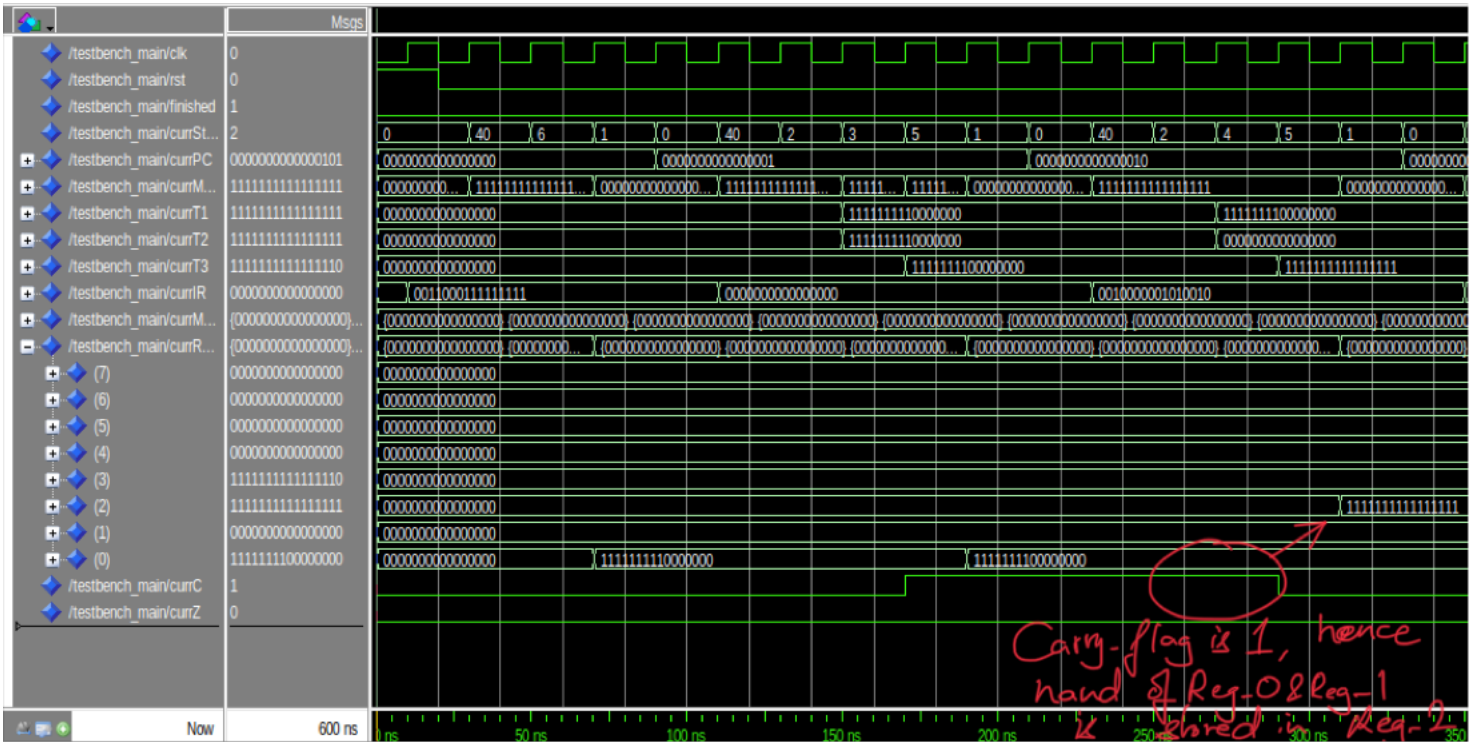
Add Reg-0 to itself & store in Reg-0

(Note that C flag will become 1 here)

(ADD)

NAND Reg-0 & Reg-1 & store in Reg-2 iff carry is 1

(NDC)



9) NDZ (z flag is set) :-

Instructions :- x"3201"; x"0000", x"2051"

Put some value in Reg-1

This will set zeroflag to 1

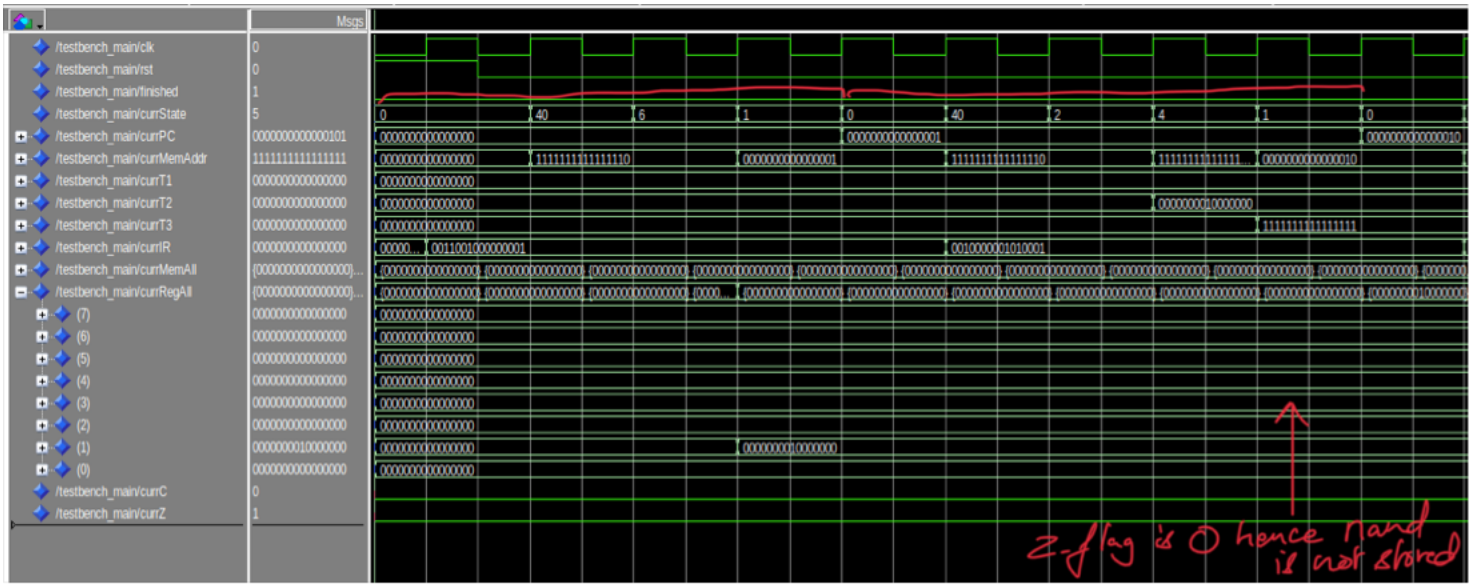
Nand Reg-1 & Reg0 & store in Reg2 iff Z flag = 1

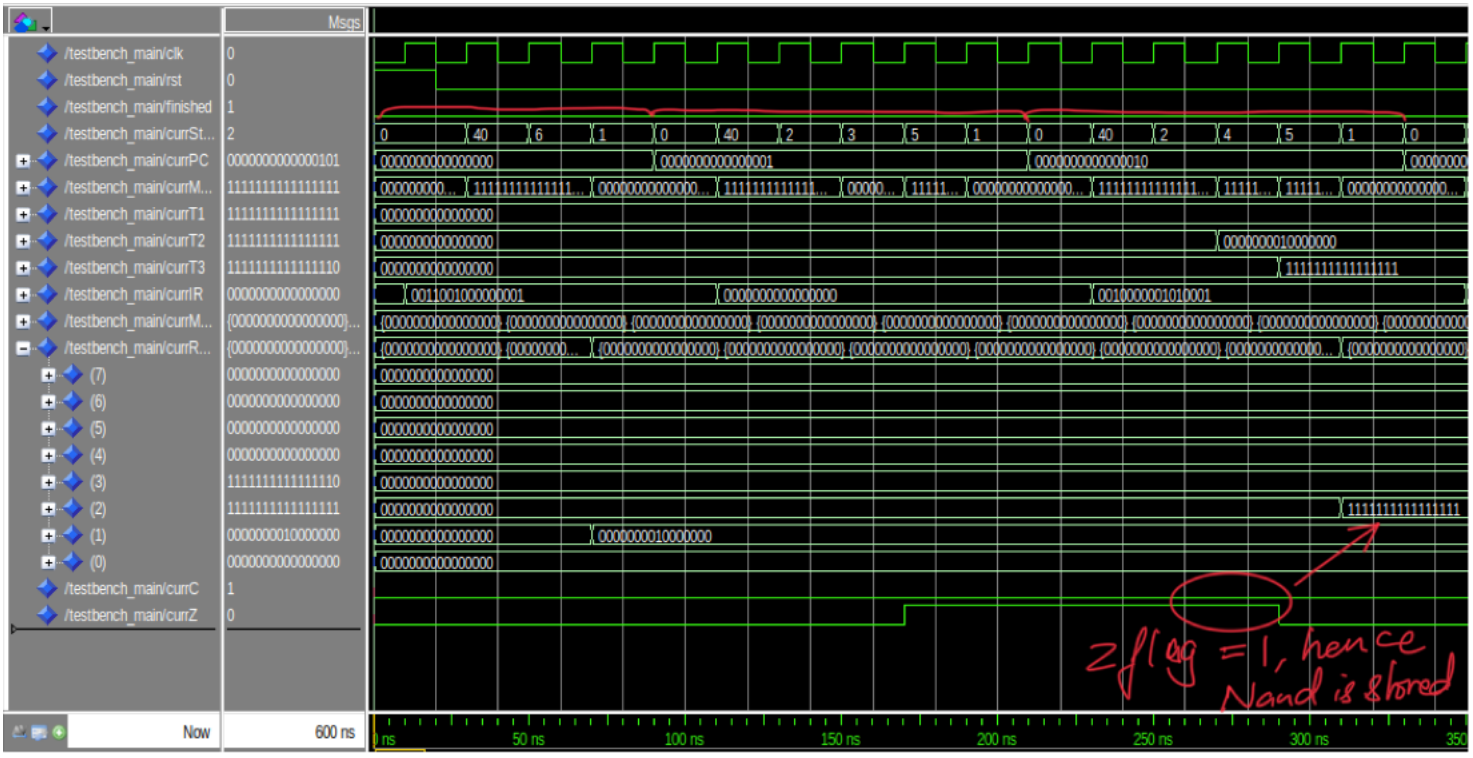
NDZ (Z flag is zero) :-

Instruction :- x"3201", x"0051".

Put some value in Reg 1

Nand Reg-0 & Reg-1 & store in Reg-2 iff Z-flag = 1


Msgs
/testbench_main/clk
/testbench_main/rst
/testbench_main/finished
/testbench_main/currState
/testbench_main/currPC
/testbench_main/currMemAddr
/testbench_main/currT1
/testbench_main/currT2
/testbench_main/currT3
/testbench_main/currIR
/testbench_main/currMemAll
/testbench_main/currRegAll
(7)
(6)
(5)
(4)
(3)
(2)
(1)
(0)
/testbench_main/currC
/testbench_main/currZ
z-flag is 0 hence nand is not stored


Msgs
/testbench_main/clk
/testbench_main/rst
/testbench_main/finished
/testbench_main/currSt...
/testbench_main/currPC
/testbench_main/currM...
/testbench_main/currT1
/testbench_main/currT2
/testbench_main/currT3
/testbench_main/currIR
/testbench_main/currM...
/testbench_main/currR...
(7)
(6)
(5)
(4)
(3)
(2)
(1)
(0)
/testbench_main/currC
/testbench_main/currZ
Now
600 ns
0 ns
50 ns
100 ns
150 ns
200 ns
250 ns
300 ns
zflag = 1, hence Nand is stored

Find other simmulations in "screenshots" folder