Toronto Metropolitan University

Department of Electrical, Computer, & Biomedical Engineering
Faculty of Engineering & Architectural Science

Course Title:	Computer Organization and Architecture
Course Number:	COE 608
Semester/Year (e.g.F2016)	W2024

Instructor:	Dr. Khalid A. Hafeez

Assignment/Lab Number:	6
Assignment/Lab Title:	CPU Control Unit Design

Submission Date:	
Due Date:	

Student LAST Name	Student FIRST Name	Student Number	Section	Signature*
Patel	Apurva	500876938	08	A.P

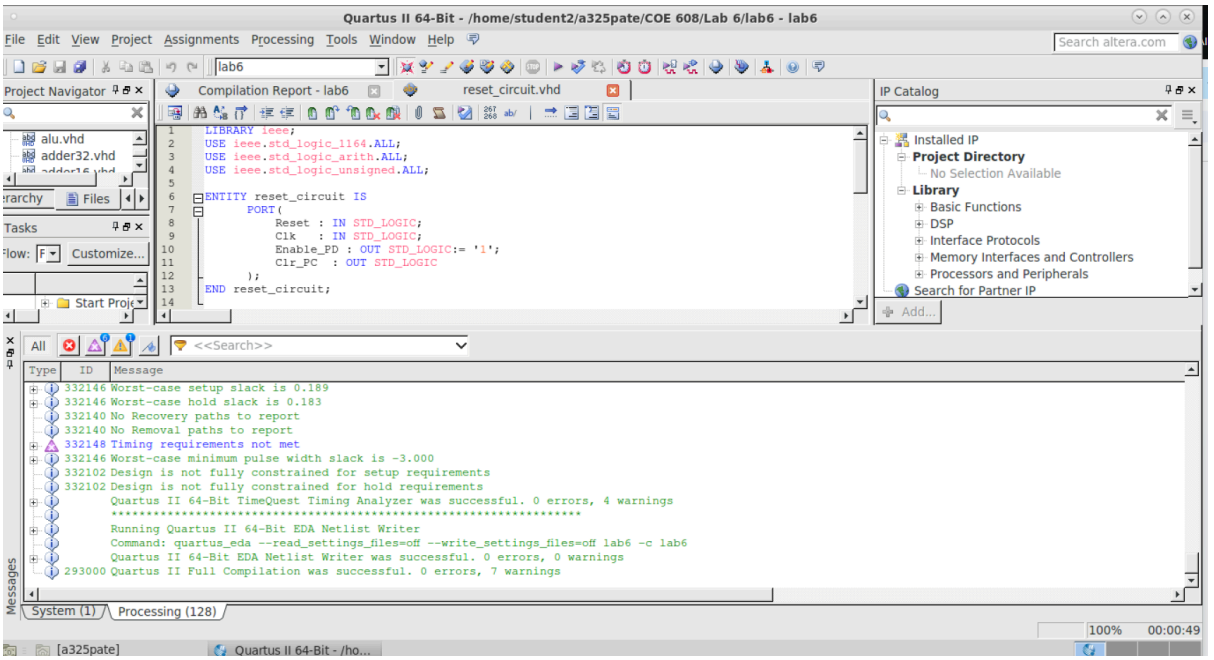
*By signing above you attest that you have contributed to this written lab report and confirm that all work you have contributed to this lab report is your own work. Any suspicion of copying or plagiarism in this work will result in an investigation of Academic Misconduct and may result in a "0" on the work, an "F" in the course, or possibly more severe penalties, as well as a Disciplinary Notice on your academic record under the Student Code of Academic Conduct, which can be found online at: https://www.torontomu.ca/content/dam/senate/policies/pol60.pdf

> Objective:
In this lab, a complete CPU will be implemented whose main components: the data path and control unit have been designed and implemented in the previous labs. We will combine the control unit and data-path with a reset circuit and when complete, the over-all design will be able to implement the features described in the CPU specification document.

> Brief:
The design and implementation of a CPU reset circuit is an essential component ensuring proper initialization and stabilisation of the CPU's components before performing operation on the designed CPU. The objectives included integrating the reset circuit seamlessly into the CPU design, synchronising its operation with the CPU clock, and facilitating a smooth transition into a stable state upon reset. The design considerations emphasised efficient resource utilisation, flexibility to accommodate variations in CPU architecture, and robust handling of the RESET signal. The methodology involves analysing the CPU architecture, developing VHDL code for the reset circuit, and conducting thorough testing and debugging to validate its functionality. By successfully implementing the reset circuit, the CPU can initialise its components reliably, paving the way for efficient execution of instructions and overall system performance.

> VHDL Module Design:

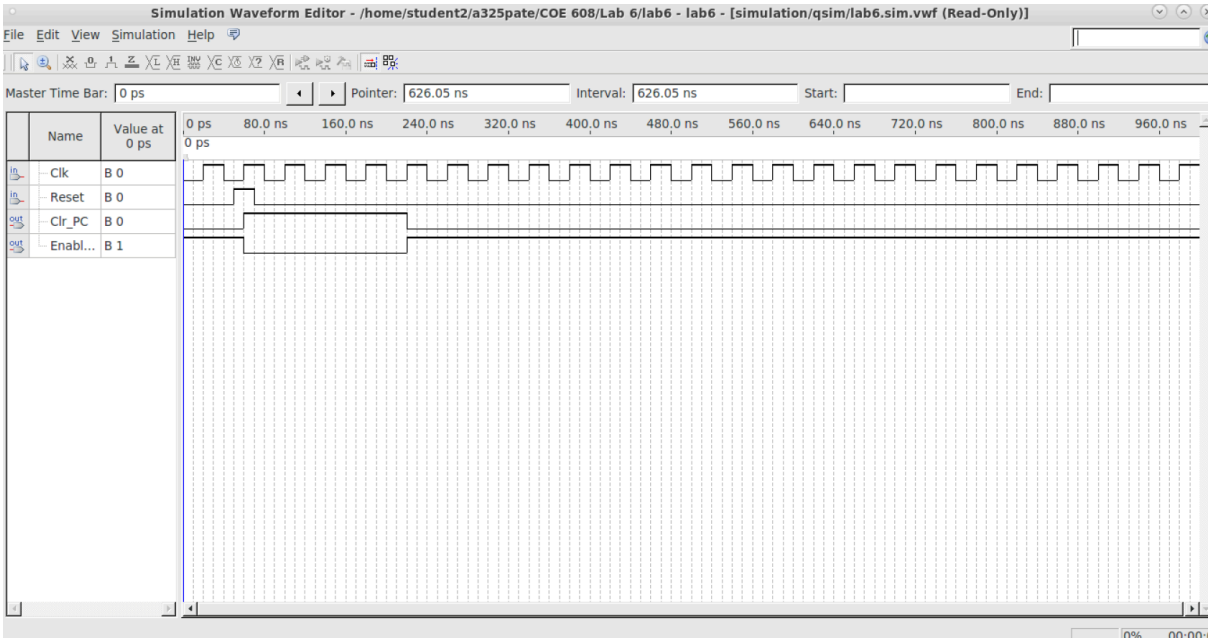
- reset_circuit.vhd: Compiles without any errors:

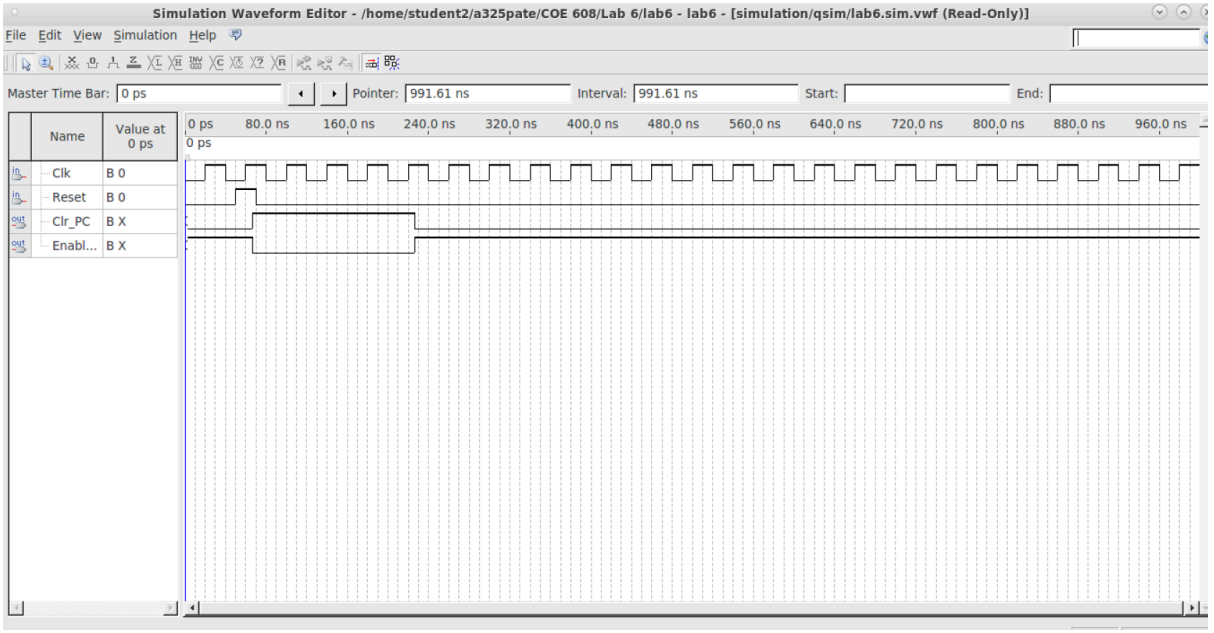



```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.std_logic_arith.ALL;
USE ieee.std_logic_unsigned.ALL;

ENTITY reset_circuit IS
                PORT(
                        Reset : IN STD_LOGIC;
```

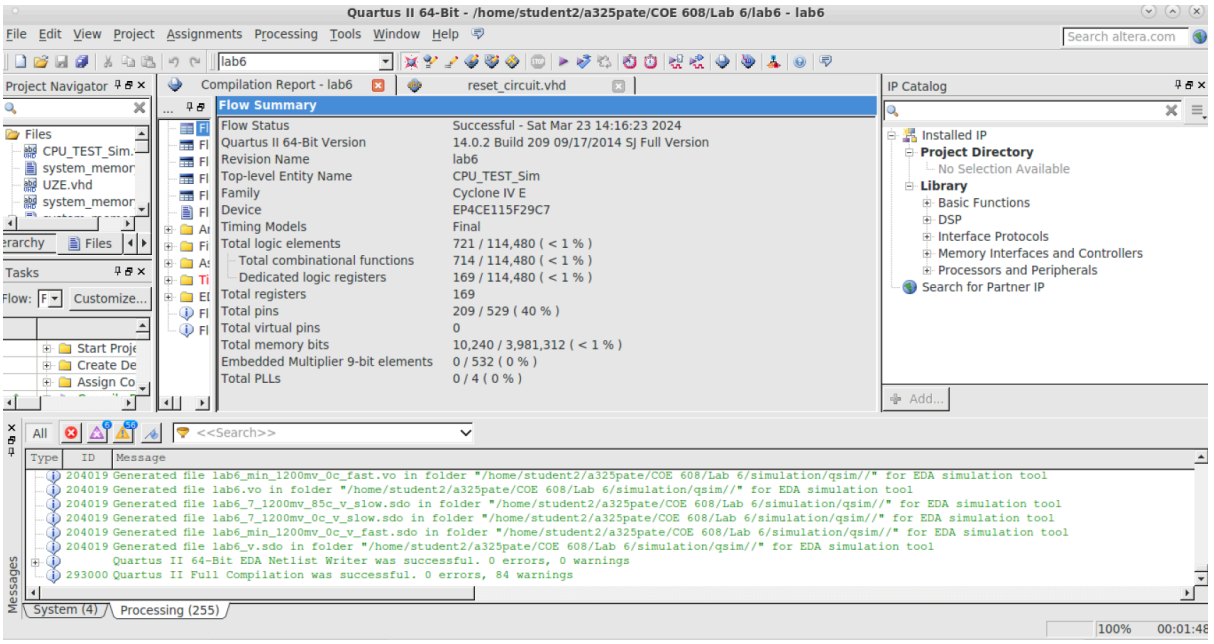
```
                    Clk      : IN STD_LOGIC;
                    Enable_PD : OUT STD_LOGIC:= '1';
                    Clr_PC          : OUT STD_LOGIC
                );
END reset_circuit;

ARCHITECTURE Behavior OF reset_circuit IS
        TYPE clkNum IS (clk0, clk1, clk2, clk3);
        SIGNAL present_clk: clkNum;
BEGIN
    process(Clk)begin
        if rising_edge(Clk) then
            if Reset = '1' then
                Clr_PC <= '1';
                Enable_PD <= '0';
                present_clk <= clk0;
            elsif present_clk <= clk0 then
                present_clk <= clk1;
            elsif present_clk <= clk1 then
                present_clk <= clk2;
            elsif present_clk <= clk2 then
                present_clk <= clk3;
            elsif present_clk <= clk3 then
                Clr_PC <= '0';
                Enable_PD <= '1';
            end if;
        end if;
    end process;
END Behavior;
```

- Reset_circuit.vwf Functional and Time Simulation:





- CPUT_TEST_SIM.vhd: Compiles without any errors.




```
library ieee;

use ieee.std_logic_1164.all;


ENTITY CPU_TEST_Sim IS

        PORT(

                cpuClk : in std_logic;

                memClk : in std_logic;
```

```
		rst : in std_logic;
		-- Debug data.
		outA, outB : out std_logic_vector(31 downto 0);
		outC, outZ : out std_logic;
		outIR : out std_logic_vector(31 downto 0);
		outPC : out std_logic_vector(31 downto 0);
		-- Processor-Inst Memory Interface.
		addrOut : out std_logic_vector(5 downto 0);
		wEn : out std_logic;
		memDataOut : out std_logic_vector(31 downto 0);
		memDataIn : out std_logic_vector(31 downto 0);
		-- Processor State
		T_Info : out std_logic_vector(2 downto 0);
		--data Memory Interface
		wen_mem, en_mem : out std_logic);
END CPU_TEST_Sim;

ARCHITECTURE behavior OF CPU_TEST_Sim IS
	COMPONENT system_memory
		PORT(
			address		: IN	STD_LOGIC_VECTOR (5 DOWNTO 0);
			clock	: IN	STD_LOGIC ;
			data		: IN	STD_LOGIC_VECTOR (31 DOWNTO
0);
```

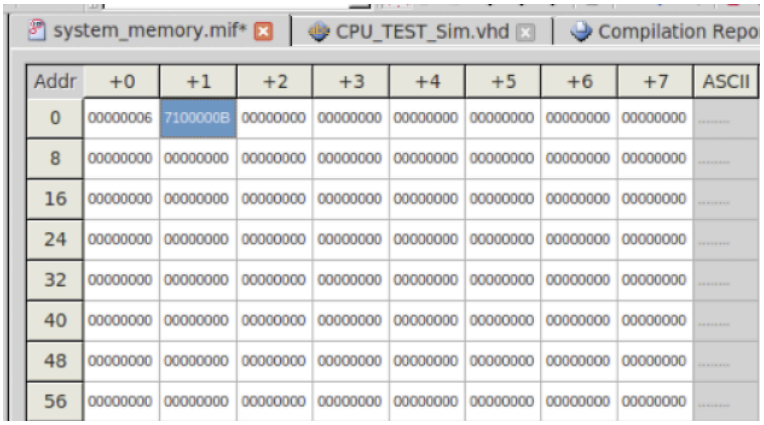
```
                wren            : IN    STD_LOGIC ;
                q               : OUT STD_LOGIC_VECTOR (31
DOWNTO 0)
);
END COMPONENT;

    COMPONENT cpu1
        PORT(
            clk             : in std_logic;
            mem_clk         : in std_logic;
            rst             : in std_logic;
            dataIn  : in std_logic_vector(31 downto 0);
            dataOut         : out std_logic_vector(31 downto 0);
            addrOut         : out std_logic_vector(31 downto 0);
            wEn : out std_logic;
            dOutA, dOutB : out std_logic_vector(31 downto 0);
            dOutC, dOutZ : out std_logic;
            dOutIR : out std_logic_vector(31 downto 0);
            dOutPC : out std_logic_vector(31 downto 0);
            outT : out std_logic_vector(2 downto 0);
            wen_mem, en_mem : out std_logic);
    END COMPONENT;

    signal cpu_to_mem: std_logic_vector(31 downto 0);
    signal mem_to_cpu: std_logic_vector(31 downto 0);
```

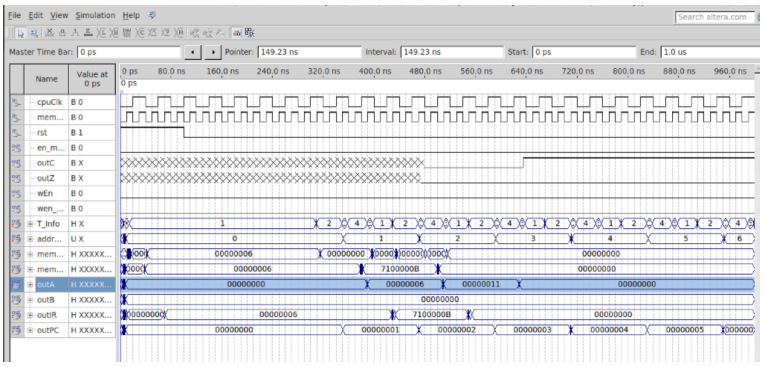
```
        signal add_from_cpu: std_logic_vector(31 downto 0);
        signal wen_from_cpu: std_logic;


BEGIN
 -- Component instantiations.
        main_memory : system_memory
                PORT MAP(
                        address => add_from_cpu(5 downto 0),
                        clock => memClk,
                        data => cpu_to_mem,
                        wren => wen_from_cpu,
                        q => mem_to_cpu
                );
        main_processor : cpu1
                PORT MAP(
                 clk => cpuClk,
                 mem_clk => memClk,
                 rst => rst,
                 dataIn => mem_to_cpu,
                 dataOut => cpu_to_mem,
                 addrOut => add_from_cpu,
                 wEn => wen_from_cpu,
                 dOutA => outA,
                 dOutB => outB,
```

```
                dOutC => outC,
                dOutZ => outZ,
                dOutIR => outIR,
                dOutPC => outPC,
                outT => T_Info,
                wen_mem => wen_mem,
                en_mem => en_mem
                );

        addrOut <= add_from_cpu(5 downto 0);
        wEn <= wen_from_cpu;
        memDataOut <= mem_to_cpu;
        memDataIn <= cpu_to_mem;
END behavior;
```

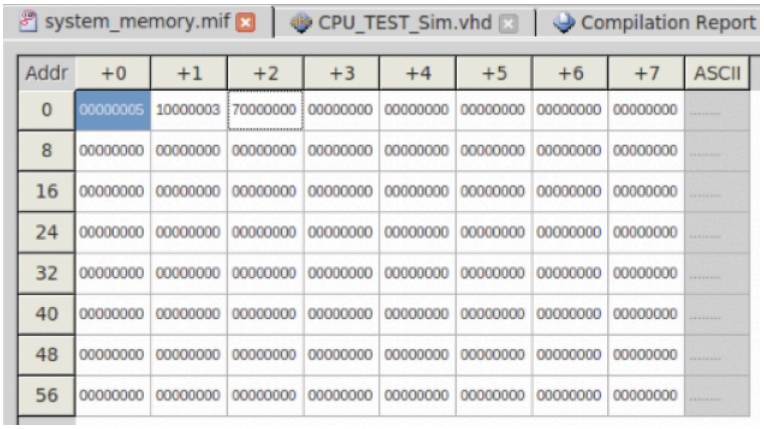
> SysMemory and Time Simulations:

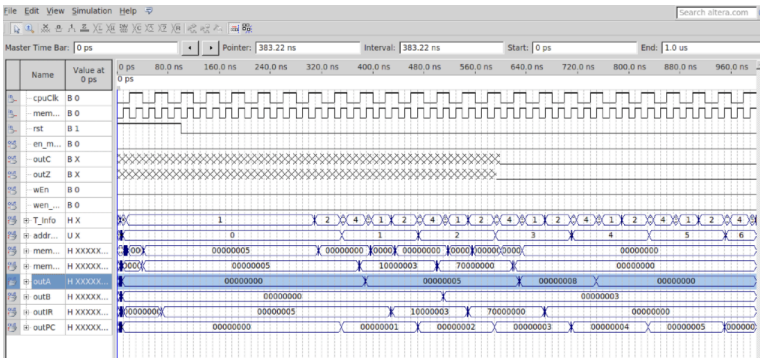
- ADDI:



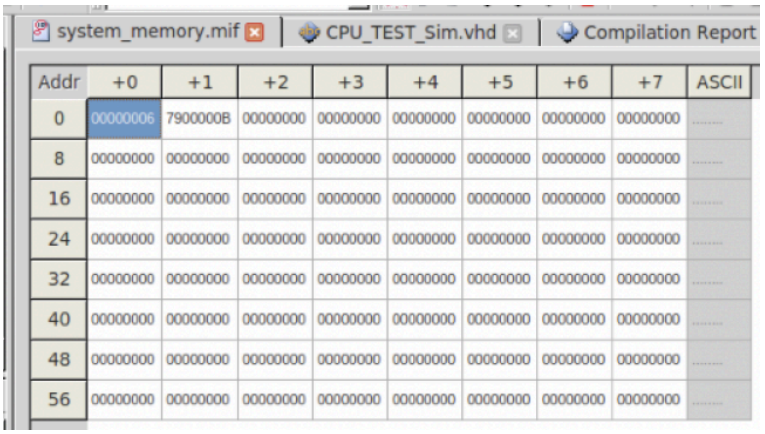


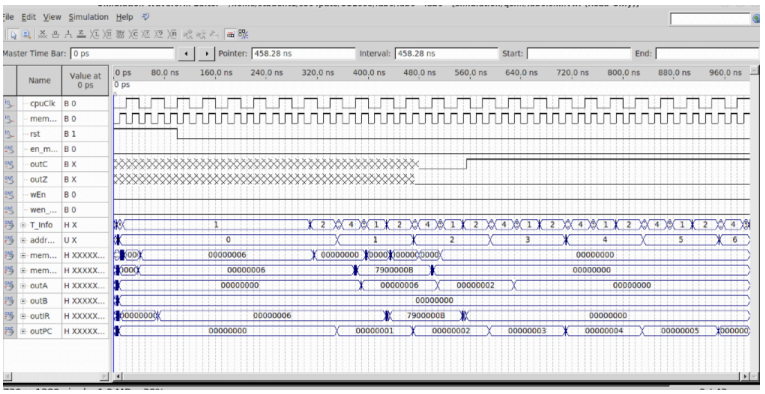
- ADD:



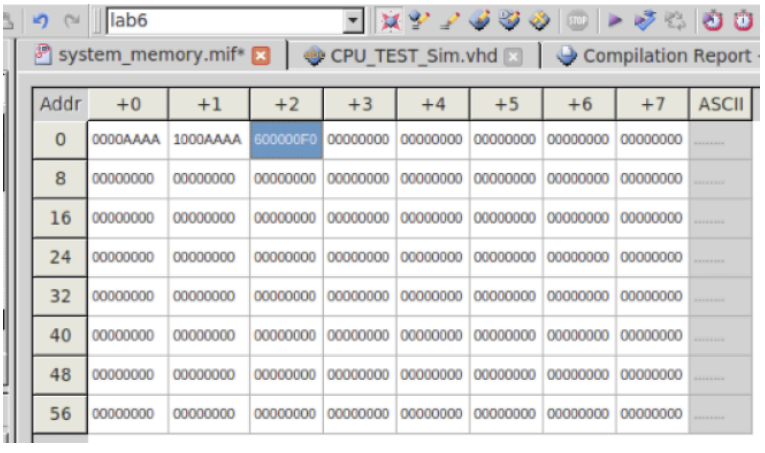


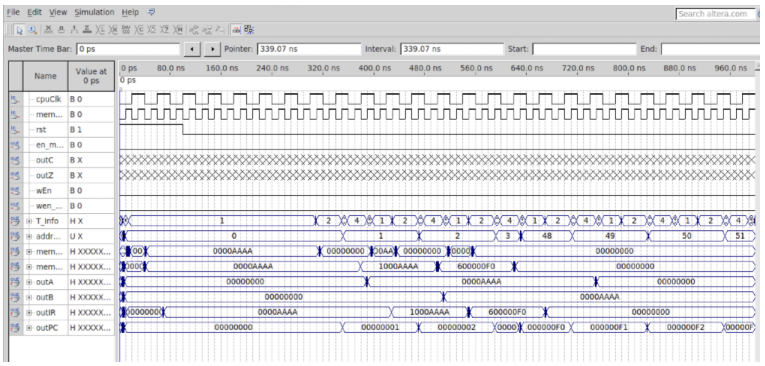
- ANDI:


system_memory.mif
CPU_TEST_Sim.vhd
Compilation Report
Addr +0 +1 +2 +3 +4 +5 +6 +7 ASCII
0 00000006 7900000B 00000000 00000000 00000000 00000000 00000000 00000000
8 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
16 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
24 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
32 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
40 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
48 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
56 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

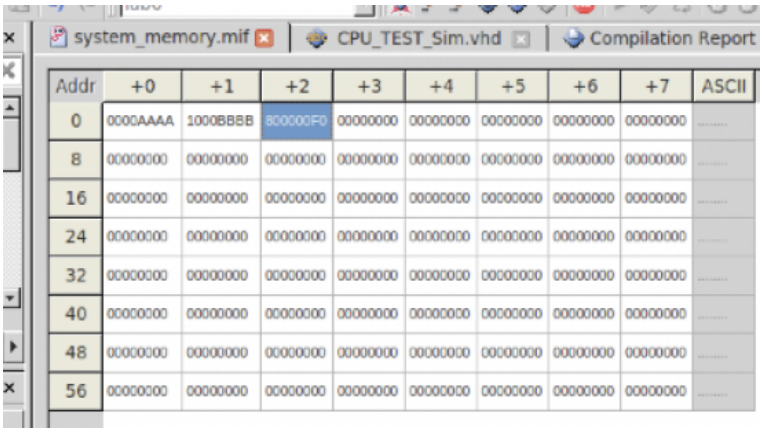


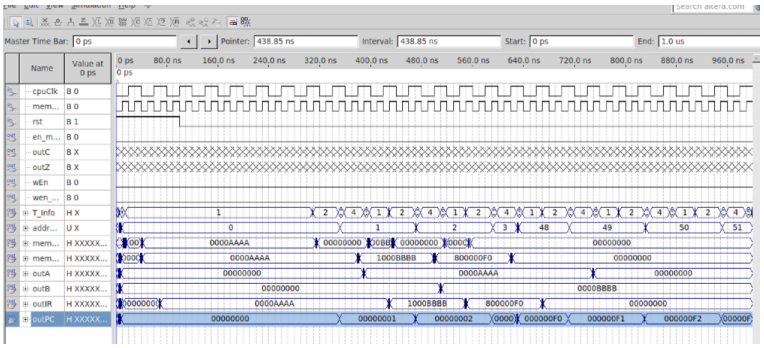
- BEQ:


lab6
system_memory.mif*
CPU_TEST_Sim.vhd
Compilation Report
Addr +0 +1 +2 +3 +4 +5 +6 +7 ASCII
0 0000AAAA 1000AAAA 600000F0 00000000 00000000 00000000 00000000 00000000
8 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
16 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
24 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
32 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
40 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
48 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
56 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

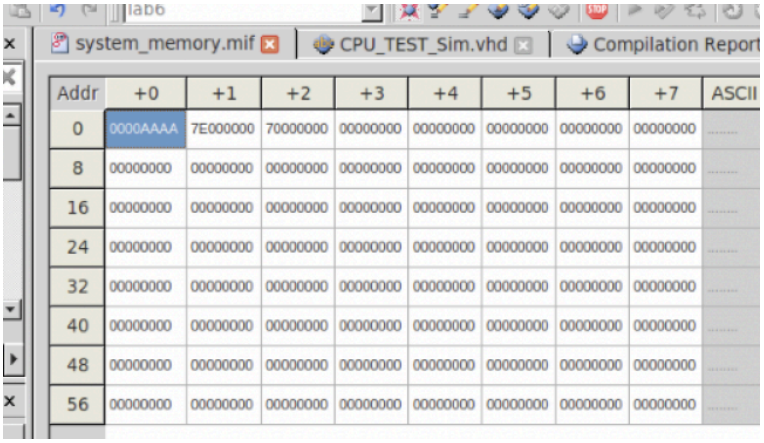


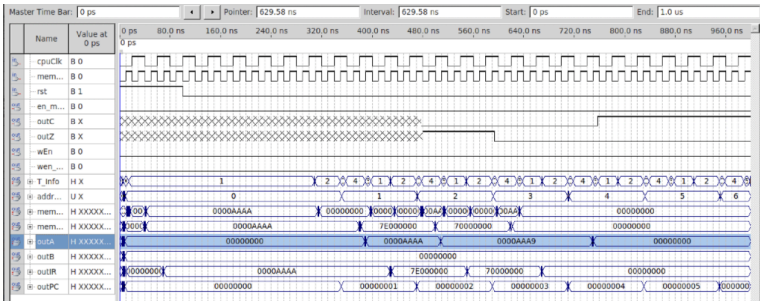
- BNE:


system_memory.mif
CPU_TEST_Sim.vhd
Compilation Report
Addr +0 +1 +2 +3 +4 +5 +6 +7 ASCII
0 0000AAAA 1000BBBB 800000F0 00000000 00000000 00000000 00000000 00000000
8 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
16 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
24 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
32 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
40 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
48 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
56 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

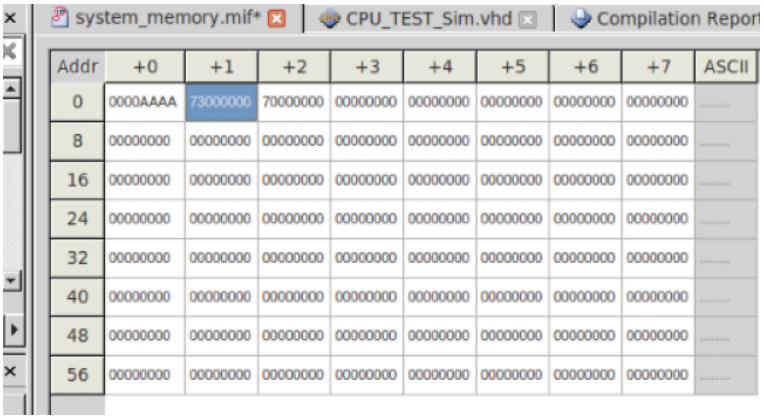


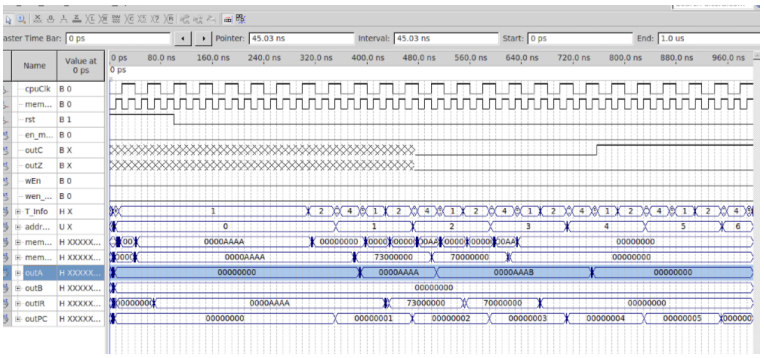
- DECA:


system_memory.mif
CPU_TEST_Sim.vhd
Compilation Report
Addr +0 +1 +2 +3 +4 +5 +6 +7 ASCII
0 0000AAAA 7E000000 70000000 00000000 00000000 00000000 00000000 00000000
8 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
16 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
24 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
32 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
40 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
48 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
56 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

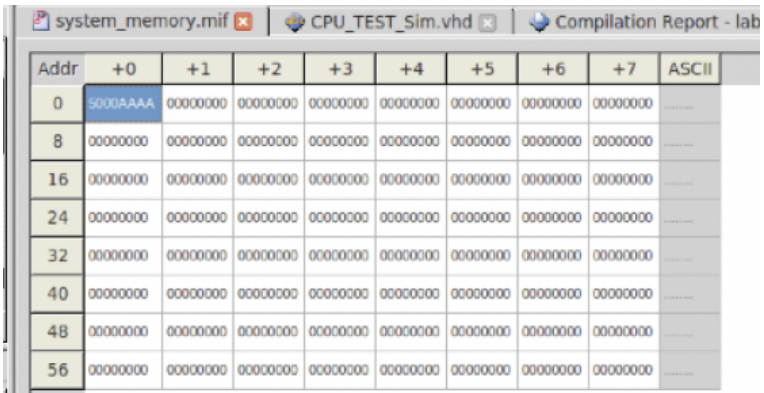


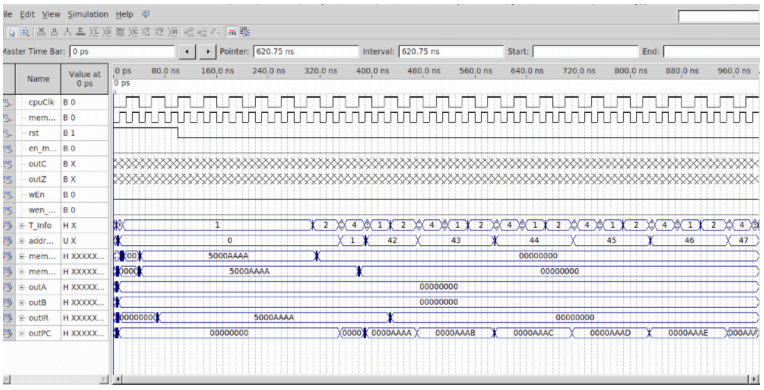
- INCA:


system_memory.mif*
CPU_TEST_Sim.vhd
Compilation Repor
Addr +0 +1 +2 +3 +4 +5 +6 +7 ASCII
0 0000AAAA 73000000 70000000 00000000 00000000 00000000 00000000 00000000
8 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
16 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
24 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
32 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
40 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
48 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
56 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000


cpuClk
rst
outC
outZ
wEn
T_Info
addr...
outA
outB
outIR
outPC

- JMP:

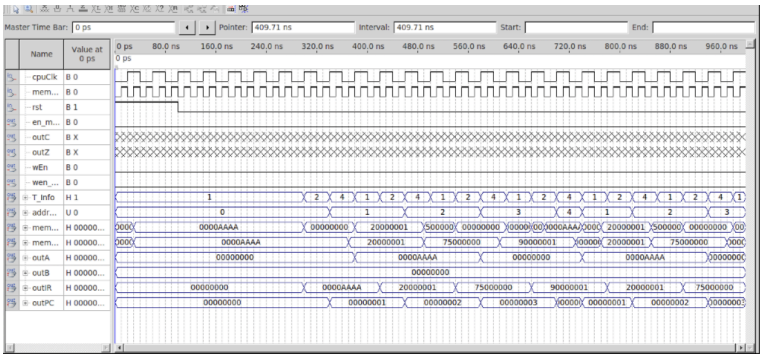

system_memory.mif
CPU_TEST_Sim.vhd
Compilation Report - lab
Addr +0 +1 +2 +3 +4 +5 +6 +7 ASCII
0 5000AAAA 00000000 00000000 00000000 00000000 00000000 00000000 00000000
8 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
16 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
24 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
32 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
40 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
48 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
56 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000


cpuClk
rst
outC
outZ
wEn
T_Info
addr...
outA
outB
outIR
outPC

- LDAI, STA, CLRA, LDA:

CPU_TEST_Sim.vhd | system_memory.mif

Addr	+0	+1	+2	+3	+4	+5	+6	+7	ASCII
0	0000AAAA	20000001	75000000	90000001	00000000	00000000	00000000	00000000	.....
8	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	.....
16	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	.....
24	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	.....
32	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	.....
40	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	.....
48	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	.....
56	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	.....

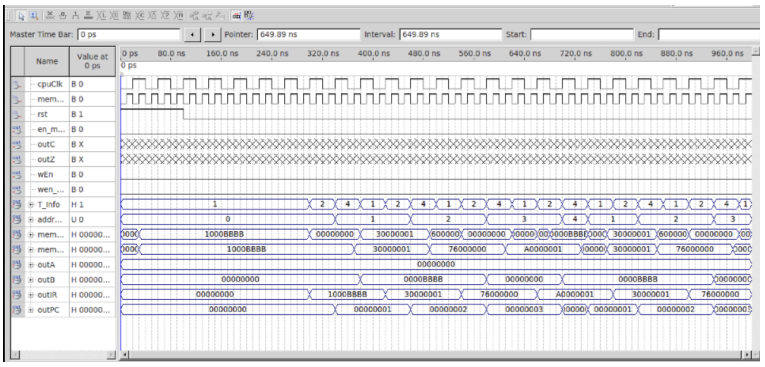


- LDBI, STB, CLRB, LDB:

lab6

CPU_TEST_Sim.vhd | system_memory.mif*

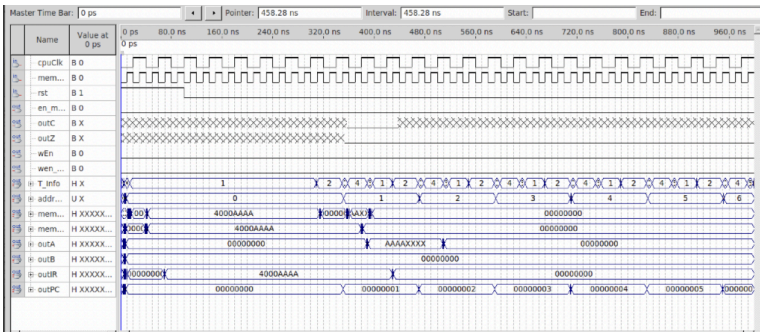
Addr	+0	+1	+2	+3	+4	+5	+6	+7	ASCII
0	1000BBBB	30000001	76000000	A0000001	00000000	00000000	00000000	00000000	.....
8	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	.....
16	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	.....
24	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	.....
32	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	.....
40	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	.....
48	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	.....
56	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	.....



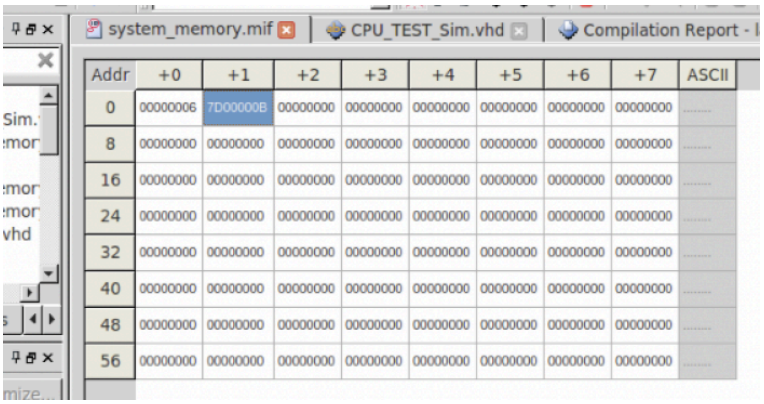
- LUI:

CPU_TEST_Sim.vhd | system_memory.mif

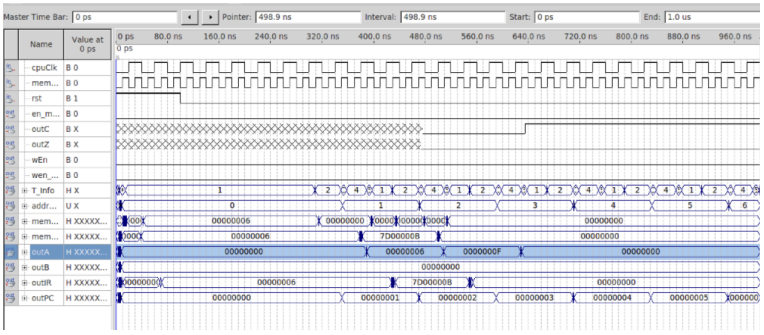
Addr	+0	+1	+2	+3	+4	+5	+6	+7	ASCII
0	4000AAAA	00000000	00000000	00000000	00000000	00000000	00000000	00000000	.....
8	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	.....
16	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	.....
24	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	.....
32	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	.....
40	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	.....
48	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	.....
56	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	.....



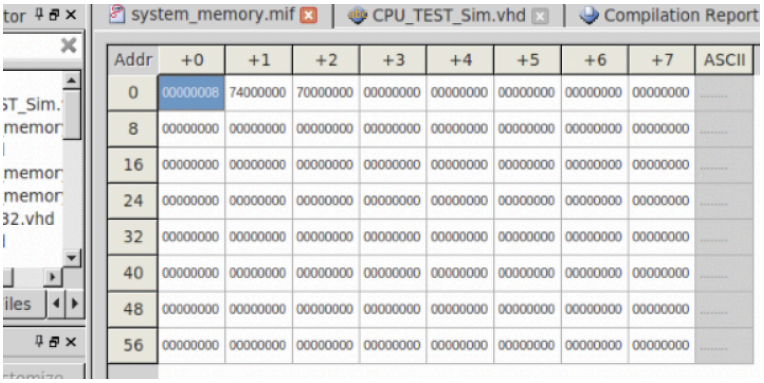
- ORI:

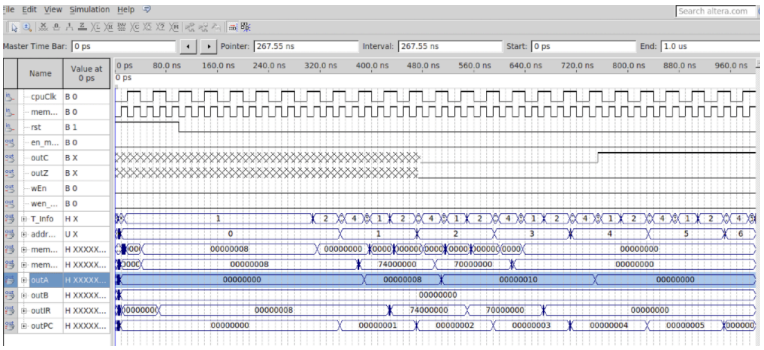


Addr	+0	+1	+2	+3	+4	+5	+6	+7	ASCII
0	00000006	7D00000B	00000000	00000000	00000000	00000000	00000000	00000000	.....
8	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	.....
16	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	.....
24	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	.....
32	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	.....
40	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	.....
48	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	.....
56	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	.....

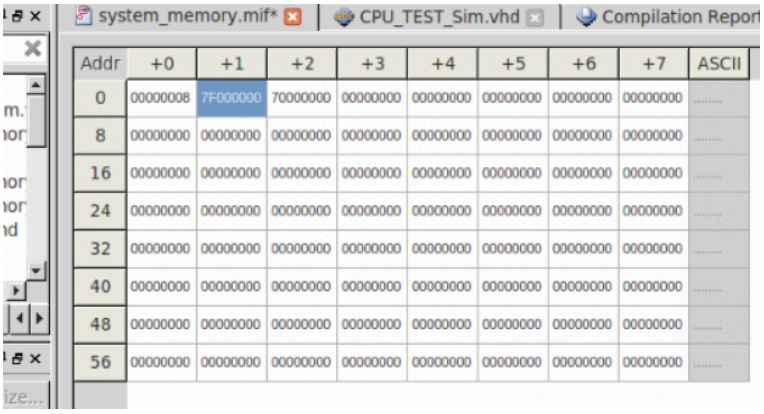


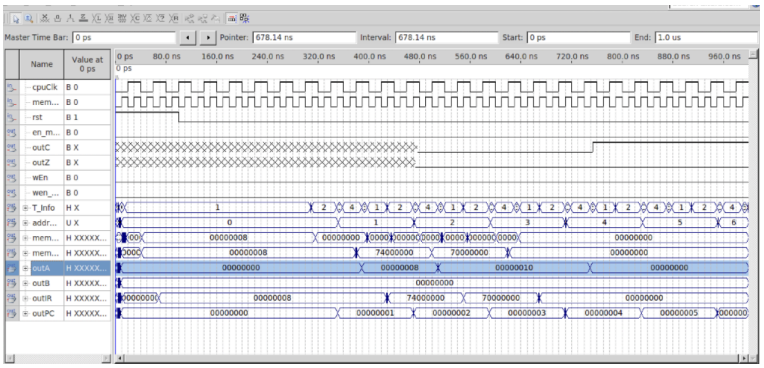
- ROL:


system_memory.mif
CPU_TEST_Sim.vhd
Compilation Report
Addr +0 +1 +2 +3 +4 +5 +6 +7 ASCII
0 00000008 74000000 70000000 00000000 00000000 00000000 00000000 00000000
8 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
16 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
24 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
32 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
40 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
48 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
56 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

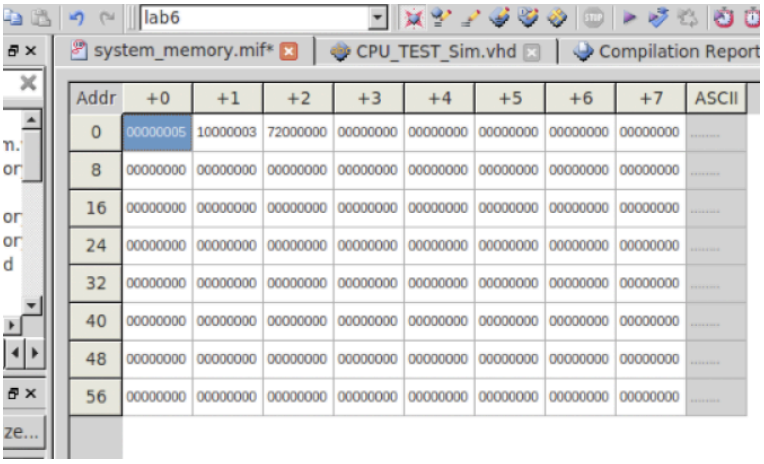


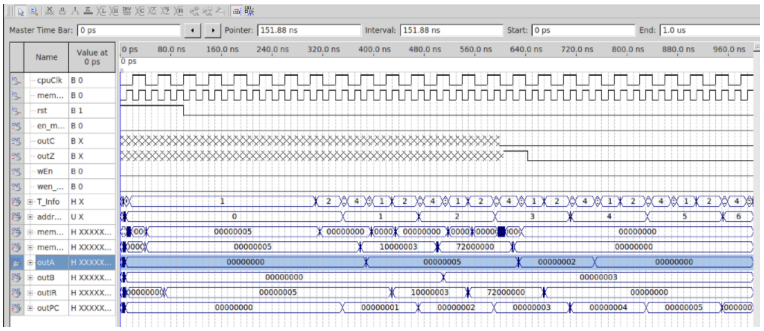
- ROR:


system_memory.mif*
CPU_TEST_Sim.vhd
Compilation Report
Addr +0 +1 +2 +3 +4 +5 +6 +7 ASCII
0 00000008 7F000000 70000000 00000000 00000000 00000000 00000000 00000000
8 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
16 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
24 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
32 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
40 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
48 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
56 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000



- SUB:





> Conclusion:
The lab was successfully completed and the complete CPU was designed, implemented and tested as instructed. The output waveforms were generated as expected and required.

> Reference (Informal):
- COE 608 Lab 6 Manual on D2L.
- COE 608 Lab 6 Tutorial on D2L.