

Course Title:	Computer Organization and Architecture
Course Number:	COE 608
Semester/Year (e.g.F2016)	W2025
Instructor:	Dr. Vadim Geurkov
<i>Assignment/Lab Number:</i>	<i>Lab 6</i>
<i>Assignment/Lab Title:</i>	<i>The Complete CPU</i>
<i>Submission Date</i>	04/03/2025
<i>Due Date:</i>	04/04/2025

Student LAST Name	Student FIRST Name	Student Number	Section	Signature*
Su	Jason	501158090	3	J.S.

*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>

Table of Contents:

Cover Page.....	1
Table of Contents.....	2
Introduction.....	3
Part 1.....	3
Part 2.....	4
Conclusion.....	21

Introduction

The purpose of this lab is to design a complete CPU which implements the designs of the datapath and control unit of the previous labs. Part 1 focuses on the design of the CPU Reset Circuitry which allows the CPU to clear its program counter and start at its base address. By combining all previously designed components, the final CPU design is able to implement features described in the CPU specification document.

Part 1 - CPU Reset Circuitry:

In order for the CPU to function properly, a reset function needs to be implemented. The design for the CPU reset circuit is shown with the VHDL code below. It has 2 inputs, clock and reset. With reset set to 1, ENABLE_PD becomes low and it forces the control unit into state T0 and sets CLR_PC to high, causing the program counter to clear. Once the reset signal returns to 0, ENABLE_PD and CLR_PC remain low and high respectively for 4 clock cycles (states T0, T1, and T2), allowing the system to stabilize.

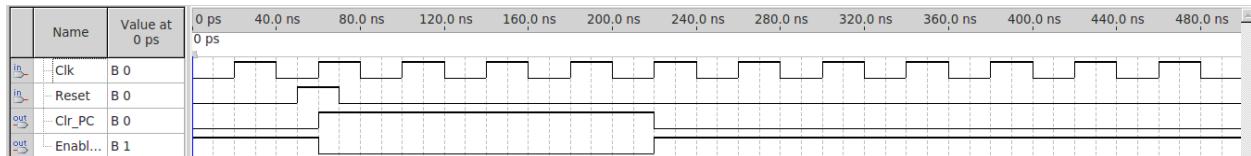
```
1  LIBRARY ieee;
2  USE ieee.std_logic_1164.ALL;
3  USE ieee.std_logic_arith.ALL;
4  USE ieee.std_logic_unsigned.ALL;
5
6  ENTITY reset_circuit IS
7  PORT(
8      Reset : IN STD_LOGIC;
9      Clk : IN STD_LOGIC;
10     Enable_PD : OUT STD_LOGIC := '1';
11     Clr_PC : OUT STD_LOGIC
12 );
13 END reset_circuit;
14
15 ARCHITECTURE Behavior OF reset_circuit IS
16   TYPE clkNum IS (clk0, clk1, clk2, clk3);
17   SIGNAL present_clk: clkNum;
18 BEGIN
19   process(Clk)begin
20     if rising_edge(Clk) then
21       if Reset = '1' then
22         Clr_PC <= '1';
23         Enable_PD <= '0';
24         present_clk <= clk0;
25       elsif present_clk <= clk0 then
26         present_clk <= clk1;
27       elsif present_clk <= clk1 then
28         present_clk <= clk1;
```

```

29      elsif present_clk <= clk2 then
30          present_clk <= clk1;
31      elsif present_clk <= clk3 then
32          Clr_PC <='0';
33          Enable_PD <= '1';
34          end if;
35      end if;
36  end process;
37 END Behavior;

```

Reset_circuit VHDL Code



Reset_circuit Waveform

Part 2 - The Complete CPU System:

This lab integrates all previously developed sub-systems: the datapath, control unit, and reset circuit, into a complete CPU system. The goal is to ensure proper interconnection between these components, verify correct functionality through simulation, and demonstrate a working CPU as specified in the CPU Testing and CPU Specification documents. The CPU is capable of performing many different functions, determined by the control unit and data path. The reset circuit allows the CPU to reset by clearing the program counter. Further detail on the previous components are explained in the previous labs.

```

1  library ieee;
2  use ieee.std_logic_1164.all;
3
4  ENTITY CPU_TEST_Sim IS
5    PORT(
6      cpuClk : in std_logic;
7      memClk : in std_logic;
8      rst : in std_logic;
9      -- Debug data.
10     outA, outB : out std_logic_vector(31 downto 0);
11     outC, outZ : out std_logic;
12     outIR : out std_logic_vector(31 downto 0);
13     outPC : out std_logic_vector(31 downto 0);
14     -- Processor-Inst Memory Interface.
15     addrOut : out std_logic_vector(5 downto 0);
16     wEn : out std_logic;
17     memDataOut : out std_logic_vector(31 downto 0);
18     memDataIn : out std_logic_vector(31 downto 0);
19     -- Processor State
20     T_Info : out std_logic_vector(2 downto 0);
21     --data Memory Interface
22     wen_mem, en_mem : out std_logic);
23 END CPU_TEST_Sim;
24
25 ARCHITECTURE behavior OF CPU_TEST_Sim IS
26   COMPONENT system_memory
27     PORT(
28       address : IN STD_LOGIC_VECTOR (5 DOWNTO 0);
29
30       clock : IN STD_LOGIC ;
31       data : IN STD_LOGIC_VECTOR (31 DOWNTO 0);
32       wren : IN STD_LOGIC ;
33       q : OUT STD_LOGIC_VECTOR (31 DOWNTO 0)
34     );
35   END COMPONENT;
36
37   COMPONENT cpul
38     PORT(
39       clk : in std_logic;
40       mem_clk : in std_logic;
41       rst : in std_logic;
42       dataIn : in std_logic_vector(31 downto 0);
43       dataOut : out std_logic_vector(31 downto 0);
44       addrOut : out std_logic_vector(31 downto 0);
45       wEn : out std_logic;
46       dOutA, dOutB : out std_logic_vector(31 downto 0);
47       dOutC, dOutZ : out std_logic;
48       dOutIR : out std_logic_vector(31 downto 0);
49       dOutPC : out std_logic_vector(31 downto 0);
50       outT : out std_logic_vector(2 downto 0);
51       wen_mem, en_mem : out std_logic);
52   END COMPONENT;
53
54   signal cpu_to_mem: std_logic_vector(31 downto 0);
55   signal mem_to_cpu: std_logic_vector(31 downto 0);
56   signal add_from_cpu: std_logic_vector(31 downto 0);
57   signal wen_from_cpu: std_logic;

```

```

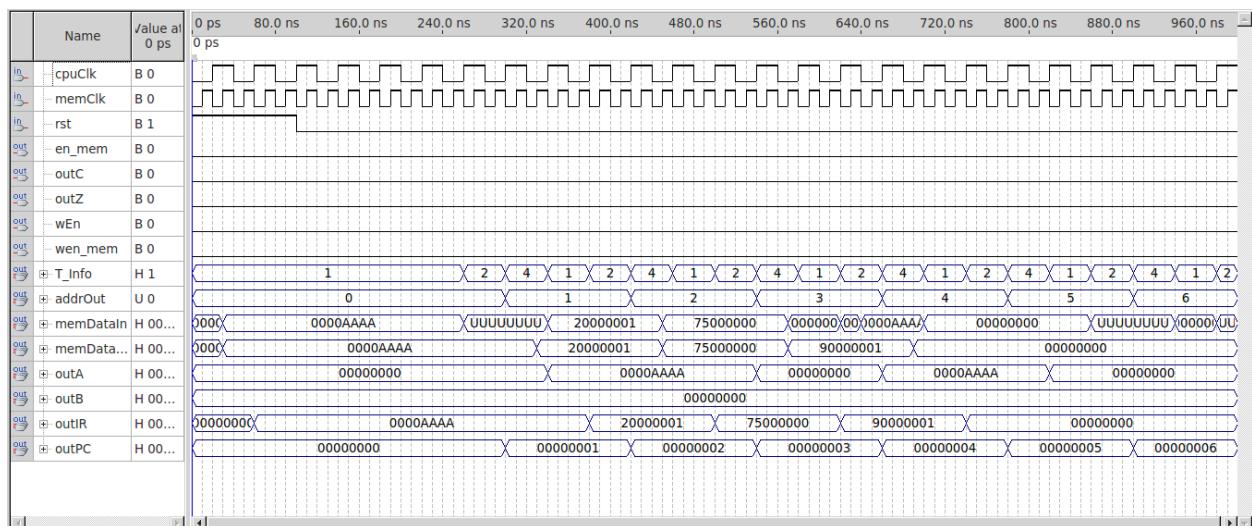
57
58      BEGIN
59          -- Component instantiations.
60          main_memory : system_memory
61          PORT MAP (
62              address => add_from_cpu(5 downto 0),
63              clock => memClk,
64              data => cpu_to_mem,
65              wren => wen_from_cpu,
66              q => mem_to_cpu
67          );
68          main_processor : cpul
69          PORT MAP (
70              clk => cpuClk,
71              mem_clk => memClk,
72              rst => rst,
73              dataIn => mem_to_cpu,
74              dataOut => cpu_to_mem,
75              addrOut => add_from_cpu,
76              wEn => wen_from_cpu,
77              dOutA => outA,
78              dOutB => outB,
79              dOutC => outC,
80              dOutZ => outZ,
81              dOutIR => outIR,
82              dOutPC => outPC,
83              outT => T_Info,
84              wen_mem => wen_mem,
85              outT => T_Info,
86              wen_mem => wen_mem,
87              en_mem => en_mem
88          );
89          addrOut <= add_from_cpu(5 downto 0);
90          wEn <= wen_from_cpu;
91          memDataOut <= mem_to_cpu;
92          memDataIn <= cpu_to_mem;
93      END behavior;

```

CPU Test Simulation VHDL Code

LDAL STA CLRA LDA:

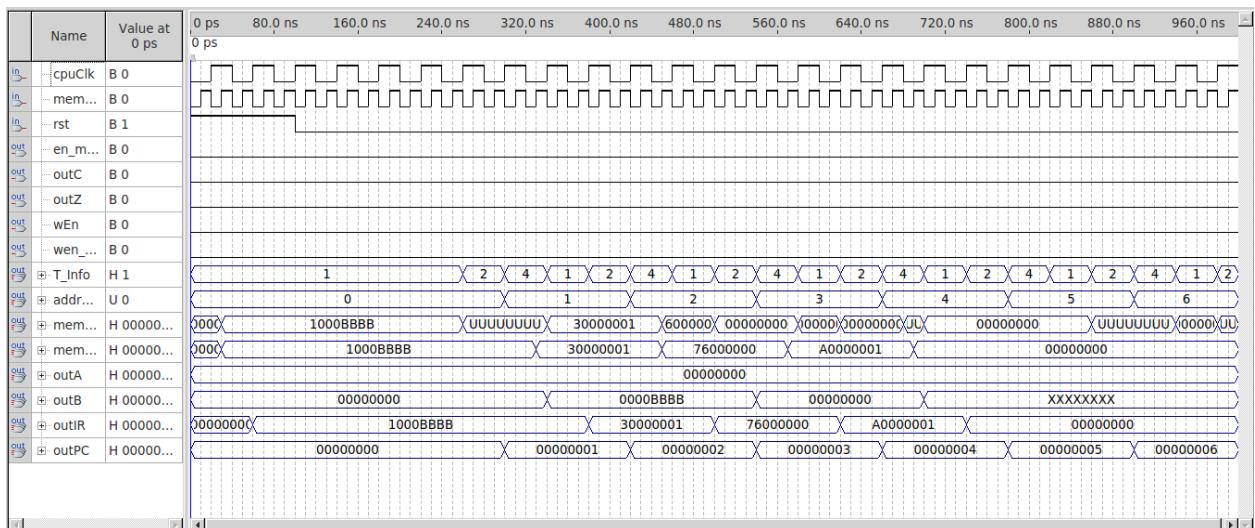
Memory Initialization File for LDAI, STA, CLRA, LDA.



Waveform for LDAI, STA, CLRA, LDA.

LDBI, STB, CLR&B, LDB:

Memory Initialization File for LDBI, STB, CLRB, LDB.

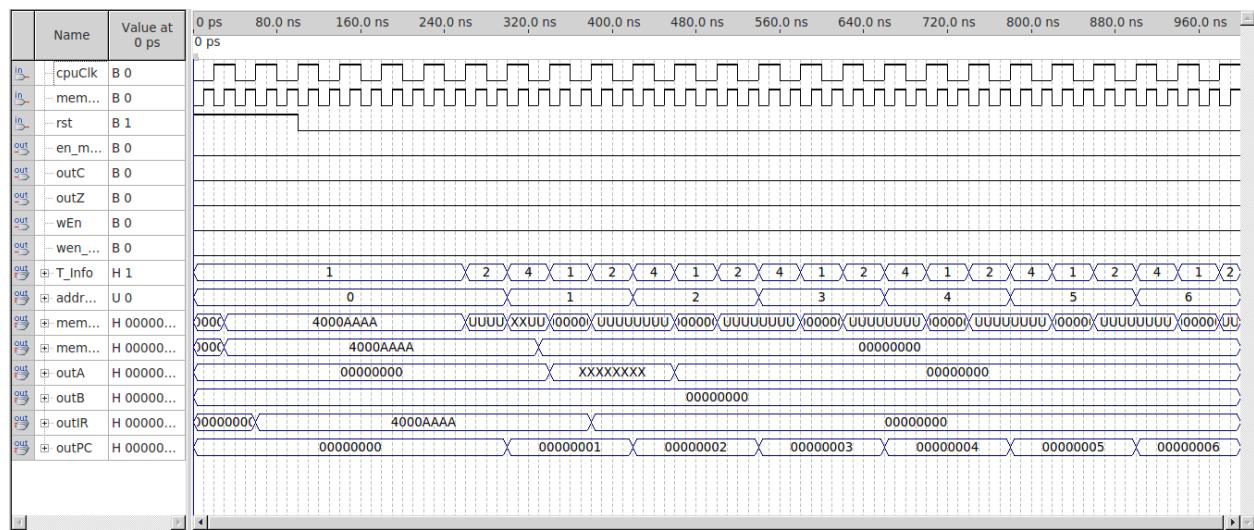


Waveform for LDBI, STB, CLRB, LDB.

LUI:

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

Memory Initialization File for LUI.

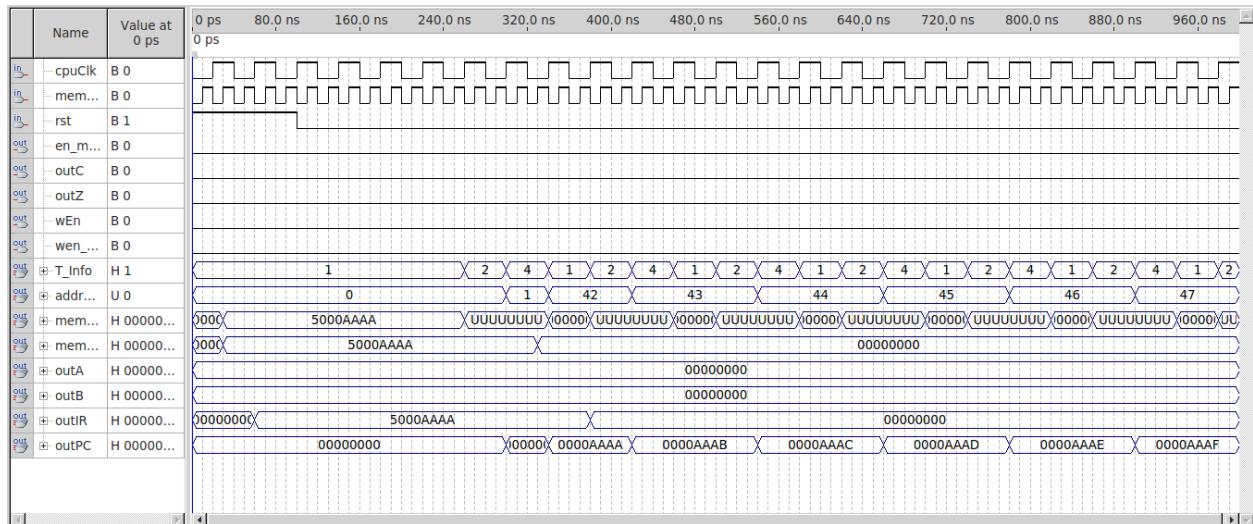


Waveform for LUI.

JMP:

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

Memory Initialization File for JMP.

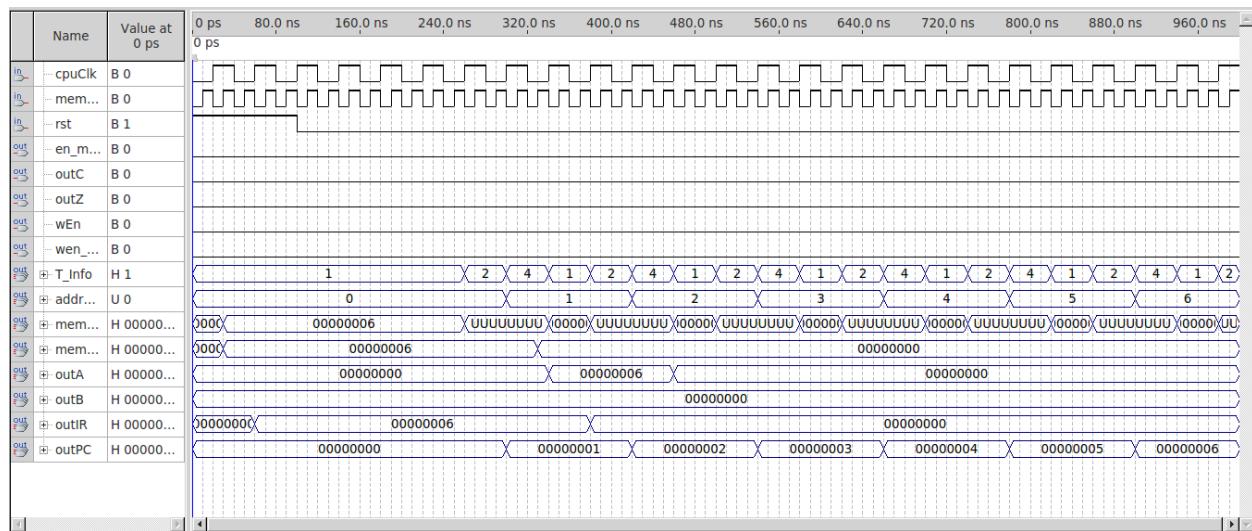


Waveform for JMP.

ANDI:

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

Memory Initialization File for ANDI.

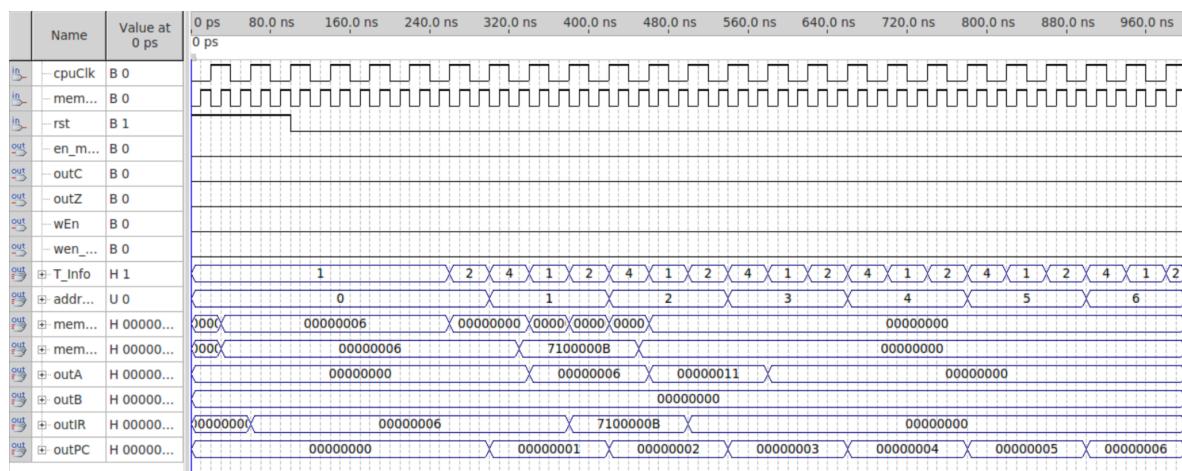


Waveform for ANDI.

ADDI:

Addr	+0	+1	+2	+3	+4	+5	+6	+7	ASCII
0	00000006	7100000B	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

Memory Initialization File for ADDI.

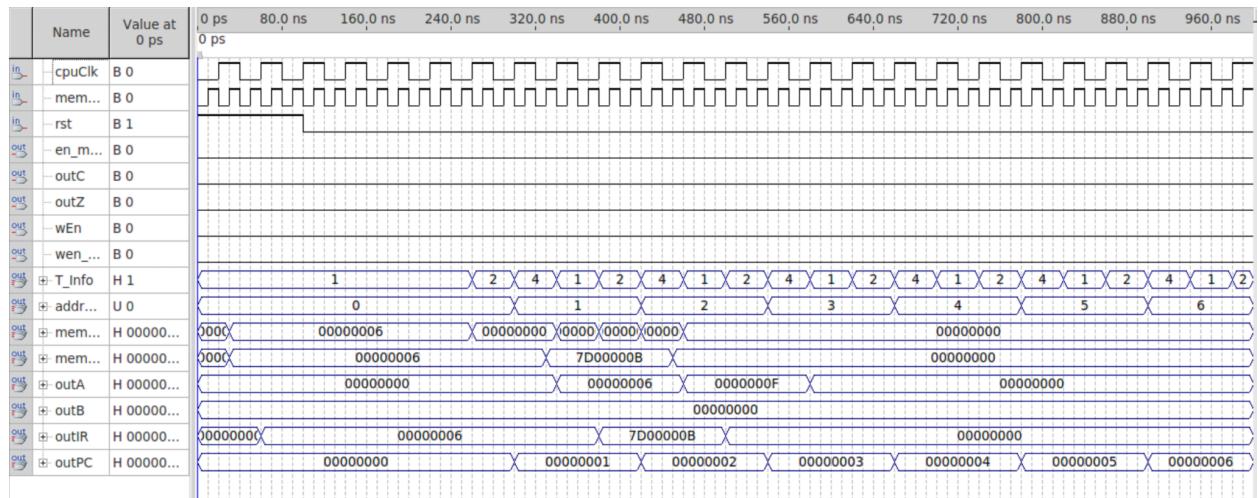


Waveform for ADDI.

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

Memory Initialization File for ORI.

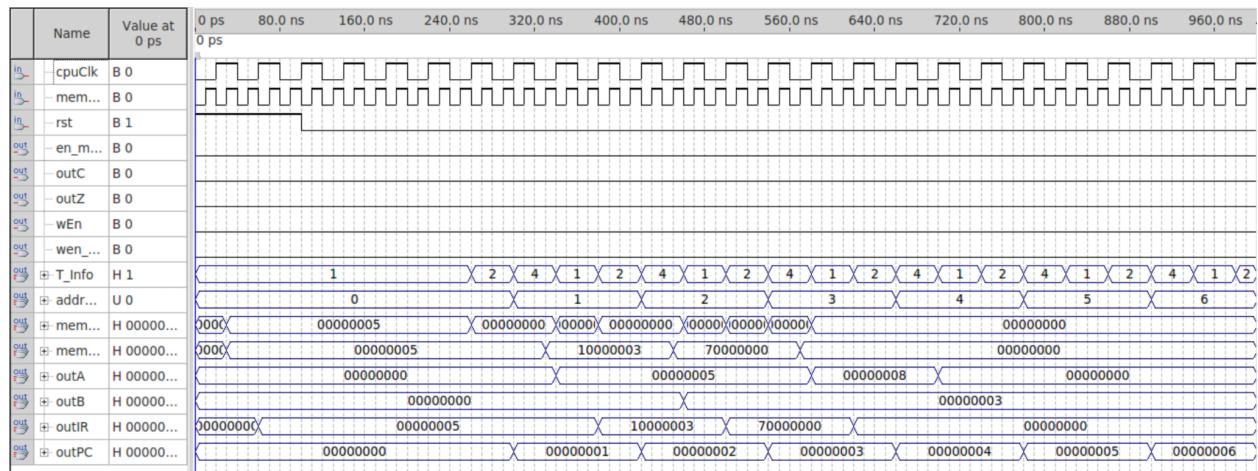


Waveform for ORI.

ADD:

Addr	+0	+1	+2	+3	+4	+5	+6	+7	ASCII
0	00000005	10000003	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

Memory Initialization File for ADD.

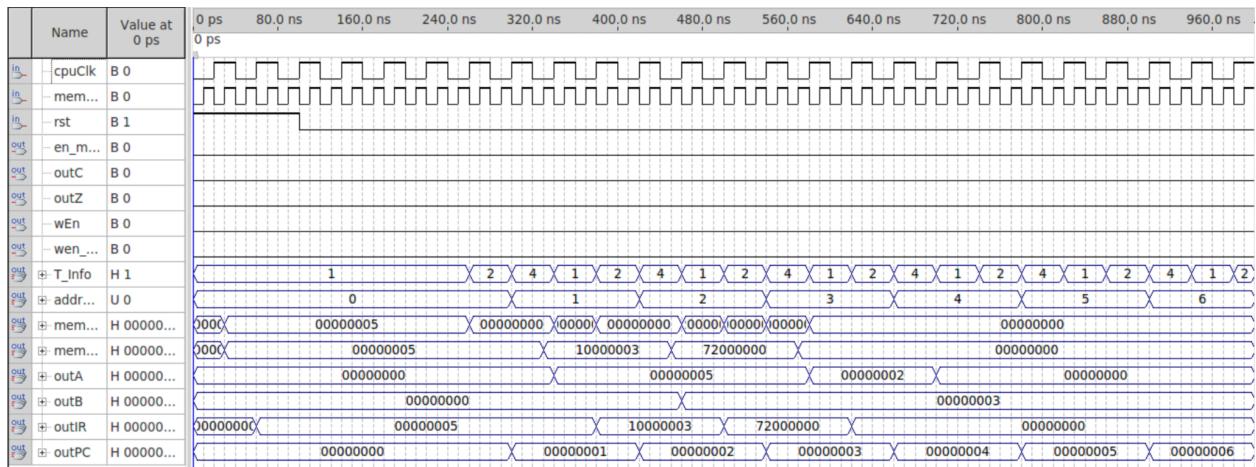


Waveform for ADD.

SUB:

Addr	+0	+1	+2	+3	+4	+5	+6	+7	ASCII
0	00000005	10000003	72000000	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

Memory Initialization File for SUB.

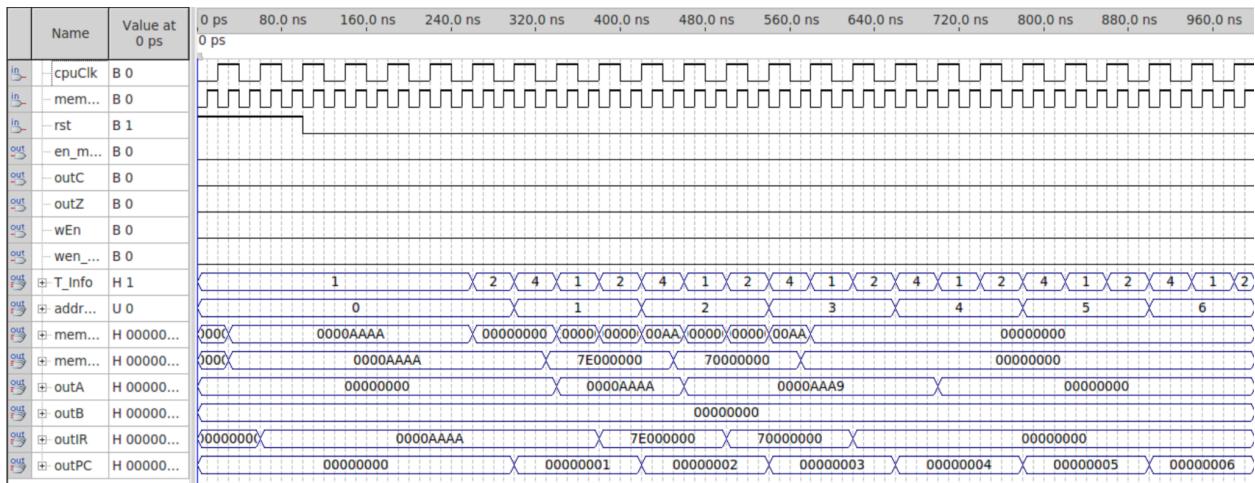


Waveform for SUB.

DECA:

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

Memory Initialization File for DECA.

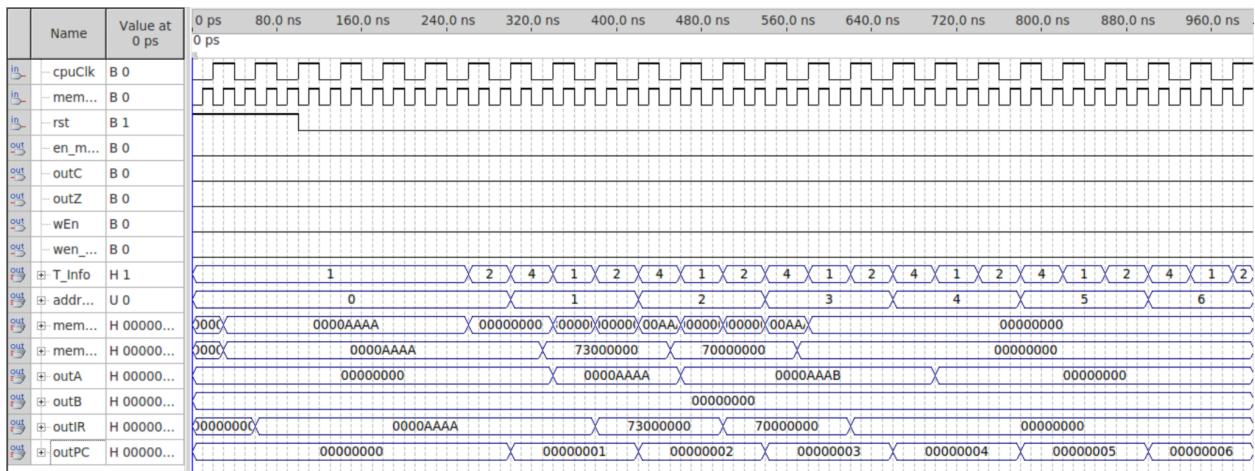


Waveform for DECA.

INCA:

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

Memory Initialization File for INCA.

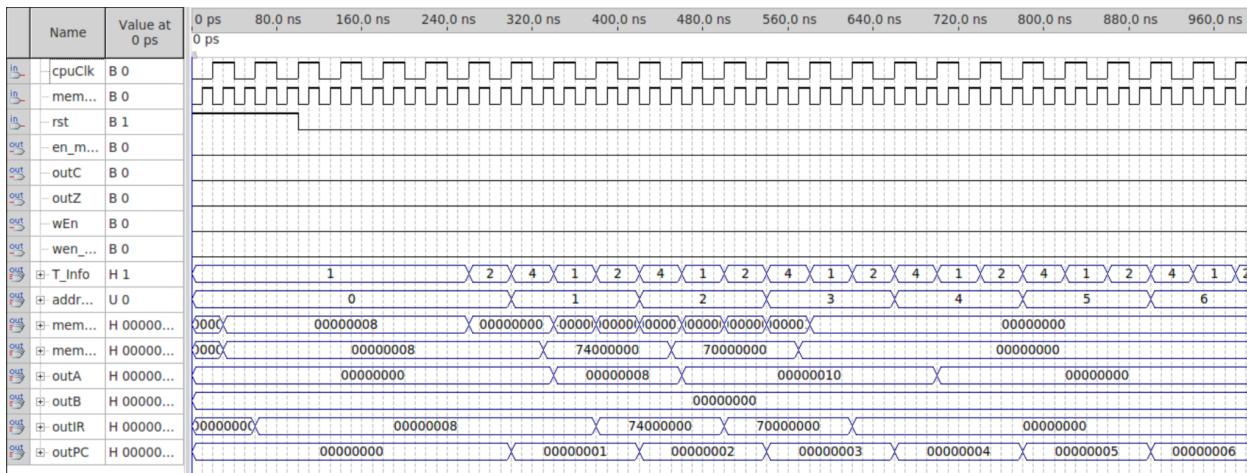


Waveform for INCA.

ROL:

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

Memory Initialization File for ROL.

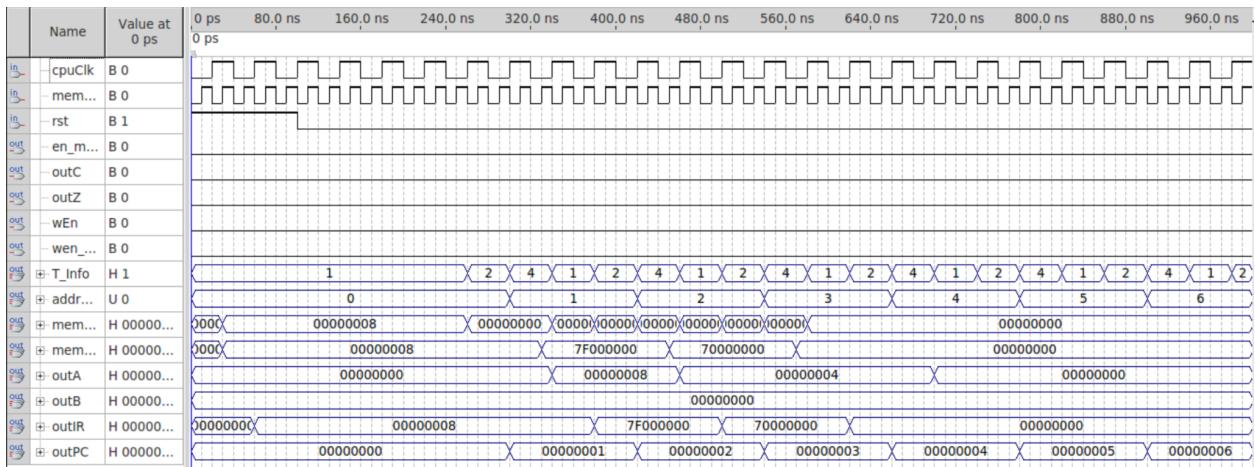


Waveform for ROL.

ROR:

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

Memory Initialization File for ROR.

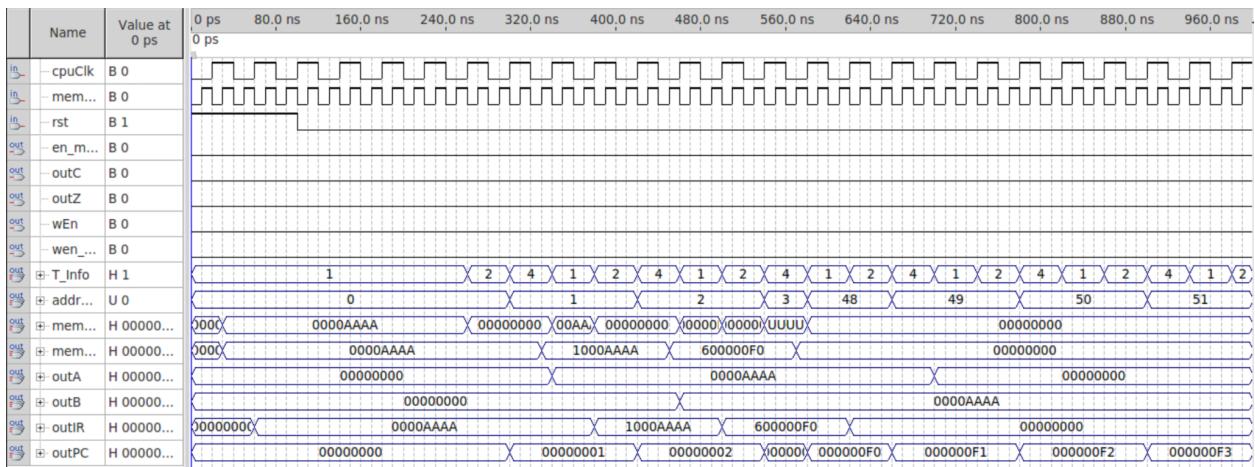


Waveform for ROR.

BEQ:

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

Memory Initialization File for BEQ.

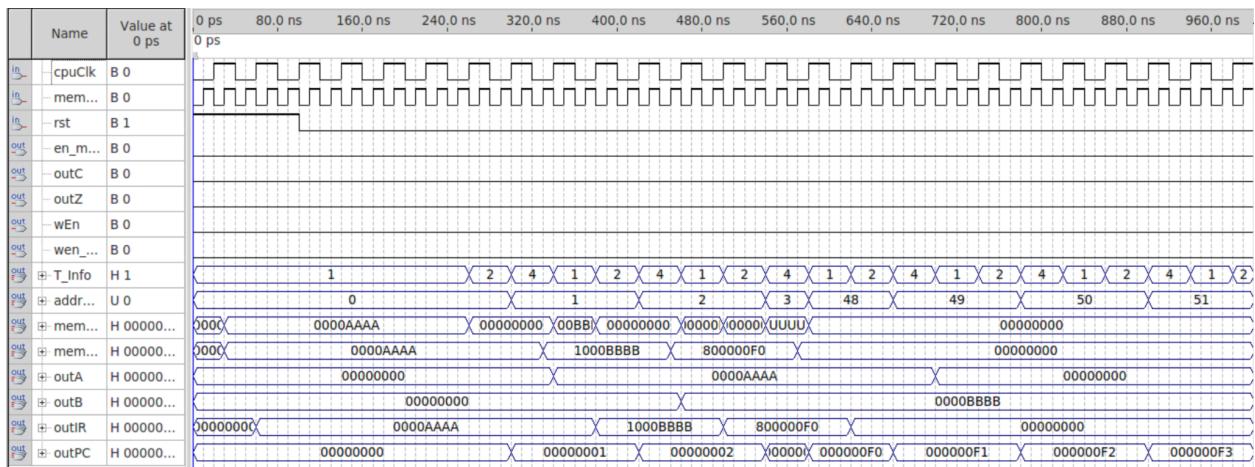


Waveform for BEQ.

BNE:

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	000000F0

Memory Initialization File for BNE.



Waveform for BNE.

Conclusion:

As the final lab for COE608, I was successfully able to integrate a fully functional CPU. Utilizing and designing components such as the datapath, control unit, and reset circuit, the CPU is able to cycle through different operations, read and write to registers, and perform different computations. Through both waveform and on-board simulations, I was able to confirm that the system adheres to the specifications of the lab.