Ryerson University

**Department of Electrical & Computer Engineering**
Faculty of Engineering & Architectural Science

| | |
|---|---|
| **Course Title:** | |
| **Course Number:** | |
| **Semester/Year (e.g.F2016)** | |

| | |
|---|---|
| **Instructor:** | |

| | |
|---|---|
| *Assignment/Lab Number:* | |
| *Assignment/Lab Title:* | |

| | |
|---|---|
| *Submission Date:* | |
| *Due Date:* | |

| Student LAST Name | Student FIRST Name | Student Number | Section | Signature* |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |

Reset Form

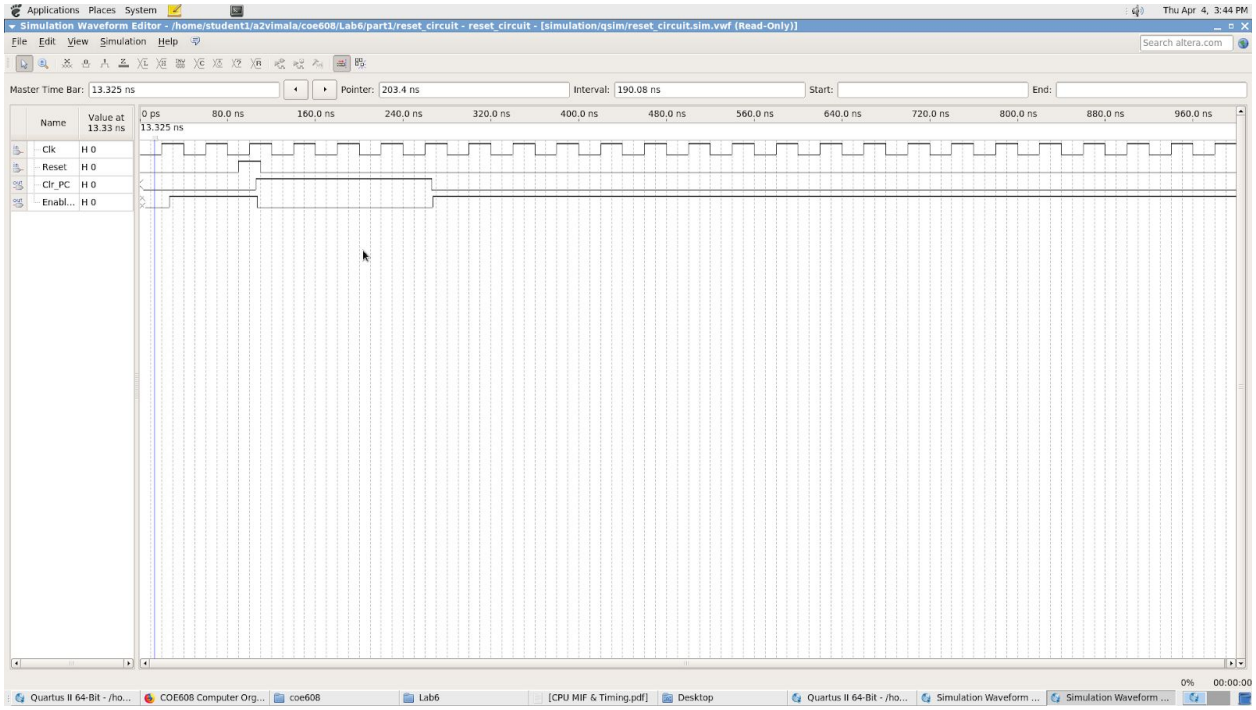
*By signing above you attest that you have contributed to this written lab report and confirm that all work you have contributed to this lab report is your own work. Any suspicion of copying or plagiarism in this work will result in an investigation of Academic Misconduct and may result in a "0" on the 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: http://www.ryerson.ca/senate/current/pol60.pdf

# PART 1: CPU Reset Circuitry

VHDL CODE OF RESET CIRCUIT

```
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;
      Clr_PC : OUT STD_LOGIC
   );
END reset_circuit;

ARCHITECTURE description OF reset_circuit IS
signal counter: integer range 0 to 5;
BEGIN
   process(clk,Reset)
   variable counter :integer:=5;
   BEGIN
      if (rising_edge(clk) and (clk='1'))
         then
         if(Reset='1')
            then
            counter:=0;
         end if;
         if(counter<4)
         then
            Enable_PD<='0';
            Clr_PC<='1';
            counter:=counter+1;
         else
            Enable_PD<='1';
            Clr_PC<='0';
         end if;
      end if;
   end process;
END description;
```

SIMULATION OUTPUT OF RESET CIRCUIT




# PART 2: The Complete CPU System

VHDL CODE OF CPU

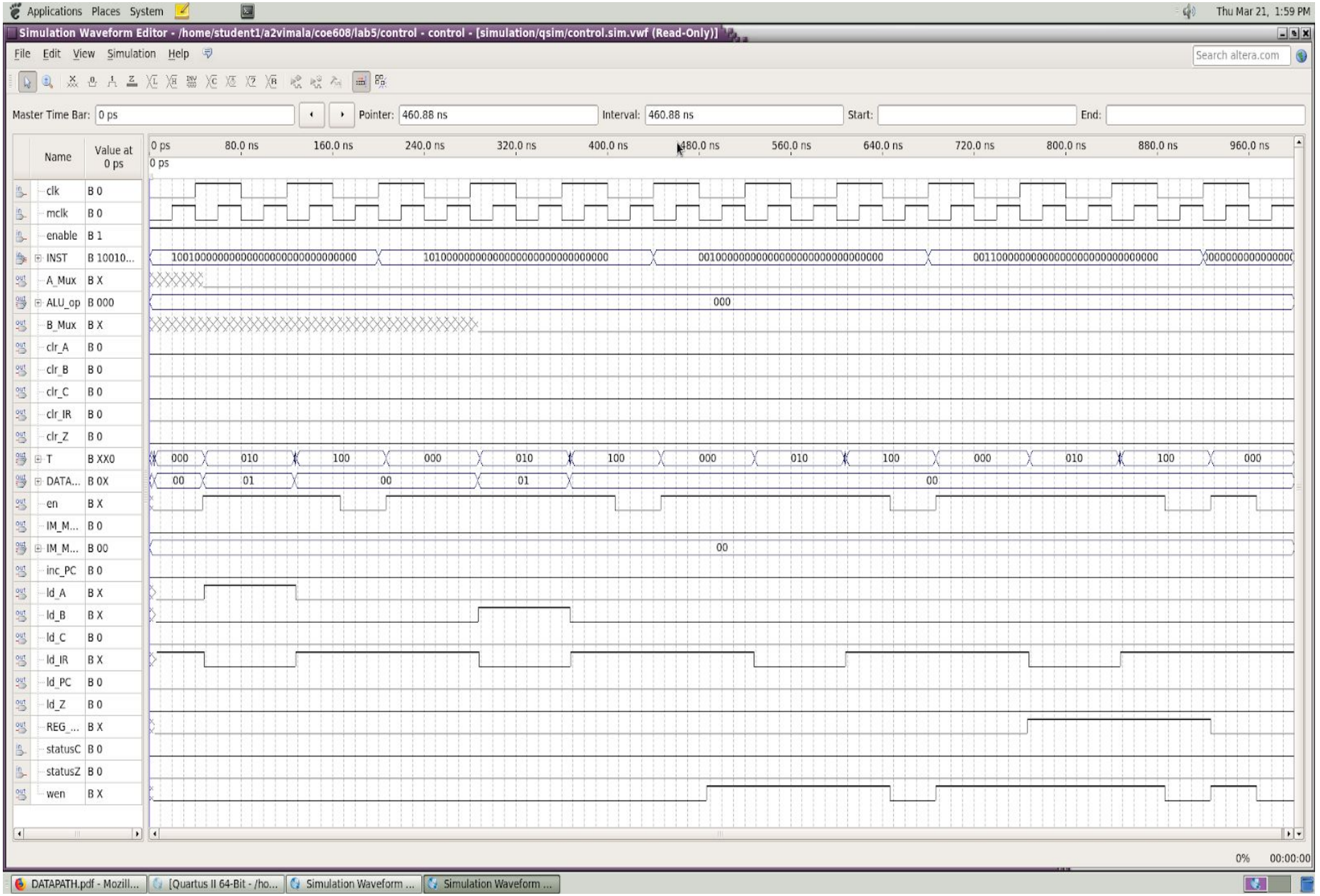
```
library ieee;
use ieee.std_logic_1164.all;

ENTITY cpu1 is
PORT
    (
        -- Input ports
        clk       : in  std_logic;
        mem_clk   : in  std_logic;
        rst       : in  std_logic;
        dataIn    : in  std_logic_vector(31 downto 0);
        -- Output ports
        dataOut      : out std_logic_vector(31 downto 0);
        addrOut      : out std_logic_vector(31 downto 0);
        wEn          : out std_logic;
        -- Debug data.
        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 cpu1;

ARCHITECTURE behavior OF cpu1 IS

    COMPONENT reset_circuit
    PORT
        (
            Reset : IN STD_LOGIC;
            Clk : IN STD_LOGIC;
            Enable_PD : OUT STD_LOGIC;
            Clr_PC : OUT STD_LOGIC
        );
    END COMPONENT;
```

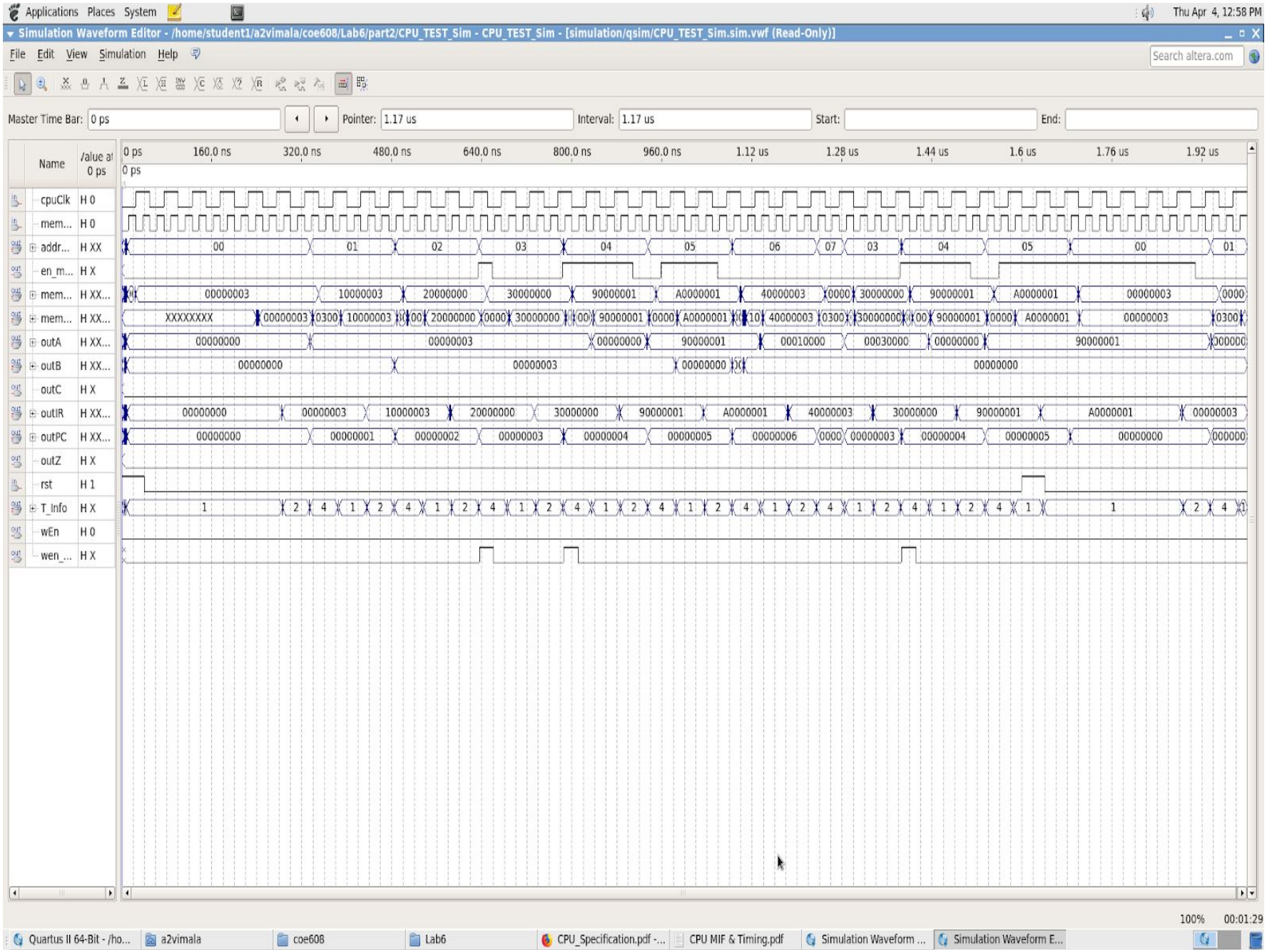
```
COMPONENT control
PORT
    (
       clk, mclk: IN STD_LOGIC;
       enable: IN STD_LOGIC;
       statusC, statusZ: IN STD_LOGIC;
       INST: IN STD_LOGIC_VECTOR(31 DOWNTO 0);
       A_Mux, B_Mux: OUT STD_LOGIC;
       IM_MUX1, REG_Mux: OUT STD_LOGIC;
       IM_MUX2, DATA_Mux: OUT STD_LOGIC_VECTOR(1 DOWNTO 0);
       ALU_op: OUT STD_LOGIC_VECTOR(2 DOWNTO 0);
       inc_PC, ld_PC: OUT STD_LOGIC;
       clr_IR: OUT STD_LOGIC;
       ld_IR: OUT STD_LOGIC;
       clr_A, clr_B, clr_C, clr_Z: OUT STD_LOGIC;
       ld_A, ld_B, ld_C, ld_Z: OUT STD_LOGIC;
       T: OUT STD_LOGIC_VECTOR(2 DOWNTO 0);
       wen, en: OUT STD_LOGIC
    );
end COMPONENT;

COMPONENT datapath
PORT
    (
       Clk,mClk: IN STD_LOGIC; -- clock Signal
       --Memory Signals
       WEN, EN : IN STD_LOGIC;
       -- Register Control Signals (CLR and LD).
       Clr_A , Ld_A : IN STD_LOGIC;
       Clr_B , Ld_B : IN STD_LOGIC;
       Clr_C , Ld_C : IN STD_LOGIC;
       Clr_Z , Ld_Z : IN STD_LOGIC;
       Clr_PC , Ld_PC : IN STD_LOGIC;
       Clr_IR , Ld_IR : IN STD_LOGIC;

       -- Register outputs (Some needed to feed back to control unit. Others pulled out for testing.
       Out_A : OUT STD_LOGIC_VECTOR(31 DOWNTO 0);
        Out_A : OUT STD_LOGIC_VECTOR(31 DOWNTO 0);
        Out_B : OUT STD_LOGIC_VECTOR(31 DOWNTO 0);
        Out_C : OUT STD_LOGIC;
        Out_Z : OUT STD_LOGIC;
        Out_PC : OUT STD_LOGIC_VECTOR(31 DOWNTO 0);
        Out_IR : OUT STD_LOGIC_VECTOR(31 DOWNTO 0);

        -- Special inputs to PC.
        Inc_PC : IN STD_LOGIC;

        -- Address and Data Bus signals for debugging.
        ADDR_OUT : OUT STD_LOGIC_VECTOR(31 DOWNTO 0);
        DATA_IN : IN STD_LOGIC_VECTOR(31 DOWNTO 0);
        DATA_OUT: OUT STD_LOGIC_VECTOR(31 DOWNTO 0);--, MEM_OUT, MEM_IN
  --    MEM_ADDR : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);

        -- Various MUX controls.
        DATA_MUX: IN STD_LOGIC_VECTOR(1 DOWNTO 0);
        Reg_Mux : IN STD_LOGIC;
        A_Mux, B_Mux : IN STD_LOGIC;
        IM_MUX1 : IN STD_LOGIC;
        IM_MUX2 : IN STD_LOGIC_VECTOR(1 DOWNTO 0);

        -- ALU Operations.
        ALU_OP : IN STD_LOGIC_VECTOR(2 DOWNTO 0)

    );
  END COMPONENT;
-- Internal signals--------
signal         Clr_A1 , Ld_A1              :  STD_LOGIC;
signal         Clr_B1 , Ld_B1              :  STD_LOGIC;
signal         Clr_C1 , Ld_C1              :  STD_LOGIC;
signal         Clr_Z1 , Ld_Z1              :  STD_LOGIC;
signal         Clr_PC1 , Ld_PC1,Inc_PC1    :  STD_LOGIC;
signal         Clr_IR1 , Ld_IR1         :  STD_LOGIC;
signal         WEN1, EN1                :  STD_LOGIC;
signal         A_Mux1,B_Mux1            :  STD_LOGIC;
```

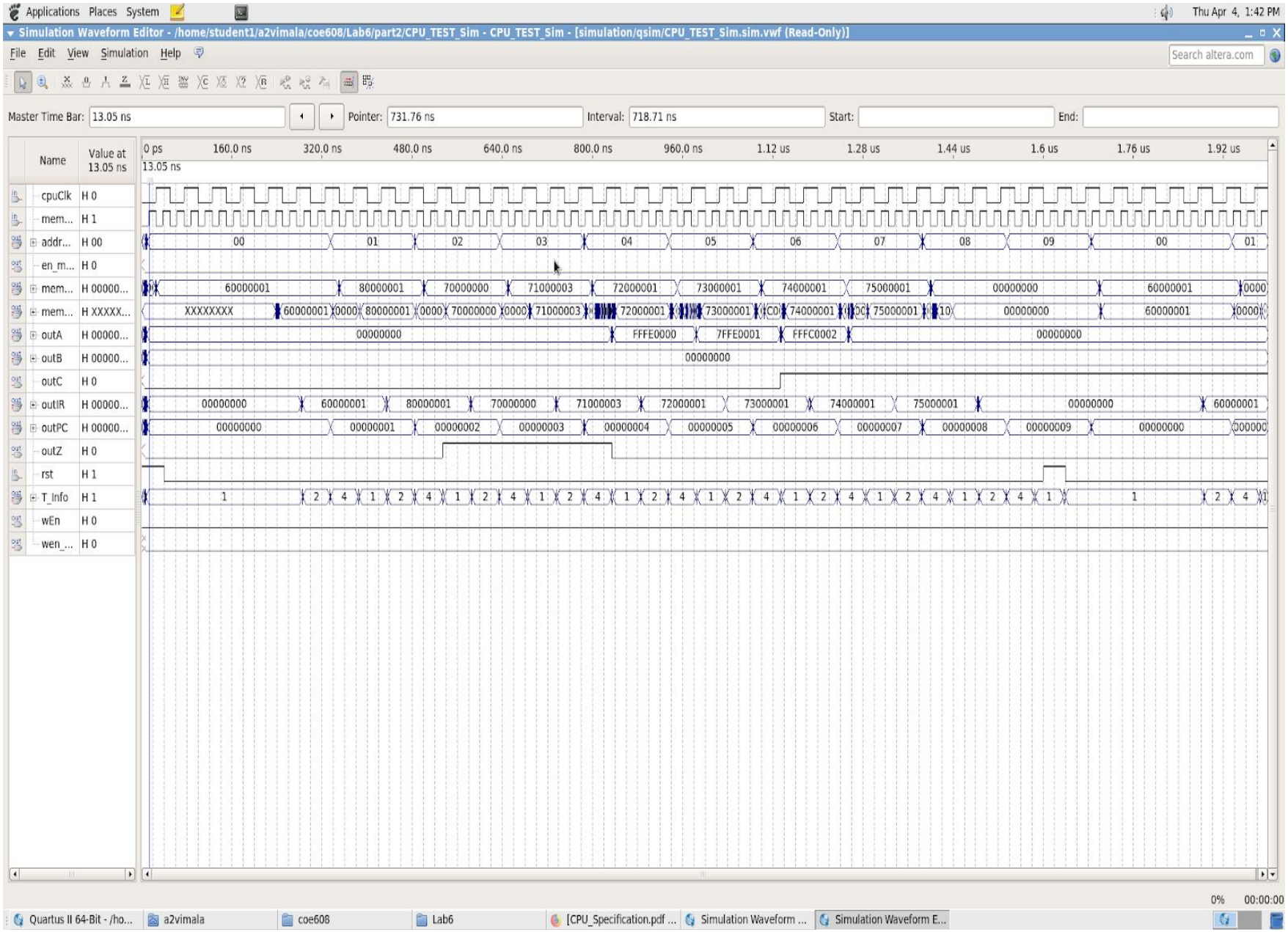
```
signal         Clr_PC1 , Ld_PC1,Inc_PC1   :  STD_LOGIC;
signal         Clr_IR1 , Ld_IR1        :  STD_LOGIC;
signal         WEN1, EN1               :  STD_LOGIC;
signal         A_Mux1,B_Mux1           :  STD_LOGIC;
signal         IM_MUX11, REG_Mux1         :  STD_LOGIC;
signal         IM_MUX21, DATA_Mux1        :  STD_LOGIC_VECTOR(1 DOWNTO 0);
signal         ALU_OP1                 :  STD_LOGIC_VECTOR(2 DOWNTO 0);
signal         enable1                 :  STD_LOGIC;
signal         statusC1,statusZ1       :  STD_LOGIC;
signal         Out_C1,Out_Z1: STD_LOGIC;
signal         useless1,useless2,useless3,useless4,useless5 :  STD_LOGIC_VECTOR(31 DOWNTO 0);
signal         us7                  :  STD_LOGIC_VECTOR(7 DOWNTO 0);

  Begin

  Dpth:datapath
  PORT MAP
     (   -- clock Signal
        Clk=>clk,
        mClk=>mem_clk,
        --Memory Signals
        WEN=>WEN1,
        EN=>EN1,
        -- Register Control Signals (CLR and LD).
        Clr_A=>CLR_A1,
        Ld_A=>Ld_A1,
        Clr_B=>Clr_B1 ,
        Ld_B=>Ld_B1,
        Clr_C=>Clr_C1 ,
        Ld_C=>Ld_C1,
        Clr_Z=>Clr_Z1 ,
        Ld_Z=>Ld_Z1,
        Clr_PC=>Clr_PC1,
        Ld_PC=>Ld_PC1,
        Clr_IR=>Clr_IR1 ,
        Ld_IR =>Ld_IR1,
        -- Register outputs (Some needed to feed back to control unit. Others pulled out for testing.
```

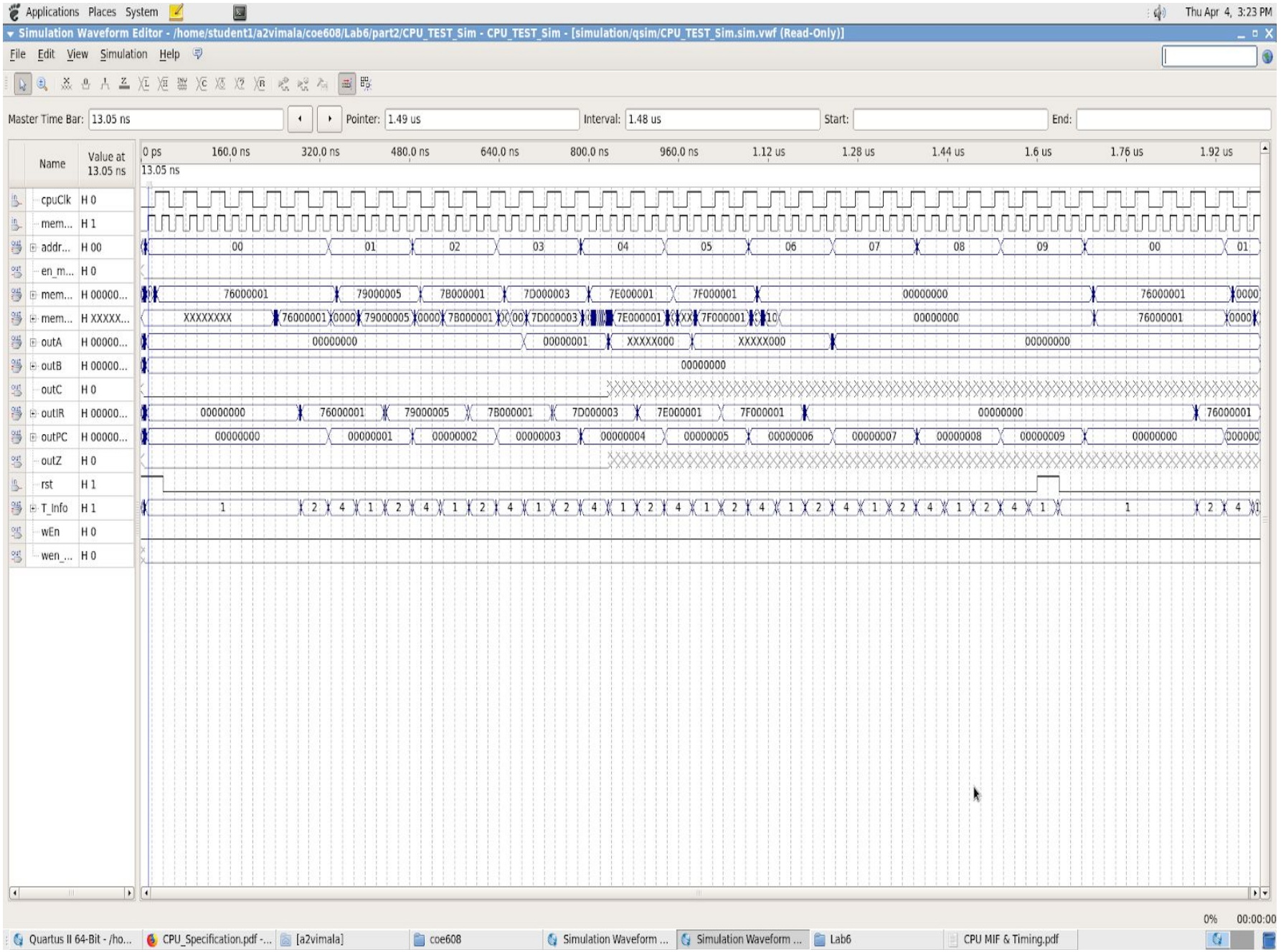
```
        -- Register outputs (Some needed to feed back to control unit. Others pulled out for testing.
        Out_A=>dOutA,
        Out_B=>dOutB,
        Out_C=>Out_C1,
        Out_Z=>Out_Z1,
        Out_PC=>dOutPC,
        Out_IR=>dOutIR,
        -- Special inputs to PC.
        Inc_PC=>Inc_PC1,
        -- Address and Data Bus signals for debugging.
        ADDR_OUT=>addrOut,
         DATA_IN=>dataIn,
        DATA_OUT=>dataOut,
        -- Various MUX controls.
        DATA_MUX=>DATA_MUX1,
        Reg_Mux=>Reg_Mux1,
        A_Mux=>A_Mux1,
        B_Mux=>B_Mux1,
        IM_MUX1=>IM_MUX11,
        IM_MUX2=>IM_MUX21,
        -- ALU Operations.
        ALU_OP=>ALU_OP1
     );

  C1:control
  PORT MAP
     (
        enable=>enable1,
        statusC=>Out_C1,
        statusZ=>Out_Z1,
        INST=>dataIn,
        A_Mux=>A_Mux1,
        B_Mux=>B_Mux1,
        IM_MUX1=>IM_MUX11,
        REG_Mux=>REG_Mux1,
        IM_MUX2=>IM_MUX21,
        DATA Mux=> DATA Mux1,
```

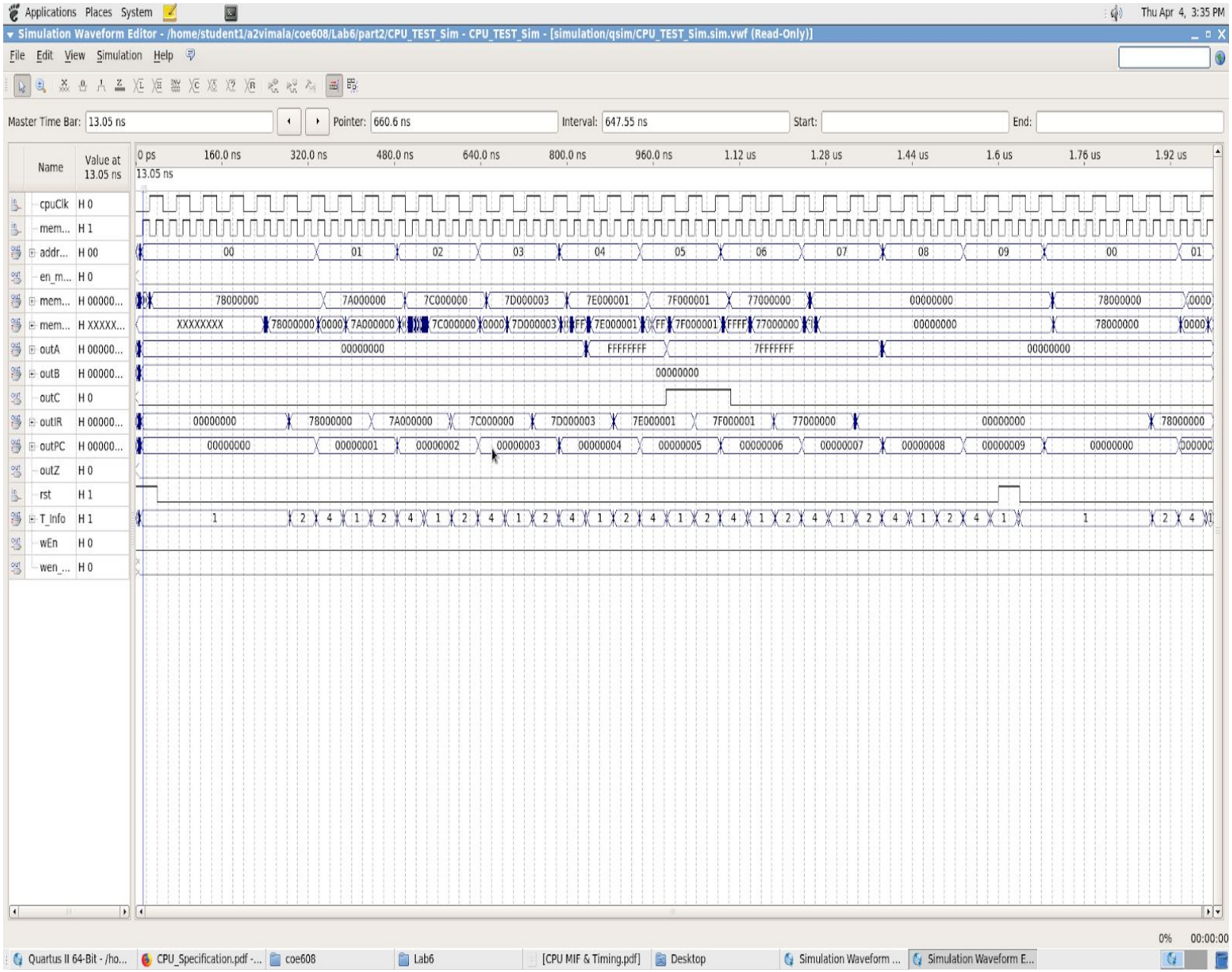
```
        DATA_Mux=> DATA_Mux1,
        ALU_OP=>ALU_OP1,
        INC_PC=>INC_PC1,
        LD_PC=>  LD_PC1,
        CLR_IR=>CLR_IR1,
        LD_IR=>  LD_IR1,
        CLR_A=>CLR_A1,
        CLR_B=>CLR_B1,
        CLR_C=>CLR_C1,
        CLR_Z=>CLR_Z1,
        LD_A=>LD_A1,
        LD_B=>LD_B1,
        LD_C=>LD_C1,
        LD_Z=>LD_Z1,
        T=>outT,
        wen=>WEN1,
        en=>EN1,
        clk=>clk,
        mclk=>mem_clk
    );

  R1:reset_circuit
  PORT MAP
    (
        Reset=>rst,
        Clk=>clk,
        Enable_PD=>enable1,
        Clr_PC=>Clr_PC1
    );
    --Final assignments
    dOutC<=Out_C1;
    dOutZ<=Out_Z1;
    wen_mem<=WEN1;
    en_mem<=EN1;
    wEn <='0';

END behavior;
```


Applications Places System
Thu Mar 21, 1:59 PM
Simulation Waveform Editor - /home/student1/a2vimala/coe608/lab5/control - control - [simulation/qsim/control.sim.vwf (Read-Only)]
File Edit View Simulation Help
Search altera.com
Master Time Bar: 0 ps
Pointer: 460.88 ns
Interval: 460.88 ns
Start:
End:
Name
Value at 0 ps
0 ps
80.0 ns
160.0 ns
240.0 ns
320.0 ns
400.0 ns
480.0 ns
560.0 ns
640.0 ns
720.0 ns
800.0 ns
880.0 ns
960.0 ns
clk B 0
mclk B 0
enable B 1
INST B 10010...
A_Mux B X
ALU_op B 000
B_Mux B X
clr_A B 0
clr_B B 0
clr_C B 0
clr_IR B 0
clr_Z B 0
T B XX0
DATA... B 0X
en B X
IM_M... B 0
IM_M... B 00
inc_PC B 0
ld_A B X
ld_B B X
ld_C B 0
ld_IR B X
ld_PC B 0
ld_Z B 0
REG_... B X
statusC B 0
statusZ B 0
wen B X
000
010
100
00
01
0%
00:00:00
DATAPATH.pdf - Mozill...
[Quartus II 64-Bit - /ho...
Simulation Waveform ...
Simulation Waveform ...


Applications Places System
Thu Apr 4, 12:58 PM
Simulation Waveform Editor - /home/student1/a2vimala/coe608/Lab6/part2/CPU_TEST_Sim - CPU_TEST_Sim - [simulation/qsim/CPU_TEST_Sim.sim.vwf (Read-Only)]
File Edit View Simulation Help
Search altera.com
Master Time Bar: 0 ps
Pointer: 1.17 us
Interval: 1.17 us
Start:
End:
Name
/alue at 0 ps
0 ps
160.0 ns
320.0 ns
480.0 ns
640.0 ns
800.0 ns
960.0 ns
1.12 us
1.28 us
1.44 us
1.6 us
1.76 us
1.92 us
cpuClk H 0
mem... H 0
addr... H XX
en_m... H X
mem... H XX...
mem... H XX...
outA H XX...
outB H XX...
outC H X
outIR H XX...
outPC H XX...
outZ H X
rst H 1
T_Info H X
wEn H 0
wen_... H X
100%
00:01:29
Quartus II 64-Bit - /ho...
a2vimala
coe608
Lab6
CPU_Specification.pdf -...
CPU MIF & Timing.pdf
Simulation Waveform ...
Simulation Waveform E...


Applications Places System
Thu Apr 4, 1:42 PM
Simulation Waveform Editor - /home/student1/a2vimala/coe608/Lab6/part2/CPU_TEST_Sim - CPU_TEST_Sim - [simulation/qsim/CPU_TEST_Sim.sim.vwf (Read-Only)]
File Edit View Simulation Help
Search altera.com
Master Time Bar: 13.05 ns
Pointer: 731.76 ns
Interval: 718.71 ns
Start:
End:
Name
Value at 13.05 ns
0 ps
160.0 ns
320.0 ns
480.0 ns
640.0 ns
800.0 ns
960.0 ns
1.12 us
1.28 us
1.44 us
1.6 us
1.76 us
1.92 us
13.05 ns
cpuClk H 0
mem... H 1
addr... H 00
en_m... H 0
mem... H 00000...
mem... H XXXXX...
outA H 00000...
outB H 00000...
outC H 0
outIR H 00000...
outPC H 00000...
outZ H 0
rst H 1
T_Info H 1
wEn H 0
wen_... H 0
0%
00:00:00
Quartus II 64-Bit - /ho...
a2vimala
coe608
Lab6
[CPU_Specification.pdf ...
Simulation Waveform ...
Simulation Waveform E...


Applications Places System
Thu Apr 4, 3:23 PM
Simulation Waveform Editor - /home/student1/a2vimala/coe608/Lab6/part2/CPU_TEST_Sim - CPU_TEST_Sim - [simulation/qsim/CPU_TEST_Sim.sim.vwf (Read-Only)]
File Edit View Simulation Help
Master Time Bar: 13.05 ns
Pointer: 1.49 us
Interval: 1.48 us
Start:
End:
Name
Value at 13.05 ns
cpuClk
mem...
addr...
en_m...
mem...
mem...
outA
outB
outC
outIR
outPC
outZ
rst
T_Info
wEn
wen_...
0 ps
160.0 ns
320.0 ns
480.0 ns
640.0 ns
800.0 ns
960.0 ns
1.12 us
1.28 us
1.44 us
1.6 us
1.76 us
1.92 us
13.05 ns
0%
00:00:00
Quartus II 64-Bit - /ho...
CPU_Specification.pdf -...
[a2vimala]
coe608
Simulation Waveform ...
Simulation Waveform ...
Lab6
CPU MIF & Timing.pdf


Applications Places System
Thu Apr 4, 3:35 PM
Simulation Waveform Editor - /home/student1/a2vimala/coe608/Lab6/part2/CPU_TEST_Sim - CPU_TEST_Sim - [simulation/qsim/CPU_TEST_Sim.sim.vwf (Read-Only)]
File Edit View Simulation Help
Master Time Bar: 13.05 ns
Pointer: 660.6 ns
Interval: 647.55 ns
Start:
End:
Name
Value at 13.05 ns
cpuClk H 0
mem... H 1
addr... H 00
en_m... H 0
mem... H 00000...
mem... H XXXXX...
outA H 00000...
outB H 00000...
outC H 0
outIR H 00000...
outPC H 00000...
outZ H 0
rst H 1
T_Info H 1
wEn H 0
wen_... H 0
0 ps
160.0 ns
320.0 ns
480.0 ns
640.0 ns
800.0 ns
960.0 ns
1.12 us
1.28 us
1.44 us
1.6 us
1.76 us
1.92 us
13.05 ns
0%
00:00:00
Quartus II 64-Bit - /ho...
CPU_Specification.pdf -...
coe608
Lab6
[CPU MIF & Timing.pdf]
Desktop
Simulation Waveform ...
Simulation Waveform E...