

Below, you will see the implementation of 1-bit, 32-bit registers and program counter (32-bit register with additional control signals) using VHDL.

Part 1:

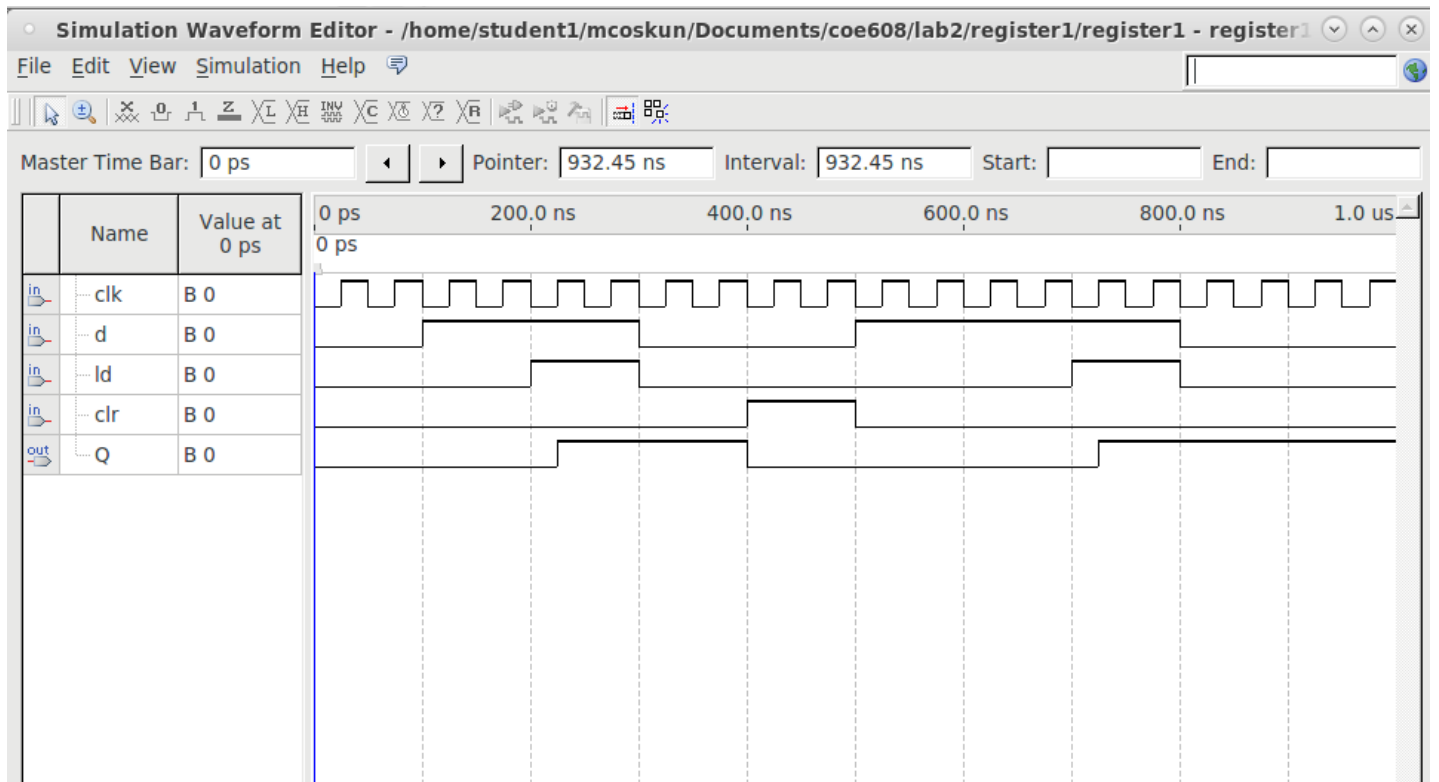
The VHDL code of the first register (Register1) can be seen below.

The screenshot displays the Quartus II IDE interface. The top toolbar shows various icons for file operations and simulation. The 'Project Navigator' on the left lists 'register1.vhd' and 'register1.vwf'. The main editor window shows the VHDL code for 'register1.vhd'. The code defines an entity 'register1' with inputs 'd', 'ld', 'clr', and 'clk', and an output 'Q'. It includes a process that implements the register logic, where 'Q' is set to '0' on a clear signal, and to 'd' on a load signal triggered by the clock. The bottom panel shows the compilation messages, indicating a successful compilation with 0 errors and 7 warnings.

```
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 register1 IS
7  PORT(
8      d : IN STD_LOGIC;
9      ld : IN STD_LOGIC;
10     clr : IN STD_LOGIC;
11     clk : IN STD_LOGIC;
12     Q : OUT STD_LOGIC
13 );
14 END register1;
15 ARCHITECTURE Behaviour OF register1 IS
16 BEGIN
17     process (ld, clr, clk)
18     begin
19         if (clr = '1') then
20             Q <= '0';
21         elsif ((clk'event and clk = '1') and (ld = '1')) then
22             Q <= d;
23         end if;
24     end process;
25 END Behaviour;
```

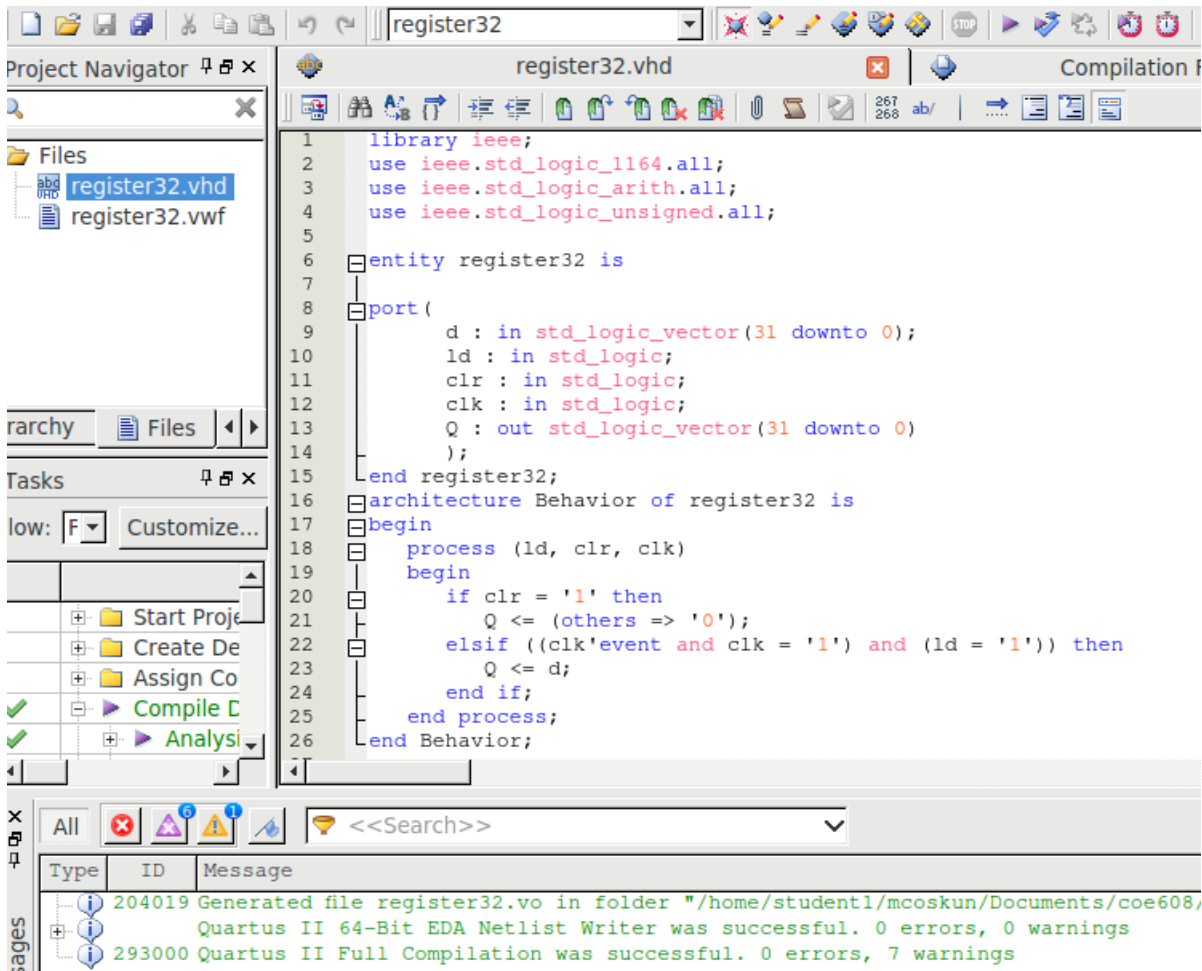
Type	ID	Message
Information	204019	Command: quartus_eda --read_settings_files=off --write_settings_files=off registe:
Information	204019	Generated file register1.vo in folder "/home/student1/mcoskun/Documents/coe608.
Information	204019	Quartus II 64-Bit EDA Netlist Writer was successful. 0 errors, 0 warnings
Information	293000	Quartus II Full Compilation was successful. 0 errors, 7 warnings

After compiling successfully, the waveform of register1 created following the lab manual instructions is displayed below.

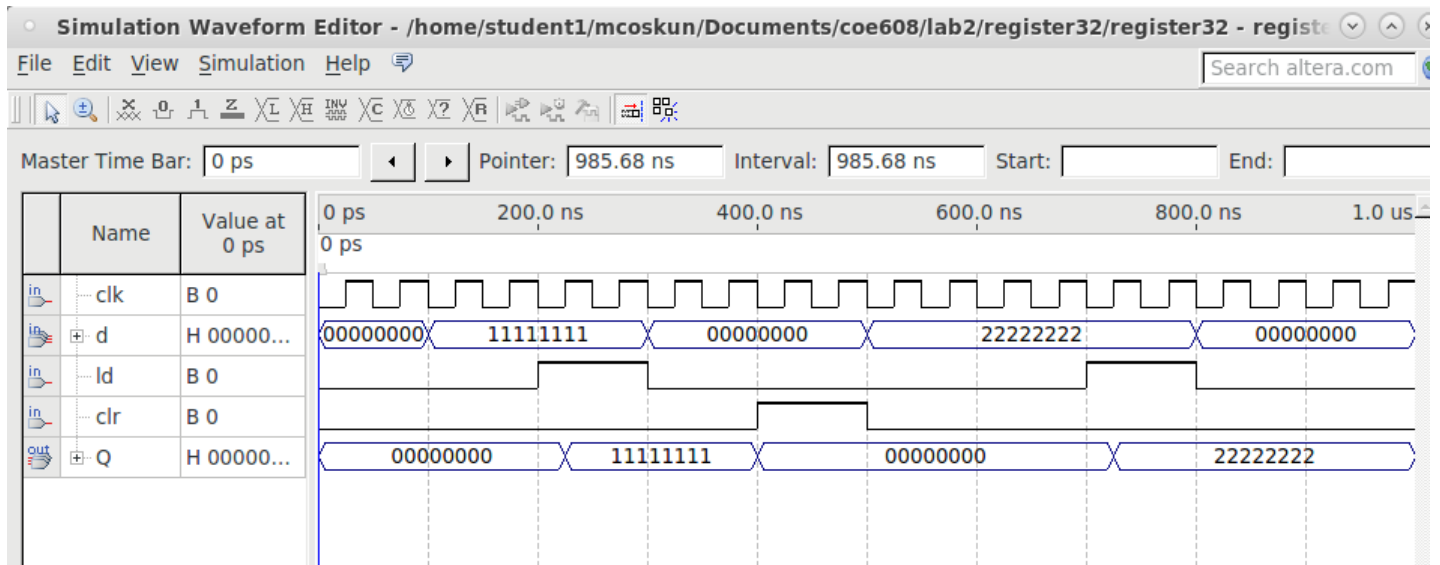


Part 2:

The VHDL code of the second register (Register32) can be seen below.



After compiling successfully, the waveform of register32 created following the lab manual instructions is displayed below.



Part 3:

The program counter consists of a multiplexer, addition code and register32 from the previous part (displayed below).

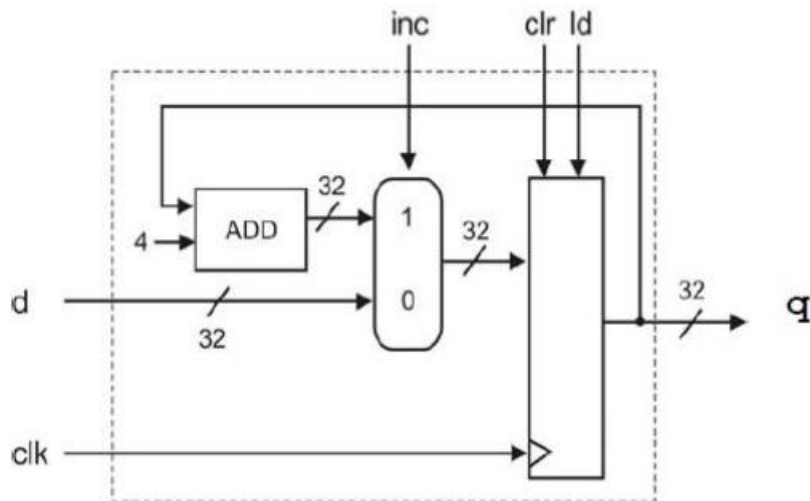


Figure 3: 32-bit Program Counter Internal Structure

The VHDL code of the addition (add) can be seen below.

Figure 4 shows the Quartus II IDE interface. The top window displays the VHDL code for the 'add' entity, which implements a 32-bit adder. The code includes library declarations for IEEE standard logic and arithmetic packages, followed by the entity definition and its behavior.

```

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 add is
7     port (A : in std_logic_vector(31 downto 0);
8           B : out std_logic_vector(31 downto 0))
9 end add;
10
11 architecture Behavior of add is
12 begin
13     B <= A + 4;
14 end Behavior;

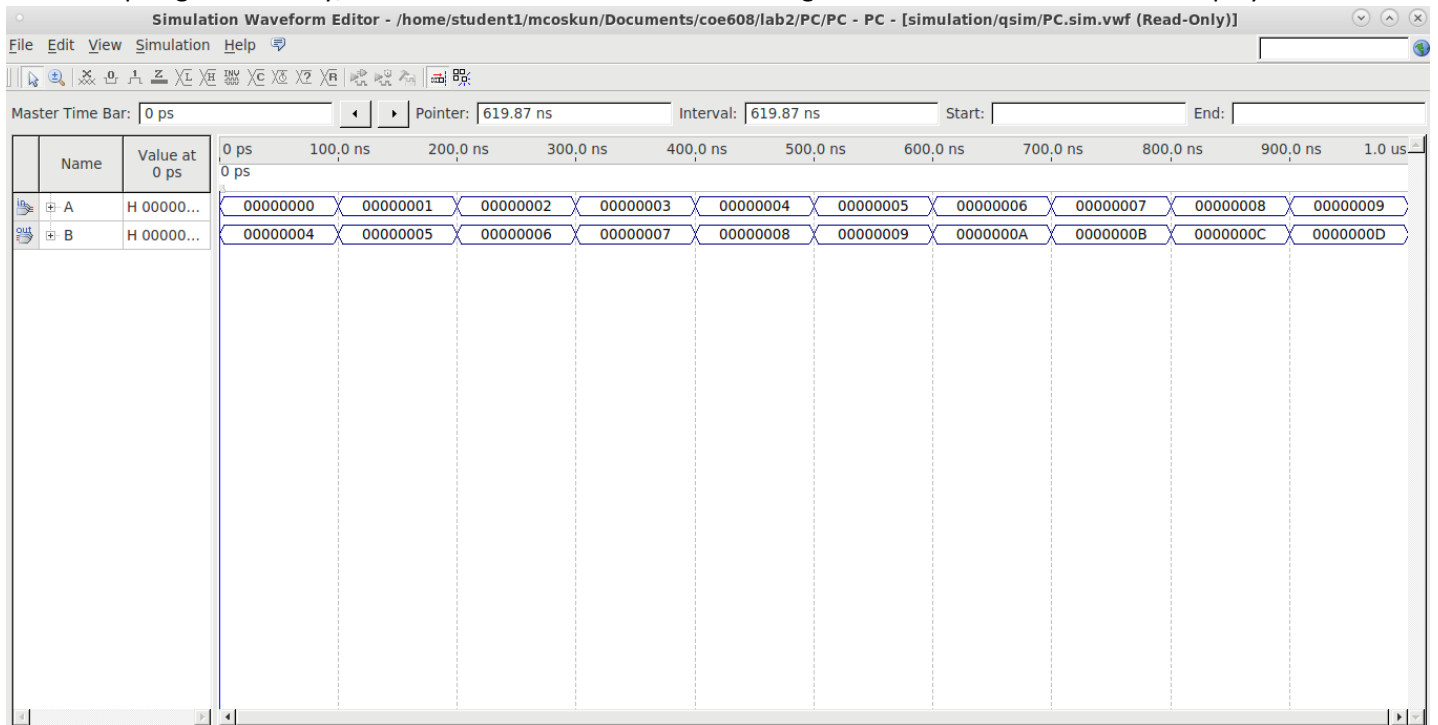
```

The bottom window displays the Messages pane, showing the compilation results. The messages indicate that the design was successfully compiled with 0 errors and 8 warnings. The messages include information about the design constraints, the compilation process, and the generation of the output files.

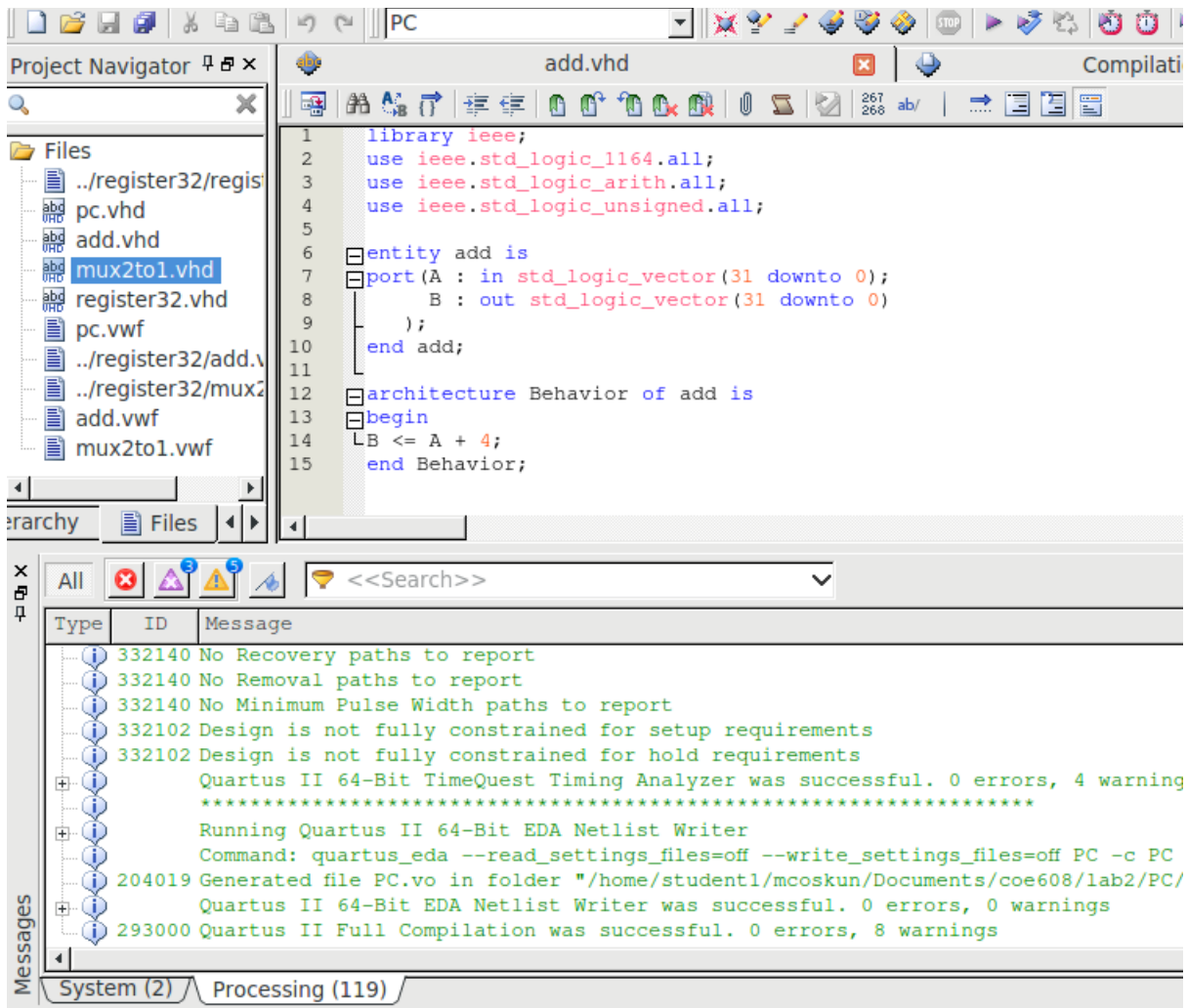
Type	ID	Message
Information	332140	No Recovery paths to report
Information	332140	No Removal paths to report
Information	332140	No Minimum Pulse Width paths to report
Information	332102	Design is not fully constrained for setup requirements
Information	332102	Design is not fully constrained for hold requirements
Information		Quartus II 64-Bit TimeQuest Timing Analyzer was successful. 0 errors, 4 warnings
Information		Running Quartus II 64-Bit EDA Netlist Writer
Information		Command: quartus_eda --read_settings_files=off --write_settings_files=off PC -c PC
Information	204019	Generated file PC.vo in folder "/home/student1/mcoskun/Documents/coe608/lab2/PC/simula
Information		Quartus II 64-Bit EDA Netlist Writer was successful. 0 errors, 0 warnings
Information	293000	Quartus II Full Compilation was successful. 0 errors, 8 warnings

The status bar at the bottom indicates that the system is processing 118 items.

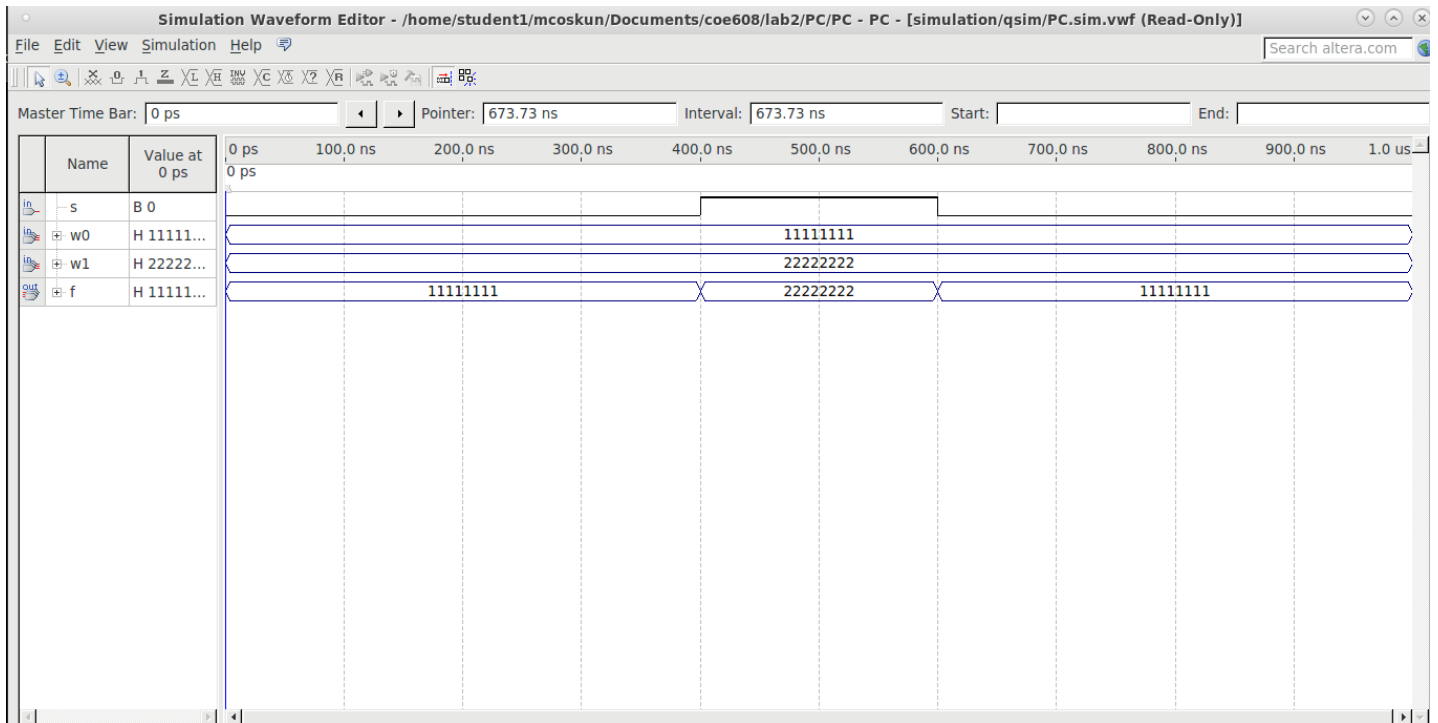
After compiling successfully, the waveform of add created following the lab manual instructions is displayed below.



The VHDL code of the multiplexer 2 to 1 (mux2to1) can be seen below.



After compiling successfully, the waveform of mux2to1 created following the lab manual instructions is displayed below.



The VHDL code of the program counter (pc) can be seen below.

First section:

The image shows a VHDL code editor window for a file named "add.vhd". The code is written in VHDL and defines the entity "pc" and its architecture "Behavior of pc". The code includes the following sections:

```

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 pc is
7
8     port (
9         d : in std_logic_vector(31 downto 0);
10        ld : in std_logic;
11        clr : in std_logic;
12        clk : in std_logic;
13        inc : in std_logic;
14        q : out std_logic_vector(31 downto 0)
15    );
16 end pc;
17
18 architecture Behavior of pc is
19     component add
20     port (
21         A : in std_logic_vector(31 downto 0);
22         B : out std_logic_vector(31 downto 0)
23     );
24 end component;
25     component mux2to1
26     port (
27         s : in std_logic;
28         w0, w1 : in std_logic_vector(31 downto 0);

```

Second section:

The screenshot displays the Quartus II IDE interface. The 'Project Navigator' on the left shows a file tree with 'pc.vhd' selected. The main editor window shows the VHDL code for 'add.vhd'. The code defines a component 'mux2to1', a component 'register32', and a top-level entity 'pc'. The 'pc' entity has three 32-bit output signals: 'add_out', 'mux_out', and 'q_out'. The 'begin' block contains three port maps: 'add0' for the 'add' component, 'mux0' for the 'mux2to1' component, and 'reg0' for the 'register32' component. The output 'q' is assigned the value of 'q_out'.

```

24   end component;
25   component mux2to1
26   port (
27       s : in std_logic;
28       w0, w1 : in std_logic_vector(31 downto 0);
29       f : out std_logic_vector(31 downto 0)
30   );
31   end component;
32   component register32
33   port (
34       d : in std_logic_vector(31 downto 0);
35       ld : in std_logic;
36       clr : in std_logic;
37       clk : in std_logic;
38       Q : out std_logic_vector(31 downto 0)
39   );
40   end component;
41   signal add_out : std_logic_vector(31 downto 0);
42   signal mux_out : std_logic_vector(31 downto 0);
43   signal q_out : std_logic_vector(31 downto 0);
44   begin
45   add0 : add port map (q_out, add_out);
46   mux0 : mux2to1 port map (inc, d, add_out, mux_out);
47   reg0 : register32 port map (mux_out, ld, clr, clk, q_out);
48
49   q <= q_out;
50
51   end Behavior;

```

The 'Messages' window at the bottom shows a successful compilation message: '293000 Quartus II Full Compilation was successful. 0 errors, 7 warnings'. The status bar at the bottom indicates 'System (3) / Processing (115)'.

After compiling successfully, the waveform of pc created following the lab manual instructions is displayed below.

