Ismail Akram

CS 343, FALL 2021

Laboratory Project 2

Introduction to VHDL, ModelSim and Quartus using Comparators

Ismail Akram

9/1/2021

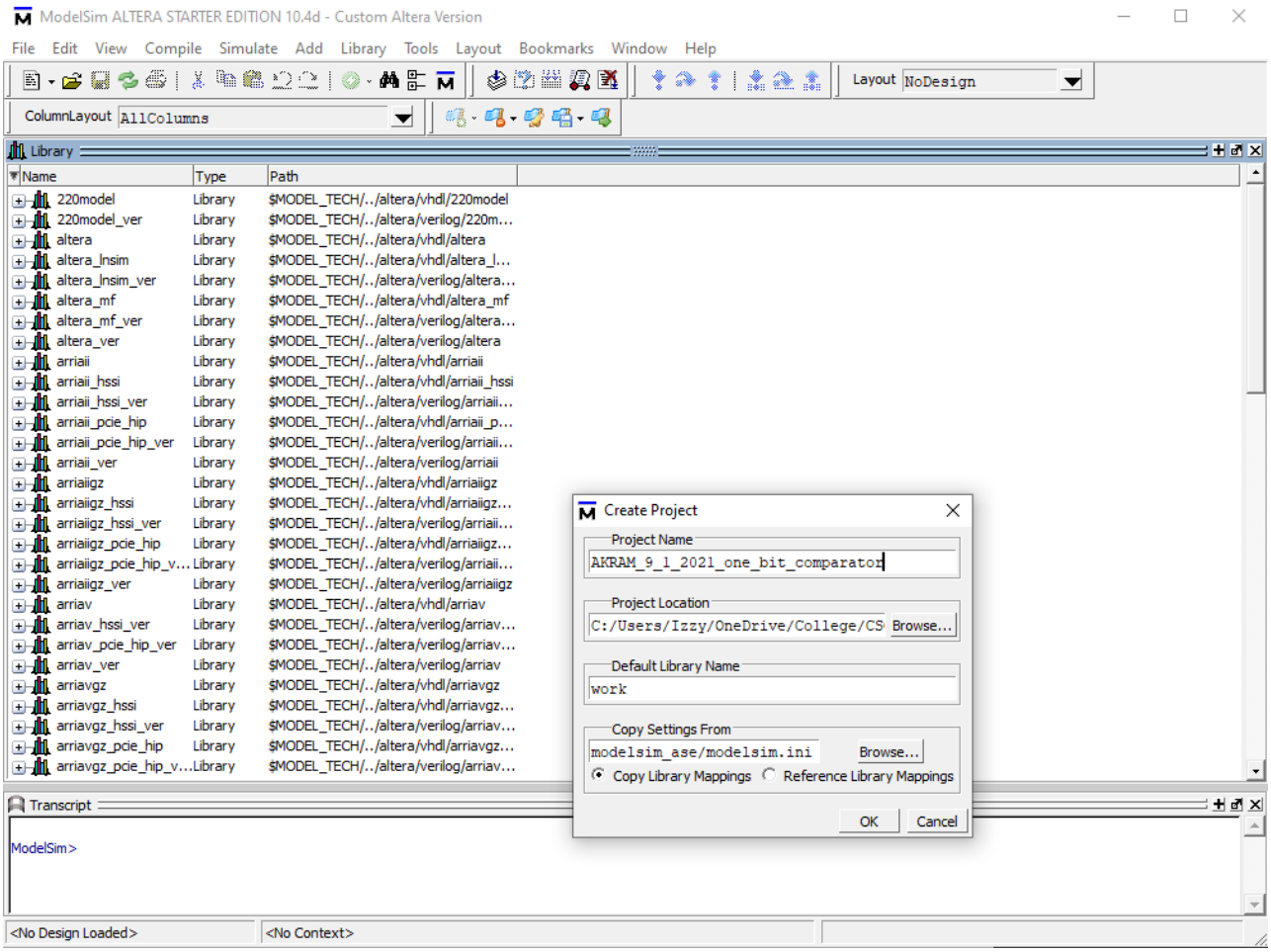


Fig 1. Creating New Project dialog in ModelSim

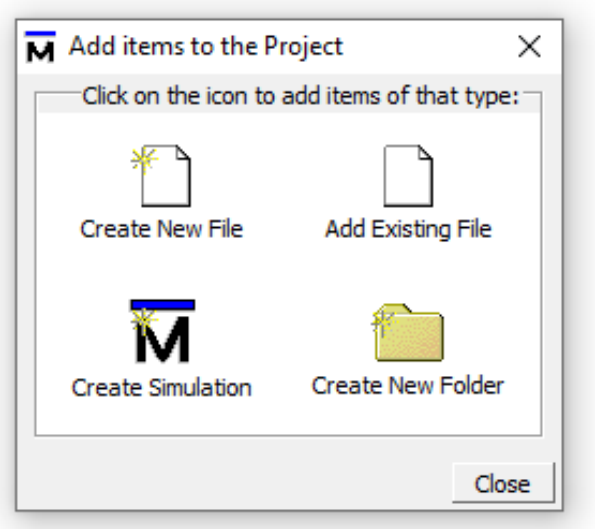


Fig 2. creating file

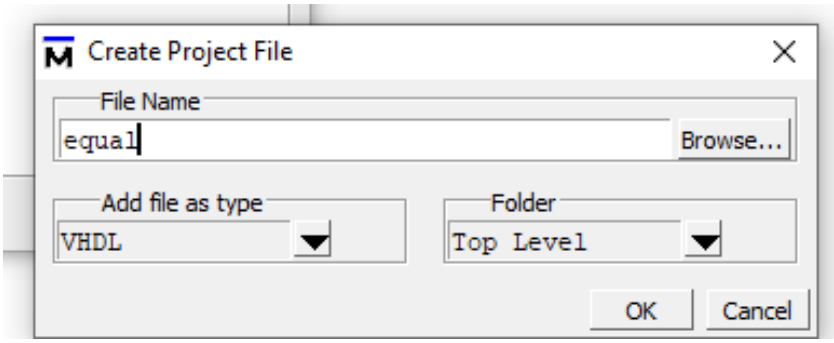


Fig 3. naming file as equal

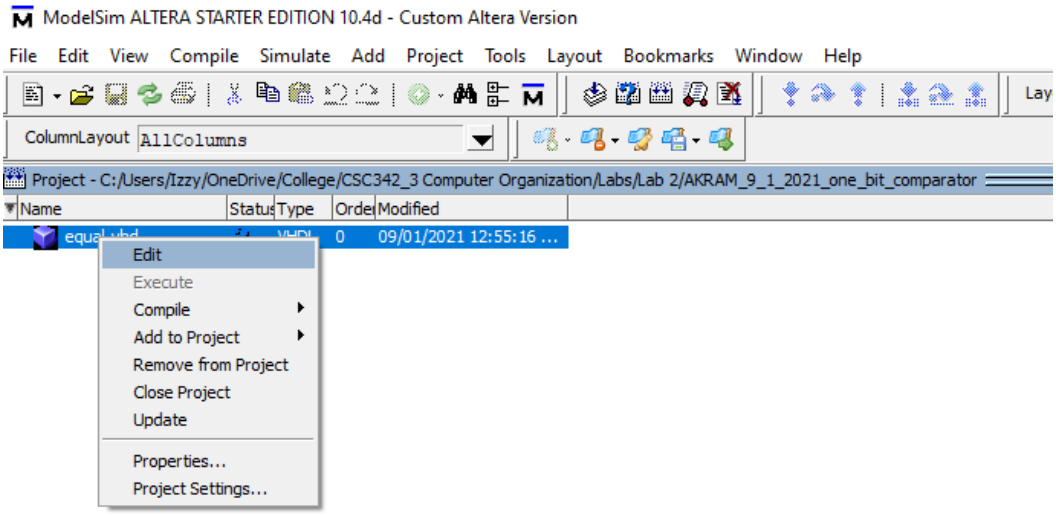


Fig 4. editing equal file

Ismail Akram

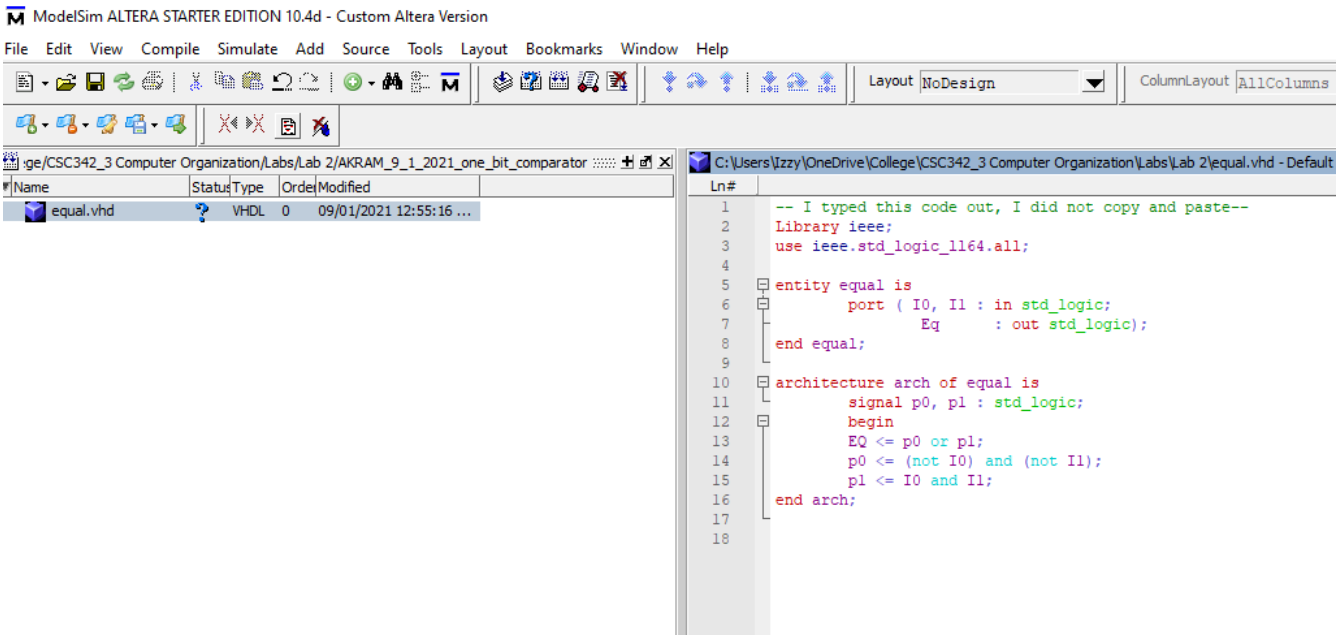


Fig 5. typing out code manually

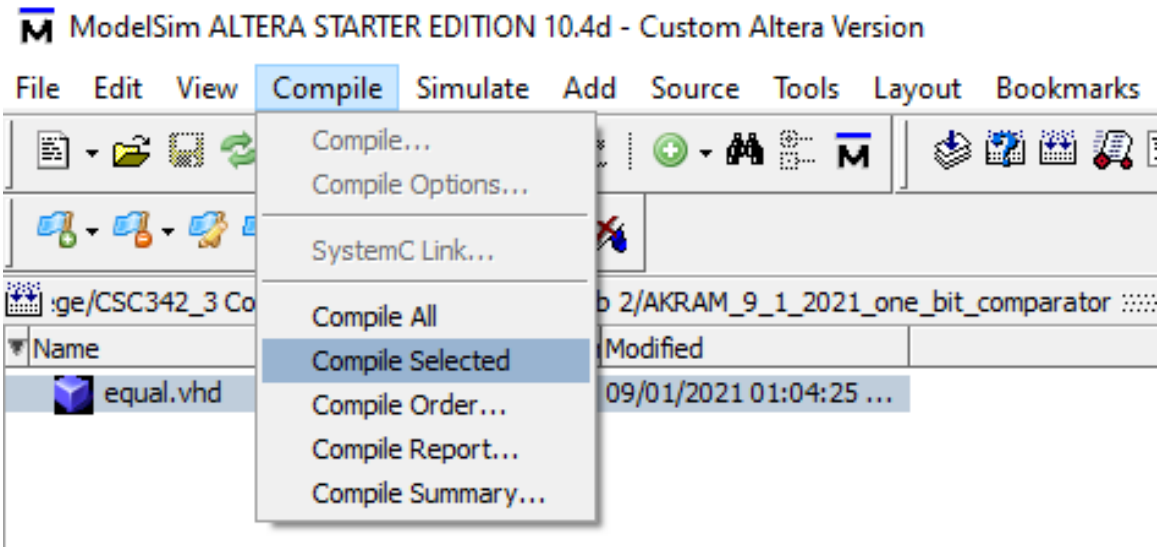


Fig 6. Compiling selected

Ismail Akram

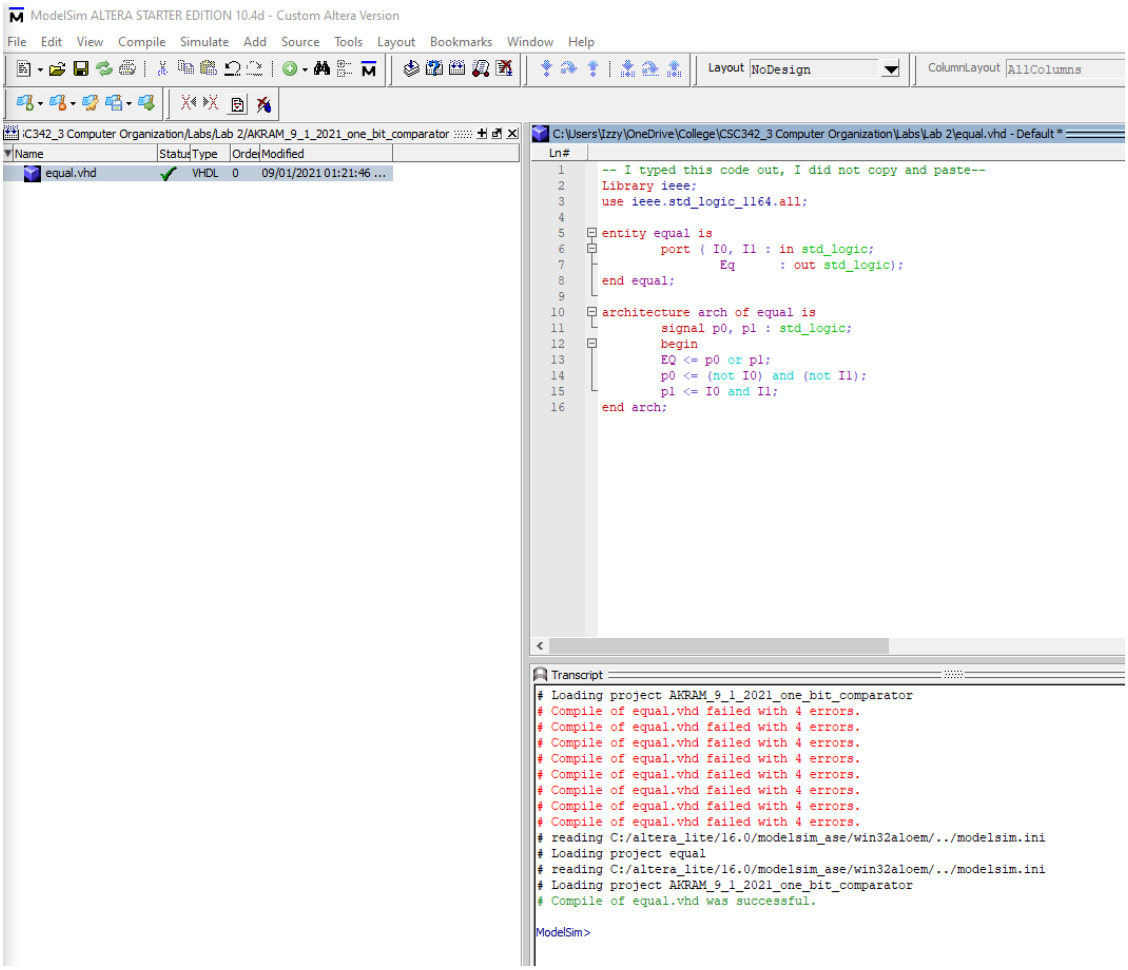


Fig 7. Compiled successfully (after several attempts).

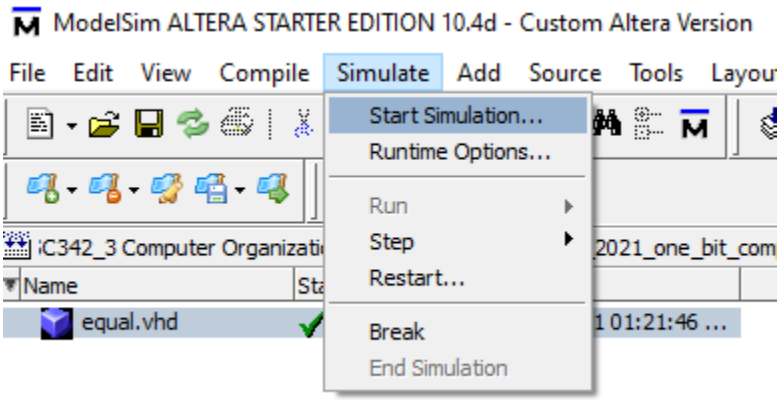


Fig 8. Simulating

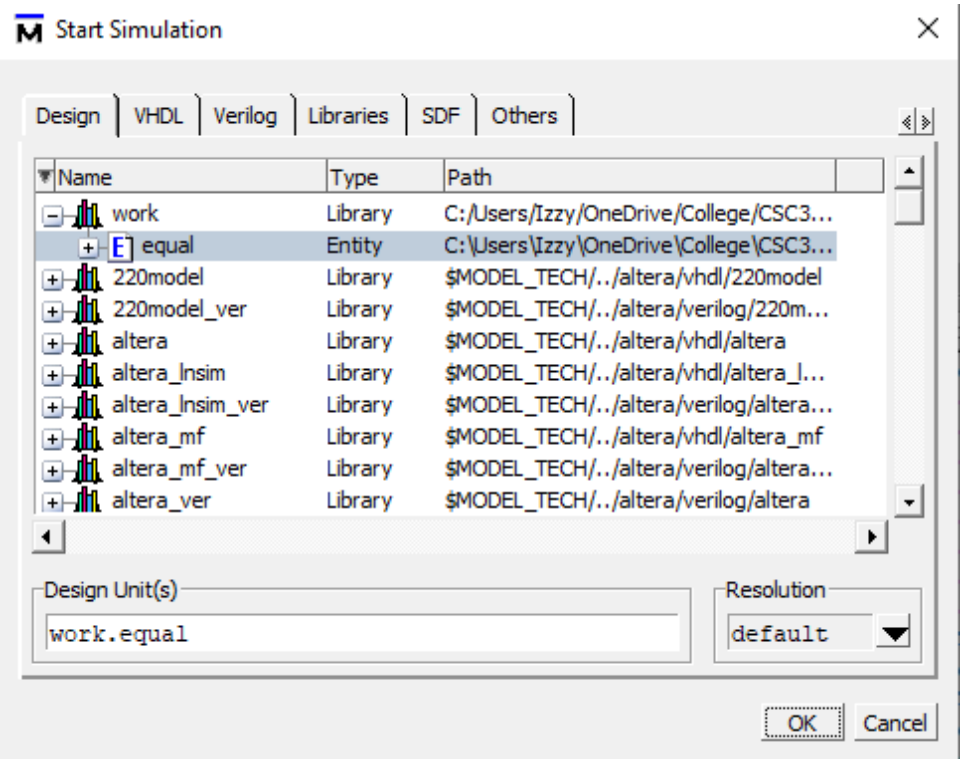


Fig 9. Choosing equal entity for simulation.

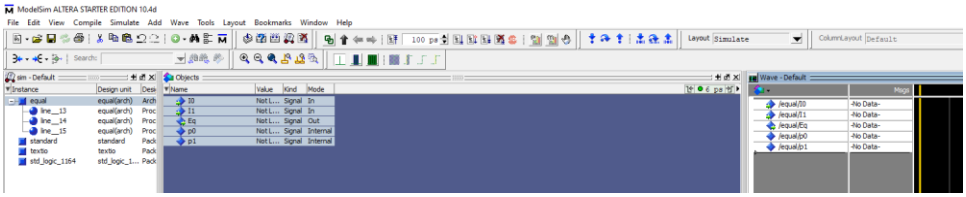


Fig 10. Dragging inputs and outputs to waveform panel

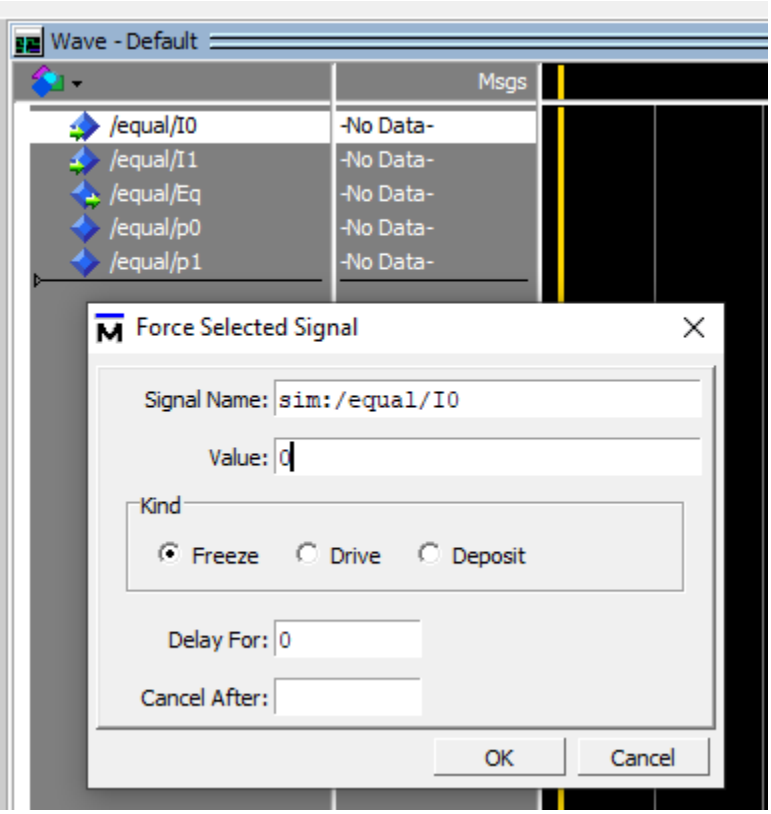


Fig 11. Forcing value input for i0

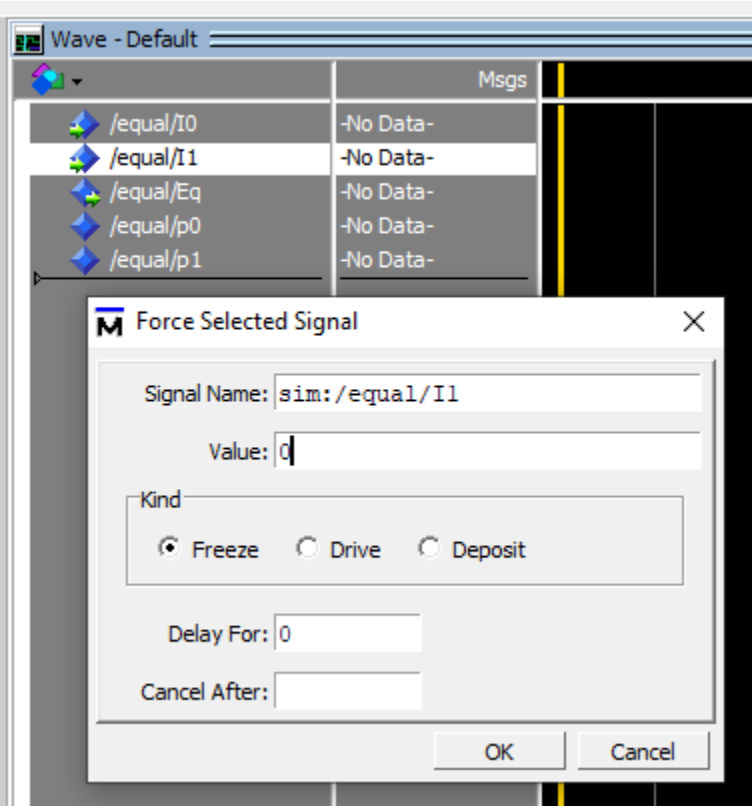


Fig 12. Forcing value input for i1

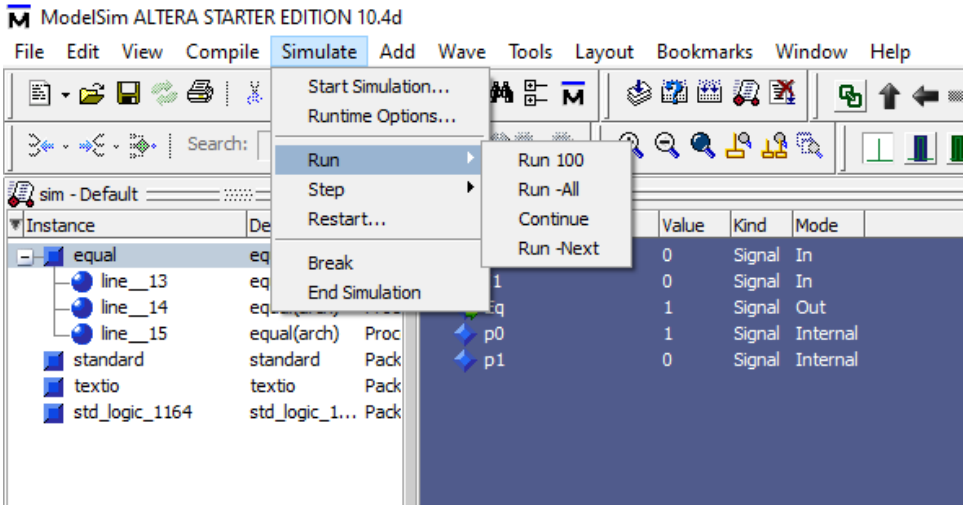


Fig 13. Simulation > Run > Run 100

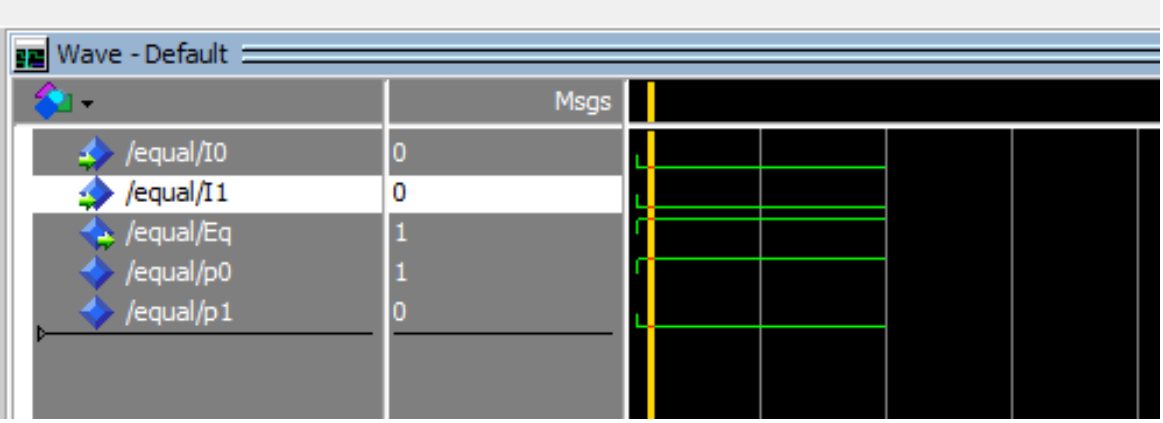


Fig 14. Result of first run (Forced values i0=0, i1=0). Eq=1 since the values of the inputs are equal.

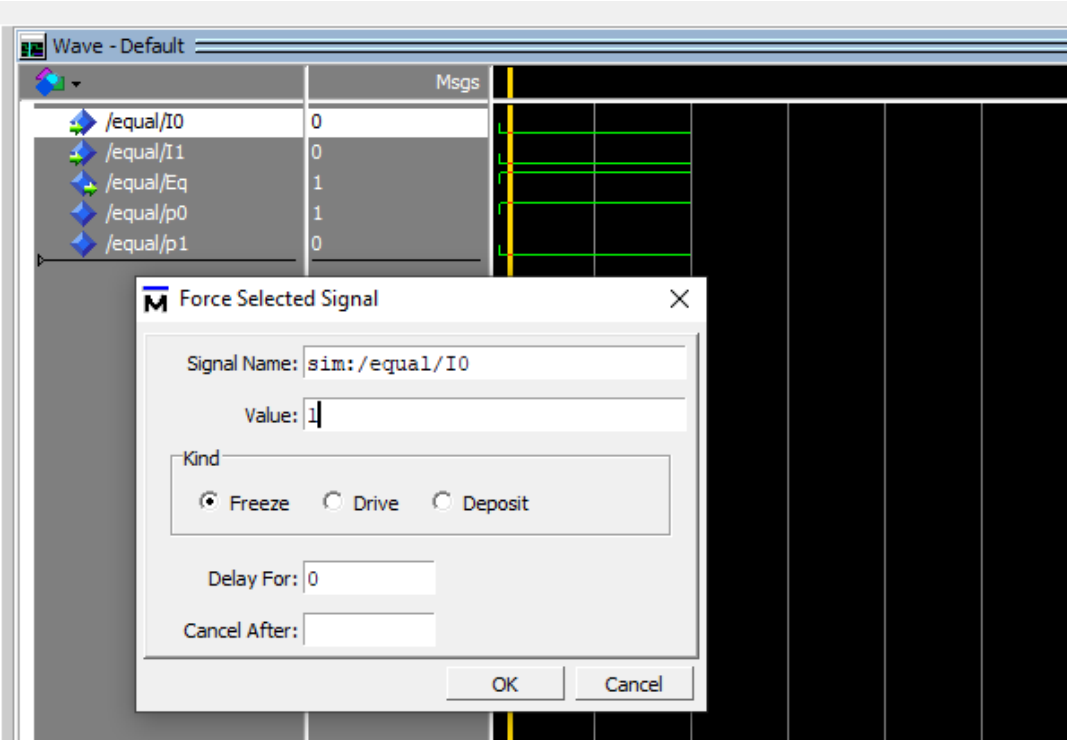


Fig 15. Changing i0 to 1

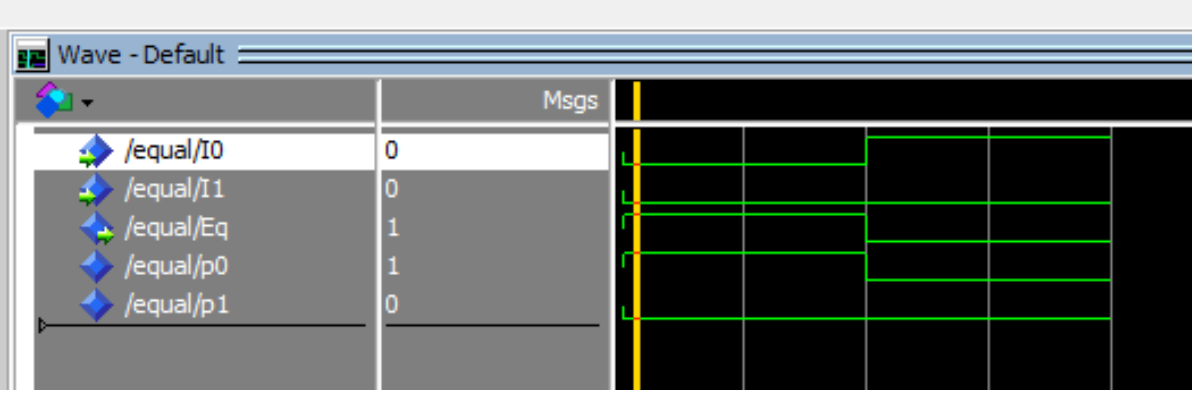


Fig 16. Second Run, waveform result

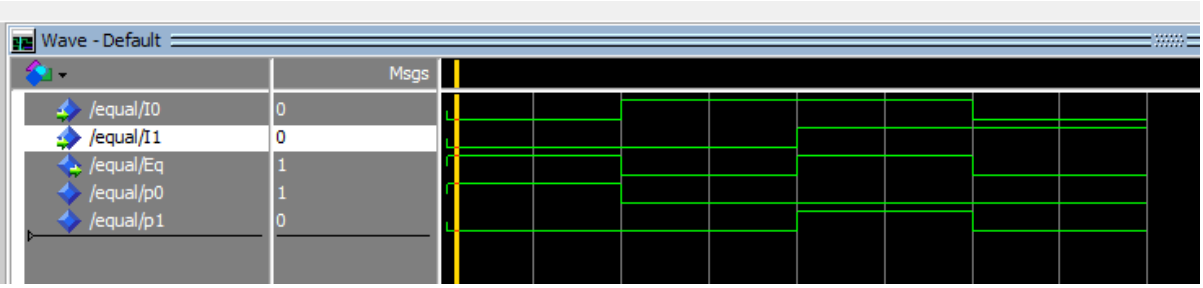
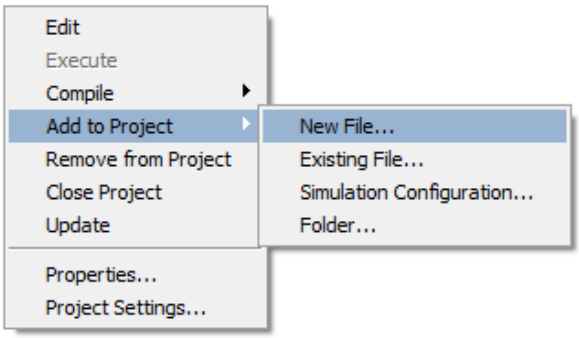


Fig 17. Exploring every combination of 1 and 0 for i0 and i1 (00, 10, 11, 01 respectively) and then Ended Simulation (Simulation > End Simulation).

Test Bench



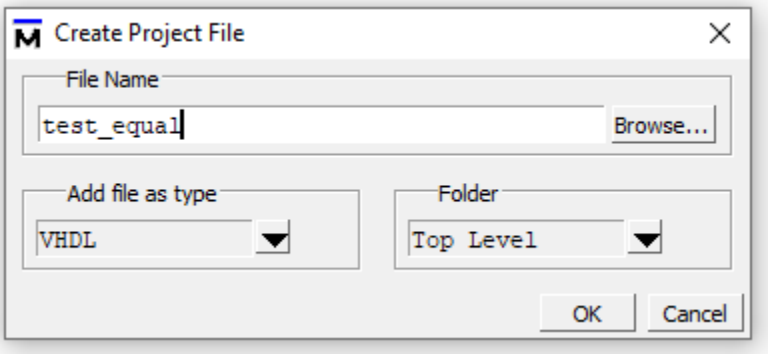
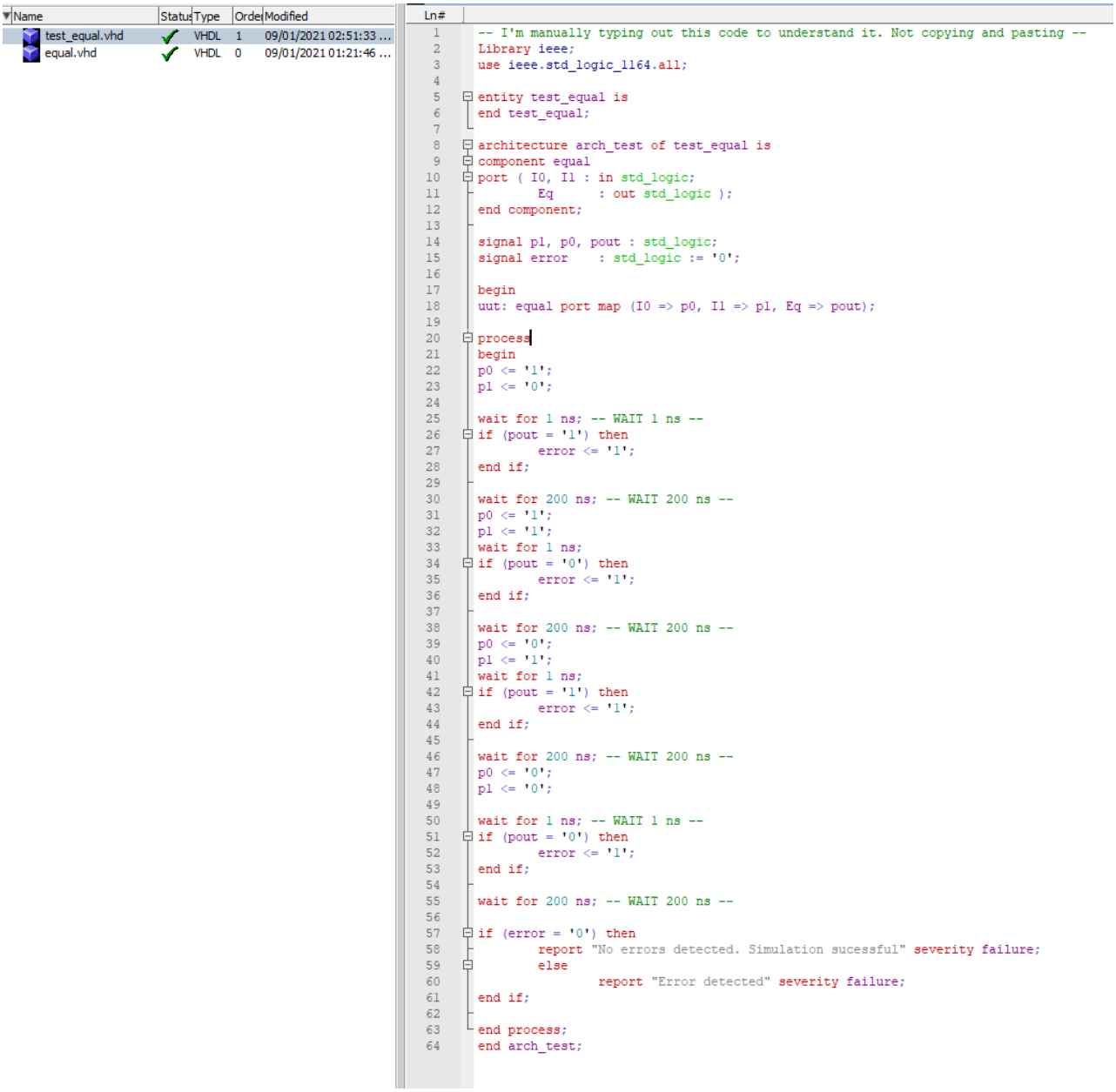


Fig 17. Creating a new VHDL file



Name	Status	Type	Order	Modified
test_equal.vhd	✓	VHDL	1	09/01/2021 02:51:33 ...
equal.vhd	✓	VHDL	0	09/01/2021 01:21:46 ...

```
Ln#
 1  -- I'm manually typing out this code to understand it. Not copying and pasting --
 2  Library ieee;
 3  use ieee.std_logic_1164.all;
 4
 5  entity test_equal is
 6  end test_equal;
 7
 8  architecture arch_test of test_equal is
 9  component equal
10  port ( I0, I1 : in std_logic;
11         Eq     : out std_logic );
12  end component;
13
14  signal p1, p0, pout : std_logic;
15  signal error    : std_logic := '0';
16
17  begin
18  uut: equal port map (I0 => p0, I1 => p1, Eq => pout);
19
20  process
21  begin
22  p0 <= '1';
23  p1 <= '0';
24
25  wait for 1 ns; -- WAIT 1 ns --
26  if (pout = '1') then
27         error <= '1';
28  end if;
29
30  wait for 200 ns; -- WAIT 200 ns --
31  p0 <= '1';
32  p1 <= '1';
33  wait for 1 ns;
34  if (pout = '0') then
35         error <= '1';
36  end if;
37
38  wait for 200 ns; -- WAIT 200 ns --
39  p0 <= '0';
40  p1 <= '1';
41  wait for 1 ns;
42  if (pout = '1') then
43         error <= '1';
44  end if;
45
46  wait for 200 ns; -- WAIT 200 ns --
47  p0 <= '0';
48  p1 <= '0';
49
50  wait for 1 ns; -- WAIT 1 ns --
51  if (pout = '0') then
52         error <= '1';
53  end if;
54
55  wait for 200 ns; -- WAIT 200 ns --
56
57  if (error = '0') then
58         report "No errors detected. Simulation sucessful" severity failure;
59         else
60                 report "Error detected" severity failure;
61  end if;
62
63  end process;
64  end arch_test;
```

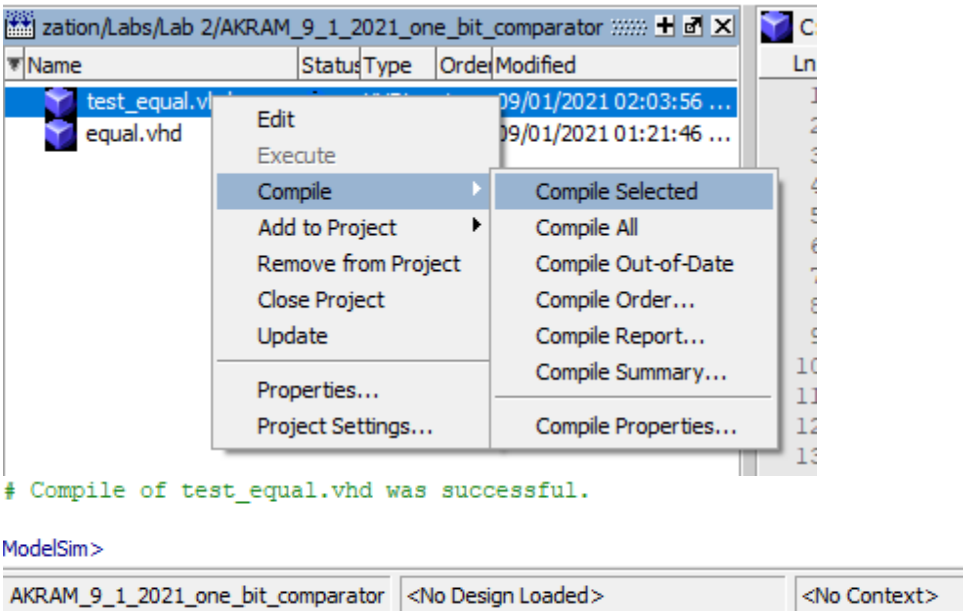


Fig 18. Compilation successful for test_equal.vhd

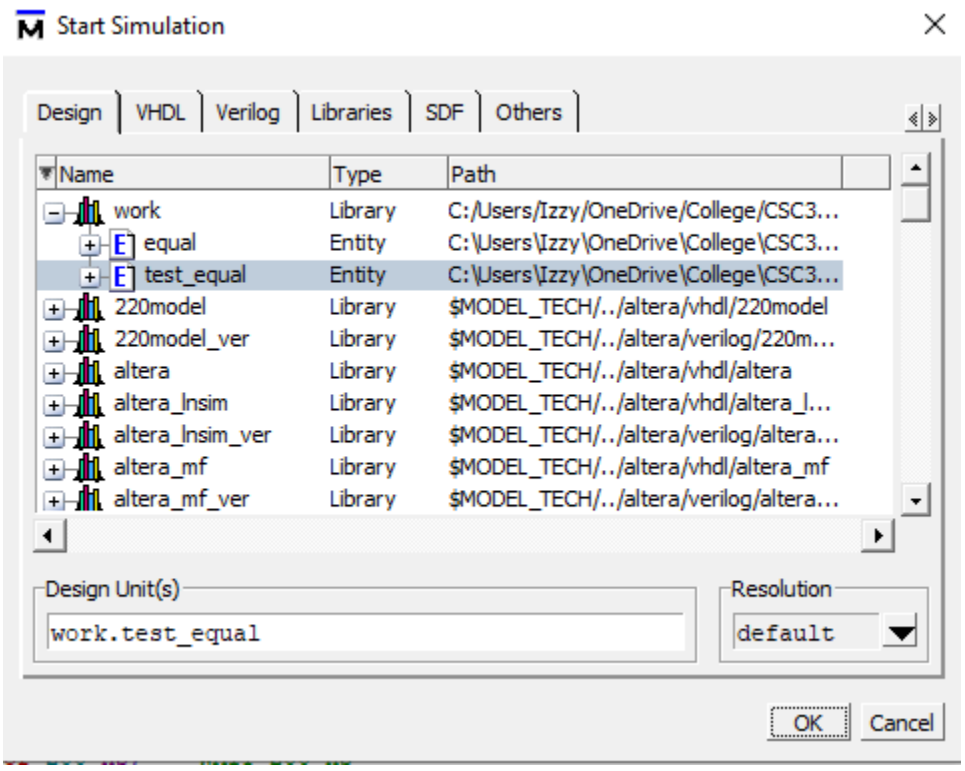


Fig 19. Starting Simulation

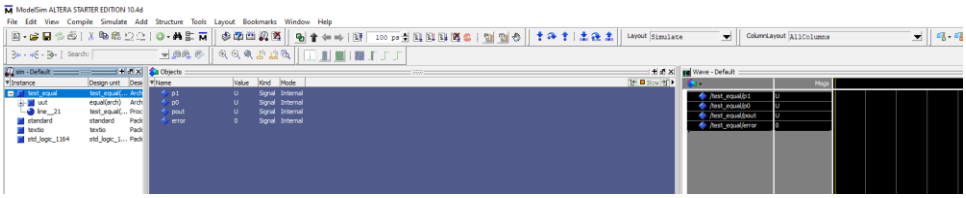


Fig 20. Added wave to simulation environment of test_equal.vhd

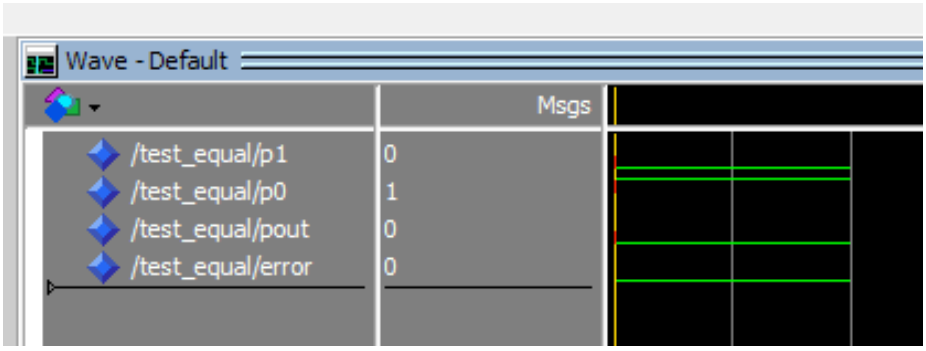


Fig 21. Running the simulation for test_equal.vhd

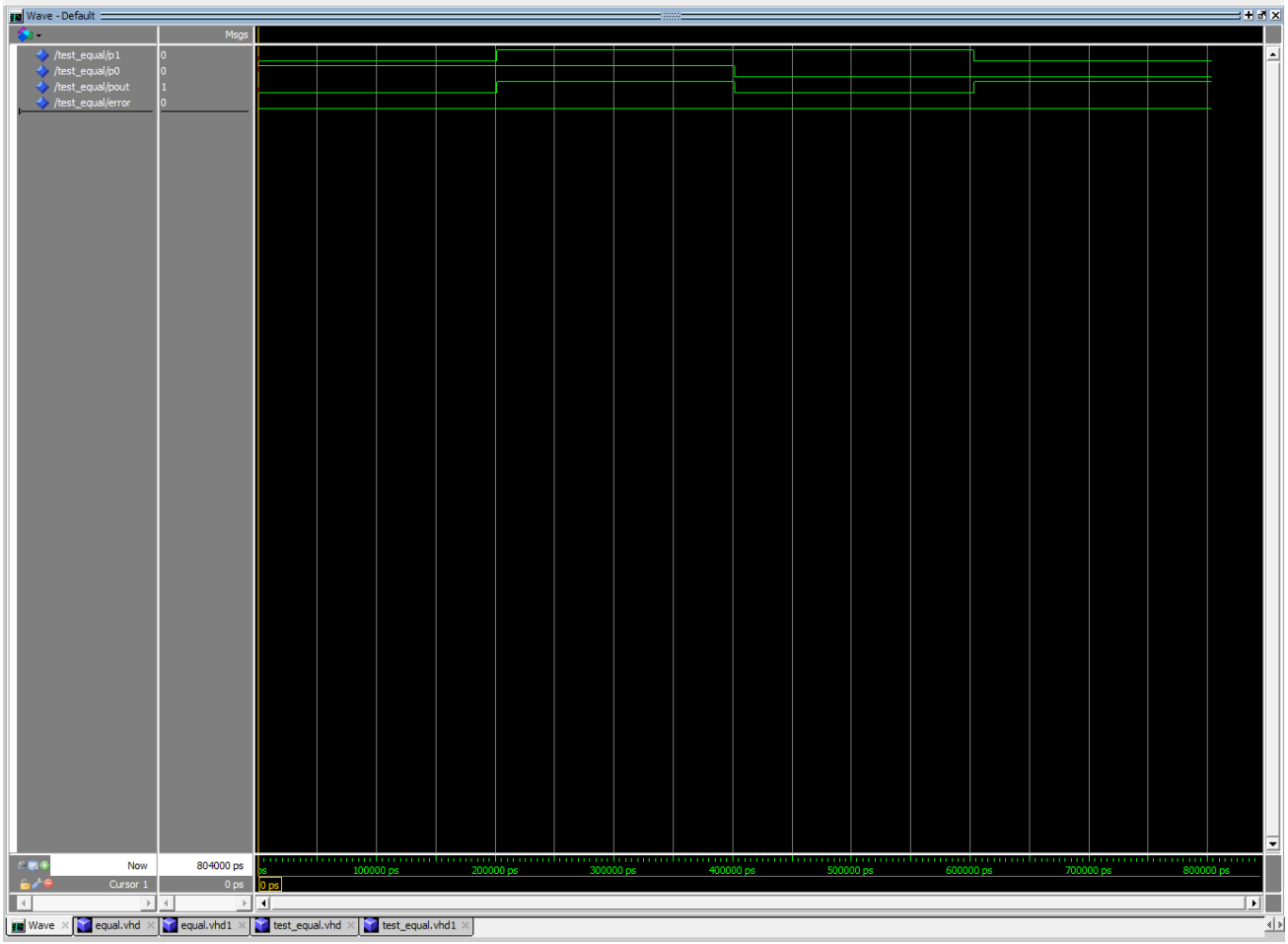


Fig 22. Ran for all combinations for i0 and i1 using Run-all and Full Zoom to see everything.

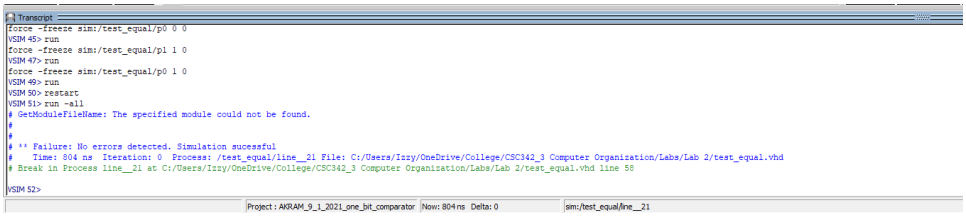


Fig 23. Transcript simulation successful.

One bit comparator assignment complete!

I'm repeating the same steps for 2-bit and 8-bit comparator.

2-Bit comparator

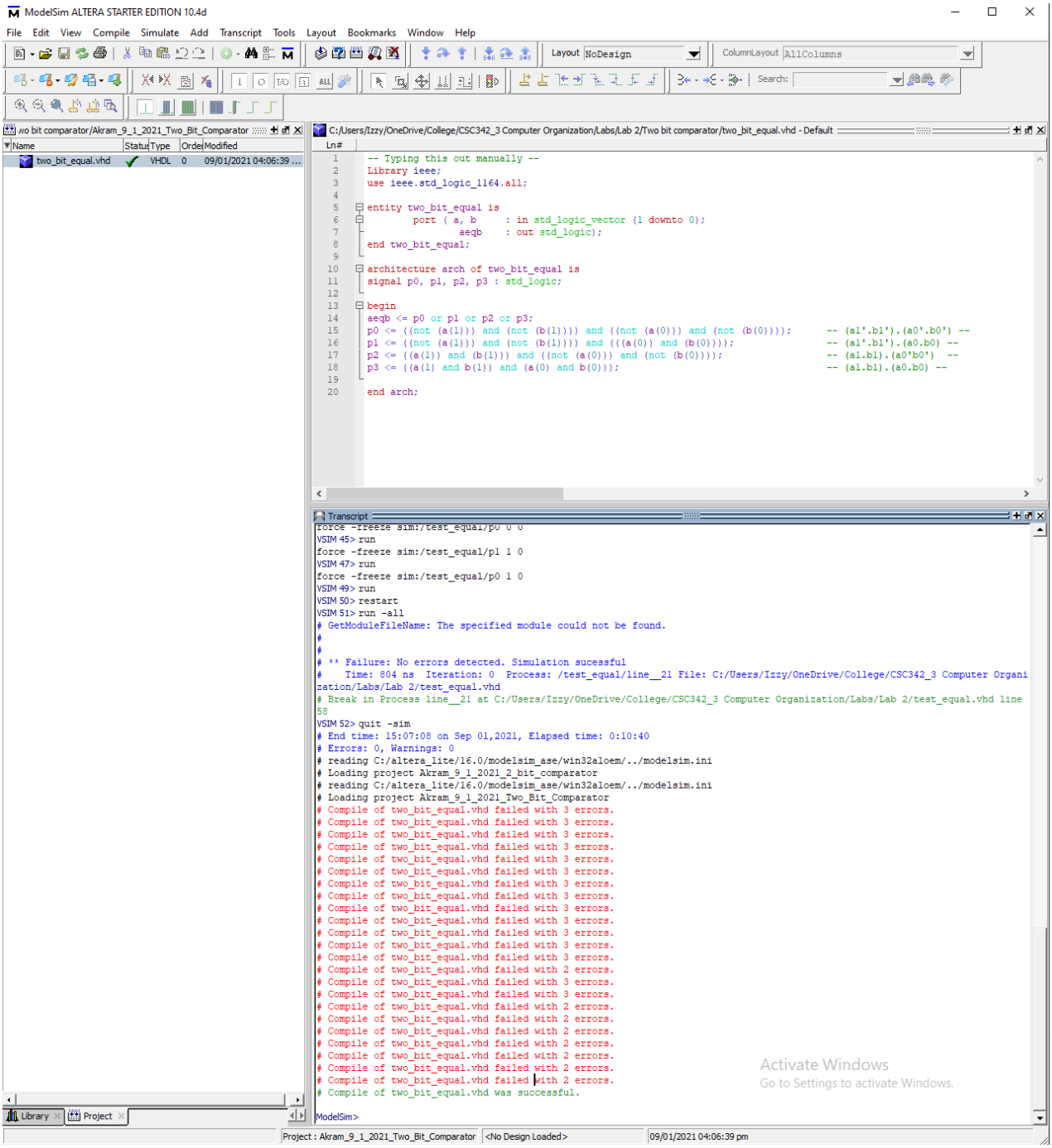


Fig 24. Logic Vector for 2-bit comparator compiled succesfully

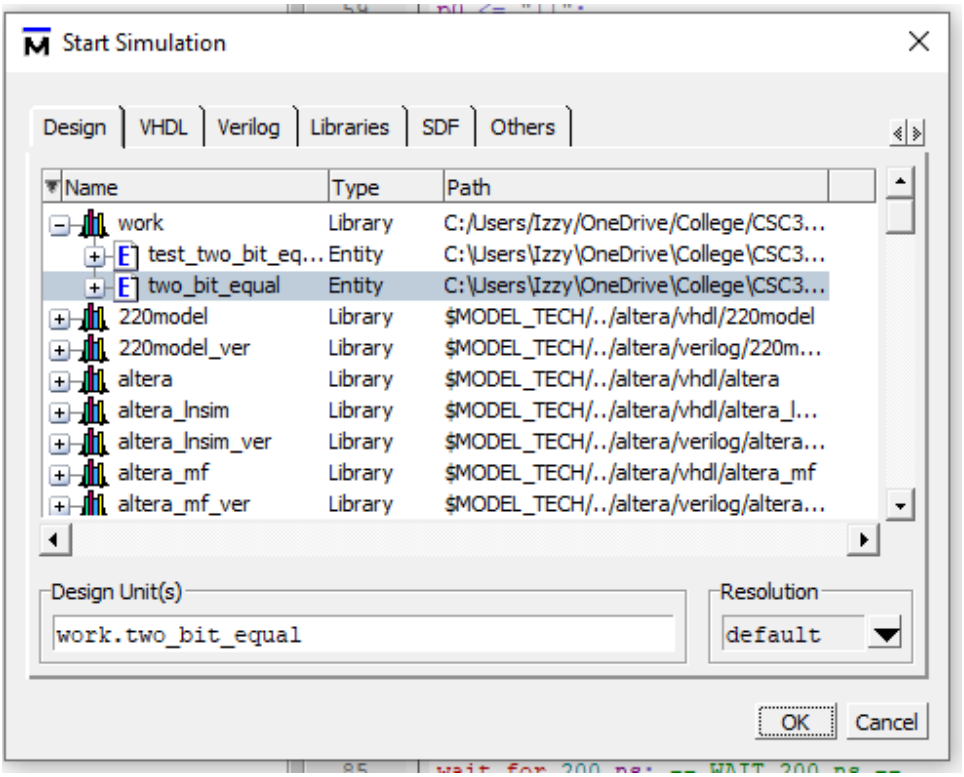


Fig 25. Running Simulation for two_bit_equal.vhd

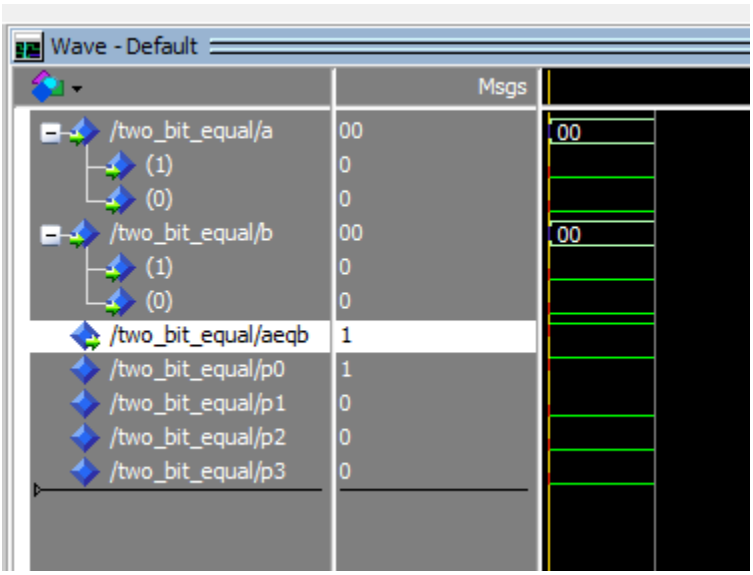


Fig 26. Forcing 0s to all inputs for the first run

Ismail Akram

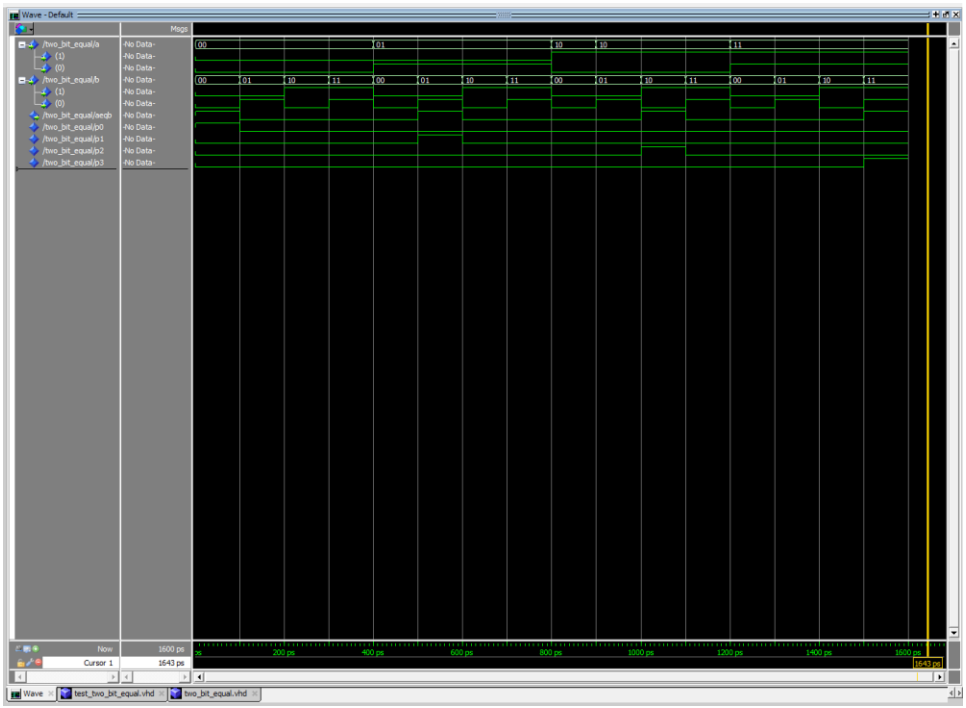


Fig 27. Run for every combination of inputs for 2-bits for two_bit_comparator (manually, run-all wouldn't work). Simulation successful.

Ismail Akram

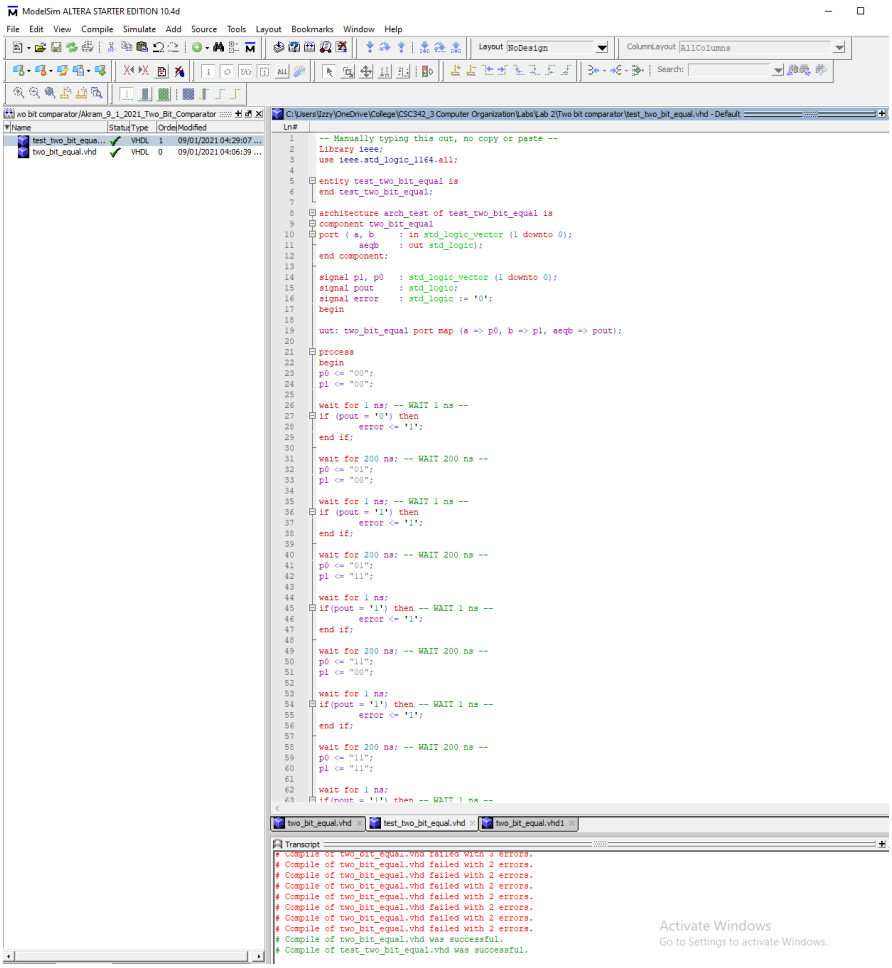


Fig 28. Compile successful for test_two_bit_equal.vhd

Ismail Akram

Full code:

C:\Users\Izzy\OneDrive\College\CSC342_3 Computer Organization\Labs\Lab 2\Two bit comparator\test_two_bit_equal.vhd - Default

```
Ln#
 1  -- Manually typing this out, no copy or paste --
 2  Library ieee;
 3  use ieee.std_logic_1164.all;
 4
 5  entity test_two_bit_equal is
 6  end test_two_bit_equal;
 7
 8  architecture arch_test of test_two_bit_equal is
 9  component two_bit_equal
10  port ( a, b     : in std_logic_vector (1 downto 0);
11         aeqb     : out std_logic);
12  end component;
13
14  signal p1, p0   : std_logic_vector (1 downto 0);
15  signal pout     : std_logic;
16  signal error    : std_logic := '0';
17  begin
18
19  uut: two_bit_equal port map (a => p0, b => p1, aeqb => pout);
20
21  process
22  begin
23  p0 <= "00";
24  p1 <= "00";
25
26  wait for 1 ns; -- WAIT 1 ns --
27  if (pout = '0') then
28         error <= '1';
29  end if;
30
31  wait for 200 ns; -- WAIT 200 ns --
32  p0 <= "01";
33  p1 <= "00";
34
35  wait for 1 ns; -- WAIT 1 ns --
36  if (pout = '1') then
37         error <= '1';
38  end if;
39
40  wait for 200 ns; -- WAIT 200 ns --
41  p0 <= "01";
42  p1 <= "11";
43
44  wait for 1 ns;
45  if(pout = '1') then -- WAIT 1 ns --
46         error <= '1';
47  end if;
48
49  wait for 200 ns; -- WAIT 200 ns --
50  p0 <= "11";
51  p1 <= "00";
52
53  wait for 1 ns;
54  if(pout = '1') then -- WAIT 1 ns --
55         error <= '1';
56  end if;
57
58  wait for 200 ns; -- WAIT 200 ns --
59  p0 <= "11";
60  p1 <= "11";
61
```

```
62    wait for 1 ns;
63    if(pout = '1') then -- WAIT 1 ns --
64            error <= '1';
65    end if;
66
67    wait for 200 ns; -- WAIT 200 ns --
68    p0 <= "10";
69    p1 <= "11";
70
71    wait for 1 ns;
72    if(pout = '1') then -- WAIT 1 ns --
73            error <= '1';
74    end if;
75
76    wait for 200 ns; -- WAIT 200 ns --
77    p0 <= "10";
78    p1 <= "10";
79
80    wait for 1 ns;
81    if(pout = '1') then -- WAIT 1 ns --
82            error <= '1';
83    end if;
84
85    wait for 200 ns; -- WAIT 200 ns --
86    p0 <= "11";
87    p1 <= "01";
88
89    wait for 1 ns;
90    if(pout = '1') then -- WAIT 1 ns --
91            error <= '1';
92    end if;
93
94    wait for 200 ns; -- WAIT 200 ns --
95
96    if (error = '0') then
97            report "No errors detected. Simulation successful" severity failure;
98            else
99            report "Error detected" severity failure;
100   end if;
101   end process;
102   end arch_test;
```

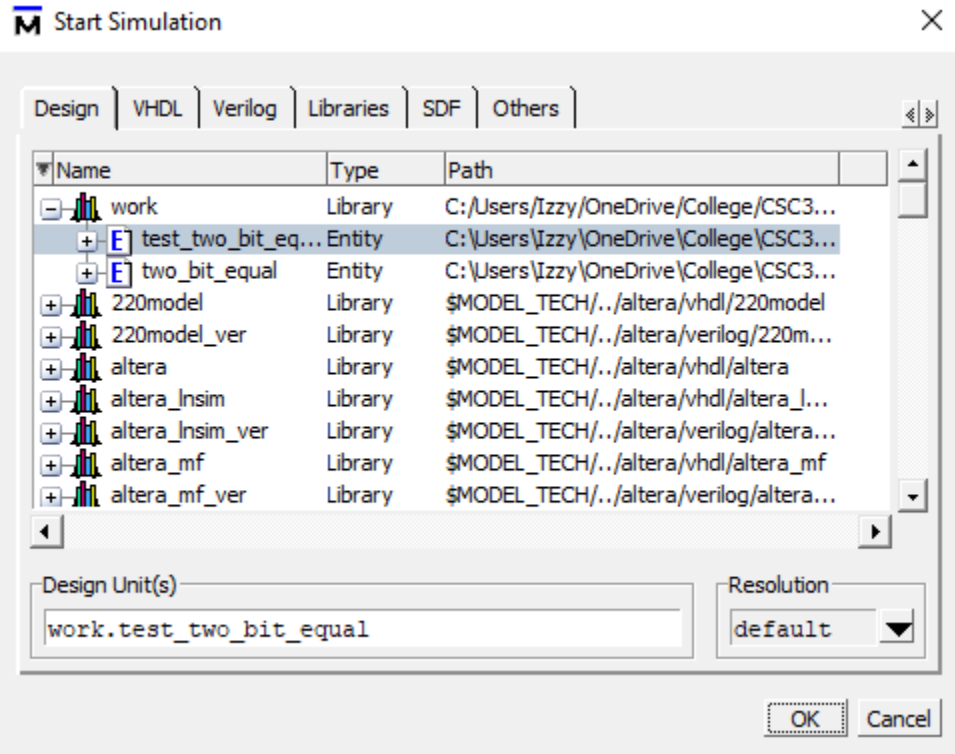


Fig 29. Running simulation for test_two_bit_comparator.vhd

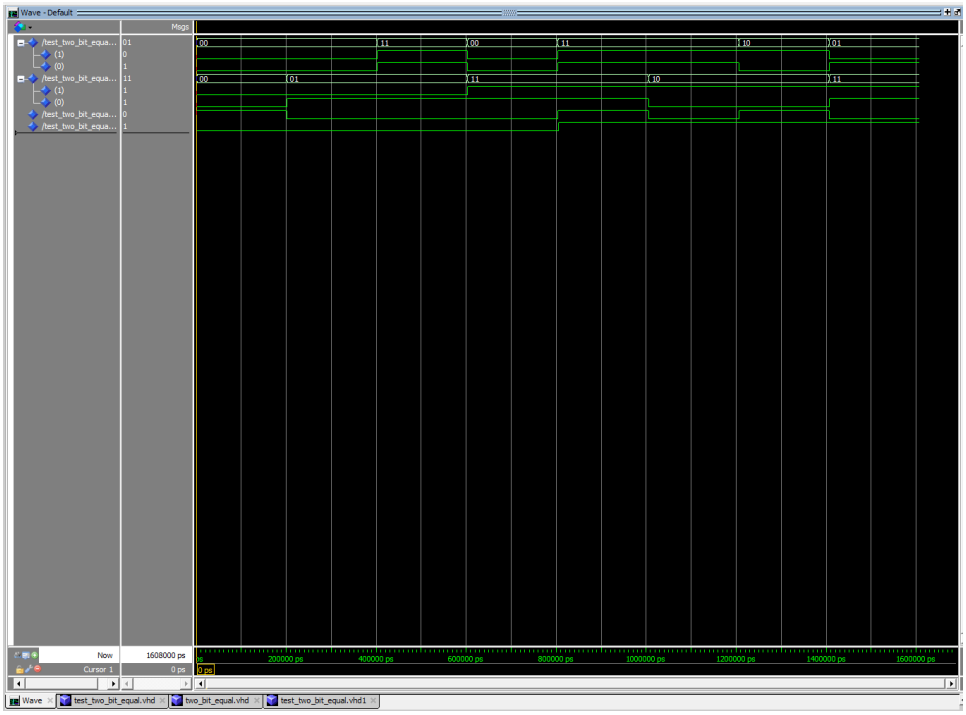


Fig 30. Running simulation for test_two_bit_comparator.vhd using run-all (this time it worked)

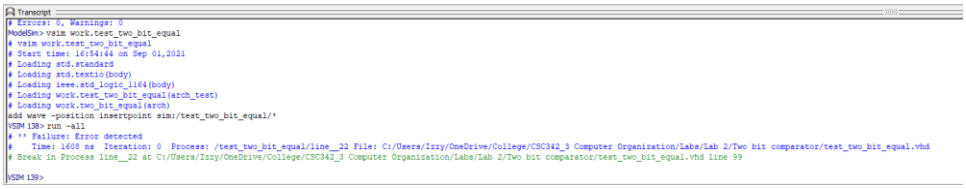


Fig 31. Transcript

PORT MAPS

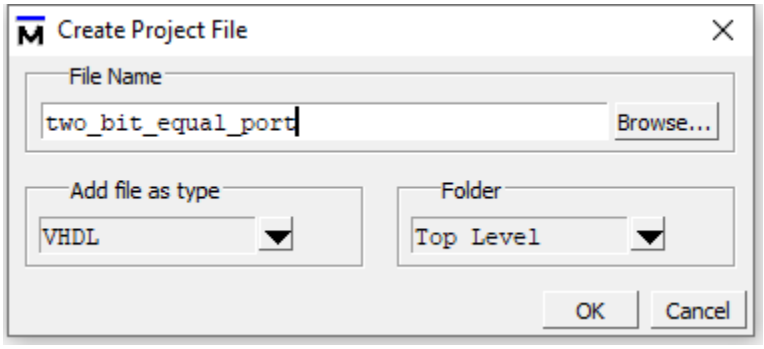


Fig 32. Creating a new file for two_bit_equal_port.vhd

Ismail Akram

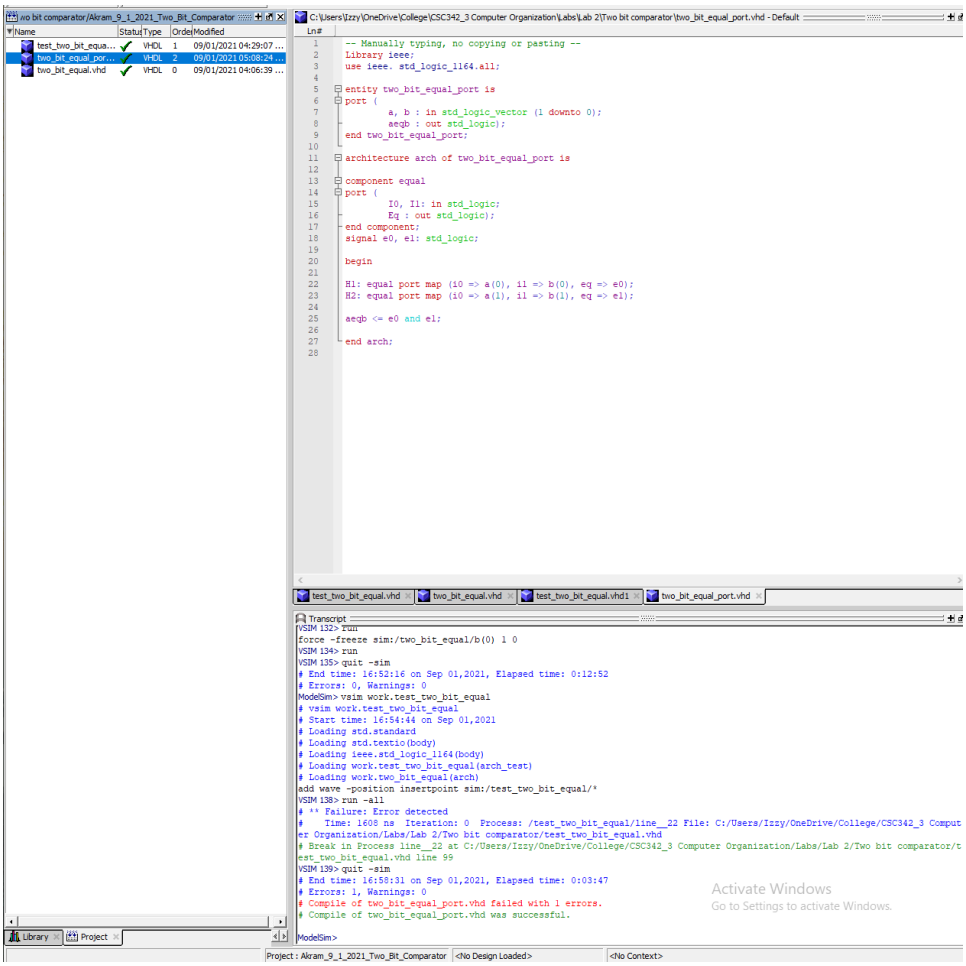


Fig 32. Compiled two_bit_equal_port.vhd

Ismail Akram

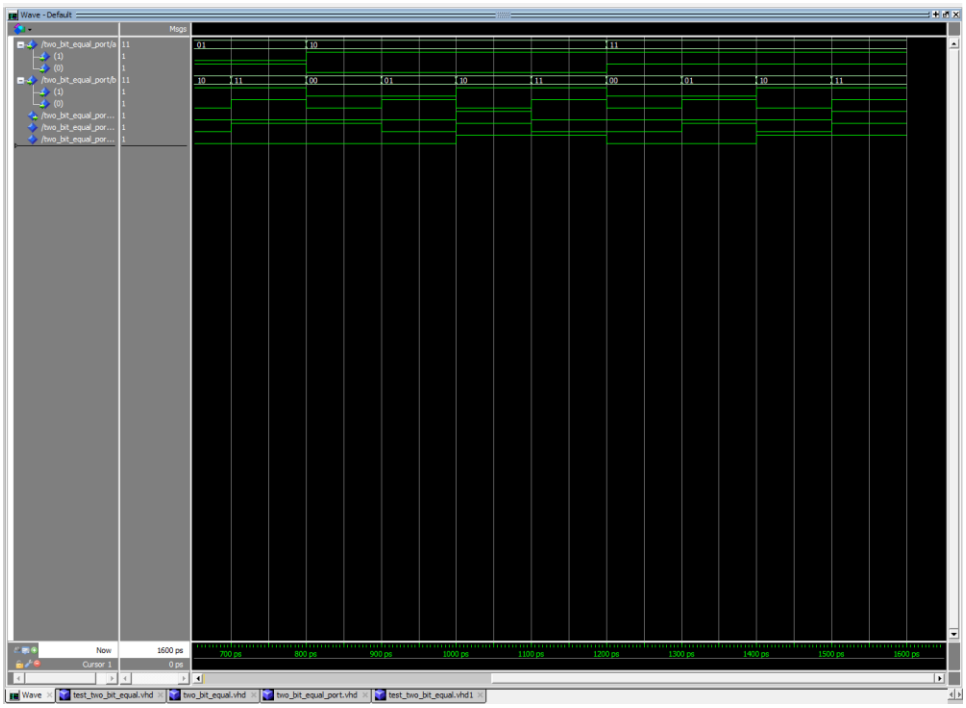


Fig 33. Ran Simulation for test_bit_equal_port.vhd for every 2-bit combination (run-all didn't work so I manually input every combination of 1s and 0s).