

Objectives:

In this lab we learned the basic skeleton of VHDL. We would learn how to design one-bit and two-bit comparators and learn to simulate our design in Model-SIM using test files.

Specifications:

One-bit comparator: two input ports, two signals and one output port.

Two-bit comparator: two input ports, four signals and one output port.

Two-bit comparator with port maps: two input ports, two signals and one output port.

Eight-bit comparator: two input ports, eight signals and one output port.

Functionality:

One-bit comparator takes two inputs and compare them against each other, if they are equal then the output would be 1 otherwise 0.

Two-bit comparator checks for equality in a similar way as one-bit comparator. However, two-bit comparator uses a logical statement given below:

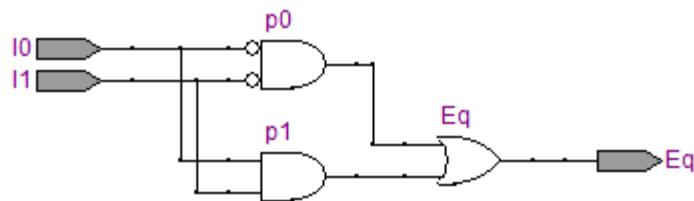
$$aeqb = (a1'.b1').(a0'.b0') + (a1'.b1).(a0.b0) + (a1.b1).(a0'b0') + (a1.b1).(a0.b0)$$

Eight-bit comparator works using port mapping in the same way as the two-bit comparator, but eight-bit comparator uses eight one-bit comparator.

Design:

One-bit comparator:

```
1  Library ieee;
2  Use ieee.std_logic_1164.all;
3
4  Entity equal is
5    Port (  I0, I1: in std_logic;
6              Eq : out std_logic);
7  End equal;
8
9  Architecture arch of equal is
10 Signal p0, p1 : std_logic;
11 begin
12   Eq <= p0 or p1;
13   P0 <= (not I0) and (not I1);
14   P1<= i0 and i1;
15 End arch;
```

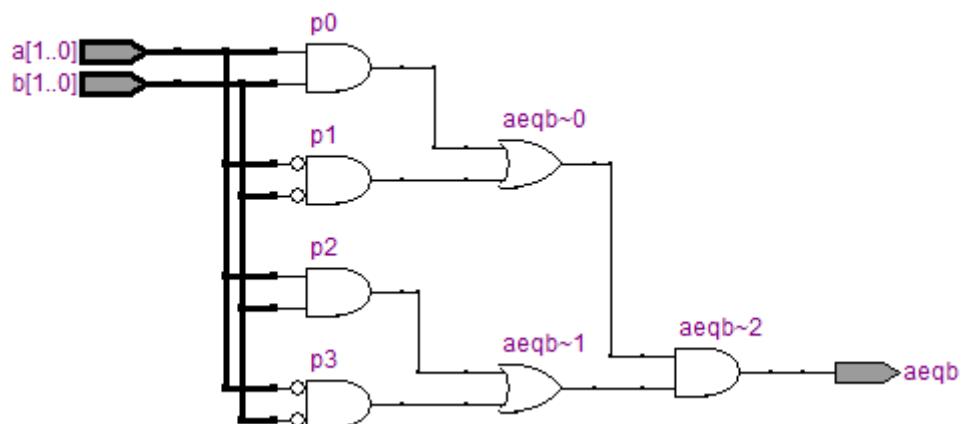


Two-bit comparator:

```

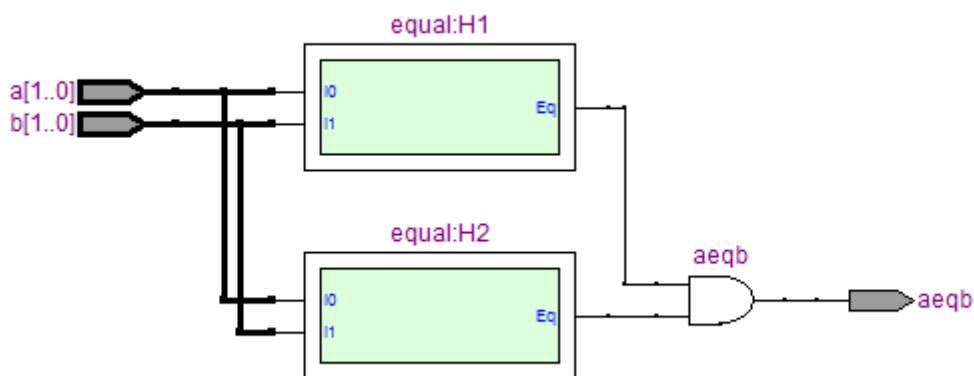
1  Library ieee;
2  Use ieee.std_logic_1164.all;
3
4  Entity two_bit_equal is
5    Port (  a, b: in std_logic_vector(1 downto 0);
6            aeqb : out std_logic);
7  End two_bit_equal;
8
9  Architecture arch of two_bit_equal is
10 Signal p0, p1,p2,p3 : std_logic;
11 begin
12   aeqb <=(p0 or p1) and (p2 or p3);
13   P0 <=a(0) and b(0);
14   P1<=(not a(0)) and (not b(0));
15   P2<=a(1) and b(1);
16   P3 <=(not a(1)) and (not b(1));
17 End arch;

```



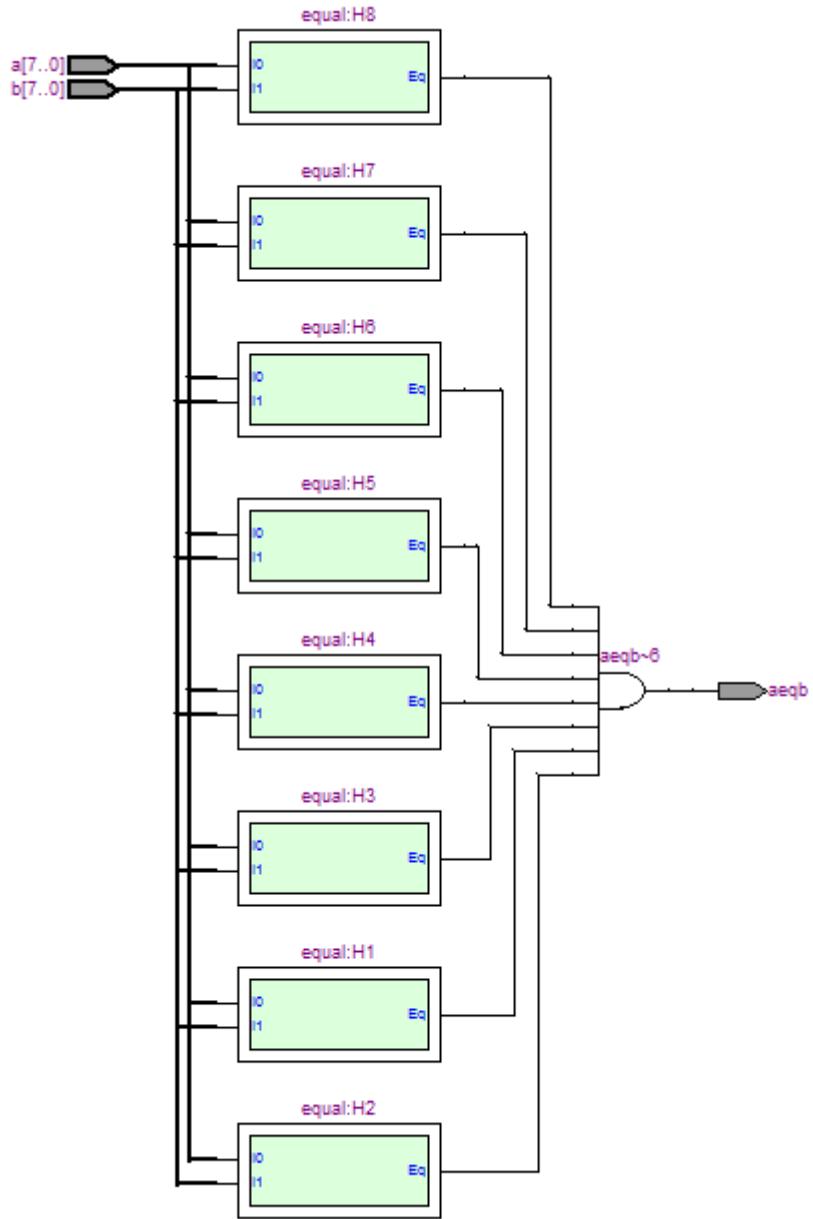
Two-bit comparator with port maps:

```
1  Library ieee;
2  Use ieee.std_logic_1164.all;
3
4  Entity two_bit_equal_port is
5  Port (
6    a, b: in std_logic_vector(1 downto 0); aeqb : out std_logic);
7  End two_bit_equal_port;
8
9  Architecture arch of two_bit_equal_port is
10
11  Component equal
12  Port (
13    i0, i1: in std_logic;
14    eq : out std_logic); End component;
15
16  signal e0,e1: std_logic;
17  begin
18
19  H1: equal
20  port map(i0=>a(0), i1=>b(0), eq=>e0);
21  H2: equal
22  port map(i0=>a(1), i1=>b(1), eq=>e1);
23
24  aeqb <= e0 and e1;
25  end arch;
```



Eight-bit comparator:

```
1  Library ieee;
2  Use ieee.std_logic_1164.all;
3
4  Entity eight_bit_equal_port is
5  Port (
6    a, b: in std_logic_vector(7 downto 0);
7    aeqb : out std_logic);
8  End eight_bit_equal_port;
9
10 Architecture arch of eight_bit_equal_port is
11
12 component equal Port (
13
14   I0, I1: in std_logic;
15   Eq : out std_logic); End component;
16
17 signal e0,e1,e2,e3,e4,e5,e6,e7: std_logic;
18 begin
19
20 H1: equal
21 port map(i0=>a(0), i1=>b(0), eq=>e0); H2: equal
22 port map(i0=>a(1), i1=>b(1), eq=>e1); H3: equal
23 port map(i0=>a(2), i1=>b(2), eq=>e2); H4: equal
24 port map(i0=>a(3), i1=>b(3), eq=>e3); H5: equal
25 port map(i0=>a(4), i1=>b(4), eq=>e4); H6: equal
26 port map(i0=>a(5), i1=>b(5), eq=>e5); H7: equal
27 port map(i0=>a(6), i1=>b(6), eq=>e6); H8: equal
28 port map(i0=>a(7), i1=>b(7), eq=>e7);
29
30 aeqb <= e0 and e1 and e2 and e3 and e4 and e5 and e6 and e7;
31 end arch;
```

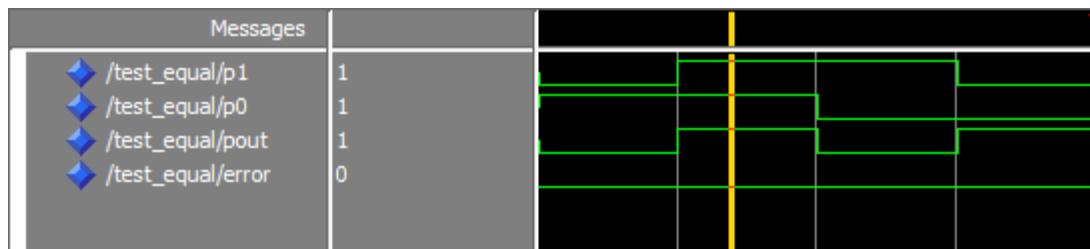


Simulation:

One-bit comparator test file:

```
1  Library ieee;
2  Use ieee.std_logic_1164.all;
3
4  Entity test_equal is
5  End test_equal;
6
7  Architecture arch_test of test_equal is
8
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  begin
17      uut: equal port map(I0 => p0, I1 => p1, Eq => pout);
18  process
19  begin
20      p0 <= '1';
21      p1 <= '0';
22      wait for 1 ns;
23      if (pout = '1') then
24          error <= '1'; end if;
25      wait for 200 ns; p0 <= '1';
26      p1 <= '1';
27      wait for 1 ns;
28      if (pout = '0') then
29          error <= '1';
30      end if;
31      wait for 200 ns; p0 <= '0';
32      p1 <= '1';
33      wait for 1 ns;
34      if (pout = '1') then
35          error <= '1'; end if;
36      wait for 200 ns; p0 <= '0';
37      p1 <= '0';
38      wait for 1 ns;
39      if (pout = '0') then
40          error <= '1'; end if;
41      wait for 200 ns;
42      if (error = '0') then
43          report "No errors detected. Simulation successful" severity failure;
44      else
45          report "Error detected" severity failure;
46      end if;
47  end process; End arch_test;
```

One-bit comparator waveform:



The outputs are correct and it contains 0 errors.

Two-bit comparator test file:

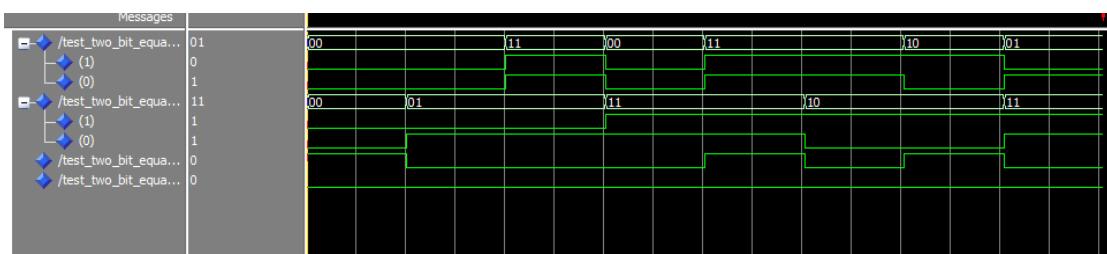
```
1  Library ieee;
2  Use ieee.std_logic_1164.all;
3
4  Entity Test_two_bit_equal is
5  End Test_two_bit_equal;
6
7  Architecture arch_test of Test_two_bit_equal is
8
9  component two_bit_equal
10 Port (
11   a, b: in std_logic_vector(1 downto 0); aeqb : out std_logic);
12 End component;
13
14 Signal p1, p0 : std_logic_vector(1 downto 0); Signal pout : std_logic;
15 Signal error : std_logic := '0';
16 begin
17  uut: two_bit_equal port map(a => p0, b => p1, aeqb => pout); process
18  begin
19    p0 <= "00";
20    p1 <= "00";
21    wait for 1 ns;
22    if (pout = '0') then
23      error <= '1'; end if;
24    wait for 200 ns; p0 <= "01"; p1 <= "00"; wait for 1 ns;
25    if (pout = '1') then error <= '1';
26    end if;
27    wait for 200 ns; p0 <= "01";
28    p1 <= "11";
29    wait for 1 ns;
30    if (pout = '1') then
31      error <= '1'; end if;
32    wait for 200 ns; p0 <= "11"; p1 <= "00"; wait for 1 ns;
33    if (pout = '1') then error <= '1';
34    end if;
35    wait for 200 ns; p0 <= "11"; p1 <= "11"; wait for 1 ns;
36    if (pout = '0') then error <= '1';
37    end if;
38    wait for 200 ns; p0 <= "10";
39    p1 <= "11";
40    wait for 1 ns;
41    if (pout = '1') then
42      error <= '1'; end if;
```

```

43   wait for 200 ns; p0 <= "10"; p1 <= "10"; wait for 1 ns;
44   if (pout = '0') then error <= '1';
45   end if;
46   wait for 200 ns; p0 <= "11";
47   p1 <= "01";
48   wait for 1 ns;
49   if (pout = '1') then
50     error <= '1'; end if;
51   wait for 200 ns;
52   if (error = '0') then
53     report "No errors detected. Simulation successful" severity failure;
54   else
55     report "Error detected" severity failure;
56   end if;
57 end process; End arch_test;

```

Two-bit comparator waveform:

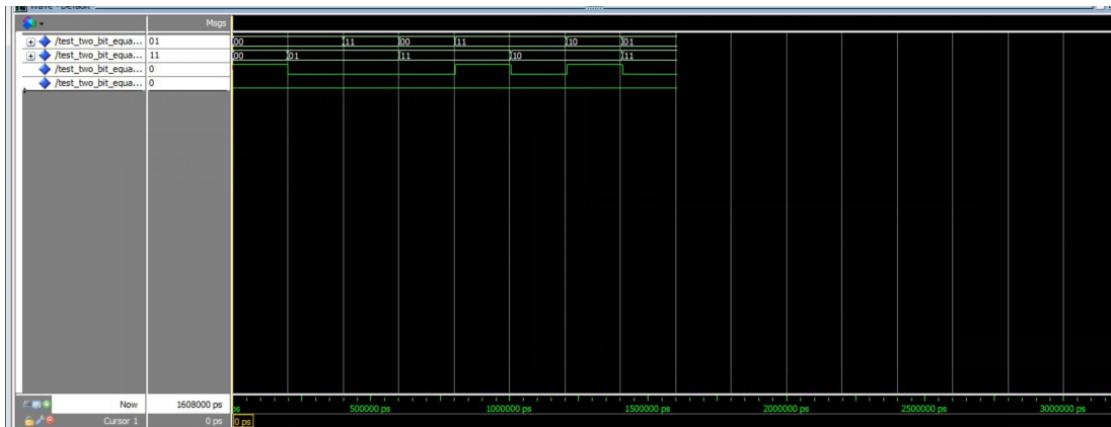


The outputs are correct and it contains 0 errors.

Two-bit comparator with port maps test file:

```
1  Library ieee;
2  Use ieee.std_logic_1164.all;
3
4  Entity Test_two_bit_equal_port is
5  End Test_two_bit_equal_port;
6
7  Architecture arch_test of Test_two_bit_equal_port is
8
9  Component two_bit_equal_port Port (
10   a, b: in std_logic_vector(1 downto 0); aeqb : out std_logic);
11  End component;
12
13  Signal p1, p0 : std_logic_vector(1 downto 0); Signal pout : std_logic;
14  Signal error : std_logic := '0';
15  begin
16  uut: two_bit_equal_port port map(a => p0, b => p1, aeqb => pout); process
17  begin
18  p0 <= "00";
19  p1 <= "00";
20  wait for 1 ns;
21  if (pout = '0') then
22  error <= '1'; end if;
23  wait for 200 ns; p0 <= "01"; p1 <= "00"; wait for 1 ns;
24  if (pout = '1') then error <= '1';
25  end if;
26  wait for 200 ns; p0 <= "01";
27  p1 <= "11";
28  wait for 1 ns;
29  if (pout = '1') then
30  error <= '1'; end if;
31  wait for 200 ns; p0 <= "11"; p1 <= "00"; wait for 1 ns;
32  if (pout = '1') then error <= '1';
33  end if;
34  wait for 200 ns; p0 <= "11";
35  p1 <= "11";
36  wait for 1 ns;
37  if (pout = '0') then
38  error <= '1'; end if;
39  wait for 200 ns; p0 <= "10"; p1 <= "11"; wait for 1 ns;
40  if (pout = '1') then
41  error <= '1'; end if;
42  wait for 200 ns; p0 <= "10"; p1 <= "10"; wait for 1 ns;
43  if (pout = '0') then error <= '1';
44  end if;
45  wait for 200 ns; p0 <= "11";
46  p1 <= "01";
47  wait for 1 ns;
48  if (pout = '1') then
49  error <= '1'; end if;
50  wait for 200 ns;
51  if (error = '0') then
52  report "No errors detected. Simulation successful" severity failure;
53  else
54  report "Error detected" severity failure;
55  end if;
56  end process; End arch_test;
```

Two-bit comparator with port maps waveform:



The outputs are correct and it contains 0 errors.

Eight-bit comparator test file:

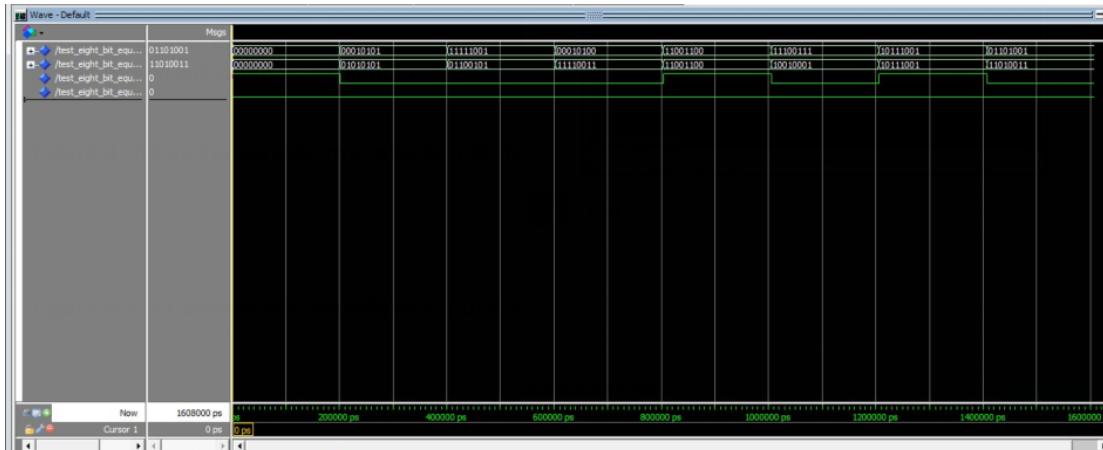
```
1  Library ieee;
2  Use ieee.std_logic_1164.all;
3
4  Entity Test_eight_bit_equal_port_map is
5  End Test_eight_bit_equal_port_map;
6
7  Architecture arch_test of Test_eight_bit_equal_port_map is
8
9  Component eight_bit_equal_port_map
10 Port (
11   a, b: in std_logic_vector(7 downto 0);
12   aeqb : out std_logic);
13 End component;
14
15 Signal p1, p0 : std_logic_vector(7 downto 0);
16 Signal pout : std_logic;
17 Signal error : std_logic := '0';
18 begin
19  uut: eight_bit_equal_port_map port map(a => p0, b => p1, aeqb => pout);
20 process
21 begin
22  p0 <= "00000000";
23  p1 <= "00000000";
24  wait for 1 ns;
25  if (pout = '0') then
26    error <= '1';
27  end if;
28  wait for 200 ns;
29  p0 <= "01010101";
30  p1 <= "00010101";
31  wait for 1 ns;
32  if (pout = '1') then
33    error <= '1';
34  end if;
35  wait for 200 ns;
36  p0 <= "01100101";
37  p1 <= "11111001";
38  wait for 1 ns;
39  if (pout = '1') then
40    error <= '1';
41  end if;
42  wait for 200 ns;
```

```

43    p0 <= "11110011";
44    p1 <= "00010100";
45    wait for 1 ns;
46    if (pout = '1') then
47        error <= '1';
48    end if;
49    wait for 200 ns;
50    p0 <= "11001100";
51    p1 <= "11001100";
52    wait for 1 ns;
53    if (pout = '0') then
54        error <= '1';
55    end if;
56    wait for 200 ns;
57    p0 <= "10010001";
58    p1 <= "11100111";
59    wait for 1 ns;
60    if (pout = '1') then
61        error <= '1';
62    end if;
63    wait for 200 ns;
64    p0 <= "10111001";
65    p1 <= "10111001";
66    wait for 1 ns;
67    if (pout = '0') then
68        error <= '1';
69    end if;
70    wait for 200 ns;
71    p0 <= "11010011";
72    p1 <= "01101001";
73    wait for 1 ns;
74    if (pout = '1') then
75        error <= '1';
76    end if;
77    wait for 200 ns;
78    if (error = '0') then
79        report "No errors detected. Simulation successful" severity failure;
80    else
81        report "Error detected" severity failure;
82    end if;
83    end process;
84 End arch_test;

```

Eight-bit comparator waveform:



The outputs are correct and it contains 0 errors.

Conclusion:

From this lab, we learned the basic code format for VHDL. We also learned how the 1-bit, 2-bit, and 8-bit comparators compare different bits of binary numbers. This lab helped me get a better understanding on using port maps and writing test file to test our code. Over all the objective of this lab was achieved. The simulations showed correct results.

Appendix:

1- bit comparator vhdl code:

Library ieee;

Use ieee.std_logic_1164.all;

Entity equal is

Port (

I0, I1: in std_logic;

```
Eq  : out std_logic);
End equal;
```

```
Architecture arch of equal is
Signal p0, p1 : std_logic;
begin
Eq <= p0 or p1;
P0 <= (not I0) and (not I1);
P1<= i0 and i1;
End arch;
```

2- 1-bit test file

```
Library ieee;
Use ieee.std_logic_1164.all;
```

```
Entity test_equal is
End test_equal;
```

```
Architecture arch_test of test_equal is
```

```
component equal
Port (
I0, I1: in std_logic;
Eq  : out std_logic);
End component;
```

```
Signal p1, p0, pout : std_logic;
Signal error      : std_logic := '0';
begin
uut: equal port map(I0 => p0, I1 => p1, Eq => pout);
process
begin
p0 <= '1';
p1 <= '0';
wait for 1 ns;
if (pout = '1') then
  error <= '1';
end if;
wait for 200 ns;
p0 <= '1';
p1 <= '1';
wait for 1 ns;
if (pout = '0') then
  error <= '1';
end if;
wait for 200 ns;
```

```

p0 <= '0';
p1 <= '1';
wait for 1 ns;
if (pout = '1') then
    error <= '1';
end if;
wait for 200 ns;
p0 <= '0';
p1 <= '0';
wait for 1 ns;
if (pout = '0') then
    error <= '1';
end if;
wait for 200 ns;

if (error = '0') then
    report "No errors detected. Simulation successful" severity failure;
else
    report "Error detected" severity failure;
end if;

end process;
End arch_test;

```

3- 2-bit comparator vhdl code:

```

Library ieee;
Use ieee.std_logic_1164.all;

Entity two_bit_equal is
Port (
a, b: in std_logic_vector(1 downto 0);
aeqb : out std_logic);
End two_bit_equal;

Architecture arch of two_bit_equal is
Signal p0, p1,p2,p3 : std_logic;
begin
aeqb <=(p0 or p1) and (p2 or p3); --Enter you code here;
P0 <=a(0) and b(0); --Enter you code here;
P1<=(not a(0)) and (not b(0)); --Enter you code here.;
P2<=a(1) and b(1);--Enter you code here;
P3 <=(not a(1)) and (not b(1));--Enter you code here;
End arch;

```

4- 2-bit test file

```

Library ieee;
```

```
Use ieee.std_logic_1164.all;
```

```
Entity Test_two_bit_equal is
End Test_two_bit_equal;
```

```
Architecture arch_test of Test_two_bit_equal is
```

```
component two_bit_equal
Port (
a, b: in std_logic_vector(1 downto 0);
aeqb : out std_logic);
End component;
```

```
Signal p1, p0 : std_logic_vector(1 downto 0);
Signal pout : std_logic;
Signal error : std_logic := '0';
begin
uut: two_bit_equal port map(a => p0, b => p1, aeqb => pout);
process
begin
p0 <= "00";
p1 <= "00";
wait for 1 ns;
if (pout = '0') then
    error <= '1';
end if;
wait for 200 ns;
p0 <= "01";
p1 <= "00";
wait for 1 ns;
if (pout = '1') then
    error <= '1';
end if;
wait for 200 ns;
p0 <= "01";
p1 <= "11";
wait for 1 ns;
if (pout = '1') then
    error <= '1';
end if;
wait for 200 ns;
p0 <= "11";
p1 <= "00";
wait for 1 ns;
if (pout = '1') then
    error <= '1';
end if;
```

```

end if;
wait for 200 ns;
p0 <= "11";
p1 <= "11";
wait for 1 ns;
if (pout = '0') then
    error <= '1';
end if;
wait for 200 ns;
p0 <= "10";
p1 <= "11";
wait for 1 ns;
if (pout = '1') then
    error <= '1';
end if;
wait for 200 ns;
p0 <= "10";
p1 <= "10";
wait for 1 ns;
if (pout = '0') then
    error <= '1';
end if;
wait for 200 ns;
p0 <= "11";
p1 <= "01";
wait for 1 ns;
if (pout = '1') then
    error <= '1';
end if;
wait for 200 ns;

if (error = '0') then
    report "No errors detected. Simulation successful" severity failure;
else
    report "Error detected" severity failure;
end if;

end process;
End arch_test;

```

5- 2-bit portmap code

```

Library ieee;
Use ieee.std_logic_1164.all;
```

```

Entity two_bit_equal_port is
Port (
```

```

a, b: in std_logic_vector(1 downto 0);
aeqb : out std_logic);
End two_bit_equal_port;

```

Architecture arch of two_bit_equal_port is

--component declaration...we are telling the compiler which components we want to use from the library.

```

component equal
Port (
I0, I1: in std_logic;
Eq : out std_logic);
End component;

signal e0,e1: std_logic;

begin

--instantiates two one-bit comparators

H1: equal
port map(i0=>a(0), i1=>b(0), eq=>e0);
H2: equal
port map(i0=>a(1), i1=>b(1), eq=>e1);

aeqb <= e0 and e1;

end arch;

```

6- 2-bit portmap test file

```

Library ieee;
Use ieee.std_logic_1164.all;

```

```

Entity Test_two_bit_equal_port is
End Test_two_bit_equal_port;

```

Architecture arch_test of Test_two_bit_equal_port is

```

component two_bit_equal_port
Port (
a, b: in std_logic_vector(1 downto 0);
aeqb : out std_logic);
End component;

```

Signal p1, p0 : std_logic_vector(1 downto 0);

```

Signal pout  : std_logic;
Signal error      : std_logic := '0';
begin
  uut: two_bit_equal_port port map(a => p0, b => p1, aeqb => pout);
process
begin
  p0 <= "00";
  p1 <= "00";
  wait for 1 ns;
  if (pout = '0') then
    error <= '1';
  end if;
  wait for 200 ns;
  p0 <= "01";
  p1 <= "00";
  wait for 1 ns;
  if (pout = '1') then
    error <= '1';
  end if;
  wait for 200 ns;
  p0 <= "01";
  p1 <= "11";
  wait for 1 ns;
  if (pout = '1') then
    error <= '1';
  end if;
  wait for 200 ns;
  p0 <= "11";
  p1 <= "00";
  wait for 1 ns;
  if (pout = '1') then
    error <= '1';
  end if;
  wait for 200 ns;
  p0 <= "11";
  p1 <= "11";
  wait for 1 ns;
  if (pout = '0') then
    error <= '1';
  end if;
  wait for 200 ns;
  p0 <= "10";
  p1 <= "11";
  wait for 1 ns;
  if (pout = '1') then
    error <= '1';

```

```

end if;
wait for 200 ns;
p0 <= "10";
p1 <= "10";
wait for 1 ns;
if (pout = '0') then
    error <= '1';
end if;
wait for 200 ns;
p0 <= "11";
p1 <= "01";
wait for 1 ns;
if (pout = '1') then
    error <= '1';
end if;
wait for 200 ns;

if (error = '0') then
    report "No errors detected. Simulation successful" severity failure;
else
    report "Error detected" severity failure;
end if;

end process;
End arch_test;

```

7- 8-bit portmap vhdl code

```

Library ieee;
Use ieee.std_logic_1164.all;
```

```

Entity eight_bit_equal_port is
Port (
a, b: in std_logic_vector(7 downto 0);
aeqb : out std_logic);
End eight_bit_equal_port;
```

Architecture arch of eight_bit_equal_port is

--component declaration...we are telling the compiler which components we want to use from the library.

```

component equal
Port (
I0, I1: in std_logic;
Eq : out std_logic);
End component;
```

```

signal e0,e1,e2,e3,e4,e5,e6,e7: std_logic;

begin

--instantiates two one-bit comparators

H1: equal
port map(i0=>a(0), i1=>b(0), eq=>e0);
H2: equal
port map(i0=>a(1), i1=>b(1), eq=>e1);
H3: equal
port map(i0=>a(2), i1=>b(2), eq=>e2);
H4: equal
port map(i0=>a(3), i1=>b(3), eq=>e3);
H5: equal
port map(i0=>a(4), i1=>b(4), eq=>e4);
H6: equal
port map(i0=>a(5), i1=>b(5), eq=>e5);
H7: equal
port map(i0=>a(6), i1=>b(6), eq=>e6);
H8: equal
port map(i0=>a(7), i1=>b(7), eq=>e7);

--a and b are equal if individual bits are equal.

aeqb <= e0 and e1 and e2 and e3 and e4 and e5 and e6 and e7;

```

end arch;

8- 8-bit pormap test file

```

Library ieee;
Use ieee.std_logic_1164.all;
```

```

Entity Test_eight_bit_equal_port_map is
End Test_eight_bit_equal_port_map;
```

```

Architecture arch_test of Test_eight_bit_equal_port_map is
```

```

component eight_bit_equal_port_map
  Port (
    a, b: in std_logic_vector(7 downto 0);
    aeqb : out std_logic);
End component;
```

```

Signal p1, p0 : std_logic_vector(7 downto 0);
```

```

Signal pout  : std_logic;
Signal error      : std_logic := '0';
begin
  uut: eight_bit_equal_port_map port map(a => p0, b => p1, aeqb => pout);
process
begin
  p0 <= "00000000";
  p1 <= "00000000";
  wait for 1 ns;
  if (pout = '0') then
    error <= '1';
  end if;
  wait for 200 ns;
  p0 <= "01010101";
  p1 <= "00010101";
  wait for 1 ns;
  if (pout = '1') then
    error <= '1';
  end if;
  wait for 200 ns;
  p0 <= "01100101";
  p1 <= "11111001";
  wait for 1 ns;
  if (pout = '1') then
    error <= '1';
  end if;
  wait for 200 ns;
  p0 <= "11110011";
  p1 <= "00010100";
  wait for 1 ns;
  if (pout = '1') then
    error <= '1';
  end if;
  wait for 200 ns;
  p0 <= "11001100";
  p1 <= "11001100";
  wait for 1 ns;
  if (pout = '0') then
    error <= '1';
  end if;
  wait for 200 ns;
  p0 <= "10010001";
  p1 <= "11100111";
  wait for 1 ns;
  if (pout = '1') then
    error <= '1';

```

```
end if;
wait for 200 ns;
p0 <= "10111001";
p1 <= "10111001";
wait for 1 ns;
if (pout = '0') then
    error <= '1';
end if;
wait for 200 ns;
p0 <= "11010011";
p1 <= "01101001";
wait for 1 ns;
if (pout = '1') then
    error <= '1';
end if;
wait for 200 ns;

if (error = '0') then
    report "No errors detected. Simulation successful" severity failure;
else
    report "Error detected" severity failure;
end if;

end process;
End arch_test;
```