Fall 2020

California State University, Northridge Department of Electrical and Computer Engineering Computer Assignment 1: Design of a Combination Multiplier

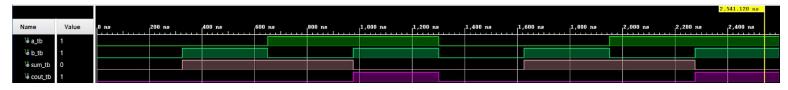
Part I:Half Adder Design (Code)

```
2 ! -- Engineer: Jose Luis Martinez
4 -- Create Date: 09/14/2020 09:33:16 AM
 5 ! -- Module Name: ha - Behavioral
    -- Project Name: Computer Assignment 1
    -- Revision 0.01 - File Created
10
11 | library IEEE;
12 use IEEE.STD LOGIC 1164.ALL;
13
14 -- Uncomment the following library declaration if using
15 -- arithmetic functions with Signed or Unsigned values
16 -- use IEEE. NUMERIC STD. ALL;
17
18 -- Uncomment the following library declaration if instantiating
19 -- any Xilinx leaf cells in this code.
20 | -- library UNISIM;
21 -- use UNISIM. VComponents.all;
23 - entity ha is
24 Port ( a : in STD LOGIC;
        b : in STD LOGIC;
             sum : out STD LOGIC;
              cout : out STD LOGIC);
28 end ha;
30 architecture Behavioral of ha is
31
32 | begin
34 sum <= a xor b;
35 | cout <= a and b;
37 end Behavioral;
```

Half Adder Design (Test Bench Code)

```
-- Engineer: Jose Luis Martinez
 3 | --
 4 -- Create Date: 09/14/2020 09:33:16 AM
 5 | -- Module Name: ha tb - Behavioral
 6 -- Project Name: Computer Assignment 1
 7 -- Revision 0.01 - File Created
 8 🖨 -----
 9
10
11 | library IEEE;
12 use IEEE.STD LOGIC 1164.ALL;
13 use IEEE.NUMERIC STD.ALL;
14
15 \ominus entity ha_tb is
16 -- Port ();
17 end ha tb;
18
19 architecture Behavioral of ha tb is
20
21 | signal a tb: std logic;
22 | signal b tb: std logic;
23 | signal sum tb: std logic;
24 | signal cout th: std logic;
25 1
26 component ha is
27 Port ( a : in STD LOGIC;
             b : in STD LOGIC;
28
29
              sum : out STD LOGIC;
30
               cout : out STD LOGIC);
31 \(\hat{\rightarrow}\) end component ha;
32
33 | begin
34
35 - uutt: ha port map ( a => a tb,
36
                        b => b tb,
37
                       sum => sum tb,
38 🖯
                        cout => cout tb);
39
40 process
41 | begin
43 ⊖ for aFor in 0 to 1 loop
44  for bFor in 0 to 1 loop
46 🖯
        if aFor = 1 then a tb <= '1';
47 :
        else a_tb <= '0';
48 🖨
         end if;
49
50 🖨
       if bFor = 1 then b_tb <= '1';
         else b_tb <= '0';
51
```

Half Adder Design (Waveforms)



Full Adder Design (Code)

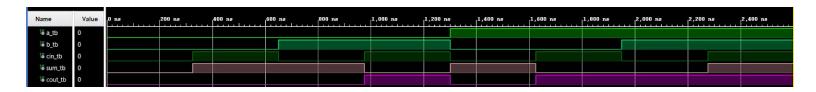
```
1 🖯 ------
 2 | -- Engineer: Jose Luis Martinez
 3 ---
 4 -- Create Date: 09/14/2020 09:33:16 AM
 5 | -- Module Name: fa - Behavioral
 6 -- Project Name: Computer Assignment 1
 7 -- Revision 0.01 - File Created
9
10
11
    library IEEE;
12
    use IEEE.STD LOGIC 1164.ALL;
13
15
        Port ( a : in STD LOGIC;
16
              b : in STD LOGIC;
               cin : in STD LOGIC;
17
18
               sum : out STD LOGIC;
19
               cout : out STD LOGIC);
20 end fa;
21
22 G architecture Behavioral of fa is
23
24
   begin
25
26
   sum <= a xor b xor cin;
27
   cout <= (a and b) or (a and cin) or (b and cin);
28
29 @ end Behavioral;
30
```

Full Adder Design (Test Bench Code)

```
2 | -- Engineer: Jose Luis Martinez
 3 ---
 4 -- Create Date: 09/14/2020 09:33:16 AM
    -- Module Name: fa tb - Behavioral
 6 -- Project Name: Computer Assignment 1
 7 -- Revision 0.01 - File Created
 8 🖨 -----
 9
10
11 | library IEEE;
12 use IEEE.STD LOGIC 1164.ALL;
13 use IEEE.NUMERIC STD.ALL;
14
15 ⊖ entity fas_tb is
16 -- Port ();
17 end fas_tb;
18
19 architecture Behavioral of fas tb is
20
21 | signal a_tb: std_logic;
22 | signal b_tb: std logic;
23 | signal cin_tb: std logic;
24 | signal sum tb: std logic;
25 signal cout_tb: std logic;
26
27 © component fa is
28 Port ( a : in STD LOGIC;
             b : in STD LOGIC;
29
30
             cin : in STD LOGIC;
31
             sum : out STD LOGIC;
              cout : out STD LOGIC);
32
33 end component fa;
34
35 | begin
37 \( \text{uutt: fa port map (a => a_tb,} \)
38
                       b => b_tb,
39
                       cin => cin tb,
40
                      sum => sum tb,
41 🖯
                      cout => cout tb);
42 !
43 process
44 begin
45 !
46 ⊖ for aFor in 0 to 1 loop
47 of for bFor in 0 to 1 loop
        for cinFor in 0 to 1 loop
48 🖯
49
50 ⊖
             if aFor = 1 then a tb <= '1';
51
             else a_tb <= '0';
52 🖨
              end if;
53
            if bFor = 1 then b tb <= '1';
54 ⊖
55
             else b tb <= '0';
56 🖯
             end if;
```

```
57
58 🖨
             if cinFor = 1 then cin tb <= '1';
59
              else cin_tb <= '0';
60 ⊝
              end if;
               wait for 325ns;
61
62
63 🖨
          end loop;
64 🖨
      end loop;
65 \(\hat{\rightarrow}\) end loop;
67 end process;
68 end Behavioral;
```

Full Adder Design (Waveforms)



4x4 Multiplier Design (Code)

```
-- Engineer: Jose Luis Martinez
 3
 4 · -- Create Date: 09/14/2020 09:33:16 AM
    -- Module Name: mult4x4 - Behavioral
    -- Project Name: Computer Assignment 1
 7 1
    -- Revision 0.01 - File Created
 8 🛆 -----
9
10
11
12 | library IEEE;
13
    use IEEE.STD LOGIC 1164.ALL;
   use IEEE.NUMERIC STD.ALL;
14
15
16 entity mult4x4 is
17
      Port ( x : in STD LOGIC VECTOR(3 downto 0);
              y : in STD LOGIC VECTOR(3 downto 0);
18
19
              p : out STD_LOGIC_VECTOR(7 downto 0));
20 end mult4x4;
21
22 @ architecture Behavioral of mult4x4 is
24 🖯 component ha is
25
       Port ( a : in STD LOGIC;
26
              b : in STD LOGIC;
              sum : out STD LOGIC;
27
28
              cout : out STD LOGIC);
29 \end component ha;
31 🖯 component fa is
      Port ( a : in STD_LOGIC;
```

```
33
                b : in STD LOGIC;
34
                cin : in STD LOGIC;
35
                sum : out STD LOGIC;
36
                cout : out STD LOGIC);
37 end component fa;
38
39
     signal x0y0, x0y1, x0y2, x0y3: std logic;
40 1
     signal x1y0, x1y1, x1y2, x1y3: std logic;
41
   signal x2y0, x2y1, x2y2, x2y3: std logic;
42
    signal x3y0, x3y1, x3y2, x3y3: std logic;
43
    signal c00, c01, c02, c03, c10, c11, c12, c13, c20, c21, c22, c23, c30, c31, c32, c33: std logic;
44
45
    signal s00, s01, s02, s03, s10, s11, s12, s13, s20, s21, s22, s23, s30, s31, s32, s33: std logic;
46
47
48
49 \times 0 y0 <= \times (0) and y(0);
50 | x0y1 \le x(0) and y(1);
51
    x0y2 \le x(0) and y(2);
52
    x0y3 \le x(0) and y(3);
53
    x1y0 \le x(1) and y(0);
54 \times 1y1 \le x(1) and y(1);
55 x1y2 \le x(1) and y(2);
55 \times 1y2 \le x(1) and y(2);
56 | x1y3 \le x(1) and y(3);
57
     x2y0 \le x(2) and y(0);
58 '
     x2y1 \le x(2) and y(1);
59 !
     x2y2 \le x(2) and y(2);
60
    x2y3 \le x(2) and y(3);
61
     x3y0 \le x(3) and y(0);
62
     x3y1 \le x(3) and y(1);
63
     x3y2 \le x(3) and y(2);
64
     x3y3 \le x(3) and y(3);
65
66 comp00: ha port map (a => x3y1, b => c01, sum => s00, cout => c00);
     comp01: fa port map (a => x3y0, b => x2y1, cin => c02, sum => s01, cout => c01);
67 1
68
     comp02: fa port map (a => x2y0, b => x1y1, cin => c03, sum => s02, cout => c02);
69
     comp03: ha port map (a => x1y0, b => x0y1, sum => s03, cout => c03);
70
     comp10: fa port map (a => c00, b => x3y2, cin => c11, sum => s10, cout => c10);
71 !
     comp11: fa port map (a => s00, b => x2y2, cin => c12, sum => s11, cout => c11);
72
     comp12: fa port map (a => s01, b => x1y2, cin => c13, sum => s12, cout => c12);
73 1
     comp13: ha port map (a => s02, b => x0y2, sum => s13, cout => c13);
74 !
     comp20: fa port map (a => c10, b => x3y3, cin => c21, sum => s20, cout => c20);
75
     comp21: fa port map (a => s10, b => x2y3, cin => c22, sum => s21, cout => c21);
76 1
     comp22: fa port map (a => s11, b => x1y3, cin => c23, sum => s22, cout => c22);
77
     comp23: ha port map (a => s12, b => x0y3, sum => s23, cout => c23);
78
79 1
     p(0) \le x(0) and y(0);
80
     p(1) \le s03;
81
     p(2) \le s13;
82 '
     p(3) \le s23;
83 | p(4) <= s22;
```

```
84 | p(5) <= s21;

85 | p(6) <= s20;

86 | p(7) <= c20;

87 |

88 \(\ho\) end Behavioral;
```

4x4 Multiplier Design (Test Bench Code)

```
2 | -- Engineer: Jose Luis Martinez
 3 ---
    -- Create Date: 09/14/2020 09:33:16 AM
 4 1
 5 | -- Module Name: 4x4mult tb - Behavioral
 6 -- Project Name: Computer Assignment 1
 7 -- Revision 0.01 - File Created
 8 🖹 -----
 9
10
11 | library IEEE;
12 use IEEE.STD LOGIC 1164.ALL;
13 use IEEE.NUMERIC STD.ALL;
14
15 entity mult4x4_tb is
16 : -- Port ();
17 end mult4x4 tb;
18
19 architecture Behavioral of mult4x4_tb is
20
21 | signal x tb : STD LOGIC VECTOR (3 downto 0);
22 | signal y tb : STD LOGIC VECTOR (3 downto 0);
23 | signal p_tb : STD LOGIC VECTOR (7 downto 0);
24
25 G component mult4x4 is
     Port ( x : in STD LOGIC VECTOR (3 downto 0);
26 !
27
             y : in STD_LOGIC_VECTOR (3 downto 0);
28
              p : out STD LOGIC VECTOR (7 downto 0));
29 end component mult4x4;
30
31 begin
32
33 - uut: mult4x4 port map ( x => x tb,
34
                          y => y_tb,
35 ⊖
                          p => p_tb);
36
37 process
38 | begin
39
40 0 for xFor in 0 to 15 loop
41 ♀
      for yFor in 0 to 15 loop
42 1
            x tb <= std logic vector(to unsigned(xFor, 4));
43 !
            y_tb <= std logic vector(to_unsigned(yFor, 4));
44
            wait for 10ns;
45 🖨
      end loop;
```

```
46 ⊕ end loop;

47 ¦

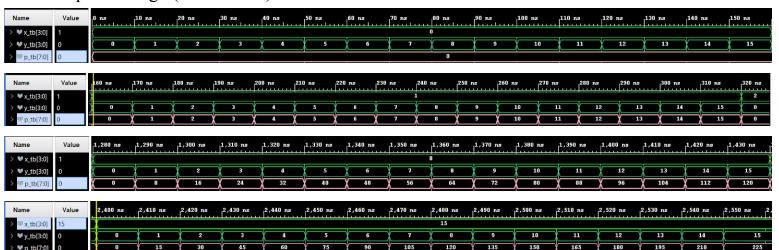
48 ¦ wait;

49 ¦

50 ⊕ end process;

51 ⊕ end Behavioral;
```

4x4 Multiplier Design (Waveforms)



Part II:

Part II uses the half and full adder designs from above as well as the half and full subtractors below.

Half Subtractor Design (Code)

```
-- Engineer: Jose Luis Martinez
 3 |
 4 -- Create Date: 09/14/2020 09:33:16 AM
    -- Module Name: hs - Behavioral
 6 -- Project Name: Computer Assignment 1
 7 -- Revision 0.01 - File Created
 8 🖨 -----
 9
10
11
    library IEEE;
12 use IEEE.STD LOGIC 1164.ALL;
13
14 entity hs is
15 Port ( a : in STD LOGIC;
            b : in STD LOGIC;
             bout : out STD_LOGIC;
17
18
             diff : out STD LOGIC);
19 end hs;
20
21 architecture Behavioral of hs is
22
23 !
    begin
24
25 | diff <= a xor b;
26 | bout <= (not a) and b;
27
28 end Behavioral;
29
```

Half Subtractor Design (Test Bench Code)

```
2 !
    -- Engineer: Jose Luis Martinez
 3
 4
    -- Create Date: 09/14/2020 09:33:16 AM
 5 | -- Module Name: hs tb - Behavioral
    -- Project Name: Computer Assignment 1
 7 1
    -- Revision 0.01 - File Created
8 🛱 -----
 9
10 | library IEEE;
    use IEEE.STD LOGIC 1164.ALL;
11
12 use IEEE.NUMERIC STD.ALL;
14 entity hs tb is
15 -- Port ();
16 end hs_tb;
```

```
18 architecture Behavioral of hs tb is
19
20
    signal a_tb: std logic;
21
    signal b_tb: std logic;
22
    signal bout th: std logic;
    signal diff th: std logic;
23
24
25 component hs is
26
        Port ( a : in STD LOGIC;
27
               b : in STD_LOGIC;
28
              bout : out STD_LOGIC;
29
              diff : out STD LOGIC);
30 end component hs;
31
32
    begin
33
34 \( \pi \) uutt: hs port map ( a => a_tb,
35
                       b => b tb,
                       bout => bout tb,
36
37 🖨
                       diff => diff_tb);
38
39 🖨 process
40 begin
41
42 🖯 for aFor in 0 to 1 loop
43 of for bFor in 0 to 1 loop
44
45 🖯
         if aFor = 1 then a_tb <= '1';
46
         else a_tb <= '0';
47 🖨
         end if;
48
49 🖨
        if bFor = 1 then b_tb <= '1';
50
         else b_tb <= '0';
51 🖨
         end if;
52
53 !
          wait for 325ns;
54
55 🖨
      end loop;
56 @ end loop;
58 end process;
59 end Behavioral;
```

Half Subtractor Design (Waveforms)



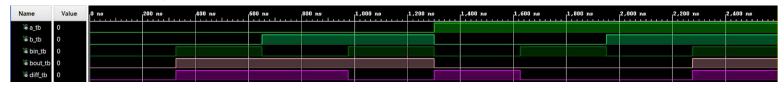
Full Subtractor Design (Code)

```
2 -- Engineer: Jose Luis Martinez
 3
 4 -- Create Date: 09/14/2020 09:33:16 AM
   -- Module Name: fs - Behavioral
 5 !
 6 : -- Project Name: Computer Assignment 1
7 -- Revision 0.01 - File Created
8 🖨 -----
9
10
11
12
    library IEEE;
13
    use IEEE.STD LOGIC 1164.ALL;
14
15 ⊖ entity fs is
       Port ( a : in STD LOGIC;
17
              b : in STD LOGIC;
18
              bin : in STD LOGIC;
19
              bout : out STD LOGIC;
20
              diff : out STD LOGIC);
21 @ end fs;
22
23 - architecture Behavioral of fs is
24
25
   begin
26
27 | diff <= (a xor b) xor bin;
28 | bout <= ((not a) and b) or ((not (not a) xor (not b)) and bin);
29
30 end Behavioral;
31
```

Full Subtractor Design (Test Bench Code)

```
13 1
14 ⊖ entity fs_tb is
15 : -- Port ();
16 end fs_tb;
18 architecture Behavioral of fs_tb is
19
20 | signal a_tb: std_logic;
21 | signal b_tb: std logic;
22 signal bin_tb: std logic;
23 | signal bout tb: std logic;
24 signal diff_tb: std logic;
25
26 © component fs is
27 Port ( a : in STD LOGIC;
              b : in STD LOGIC;
29
              bin : in STD LOGIC;
30
              bout : out STD LOGIC;
              diff : out STD LOGIC);
31
32 \(\hat{\rightarrow}\) end component fs;
34 begin
35
36 uutt: fs port map ( a => a_tb,
37
                        b => b_tb,
38
                       bin => bin tb,
39
                       diff => diff tb,
40 🖨
                       bout => bout_tb);
41
42 process
43 | begin
45 for aFor in 0 to 1 loop
46 🖯
     for bFor in 0 to 1 loop
47 ♀
         for binFor in 0 to 1 loop
48
49 🖯
             if aFor = 1 then a_tb <= '1';
50 !
             else a_tb <= '0';
51 🖒
              end if;
52
53 ⊝
             if bFor = 1 then b tb <= '1';
54
             else b_tb <= '0';
55 🖨
             end if;
56 !
           if binFor = 1 then bin_tb <= '1';
58
            else bin_tb <= '0';
59 ⊖
             end if;
60
             wait for 325ns;
61
62 🖨
        end loop;
63 end loop;
64 end loop;
66 end process;
67 end Behavioral;
68 ;
```

Full Subtractor Design (Waveforms)



4x4 Signed Multiplier Design (Code)

```
-- Engineer: Jose Luis Martinez
 3
    -- Create Date: 09/14/2020 09:33:16 AM
    -- Module Name: smult4x4 - Behavioral
   -- Project Name: Computer Assignment 1
   -- Revision 0.01 - File Created
 7
8 🗇 -----
9
10
    library IEEE;
11 !
    use IEEE.STD LOGIC 1164.ALL;
12
13
   use IEEE.NUMERIC STD.ALL;
14
15
16 entity smult3x3 is
     Port ( x : in STD LOGIC VECTOR(3 downto 0);
            y : in STD LOGIC VECTOR (3 downto 0);
19
             p : out STD_LOGIC_VECTOR(7 downto 0));
20 @ end smult3x3;
21
22 architecture Behavioral of smult3x3 is
23
24 Component ha is
      Port ( a : in STD LOGIC;
             b : in STD LOGIC;
27
              sum : out STD LOGIC;
28
               cout : out STD LOGIC);
29 end component ha;
31 \odot component fa is
    Port ( a : in STD LOGIC;
33
             b : in STD LOGIC;
             cin : in STD LOGIC;
35
              sum : out STD LOGIC;
              cout : out STD LOGIC);
37 end component fa;
39 🖯 component hs is
    Port ( a : in STD LOGIC;
41
             b : in STD LOGIC;
```

```
42
                   bout : out STD LOGIC;
43
                   diff : out STD LOGIC);
44 end component hs;
45
46 ⊖ component fs is
47
           Port ( a : in STD LOGIC;
                   b : in STD LOGIC;
48
49
                   bin : in STD LOGIC;
50
                   bout : out STD LOGIC;
51
                   diff : out STD LOGIC);
52 end component fs;
    signal x0y0, x0y1, x0y2, x0y3: std logic;
55
    signal x1y0, x1y1, x1y2, x1y3: std logic;
    signal x2y0, x2y1, x2y2, x2y3: std logic;
56 1
57
    signal x3y0, x3y1, x3y2, x3y3: std logic;
58
59
    signal c00, c01, c02, c03, c04, c05, c06: std logic;
    signal c10, c11, c12, c13, c14, c15: std logic;
60 !
    signal d20, d21, d22, d23, d24: std logic;
61
62
63 ;
    signal s00, s01, s02, s03, s04, s05, s06: std logic;
64
    signal s10, s11, s12, s13, s14, s15: std logic;
65 1
    signal b20, b21, b22, b23, b24: std logic;
66
67
    begin
68
69 !
    x0y0 \le x(0) and y(0);
70 \times 0y1 <= x(0) and y(1);
71 \times 0y2 \le x(0) and y(2);
72 !
    x0y3 \le x(0) and y(3);
73 \times 1y0 \le x(1) and y(0);
74 \times 1y1 \le x(1) and y(1);
75 !
    x1y2 \le x(1) and y(2);
76 \times 1y3 \le x(1) and y(3);
77 1
    x2y0 \le x(2) and y(0);
78 !
    x2y1 \le x(2) and y(1);
79 | x2y2 \le x(2) and y(2);
80 \times 2y3 \le x(2) and y(3);
81
    x3y0 \le x(3) and y(0);
82 ;
    x3y1 \le x(3) and y(1);
83 1
    x3y2 \le x(3) and y(2);
84
    x3y3 \le x(3) and y(3);
85
86 comp00: ha port map (a => x1y0, b => x0y1, sum => s00, cout => c00);
87 | comp01: fa port map (a => x2y0, b => x1y1, cin => c00, sum => s01, cout => c01);
88 comp02: fa port map (a => x3y0, b => x2y1, cin => c01, sum => s02, cout => c02);
89 comp03: fa port map (a => x3y0, b => x3y1, cin => c02, sum => s03, cout => c03);
90 | comp04: fa port map (a => x3y0, b => x3y1, cin => c03, sum => s04, cout => c04);
```

```
91 comp05: fa port map (a => x3y0, b => x3y1, cin => c04, sum => s05, cout => c05);
     comp06: fa port map (a => x3y0, b => x3y1, cin => c05, sum => s06, cout => c06);
93
94
    comp10: ha port map (a => s01, b => x0y2, sum => s10, cout => c10);
     comp11: fa port map (a => s02, b => x1y2, cin => c10, sum => s11, cout => c11);
95 1
96
    comp12: fa port map (a => s03, b => x2y2, cin => c11, sum => s12, cout => c12);
97
    comp13: fa port map (a => s04, b => x3y2, cin => c12, sum => s13, cout => c13);
98
     comp14: fa port map (a => s05, b => x3y2, cin => c13, sum => s14, cout => c14);
     comp15: fa port map (a => s06, b => x3y2, cin => c14, sum => s15, cout => c15);
99
100
101 | comp20: hs port map (a => s11, b => x0y3, bout => b20, diff => d20);
102 | comp21: fs port map (a => s12, b => x1y3, bin => b20, bout => b21, diff => d21);
103 comp22: fs port map (a => s13, b => x2y3, bin => b21, bout => b22, diff => d22);
104 | comp23: fs port map (a => s14, b => x3y3, bin => b22, bout => b23, diff => d23);
105 | comp24: fs port map (a => s15, b => x3y3, bin => b23, bout => b24, diff => d24);
107 \cdot p(0) \le x0y0;
108 | p(1) <= s00;
109 | p(2) <= s10;
110 \cdot p(3) \le d20;
111 p(4) \le d21;
112 \mid p(5) \le d22;
113 p(6) <= d23;
114 \mid p(7) \le d24;
115
116 \(\hat{\text{\text{-}}}\) end Behavioral;
117
```

4x4 Signed Multiplier Design (Test Bench Code)

```
1 🖯
 2 !
         -- Engineer: Jose Luis Martinez
3 1
         -- Create Date: 09/14/2020 09:33:16 AM
 4 1
5 !
         -- Module Name: 4x4smult tb - Behavioral
 6
         -- Project Name: Computer Assignment 1
7
         -- Revision 0.01 - File Created
8 (
10
11
         library IEEE;
12
         use IEEE.STD LOGIC 1164.ALL;
13
        use IEEE.NUMERIC STD.ALL;
15 🖯
         entity smult3x3 tb is
         -- Port ();
16
17 0
         end smult3x3 tb;
19 🖨
         architecture Behavioral of smult3x3 tb is
20 ;
21
         signal x tb : STD LOGIC VECTOR (3 downto 0);
```

```
22 !
         signal y_tb : STD LOGIC VECTOR (3 downto 0);
23
         signal p_tb : STD LOGIC VECTOR (7 downto 0);
24
25 ⊖
         component smult3x3 is
26
             Port ( x : in STD LOGIC VECTOR (3 downto 0);
27
                    y : in STD LOGIC VECTOR (3 downto 0);
28
                    p : out STD_LOGIC_VECTOR (7 downto 0));
29 🖨
         end component smult3x3;
30
31
         begin
32
33 ⊖
         uutttt: smult3x3 port map ( x => x_tb,
34
                                 y => y_tb,
35 🖨
                                 p => p_tb);
36
37 ⊖
         process
38
         begin
39
40 □
         for xFor in -8 to 7 loop
41 🖯
            for yFor in -8 to 7 loop
                 x_tb <= std logic_vector(to_signed(xFor, 4));</pre>
42
43
                 y_tb <= std logic vector(to_signed(yFor, 4));</pre>
44
                 wait for 10ns;
45 🖨
          end loop;
         end loop;
46 🖯
47
48
         wait;
49
50 🖨
         end process;
51 🖯
         end Behavioral;
52 !
```

4x4 Signed Multiplier Design (Waveforms)

							,										
Name	Value	0 ns	10 ns	20 ns	30 ns	40 ns	50 ns	60 ns	70 ns	80 ns	90 ns	100 ns	110 ns	120 ns	130 ns	140 ns	150 ns
> W x_tb[3:0]	-7									-8						,	
> ¥ y_tb[3:0]	-8	-8	-7	-6	-5	/-4	-3	-2	-1	0	1	2	3	4	5	6	7
> W p_tb[7:0]	56	64	56	48	40	32	24	16	8	0	-8	-16	-24	-32	-40	-48	-56
Name	Value	480 ns	490 ns	500 ns	510 ns	520 ns	530 ns	540 ns	550 ns	560 ns	570 ns	580 ns	590 ns	600 ns	610 ns	620 ns	630 ns
> ¥ x_tb[3:0]	-7	X	.,	.,	.,	.,			.,	-5		.,	.,				
> W y_tb[3:0]	-8	-8	_7	-6	-5	-4	-3	-2	-1	V 0	1	2	3	4	5	6	7
> ¥ p_tb[7:0]	56	40	35	30	25	20	15	10	5	0	-5	-10	-15	-20	-25	-30	-35
Name	Value		1,290 ns	1,300 ns	1,310 ns	1,320 ns	1,330 ns	1,340 ns	1,350 ns	1,360 ns	1,370 ns	1,380 ns	1,390 ns	1,400 ns	1,410 ns	1,420 ns	1,430 ns
> W x_tb[3:0]	-7			,	,	,		,		0	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,		,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,		,	,	,
> 💆 y_tb[3:0]	-8	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7
> W p_tb[7:0]	56									0							
								,		.,		,		,			
Name	Value		1,450 ns	1,460 ns	1,470 ns	1,480 ns	1,490 ns	1,500 ns	1,510 ns	1,520 ns	1,530 ns	1,540 ns	1,550 ns	1,560 ns	1,570 ns	1,580 ns	1,590 ns
> ₩ x_tb[3:0]	-7				,		,		,	1	,		,	1	,	1	
> W y_tb[3:0]	-8	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7
> ♥ p_tb[7:0]	56	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7
Name	Value		1,610 ns	1,620 ns	1,630 ns	1,640 ns	1,650 ns	1,660 ns	1,670 ns	1,680 ns	1,690 ns	1,700 ns	1,710 ns	1,720 ns	1,730 ns	1,740 ns	1,750 ns
> ₩ x_tb[3:0]	-7									2							
> V y_tb[3:0]	-8	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7
> W p_tb[7:0]	56	-16	-14	-12	-10	-8	-6	-4	-2	0	2	X 4	6	8	10	12	14