Jose Luis Martinez
Prof: Shahnam Mirzaei, Ph.D.
October 2, 2020
ECE 420

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)

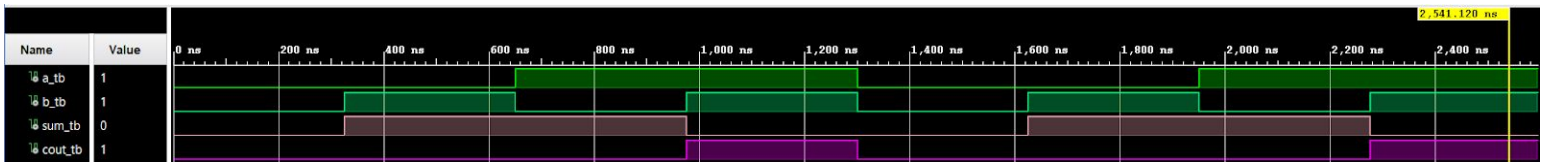
```
1  ----------------------------------------------------------------------------------
2  -- Engineer: Jose Luis Martinez
3  --
4  -- Create Date: 09/14/2020 09:33:16 AM
5  -- Module Name: ha - Behavioral
6  -- Project Name: Computer Assignment 1
7  -- Revision 0.01 - File Created
8  ----------------------------------------------------------------------------------
9
10
11 library IEEE;
12 use IEEE.STD_LOGIC_1164.ALL;
13
14 -- 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;
22
23 entity ha is
24     Port ( a : in STD_LOGIC;
25            b : in STD_LOGIC;
26            sum : out STD_LOGIC;
27            cout : out STD_LOGIC);
28 end ha;
29
30 architecture Behavioral of ha is
31
32 begin
33
34 sum <= a xor b;
35 cout <= a and b;
36
37 end Behavioral;
```

Half Adder Design (Test Bench Code)

```
----------------------------------------------------------------------------------
-- Engineer: Jose Luis Martinez
--
-- Create Date: 09/14/2020 09:33:16 AM
-- Module Name: ha_tb - Behavioral
-- Project Name: Computer Assignment 1
-- Revision 0.01 - File Created
----------------------------------------------------------------------------------


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;

entity ha_tb is
--  Port ( );
end ha_tb;

architecture Behavioral of ha_tb is

signal a_tb: std_logic;
signal b_tb: std_logic;
signal sum_tb: std_logic;
signal cout_tb: std_logic;

component ha is
    Port ( a : in STD_LOGIC;
           b : in STD_LOGIC;
           sum : out STD_LOGIC;
           cout : out STD_LOGIC);
end component ha;

begin

uutt: ha port map ( a => a_tb,
                    b => b_tb,
                    sum => sum_tb,
                    cout => cout_tb);

process
begin

for aFor in 0 to 1 loop
   for bFor in 0 to 1 loop

      if aFor = 1 then a_tb <= '1';
      else a_tb <= '0';
      end if;

      if bFor = 1 then b_tb <= '1';
      else b_tb <= '0';
```

```
52          end if;
53
54          wait for 325ns;
55
56      end loop;
57  end loop;
58
59  end process;
60  end Behavioral;
```

Half Adder Design (Waveforms)



Full Adder Design (Code)

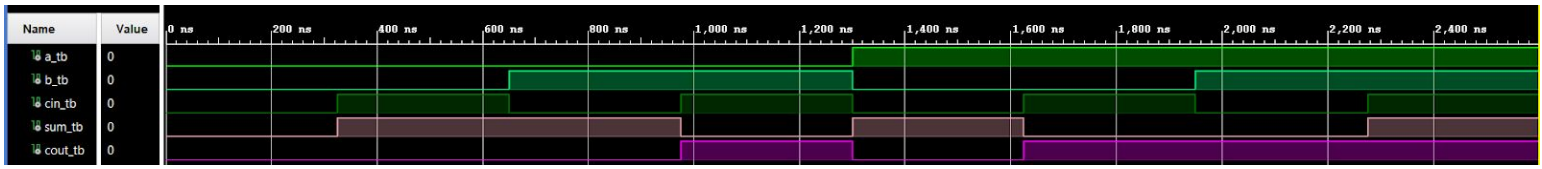
```
1   ----------------------------------------------------------------------------------
2   -- Engineer: Jose Luis Martinez
3   --
4   -- Create Date: 09/14/2020 09:33:16 AM
5   -- Module Name: fa - Behavioral
6   -- Project Name: Computer Assignment 1
7   -- Revision 0.01 - File Created
8   ----------------------------------------------------------------------------------
9
10
11  library IEEE;
12  use IEEE.STD_LOGIC_1164.ALL;
13
14  entity fa is
15      Port ( a : in STD_LOGIC;
16             b : in STD_LOGIC;
17             cin : in STD_LOGIC;
18             sum : out STD_LOGIC;
19             cout : out STD_LOGIC);
20  end fa;
21
22  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)

```
----------------------------------------------------------------------------------
-- Engineer: Jose Luis Martinez
--
-- Create Date: 09/14/2020 09:33:16 AM
-- Module Name: fa_tb - Behavioral
-- Project Name: Computer Assignment 1
-- Revision 0.01 - File Created
----------------------------------------------------------------------------------


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;

entity fas_tb is
--  Port ( );
end fas_tb;

architecture Behavioral of fas_tb is

signal a_tb: std_logic;
signal b_tb: std_logic;
signal cin_tb: std_logic;
signal sum_tb: std_logic;
signal cout_tb: std_logic;

component fa is
    Port ( a : in STD_LOGIC;
           b : in STD_LOGIC;
           cin : in STD_LOGIC;
           sum : out STD_LOGIC;
           cout : out STD_LOGIC);
end component fa;

begin

uutt: fa port map ( a => a_tb,
                    b => b_tb,
                    cin => cin_tb,
                    sum => sum_tb,
                    cout => cout_tb);


process
begin

for aFor in 0 to 1 loop
   for bFor in 0 to 1 loop
      for cinFor in 0 to 1 loop

          if aFor = 1 then a_tb <= '1';
          else a_tb <= '0';
          end if;

          if bFor = 1 then b_tb <= '1';
          else b_tb <= '0';
          end if;
```

```
57
58          if cinFor = 1 then cin_tb <= '1';
59          else cin_tb <= '0';
60          end if;
61          wait for 325ns;
62
63      end loop;
64    end loop;
65 end loop;
66
67 end process;
68 end Behavioral;
```

Full Adder Design (Waveforms)



4x4 Multiplier Design (Code)

```
1  ----------------------------------------------------------------------------------
2  -- Engineer: Jose Luis Martinez
3  --
4  -- Create Date: 09/14/2020 09:33:16 AM
5  -- Module Name: mult4x4 - Behavioral
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 use IEEE.NUMERIC_STD.ALL;
15
16 entity mult4x4 is
17     Port ( x : in STD_LOGIC_VECTOR(3 downto 0);
18            y : in STD_LOGIC_VECTOR(3 downto 0);
19            p : out STD_LOGIC_VECTOR(7 downto 0));
20 end mult4x4;
21
22 architecture Behavioral of mult4x4 is
23
24 component ha is
25     Port ( a : in STD_LOGIC;
26            b : in STD_LOGIC;
27            sum : out STD_LOGIC;
28            cout : out STD_LOGIC);
29 end component ha;
30
31 component fa is
32     Port ( a : in STD_LOGIC;
```

```
            b : in STD_LOGIC;
            cin : in STD_LOGIC;
            sum : out STD_LOGIC;
            cout : out STD_LOGIC);
end component fa;

signal x0y0, x0y1, x0y2, x0y3: std_logic;
signal x1y0, x1y1, x1y2, x1y3: std_logic;
signal x2y0, x2y1, x2y2, x2y3: std_logic;
signal x3y0, x3y1, x3y2, x3y3: std_logic;

signal c00, c01, c02, c03, c10, c11, c12, c13, c20, c21, c22, c23, c30, c31, c32, c33: std_logic;
signal s00, s01, s02, s03, s10, s11, s12, s13, s20, s21, s22, s23, s30, s31, s32, s33: std_logic;

begin

x0y0 <= x(0) and y(0);
x0y1 <= x(0) and y(1);
x0y2 <= x(0) and y(2);
x0y3 <= x(0) and y(3);
x1y0 <= x(1) and y(0);
x1y1 <= x(1) and y(1);
x1y2 <= x(1) and y(2);
```

```
x1y2 <= x(1) and y(2);
x1y3 <= x(1) and y(3);
x2y0 <= x(2) and y(0);
x2y1 <= x(2) and y(1);
x2y2 <= x(2) and y(2);
x2y3 <= x(2) and y(3);
x3y0 <= x(3) and y(0);
x3y1 <= x(3) and y(1);
x3y2 <= x(3) and y(2);
x3y3 <= x(3) and y(3);

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);
comp02: fa port map (a => x2y0, b => x1y1, cin => c03, sum => s02, cout => c02);
comp03: ha port map (a => x1y0, b => x0y1, sum => s03, cout => c03);
comp10: fa port map (a => c00, b => x3y2, cin => c11, sum => s10, cout => c10);
comp11: fa port map (a => s00, b => x2y2, cin => c12, sum => s11, cout => c11);
comp12: fa port map (a => s01, b => x1y2, cin => c13, sum => s12, cout => c12);
comp13: ha port map (a => s02, b => x0y2, sum => s13, cout => c13);
comp20: fa port map (a => c10, b => x3y3, cin => c21, sum => s20, cout => c20);
comp21: fa port map (a => s10, b => x2y3, cin => c22, sum => s21, cout => c21);
comp22: fa port map (a => s11, b => x1y3, cin => c23, sum => s22, cout => c22);
comp23: ha port map (a => s12, b => x0y3, sum => s23, cout => c23);

p(0) <= x(0) and y(0);
p(1) <= s03;
p(2) <= s13;
p(3) <= s23;
p(4) <= s22;
```

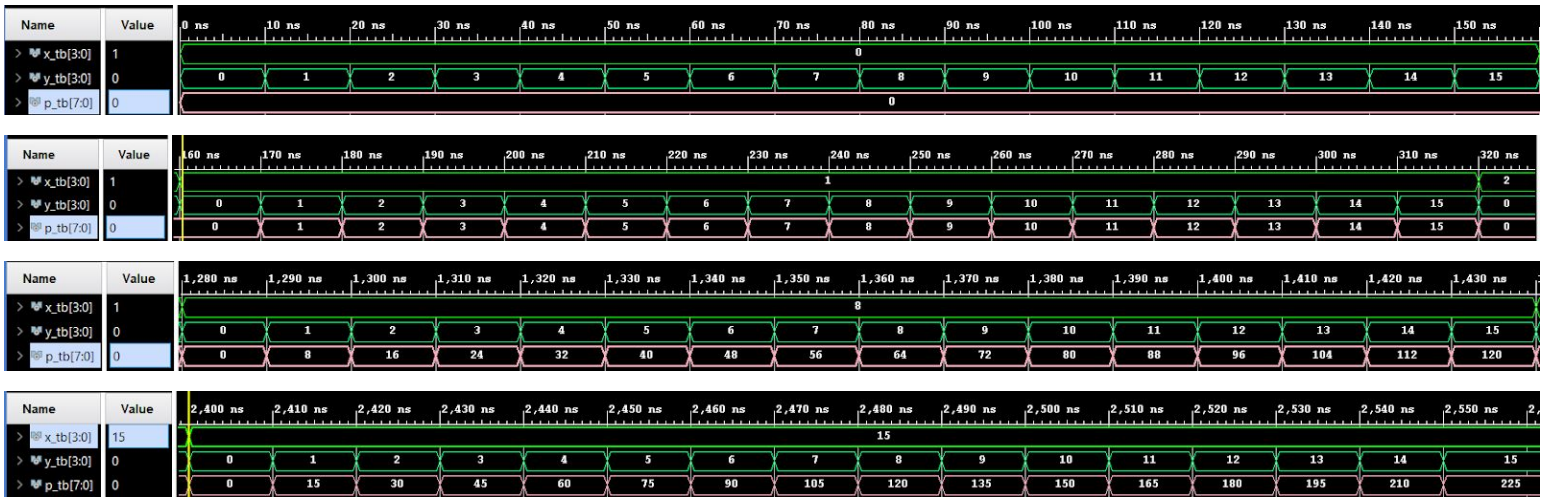
```
p(5) <= s21;
p(6) <= s20;
p(7) <= c20;

end Behavioral;
```

4x4 Multiplier Design (Test Bench Code)

```
----------------------------------------------------------------------------------
-- Engineer: Jose Luis Martinez
--
-- Create Date: 09/14/2020 09:33:16 AM
-- Module Name: 4x4mult_tb - Behavioral
-- Project Name: Computer Assignment 1
-- Revision 0.01 - File Created
----------------------------------------------------------------------------------


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;

entity mult4x4_tb is
--  Port ( );
end mult4x4_tb;

architecture Behavioral of mult4x4_tb is

signal x_tb : STD_LOGIC_VECTOR (3 downto 0);
signal y_tb : STD_LOGIC_VECTOR (3 downto 0);
signal p_tb : STD_LOGIC_VECTOR (7 downto 0);

component mult4x4 is
    Port ( x : in STD_LOGIC_VECTOR (3 downto 0);
           y : in STD_LOGIC_VECTOR (3 downto 0);
           p : out STD_LOGIC_VECTOR (7 downto 0));
end component mult4x4;

begin

uut: mult4x4 port map ( x => x_tb,
                        y => y_tb,
                        p => p_tb);

process
begin

for xFor in 0 to 15 loop
    for yFor in 0 to 15 loop

        x_tb <= std_logic_vector(to_unsigned(xFor, 4));
        y_tb <= std_logic_vector(to_unsigned(yFor, 4));
        wait for 10ns;
    end loop;
```

```
46  end loop;
47
48  wait;
49
50  end process;
51  end Behavioral;
52
```

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
--
-- Create Date: 09/14/2020 09:33:16 AM
-- Module Name: hs - Behavioral
-- Project Name: Computer Assignment 1
-- Revision 0.01 - File Created
----------------------------------------------------------------------------------


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity hs is
    Port ( a : in STD_LOGIC;
           b : in STD_LOGIC;
           bout : out STD_LOGIC;
           diff : out STD_LOGIC);
end hs;

architecture Behavioral of hs is

begin

diff <= a xor b;
bout <= (not a) and b;

end Behavioral;

```

Half Subtractor Design (Test Bench Code)

```
----------------------------------------------------------------------------------
-- Engineer: Jose Luis Martinez
--
-- Create Date: 09/14/2020 09:33:16 AM
-- Module Name: hs_tb - Behavioral
-- Project Name: Computer Assignment 1
-- Revision 0.01 - File Created
----------------------------------------------------------------------------------

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;

entity hs_tb is
--  Port ( );
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_tb: std_logic;
23  signal diff_tb: std_logic;
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  uutt: hs port map ( a => a_tb,
35                      b => b_tb,
36                      bout => bout_tb,
37                      diff => diff_tb);
38
39  process
40  begin
41
42  for aFor in 0 to 1 loop
43     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;

57
58  end process;
59  end Behavioral;
```

Half Subtractor Design (Waveforms)

Full Subtractor Design (Code)

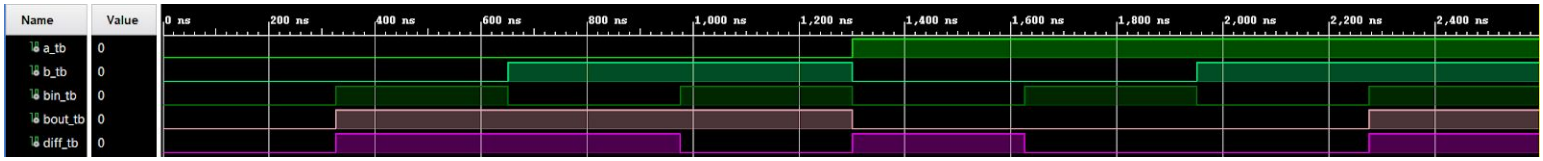
```
----------------------------------------------------------------------------------
-- Engineer: Jose Luis Martinez
--
-- Create Date: 09/14/2020 09:33:16 AM
-- Module Name: fs - Behavioral
-- Project Name: Computer Assignment 1
-- Revision 0.01 - File Created
----------------------------------------------------------------------------------



library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity fs is
    Port ( a : in STD_LOGIC;
           b : in STD_LOGIC;
           bin : in STD_LOGIC;
           bout : out STD_LOGIC;
           diff : out STD_LOGIC);
end fs;

architecture Behavioral of fs is

begin

diff <= (a xor b) xor bin;
bout <= ((not a) and b) or ((not(not a) xor (not b)) and bin);

end Behavioral;

```

Full Subtractor Design (Test Bench Code)

```
----------------------------------------------------------------------------------
-- Engineer: Jose Luis Martinez
--
-- Create Date: 09/14/2020 09:33:16 AM
-- Module Name: fs_tb - Behavioral
-- Project Name: Computer Assignment 1
-- Revision 0.01 - File Created
----------------------------------------------------------------------------------

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;

```

```
13
14  entity fs_tb is
15  --  Port ( );
16  end fs_tb;
17
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;
28             b : in STD_LOGIC;
29             bin : in STD_LOGIC;
30             bout : out STD_LOGIC;
31             diff : out STD_LOGIC);
32  end component fs;
33
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
44
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
57            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;
65
66  end process;
67  end Behavioral;
68
```

Full Subtractor Design (Waveforms)



4x4 Signed Multiplier Design (Code)

```
----------------------------------------------------------------------------------
-- Engineer: Jose Luis Martinez
--
-- Create Date: 09/14/2020 09:33:16 AM
-- Module Name: smult4x4 - Behavioral
-- Project Name: Computer Assignment 1
-- Revision 0.01 - File Created
----------------------------------------------------------------------------------

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

use IEEE.NUMERIC_STD.ALL;


entity smult3x3 is
   Port ( x : in STD_LOGIC_VECTOR(3 downto 0);
          y : in STD_LOGIC_VECTOR(3 downto 0);
          p : out STD_LOGIC_VECTOR(7 downto 0));
end smult3x3;

architecture Behavioral of smult3x3 is

component ha is
    Port ( a : in STD_LOGIC;
           b : in STD_LOGIC;
           sum : out STD_LOGIC;
           cout : out STD_LOGIC);
end component ha;

component fa is
    Port ( a : in STD_LOGIC;
           b : in STD_LOGIC;
           cin : in STD_LOGIC;
           sum : out STD_LOGIC;
           cout : out STD_LOGIC);
end component fa;

component hs is
    Port ( a : in STD_LOGIC;
           b : in STD LOGIC;
```

```
            bout : out STD_LOGIC;
            diff : out STD_LOGIC);
end component hs;

component fs is
    Port ( a : in STD_LOGIC;
           b : in STD_LOGIC;
           bin : in STD_LOGIC;
           bout : out STD_LOGIC;
           diff : out STD_LOGIC);
end component fs;

signal x0y0, x0y1, x0y2, x0y3: std_logic;
signal x1y0, x1y1, x1y2, x1y3: std_logic;
signal x2y0, x2y1, x2y2, x2y3: std_logic;
signal x3y0, x3y1, x3y2, x3y3: std_logic;

signal c00, c01, c02, c03, c04, c05, c06: std_logic;
signal c10, c11, c12, c13, c14, c15: std_logic;
signal d20, d21, d22, d23, d24: std_logic;

signal s00, s01, s02, s03, s04, s05, s06: std_logic;
signal s10, s11, s12, s13, s14, s15: std_logic;
signal b20, b21, b22, b23, b24: std_logic;

begin

x0y0 <= x(0) and y(0);
x0y1 <= x(0) and y(1);
x0y2 <= x(0) and y(2);
x0y3 <= x(0) and y(3);
x1y0 <= x(1) and y(0);
x1y1 <= x(1) and y(1);
x1y2 <= x(1) and y(2);
x1y3 <= x(1) and y(3);
x2y0 <= x(2) and y(0);
x2y1 <= x(2) and y(1);
x2y2 <= x(2) and y(2);
x2y3 <= x(2) and y(3);
x3y0 <= x(3) and y(0);
x3y1 <= x(3) and y(1);
x3y2 <= x(3) and y(2);
x3y3 <= x(3) and y(3);

comp00: ha port map (a => x1y0, b => x0y1, sum => s00, cout => c00);
comp01: fa port map (a => x2y0, b => x1y1, cin => c00, sum => s01, cout => c01);
comp02: fa port map (a => x3y0, b => x2y1, cin => c01, sum => s02, cout => c02);
comp03: fa port map (a => x3y0, b => x3y1, cin => c02, sum => s03, cout => c03);
comp04: fa port map (a => x3y0, b => x3y1, cin => c03, sum => s04, cout => c04);
```

```
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);

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);
comp12: fa port map (a => s03, b => x2y2, cin => c11, sum => s12, cout => c12);
comp13: fa port map (a => s04, b => x3y2, cin => c12, sum => s13, cout => c13);
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);

comp20: hs port map (a => s11, b => x0y3, bout => b20, diff => d20);
comp21: fs port map (a => s12, b => x1y3, bin => b20, bout => b21, diff => d21);
comp22: fs port map (a => s13, b => x2y3, bin => b21, bout => b22, diff => d22);
comp23: fs port map (a => s14, b => x3y3, bin => b22, bout => b23, diff => d23);
comp24: fs port map (a => s15, b => x3y3, bin => b23, bout => b24, diff => d24);
```

```
p(0) <= x0y0;
p(1) <= s00;
p(2) <= s10;
p(3) <= d20;
p(4) <= d21;
p(5) <= d22;
p(6) <= d23;
p(7) <= d24;

end Behavioral;
```

4x4 Signed Multiplier Design (Test Bench Code)

```
----------------------------------------------------------------------------------
-- Engineer: Jose Luis Martinez
--
-- Create Date: 09/14/2020 09:33:16 AM
-- Module Name: 4x4smult_tb - Behavioral
-- Project Name: Computer Assignment 1
-- Revision 0.01 - File Created
----------------------------------------------------------------------------------


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;

entity smult3x3_tb is
--  Port ( );
end smult3x3_tb;

architecture Behavioral of smult3x3_tb is

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
42              x_tb <= std_logic_vector(to_signed(xFor, 4));
43              y_tb <= std_logic_vector(to_signed(yFor, 4));
44              wait for 10ns;
45         end loop;
46      end loop;
47
48      wait;
49
50      end process;
51      end Behavioral;
52
```

4x4 Signed Multiplier Design (Waveforms)