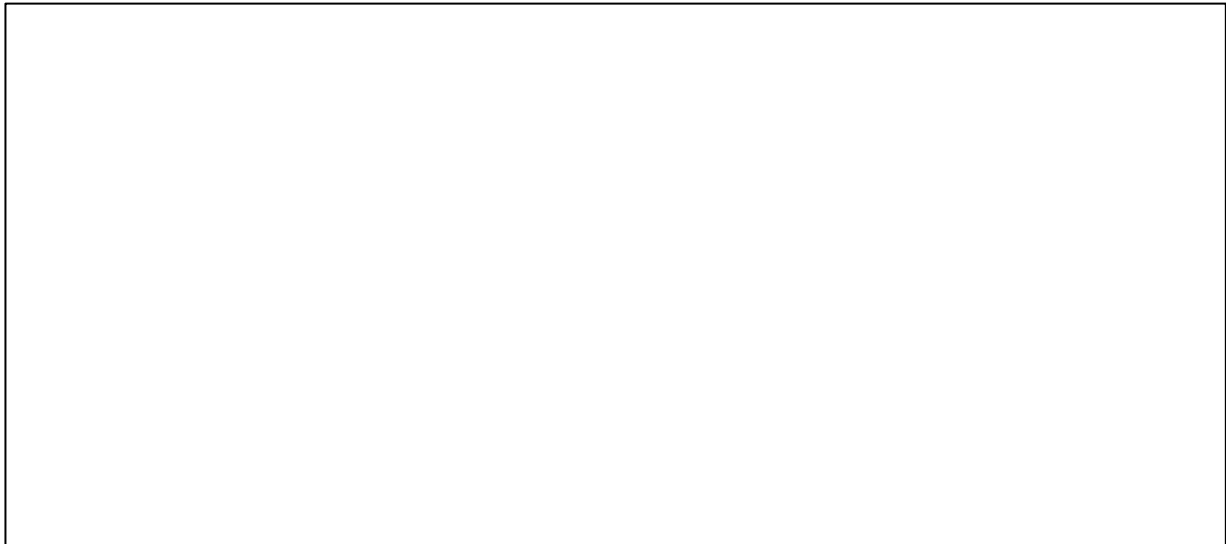**HACETTEPE UNIVERSITY**
ELECTRICAL AND ELECTRONICS ENGINEERING
ELE227 FUNDAMENTALS OF DIGITAL SYSTEMS LABORATORY
FALL 2019

1) Circuit Diagram

VHDL Implementation:

```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity fullSubtractor is
    Port ( a : in  STD_LOGIC;
           b : in  STD_LOGIC;
           cIn : in  STD_LOGIC;
           borrow : out  STD_LOGIC;
           result : out
STD_LOGIC);
end fullSubtractor;

architecture Behavioral of
fullSubtractor is
signal inSign:std_logic_vector (2
downto 0);
signal resSign:std_logic_vector (7
downto 0);
component three_to_eight_decoder is
    Port ( input : in
STD_LOGIC_VECTOR (2 downto 0);
           output : out
STD_LOGIC_VECTOR (7 downto 0));
end component;

begin

    inSign(0) <= cIn;

    inSign(1) <= b;

    inSign(2) <= a;

    decoder1:three_to_eight_deco
der port map(inSign,resSign);

    result <= (resSign(1) or
resSign(2)   or resSign(4) or
resSign(7));

    borrow <= (resSign(1) or
resSign(2) or resSign(3) or
resSign(7));

end Behavioral;
```
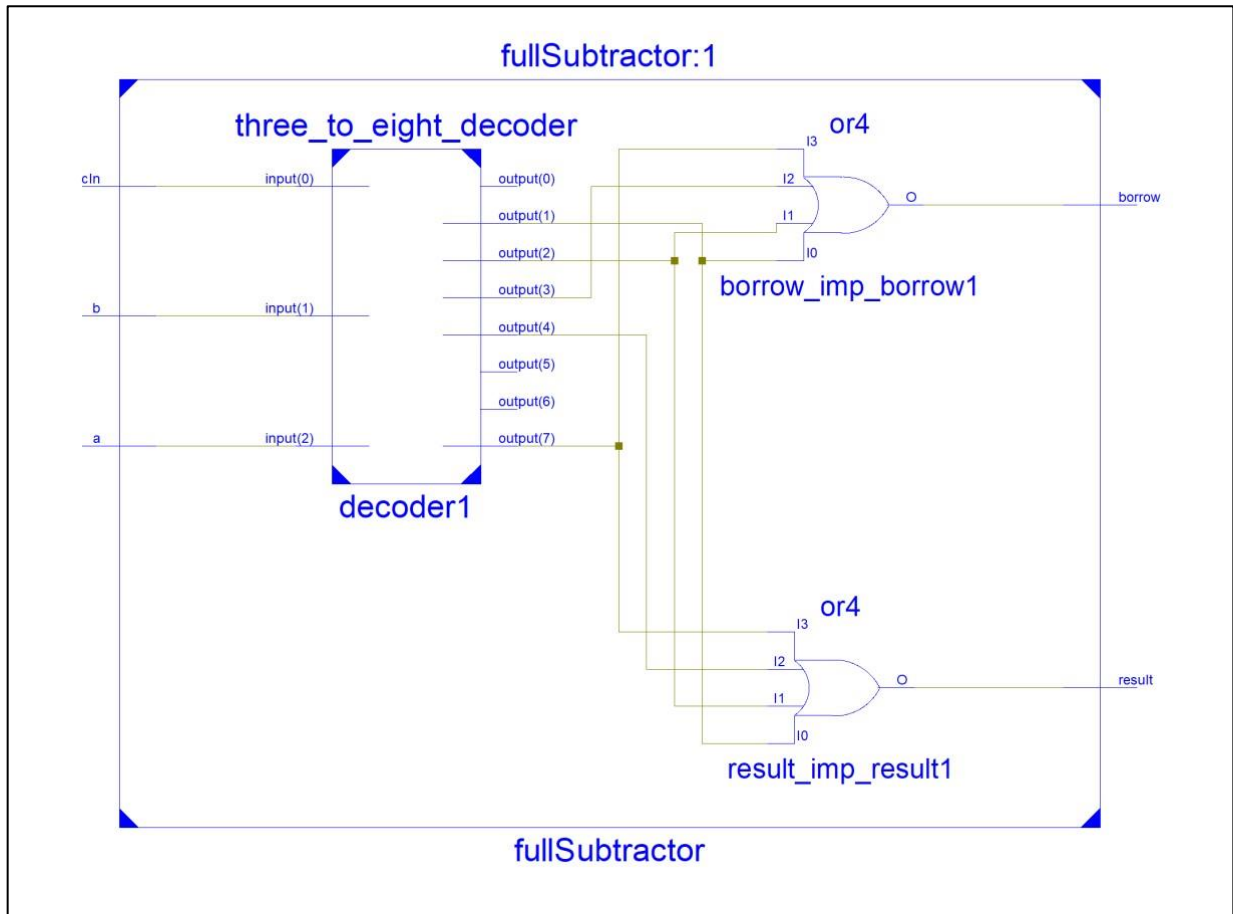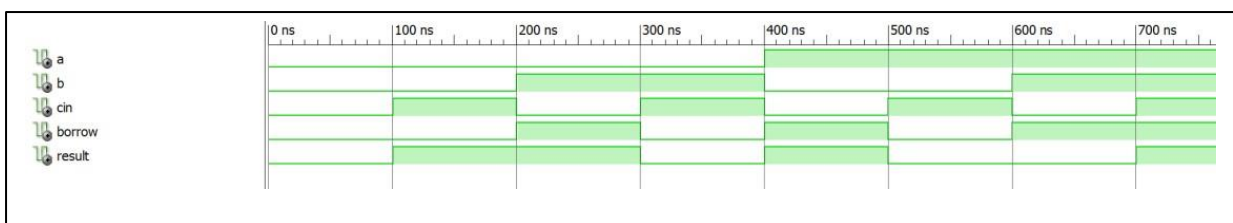
RTL Schematic:

Test Bench Input:

```
stim_proc: process
begin
        a <= '0';
        b <= '0';
        cIn <= '0';
    wait for 100 ns;
        a <= '0';
        b <= '0';
        cIn <= '1';
    wait for 100 ns;
        a <= '0';
        b <= '1';
        cIn <= '0';
    wait for 100 ns;
        a <= '0';
        b <= '1';
        cIn <= '1';
    wait for 100 ns;
```

```
        a <= '1';
        b <= '0';
        cIn <= '0';
    wait for 100 ns;
        a <= '1';
        b <= '0';
        cIn <= '1';
    wait for 100 ns;
        a <= '1';
        b <= '1';
        cIn <= '0';
    wait for 100 ns;
        a <= '1';
        b <= '1';
        cIn <= '1';
    wait for 100 ns;
    wait;
end process;
```

Test Bench Result:

2) Circuit Diagram

VHDL Implementation:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity func is
    Port ( x : in  STD_LOGIC;
           y : in  STD_LOGIC;
           z : in  STD_LOGIC;
           result : out
STD_LOGIC);
end func;


architecture Behavioral of func is
component mux is
    Port ( d3 : in  STD_LOGIC;
           d2 : in  STD_LOGIC;
           d1 : in  STD_LOGIC;
           d0 : in  STD_LOGIC;
           s1 : in  STD_LOGIC;
           s0 : in  STD_LOGIC;
           output : out
STD_LOGIC);
end component;
```
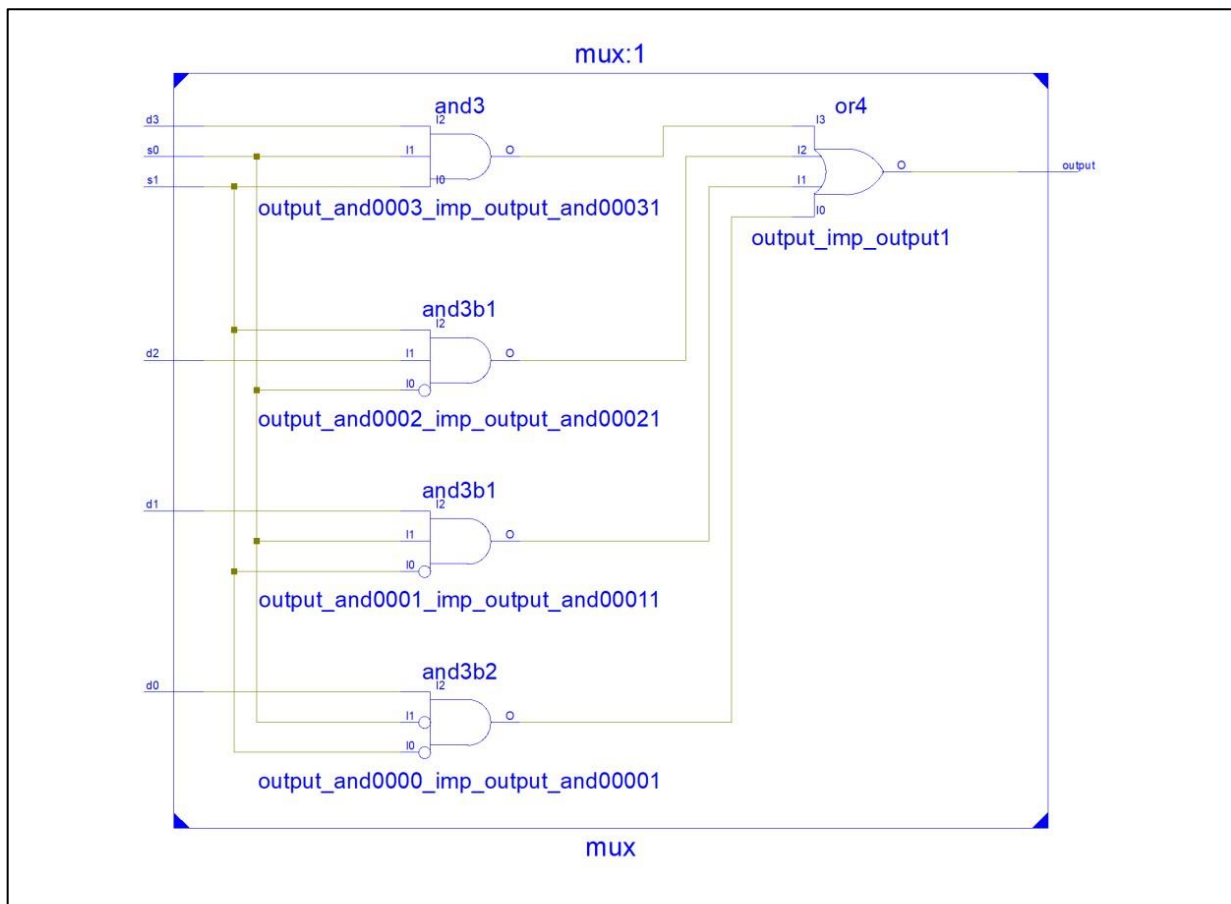
```
signal
xSign,ySign,zSign,rSign:std_logic;
begin
     xSign <= x;
     ySign <= y;
     zSign <= z;
     result <= rSign;
     mux1:mux port
map('0','1',zSign,'1',xSign,ySign,
rSign);


end Behavioral;
```
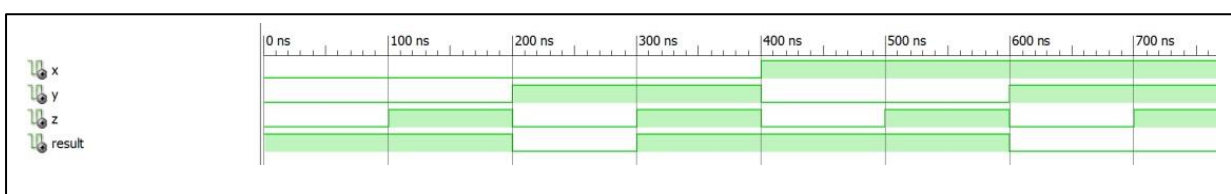
RTL Schematic:

Test Bench Input:

```
stim_proc: process
begin
        x <= '0';
        y <= '0';
        z <= '0';
        wait for 100 ns;
        x <= '0';
        y <= '0';
        z <= '1';
        wait for 100 ns;
        x <= '0';
        y <= '1';
        z <= '0';
        wait for 100 ns;
        x <= '0';
        y <= '1';
        z <= '1';
        wait for 100 ns;

        x <= '1';
        y <= '0';
        z <= '0';
        wait for 100 ns;
        x <= '1';
        y <= '0';
        z <= '1';
        wait for 100 ns;
        x <= '1';
        y <= '1';
        z <= '0';
        wait for 100 ns;
        x <= '1';
        y <= '1';
        z <= '1';
    wait;
end process;
```

Test Bench Result:

3) Circuit Diagram

VHDL Implementation:

```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;


entity comparator is
    Port ( a : in  STD_LOGIC_VECTOR (1
downto 0);
           b : in  STD_LOGIC_VECTOR (1
downto 0);
           g : out  STD_LOGIC;
           e : out  STD_LOGIC;
           l : out  STD_LOGIC);
end comparator;


architecture Behavioral of comparator
is
component decodertwo is
    Port ( x : in  STD_LOGIC_VECTOR (1
downto 0);
           result : out
STD_LOGIC_VECTOR (3 downto 0));
end component;
```

```vhdl
signal aDec,bDec : std_logic_vector(3
downto 0);

begin


decoder1 : decodertwo port map
(a,aDec);

decoder2 : decodertwo port map
(b,bDec);


g <= (aDec(3) and (bDec(0) or bDec(1)
or bDec(2))) or (aDec(2) and (bDec(0)
or bDec(1))) or (aDec(1) and bDec(0));

e <= (aDec(0) and bDec(0)) or (aDec(1)
and bDec(1)) or (aDec(2) and bDec(2))
or (aDec(3) and bDec(3));

l <= (aDec(0) and (bDec(1) or bDec(2)
or bDec(3))) or (aDec(1) and (bDec(2)
or bDec(3))) or (aDec(2) and bDec(3));


end Behavioral;
```
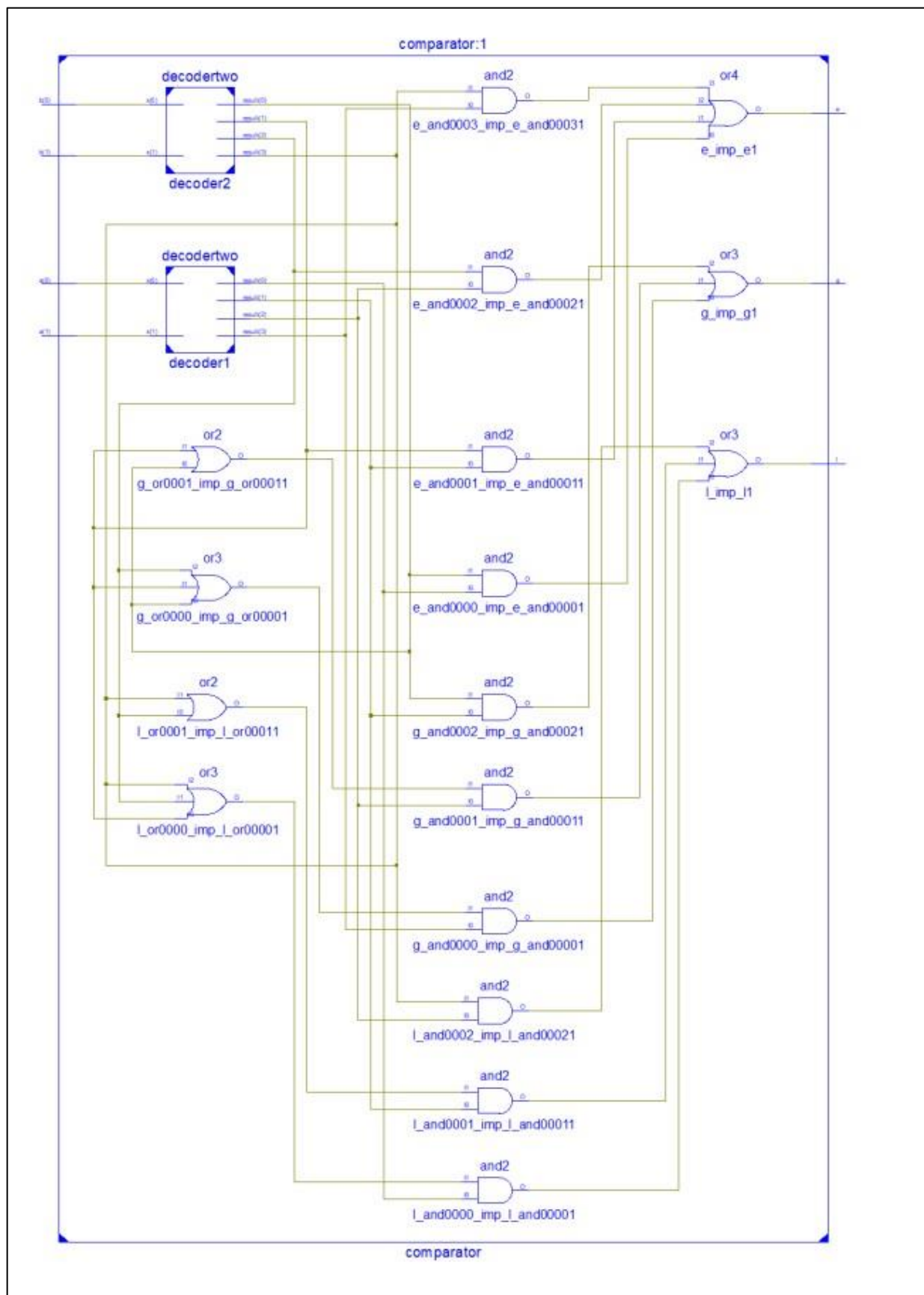
RTL Schematic:
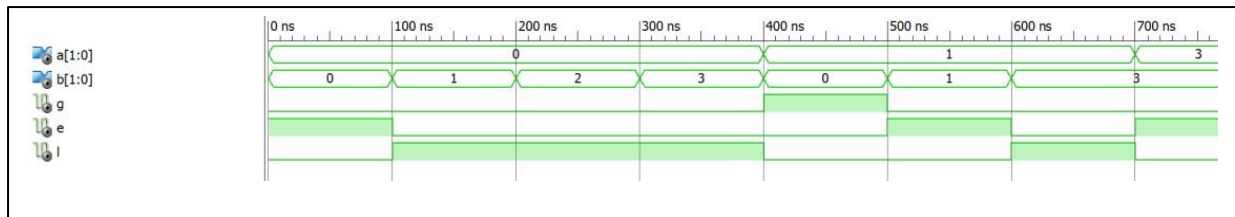
Test Bench Input:
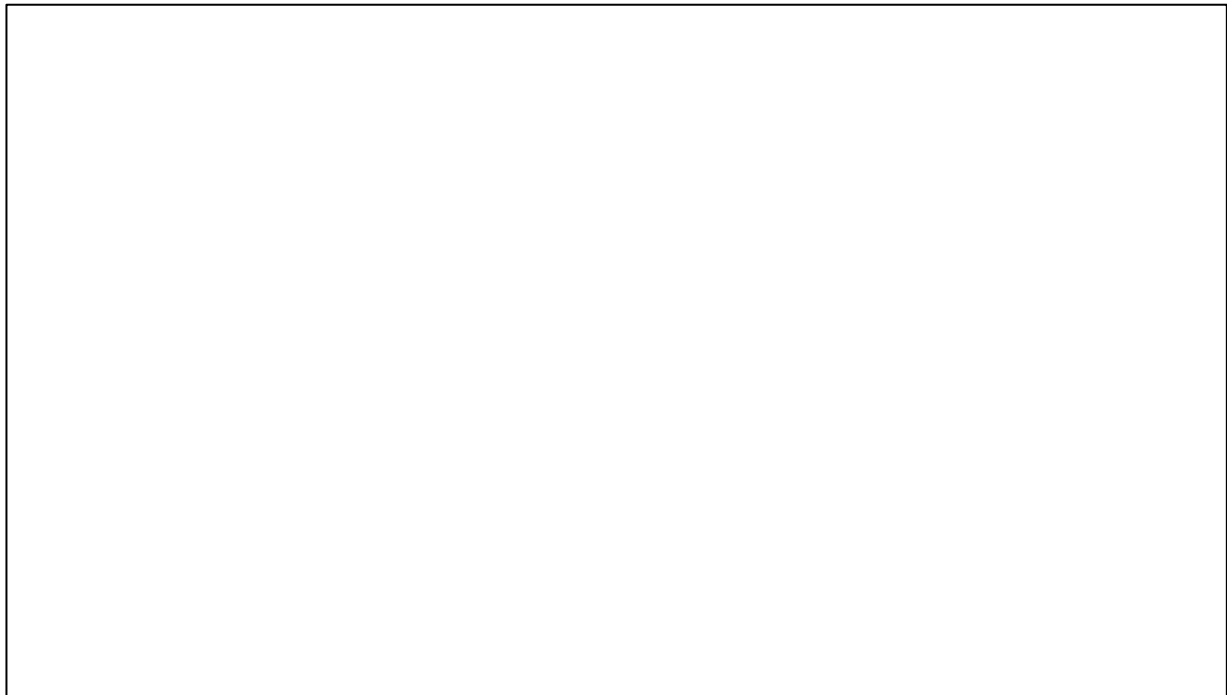
```
    stim_proc: process
begin
        a <= "00";
        b <= "00";
    wait for 100 ns;
        a <= "00";
        b <= "01";
    wait for 100 ns;
        a <= "00";
        b <= "10";
    wait for 100 ns;
        a <= "00";
        b <= "11";
    wait for 100 ns;
```

```
        a <= "01";
        b <= "00";
    wait for 100 ns;
        a <= "01";
        b <= "01";
    wait for 100 ns;
        a <= "01";
        b <= "11";
    wait for 100 ns;
        a <= "11";
        b <= "11";
    wait;
end process;
```

Test Bench Result:

4) Circuit Diagram

VHDL Implementation:

```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;


entity bin2bcd is
    Port ( d : in  STD_LOGIC_VECTOR (5
downto 0);
          result : out
STD_LOGIC_VECTOR (7 downto 0));
end bin2bcd;


architecture Behavioral of bin2bcd is

component shifter is
    Port ( input : in  STD_LOGIC_VECTOR
(3 downto 0);
          output : out
STD_LOGIC_VECTOR (3 downto 0));
end component;

signal sgn:std_logic_vector(5 downto
0);

begin
      result(0) <= d(0);
      result(7) <= '0';

      shifter1:shifter port map(
      input(3) => sgn(3),
      input(2) => sgn(4),
      input(1) => sgn(5),
      input(0) => d(1),
      output => result(4 downto 1));


      shifter2:shifter port map(
      input(3) => sgn(0),
      input(2) => sgn(1),
      input(1) => sgn(2),
      input(0) => d(2),
      output(3) => result(5),
      output(2) => sgn(3),
      output(1) => sgn(4),
      output(0) => sgn(5));


      shifter3:shifter port map(
      input(3) => '0',
      input(2) => d(5),
      input(1) => d(4),
      input(0) => d(3),
      output(3) => result(6),
      output(2) => sgn(0),
      output(1) => sgn(1),
      output(0) => sgn(2));

end Behavioral;
```
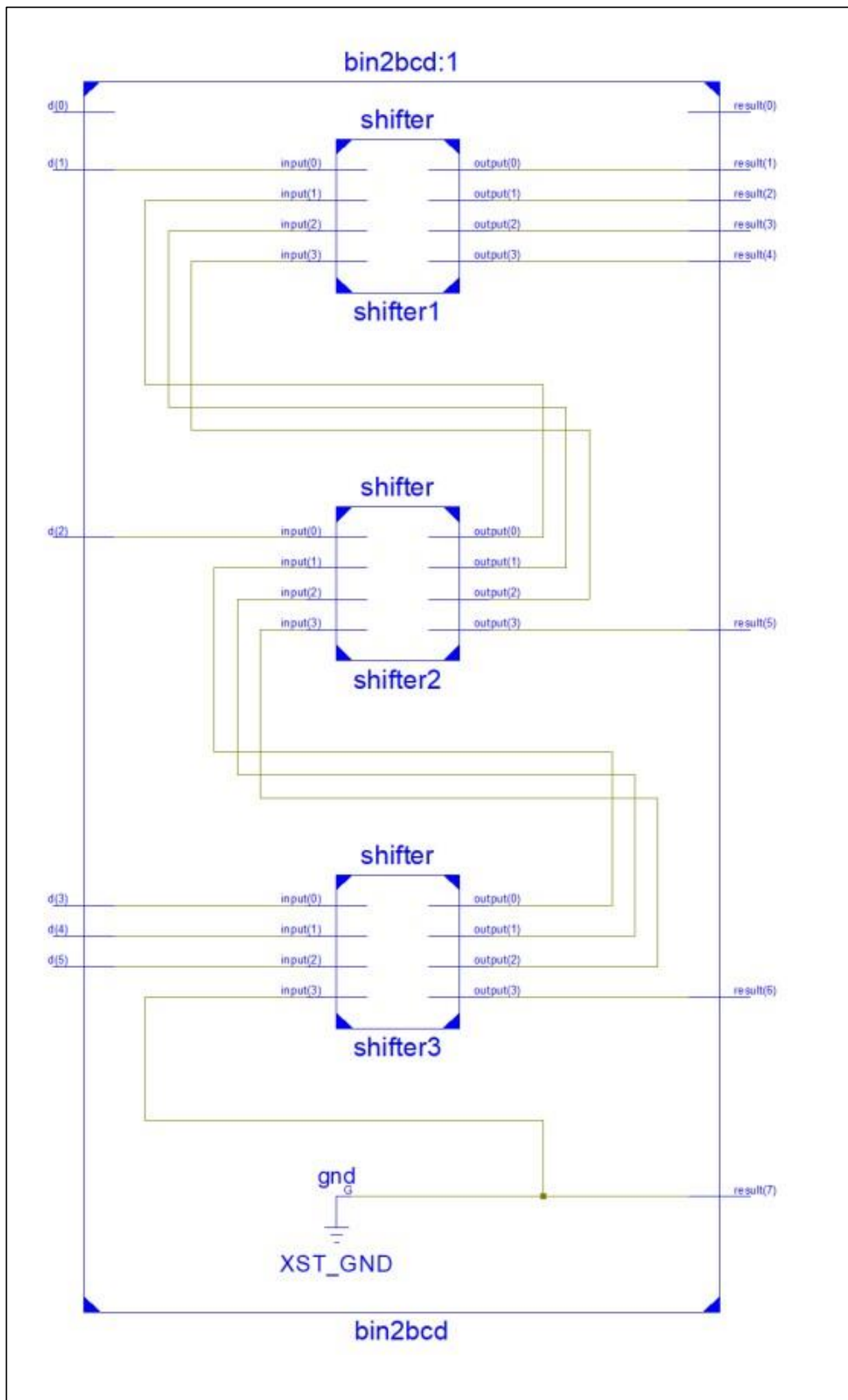
RTL Schematic:

Test Bench Input:

```
stim_proc: process
begin
   d <= "000000";
   wait for 100 ns;
   d <= "000001";
   wait for 100 ns;
   d <= "000100";
   wait for 100 ns;
   d <= "000101";
   wait for 100 ns;
   d <= "001000";
   wait for 100 ns;
   d <= "010000";
   wait for 100 ns;
   d <= "100101";
   wait for 100 ns;
   d <= "111111";
   wait;
end process;
```

Test Bench Result:

	0 ns	100 ns	200 ns	300 ns	400 ns	500 ns	600 ns	700 ns
d[5:0]	0	1	4	5	8	16	37	63
result[7:0]	00000000	00000001	00000100	00000101	00001000	00010110	00110111	01100011