```
1   library IEEE;
2   use IEEE.STD_LOGIC_1164.ALL;
3
4   -- Uncomment the following library declaration if using
5   -- arithmetic functions with Signed or Unsigned values
6   --use IEEE.NUMERIC_STD.ALL;
7
8   -- Uncomment the following library declaration if instantiating
9   -- any Xilinx leaf cells in this code.
10  --library UNISIM;
11  --use UNISIM.VComponents.all;
12
13  entity task1 is
14      Port ( a : in STD_LOGIC;
15             b : in STD_LOGIC;
16             y : out STD_LOGIC);
17  end task1;
18
19
20  architecture Behavioral of task1 is
21
22  begin
23
24  -- Enter your code here.
25      y <= a and b; -- get a and b and store it in y
26
27  end Behavioral;
```

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

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity task1_tb is
--  Port ( );
end task1_tb;

architecture Behavioral of task1_tb is

    --declaring the component
    component task1
        Port (
           a : in STD_LOGIC;
           b : in STD_LOGIC;
           y : out STD_LOGIC);
    end component;

    --declaring the signals needed
    --these a, b, and y signals are different from the
    --internal ones of the component
    signal a, b, y : std_logic;

    --signal to assign values to a and b
    signal counter : unsigned(1 downto 0) := "00";

begin
    -- component assignment
    uut: task1 port map(
        a => a,
        b => b,
        y => y
    );

    --assign a (bit 1) and b (bit 0) to the counter bits so that
    --all possible inputs are tested
    --Enter your code
    a <= counter(0);
    b <= counter(1);

    --increments the counter using a process
    --use a 20ns delay between each combination
    --Enter your code here
    tb : process is
    begin
        wait for 20ns;
        counter <= counter + 1;
    end process tb;

end Behavioral;
```

```
1   library IEEE;
2   use IEEE.STD_LOGIC_1164.ALL;
3
4   -- Uncomment the following library declaration if using
5   -- arithmetic functions with Signed or Unsigned values
6   --use IEEE.NUMERIC_STD.ALL;
7
8   -- Uncomment the following library declaration if instantiating
9   -- any Xilinx leaf cells in this code.
10  --library UNISIM;
11  --use UNISIM.VComponents.all;
12
13  entity task2 is
14      Port (
15             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 task2;
21
22  architecture Behavioral of task2 is
23
24  begin
25
26  -- Enter your code here
27      sum <= a xor b xor cin;
28      cout <= (a and b) or (a and cin) or (b and cin);
29
30  end Behavioral;
```

```
1   library IEEE;
2   use IEEE.STD_LOGIC_1164.ALL;
3
4   -- Uncomment the following library declaration if using
5   -- arithmetic functions with Signed or Unsigned values
6   use IEEE.NUMERIC_STD.ALL;
7
8   -- Uncomment the following library declaration if instantiating
9   -- any Xilinx leaf cells in this code.
10  --library UNISIM;
11  --use UNISIM.VComponents.all;
12
13  entity task2_tb is
14  --  Port ( );
15  end task2_tb;
16
17  architecture Behavioral of task2_tb is
18
19      --declaring the component
20      component task2
21          Port (
22             a   : in STD_LOGIC;
23             b   : in STD_LOGIC;
24             cin : in STD_LOGIC;
25             sum : out STD_LOGIC;
26             cout: out STD_LOGIC);
27      end component;
28
29      --declaring the signals needed
30      --these a, b, cin, sum, cout signals are different from the
31      --internal ones of the component
32      signal a, b, cin, sum, cout : std_logic;
33
34      --signal to iterate over all 3 variables and their possible values
35      signal counter : unsigned(2 downto 0) := "000";
36
37  begin
38      -- component assignment
39      uut: task2 port map(
40          a => a,
41          b => b,
42          cin => cin,
43          sum => sum,
44          cout => cout
45      );
46
47      --assign a, b, cin (order is: a [MSB], b, cin [LSB])to the counter bits so that
48      --all possible inputs are tested
49      --Enter your code here
50      a   <= counter(2);
51      b   <= counter(1);
52      cin <= counter(0);
53
54      tb : process
55      begin
56          wait for 20ns;
57          counter <= counter + 1;
58      end process tb;
59      --increments the counter using a process
60      --use a 20ns delay between each combination
61      --Enter your code here
62
63
64  end Behavioral;
```

```
1   library IEEE;
2   use IEEE.STD_LOGIC_1164.ALL;
3
4   -- Uncomment the following library declaration if using
5   -- arithmetic functions with Signed or Unsigned values
6   --use IEEE.NUMERIC_STD.ALL;
7
8   -- Uncomment the following library declaration if instantiating
9   -- any Xilinx leaf cells in this code.
10  --library UNISIM;
11  --use UNISIM.VComponents.all;
12
13  entity encoder is
14      Port (
15            hex_in : in STD_LOGIC_VECTOR(3 DOWNTO 0);
16            a : out STD_LOGIC;
17            b : out STD_LOGIC;
18            c : out STD_LOGIC;
19            d : out STD_LOGIC;
20            e : out STD_LOGIC;
21            f : out STD_LOGIC;
22            g : out STD_LOGIC
23      );
24  end encoder;
25
26  architecture Behavioral of encoder is
27
28      -- temporary signal to make our assignment of values (a through g) simpler
29      signal seven_seg    : std_logic_vector(6 downto 0);
30
31  begin
32
33      --seven_seg <= -- Enter your code to assign values for the entire 7 bits (a through
        g) based on the hex input
34      process(hex_in)
35      begin
36          case hex_in is
37              when "0000" => seven_seg <= "1111110";
38              when "0001" => seven_seg <= "0110000";
39              when "0010" => seven_seg <= "1101101";
40              when "0011" => seven_seg <= "1111001";
41
42              when "0100" => seven_seg <= "0110011";
43              when "0101" => seven_seg <= "1011011";
44              when "0110" => seven_seg <= "1011111";
45              when "0111" => seven_seg <= "1110000";
46
47              when "1000" => seven_seg <= "1111111";
48              when "1001" => seven_seg <= "1111011";
49              when "1010" => seven_seg <= "1110111";
50              when "1011" => seven_seg <= "0011111";
51
52              when "1100" => seven_seg <= "1001110";
53              when "1101" => seven_seg <= "0111101";
54              when "1110" => seven_seg <= "1001111";
55              when "1111" => seven_seg <= "1000111";
56
57              when others => seven_seg <= "0000000";
58
59          end case;
60      end process;
61      -- Extract each individual bit and assign it to the 7 outputs
62      a <= seven_seg(6);
63      b <= seven_seg(5);
64      c <= seven_seg(4);
65      d <= seven_seg(3);
```

```
66          e <= seven_seg(2);
67          f <= seven_seg(1);
68          g <= seven_seg(0);
69
70      end Behavioral;
71
```

```
1   library IEEE;
2   use IEEE.STD_LOGIC_1164.ALL;
3
4   -- Uncomment the following library declaration if using
5   -- arithmetic functions with Signed or Unsigned values
6   use IEEE.NUMERIC_STD.ALL;
7
8   -- Uncomment the following library declaration if instantiating
9   -- any Xilinx leaf cells in this code.
10  --library UNISIM;
11  --use UNISIM.VComponents.all;
12
13  entity encoder_tb is
14  --    Port ( );
15  end encoder_tb;
16
17  architecture Behavioral of encoder_tb is
18
19      component encoder
20          Port (
21                hex_in : in STD_LOGIC_VECTOR(3 DOWNTO 0);
22                a : out STD_LOGIC;
23                b : out STD_LOGIC;
24                c : out STD_LOGIC;
25                d : out STD_LOGIC;
26                e : out STD_LOGIC;
27                f : out STD_LOGIC;
28                g : out STD_LOGIC
29          );
30      end component;
31
32      signal counter      : unsigned(3 downto 0):="0000";
33      signal hex_in       : std_logic_vector(3 downto 0);
34      signal a, b, c, d   : std_logic;
35      signal e, f, g      : std_logic;
36
37  begin
38      uut: encoder port map(
39          hex_in => hex_in,
40          a   => a,
41          b   => b,
42          c   => c,
43          d   => d,
44          e   => e,
45          f   => f,
46          g   => g
47      );
48
49      hex_in <= std_logic_vector(counter);
50
51      --increments the counter using a process
52      --use a 20ns delay between each combination
53      --Enter your code here
54      tb2 : process
55      begin
56            wait for 20ns;
57            counter <= counter + 1;
58      end process tb2;
59
60  end Behavioral;
61
```