Saif Ba Madhaf - 527028

VHDL Week 2

self-test chapter 3:

1. Given:

```
ARCHITECTURE x OF gate_network IS
BEGIN
-- concurrent assignment statements operate in parallel
-- D(1) selects bit 1 of standard logic vector D
X <= A AND NOT(B OR C) AND (D(1) XOR D(2));
-- Process must declare a sensitivity list
-- Sensitivity list includes all signals which can change the outputs
PROCESS (A,B,C,D)
BEGIN
-- Statements inside a process execute in sequential order
Y <= A AND NOT(B OR C) AND (D(1) XOR D(2));
END PROCESS;
END x;
```

Is this a structural, behavioral or RTL description?

It has both a structural and behavioral description. The line *Y <= A AND NOT(B OR C) AND (D(1) XOR D(2));* describes a structure (connection between components). The line *PROCESS (A,B,C,D)*

Describes a behavior of a circuit.

2. What is the purpose of a generic?

The purpose of a generic in VHDL is like having a variable to store logic within it to use later. It is basically like having components stored in a veariable.

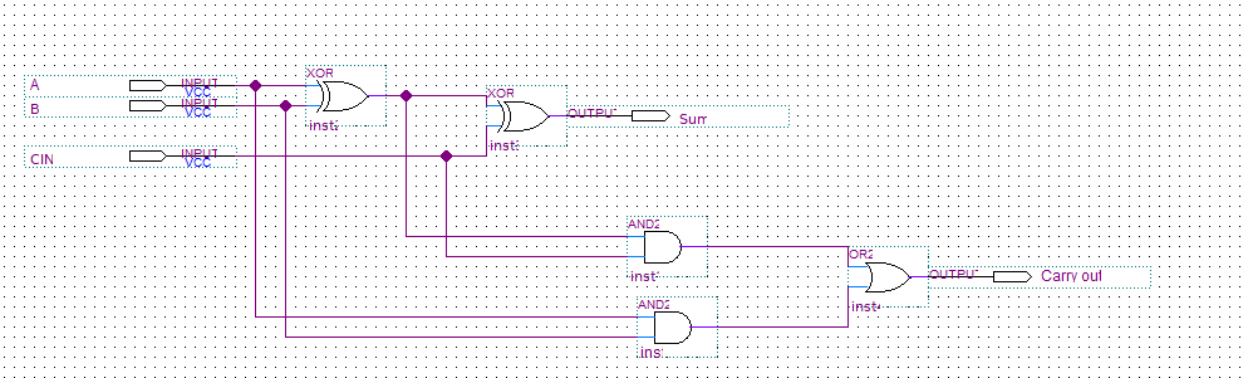
3. What is the purpose of a Configuration Statement?

The Configuration Statement in VHDL gives the ability to configure components created in entities.

4. In our tutorial there is an adder, is this a structural, behavioral or RTL description?

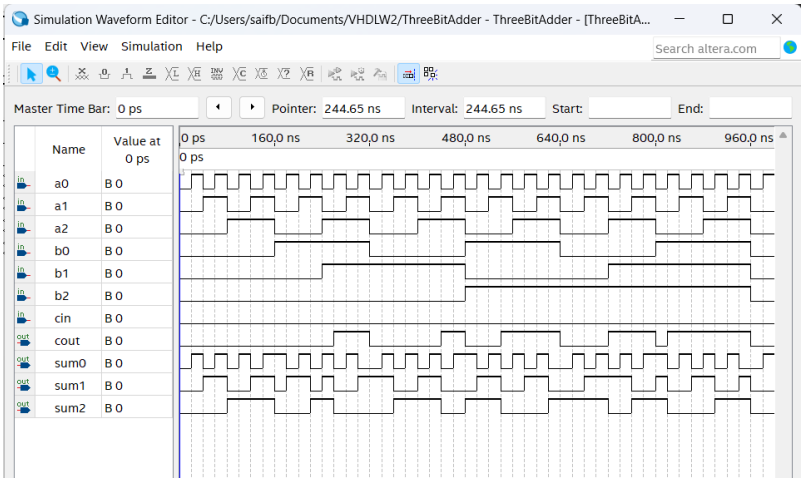
In the tutorial it was a behavioral description as it was describing the behavior of the adder.

5. Draw the schematics of a full adder.



Exercise 3:

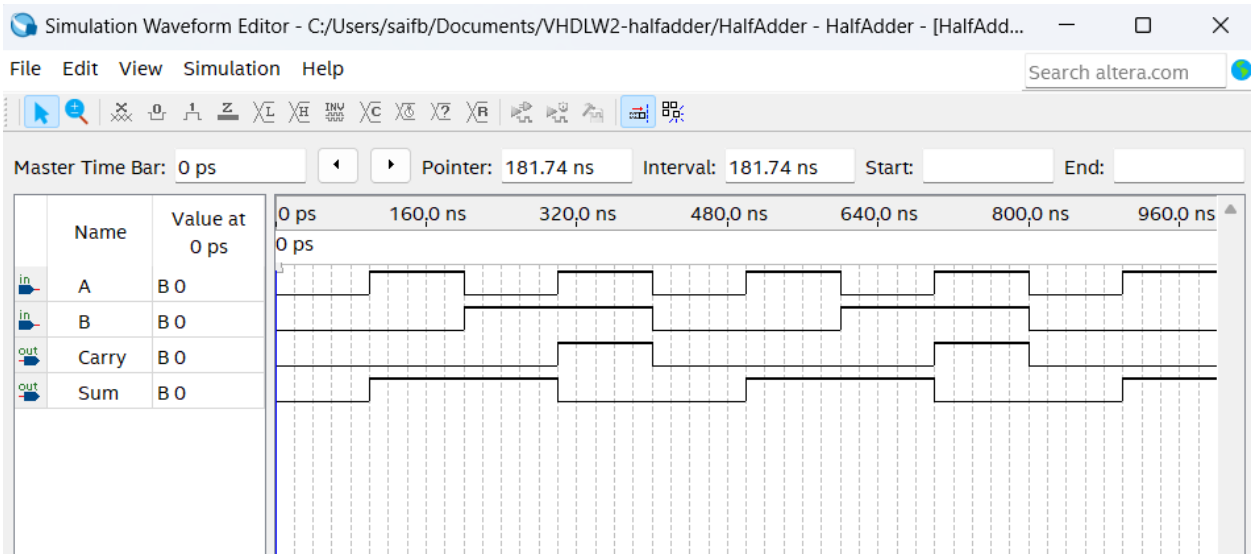
1. Simulate in Quartus the behavioral VHDL of a 3-bit adder.



2. FPGA

In this example we can see that our a0 is s0 in the fpga and our b0 is s3 and when both are one the sum 0 (led0).

3. Simulate in Quartus a structural description of a half adder.



This is the half adder simulation in Quartus. In the picture above we can see all possible outcomes. As we know a half adder adds two binary inputs in our case A, B, and it the produces 2 outputs our sum and carry out. An example of adding two binary inputs would be adding two ones:

1 1(carry out)

1

-

0 (sum)

Adding two ones in binary is zero with a carry out 1.

Code

```
entity HalfAdder is
    Port (
        A : in STD_LOGIC;
        B : in STD_LOGIC;
        Sum : out STD_LOGIC;
        Carry : out STD_LOGIC
    );
end HalfAdder;

architecture Structural of HalfAdder is
    component XOR1 is
        Port (
            X : in STD_LOGIC;
            Y : in STD_LOGIC;
            Z : out STD_LOGIC
        );
    end component;

    component AND1 is
        Port (
            X : in STD_LOGIC;
            Y : in STD_LOGIC;
            Z : out STD_LOGIC
        );
    end component;

    signal S : std_logic;
    signal C : std_logic;
begin
    gate1: XOR1 port map(A, B, S);
    gate2: AND1 port map(A, B, C);

    Sum <= S;
    Carry <= C;
end Structural;
```

self-test chapter 4:

1. given:

ARCHITECTURE behavior OF gate_network IS

BEGIN

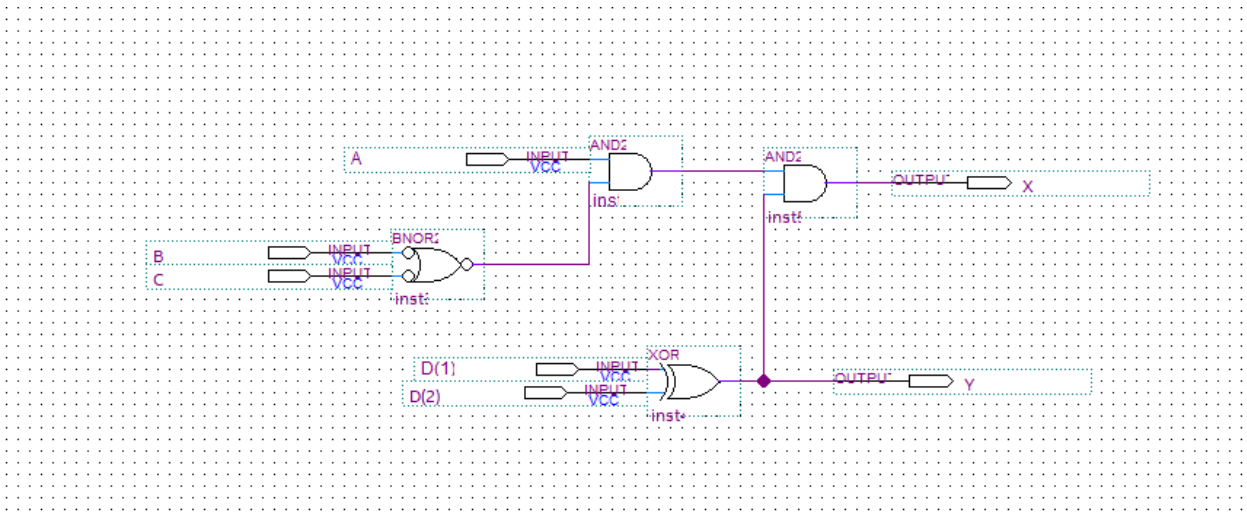
X <= A AND NOT(B OR C) AND (D(1) XOR D(2));

Y <= (D(1) XOR D(2));

Which value, X or Y is first evaluated?

In Vhdl the execution of signals is done at the same time that is why non is evaluated first.

2. Draw the schematics of the VHDL description in question 1.



3. Give the behavioral description of a 2 to 4 decoder. (use the knowledge of the module digital 1)

```
    PORT (
       a, b : IN std_logic;
       d0, d1, d2, d3 : OUT std_logic
    );
END TwoToFourDecoder;

ARCHITECTURE behavior OF TwoToFourDecoder IS

BEGIN

    process(a, b)
    begin
        if (a = '0' and b = '0') then
            d0 <= '1';
            d1 <= '0';
            d2 <= '0';
            d3 <= '0';
        elsif (a = '0' and b = '1') then
            d0 <= '0';
            d1 <= '1';
            d2 <= '0';
            d3 <= '0';
        elsif (a = '1' and b = '0') then
            d0 <= '0';
            d1 <= '0';
            d2 <= '1';
            d3 <= '0';
        elsif (a = '1' and b = '1') then
            d0 <= '0';
            d1 <= '0';
            d2 <= '0';
            d3 <= '1';
        else
            d0 <= '0';
            d1 <= '0';
            d2 <= '0';
            d3 <= '0';
        end if;
    end process;
```

Exercise 4: TwoToFourDecoder

1. **Simulate the behavioral description of a 2 to 4 decoder in Quartus.**

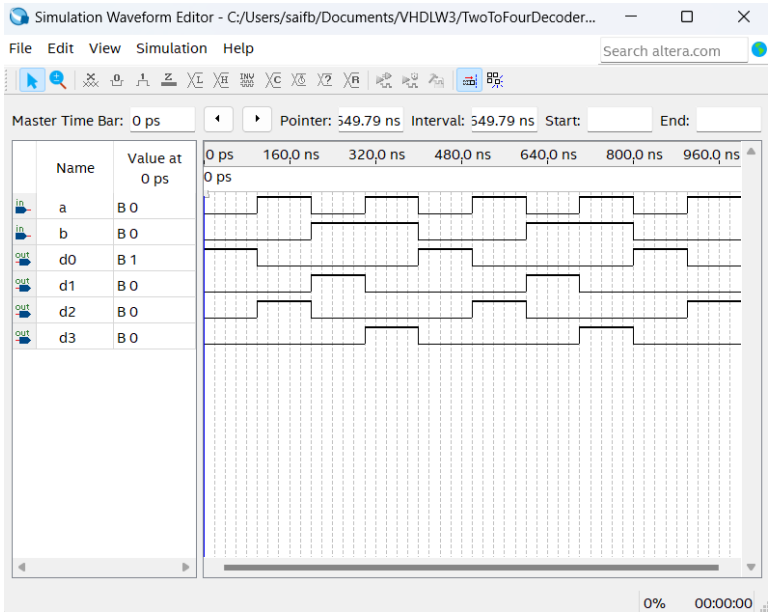
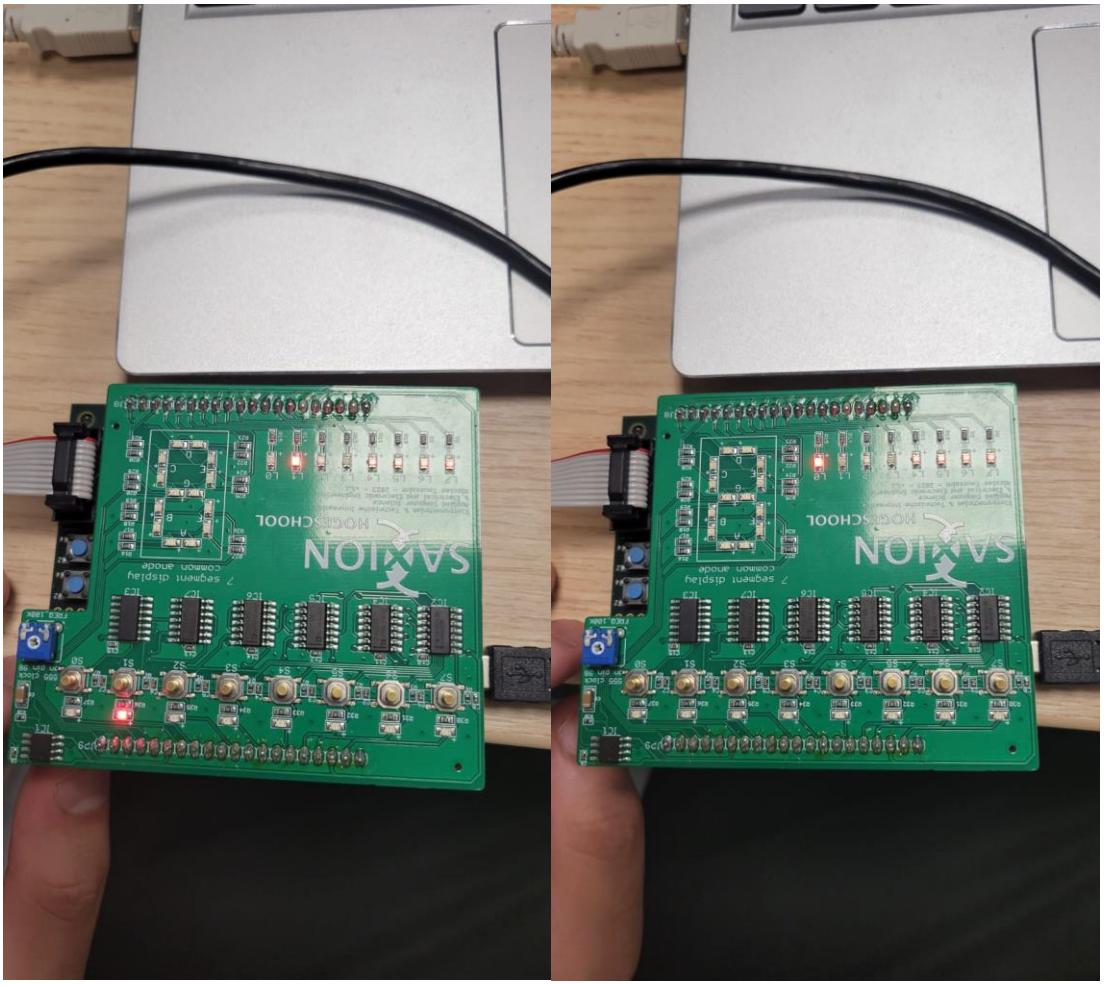


Table TwoToFourDecoder

A	B	D0	D1	D2	D3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

2. FPGA implementation



Here are two examples of the 2 to 4 decoders as we can see a(s0) and b(s1) and Led (l0-l3) are our outputs. Look at the table for reference on how it works.

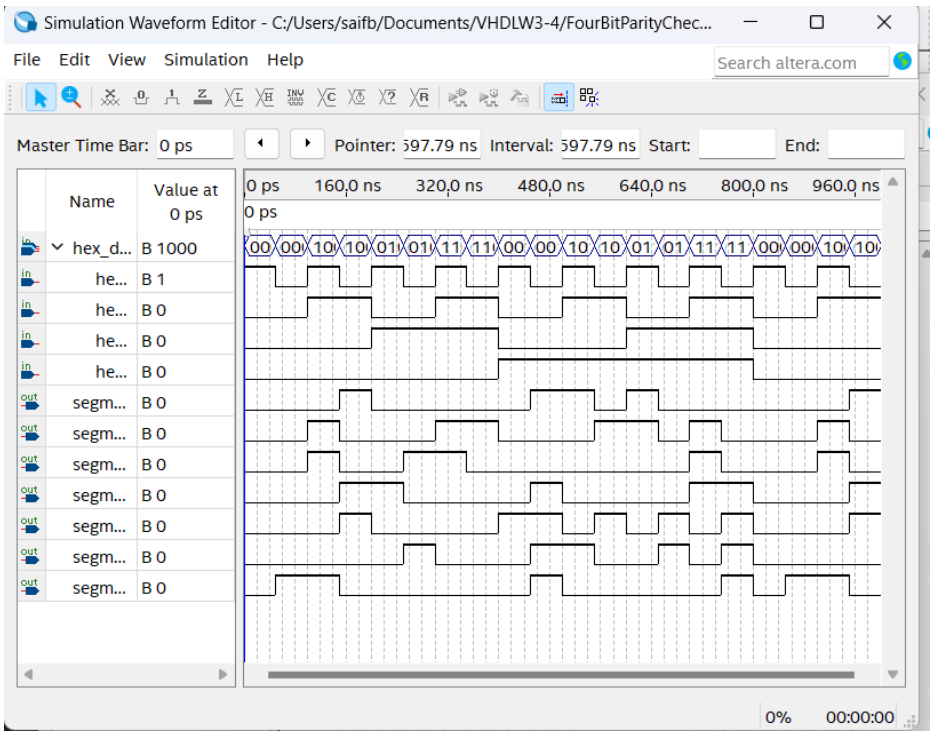
self-test chapter 5:

1. What does a concurrent assignment mean?

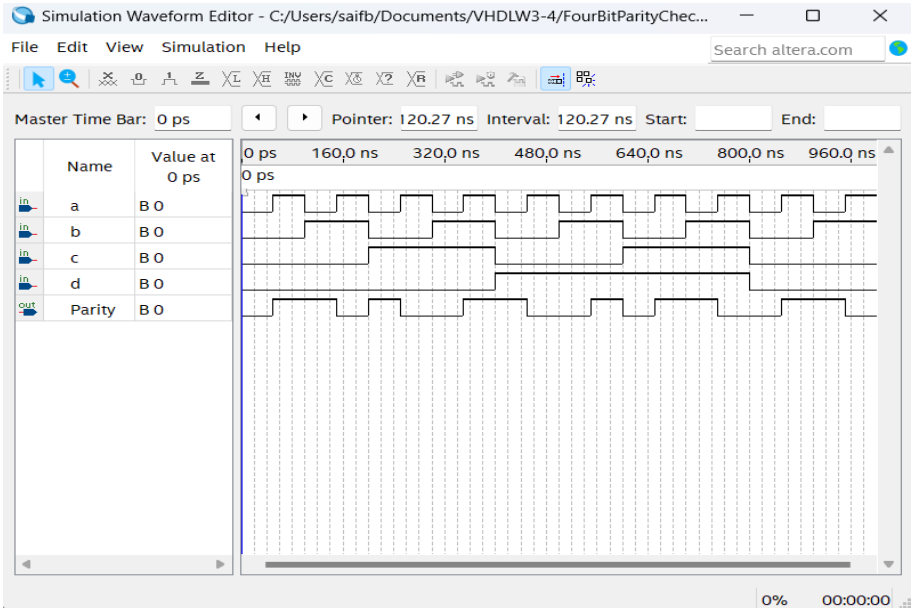
It is when operations are executed in parallel/ at the same time.

Exercise 5:

1. **Simulate the 7 segments display model.**



2. Design a 4 bit 'even parity checker' (structured) and simulate it.

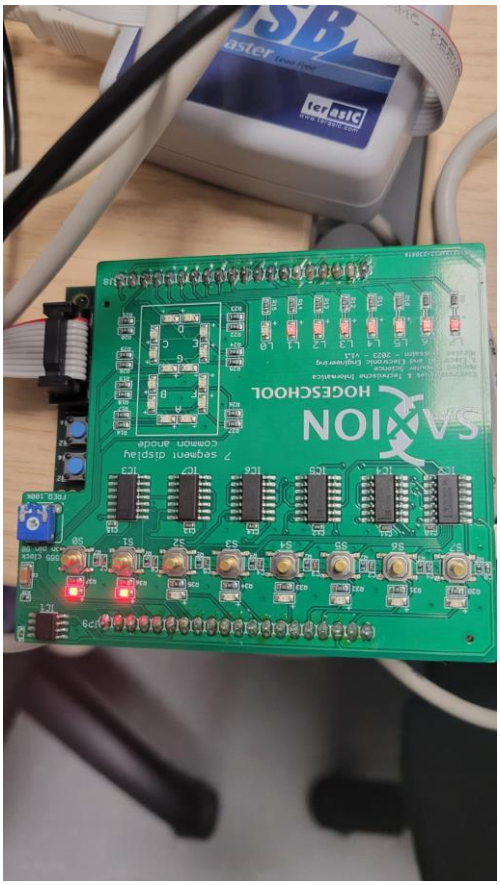


Even Parity checker is on 1 only to make the four inputs even for example if a is 1 and the rest is 0 the parity checker will be one to make it even and when two inputs are on it will be 0 to keep it even.

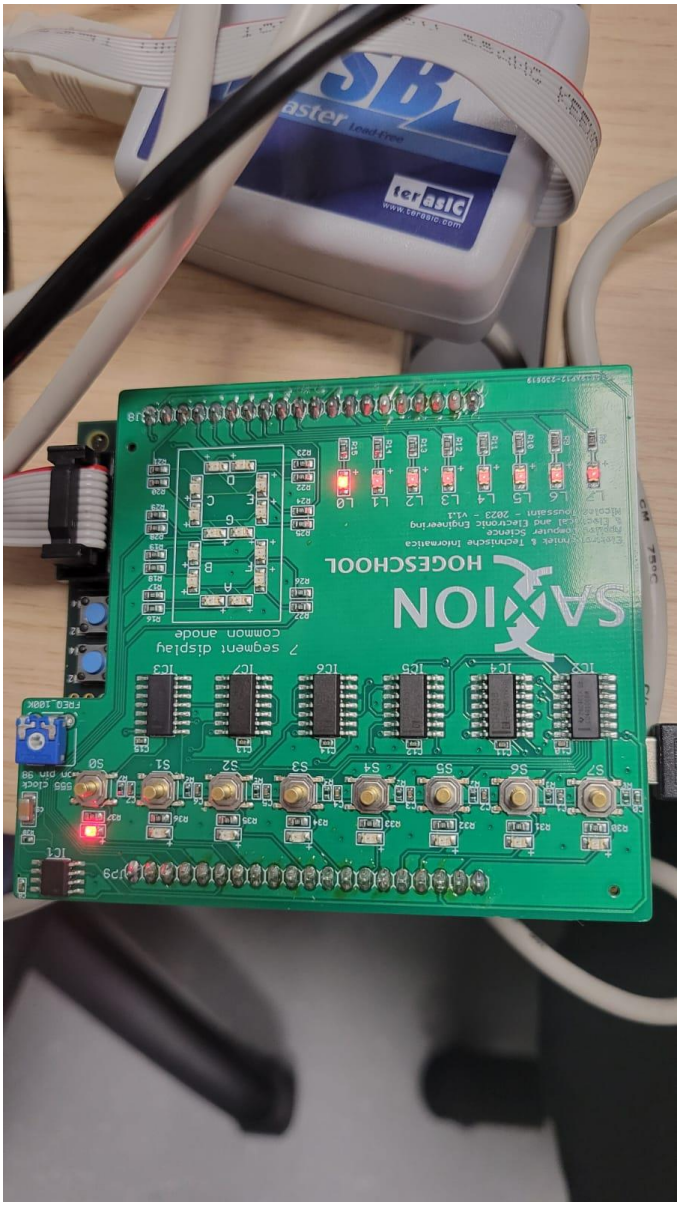
CODE

```
architecture Structural of FourBitParityCheckerEven is

    component XOR2 is
        Port (
            X : in STD_LOGIC;
            Y : in STD_LOGIC;
            Z : out STD_LOGIC
        );
    end component;

    signal c1, c2, c3 : STD_LOGIC;

begin

    xor11: XOR2 port map(a, b, c1);
    xor22: XOR2 port map(c, d, c2);
    xor33: XOR2 port map(c1, c2, Parity);


end Structural;
```

3. FPGA implementation



In this example we can see that when our a (s0) b(s1) are on the parity (led0) is off since the the parity led only turns on when the 4 bits a,b,c,d are odd for example if only a(s0) is on which is shown in the next image.


terasic
www.terasic.com
SAXION
HOGESCHOOL
7 segment display
common anode