EX. NO: 07

DATE : 24/10/2024

Comparators, Parity Generator and ALU

AIM: To verify the functionalities of Comparators, Parity Generator and ALU using Verilog HDL program.

SOFTWARE USED: Xilinx Vivado

HARDWARE USED: Basys3 FPGA Board

COMPARATOR:

Verilog HDL Code for 4 bit comparator:

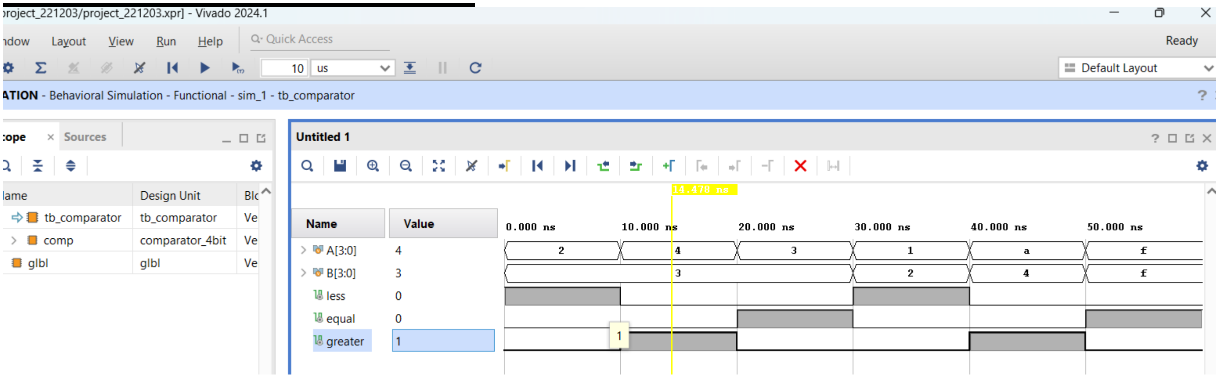
```
module comparator_4bit(A, B, less, equal, greater);
    input [3:0] A, B;
    output reg less, equal, greater;
    wire less_1, equal_1, greater_1;
    wire less_0, equal_0, greater_0;
    comparator comp1(
        .A(A[1:0]),
        .B(B[1:0]),
        .less(less_0),
        .equal(equal_0),
        .greater(greater_0)
    );
    comparator comp2(
        .A(A[3:2]),
        .B(B[3:2]),
        .less(less_1),
        .equal(equal_1),
        .greater(greater_1)
    );
 always @(*) begin
        if (A[3:2] > B[3:2]) begin
            less = 0; equal = 0; greater = 1;end
        else if (A[3:2] < B[3:2]) begin
            less = 1; equal = 0; greater = 0;end
        else begin
            less = less_0;
            equal = equal_0;
            greater = greater_0;
        end
    end
endmodule
module comparator(A,B, less, equal, greater);
          input [1:0]A,B;
```

```
     output reg less, equal, greater;
always@(A or B)
     begin
     if (A>B)
     begin
     less = 0;equal = 0;greater = 1 ;
     end
  else if (A<B)
    begin
    less = 1;equal = 0;greater = 0 ;
    end
  else
    begin
    less = 0;equal = 1;greater = 0 ;
    end
end
endmodule
```

Test bench for 4 bit Comparator:

```
module tb_comparator;
  reg [3:0] A, B;
  wire less, equal, greater;
  comparator_4bit comp(
    .A(A),
    .B(B),
    .less(less),
    .equal(equal),
    .greater(greater)
  );
  initial begin
    A = 4'b0010; B = 4'b0011; #10;
    A = 4'b0100; B = 4'b0011; #10;
    A = 4'b0011; B = 4'b0011; #10;
    A = 4'b0001; B = 4'b0010; #10;
    A = 4'b1010; B = 4'b0100; #10;
    A = 4'b1111; B = 4'b1111; #10;
    $finish;
  end
 initial begin
  $monitor("A = %b, B = %b -> less: %b, equal: %b, greater: %b", A, B, less,
equal, greater);
  end
endmodule
```

SIMULATION WAVEFORM:



roject_221203/project_221203.xpr] - Vivado 2024.1
ndow
Layout
View
Run
Help
Quick Access
Ready
10
us
Default Layout
ATION - Behavioral Simulation - Functional - sim_1 - tb_comparator
ope
Sources
Untitled 1
14.410 ns

lame	Design Unit	Blc
tb_comparator	tb_comparator	Ve
comp	comparator_4bit	Ve
glbl	glbl	Ve

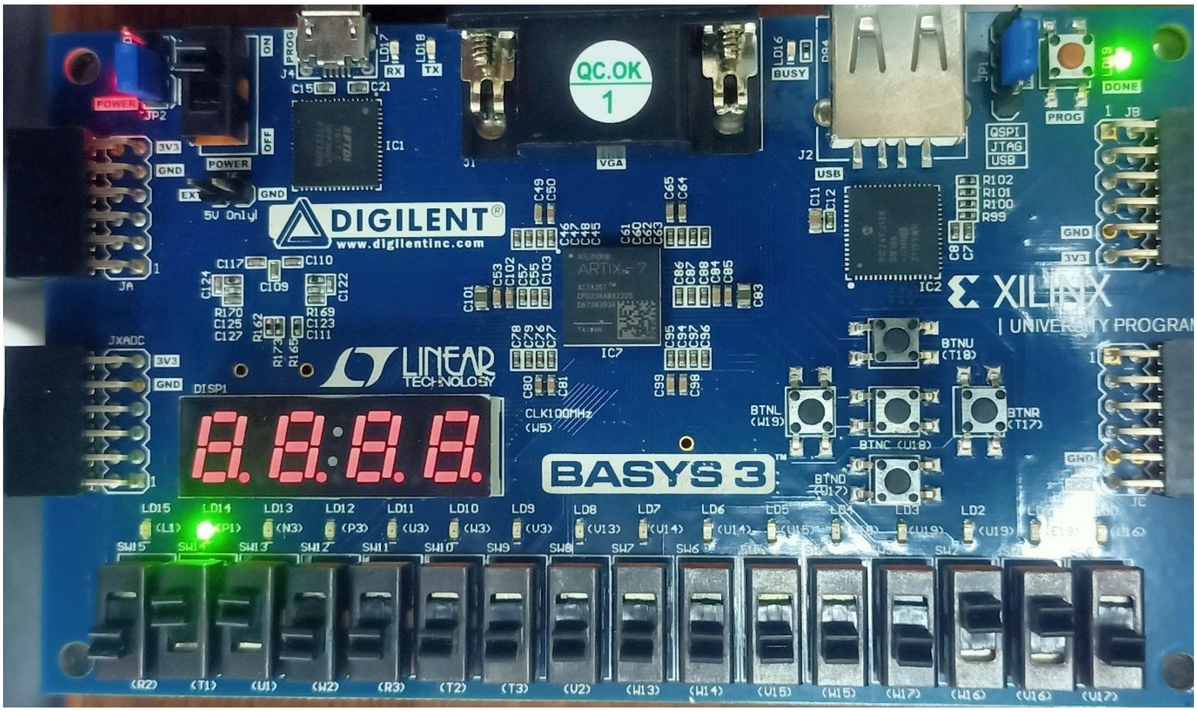
Name	Value
A[3:0]	4
B[3:0]	3
less	0
equal	0
greater	1

HARDWARE OUTPUT:
I/O ports:

Inputs:		**Outputs:**
A[3]: R2	B[3]: W17	equal: P1
A[2]: T1	B[2]: W16	less: L1
A[1]: U1	B[1]: V16	greater: N3
A[0]: W2	B[0]: V17	

For
A[3]=0, A[2]= 1, A[1]=1, A[0]=0
B[3]=0, B[2]= 1, B[1]=1, B[0]=0

Output:
equal=1, less=0, greater=0



TRUTH TABLE FOR 2 BIT COMARATOR:

A	B	Less	Greater	Equal
00	**00**	**0**	**0**	**1**
00	**01**	**1**	0	0
00	**10**	**1**	0	0
00	**11**	**1**	0	0
01	**00**	0	**1**	0
01	**01**	0	0	**1**
01	**10**	**1**	0	0
01	**11**	**1**	0	0
10	**00**	0	**1**	0
10	**01**	0	**1**	0
10	**10**	0	0	**1**
10	**11**	**1**	0	0
11	**00**	0	**1**	0
11	**01**	0	**1**	0
11	**10**	0	**1**	0
11	**11**	0	0	**1**

BLOCK DIAGRAM:

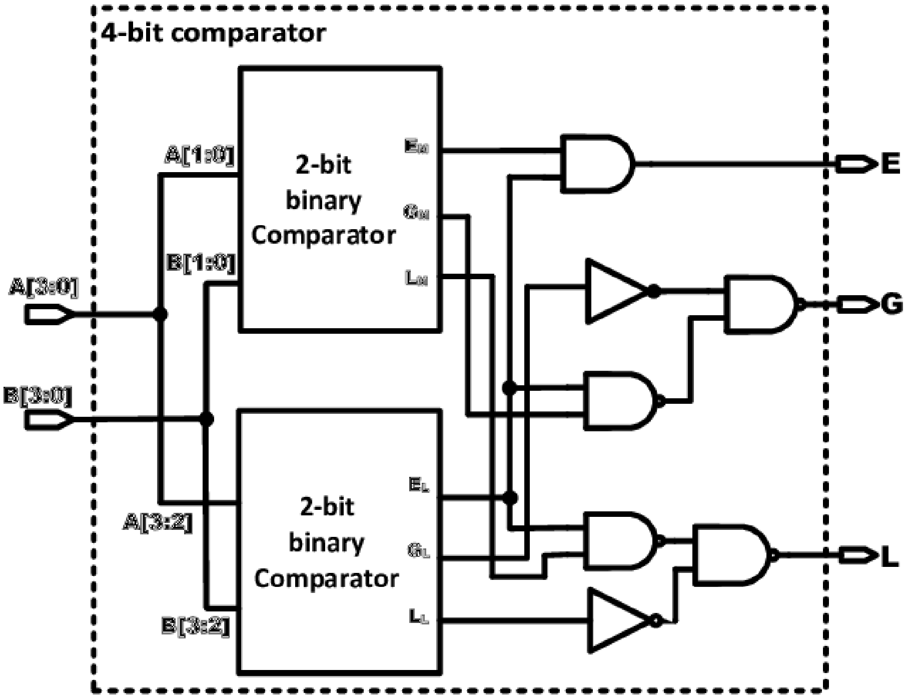


Fig. 3: The architecture for 4-bit binary comparator usi a 2-bit elementary binary comparator.

PARITY GENERATOR:

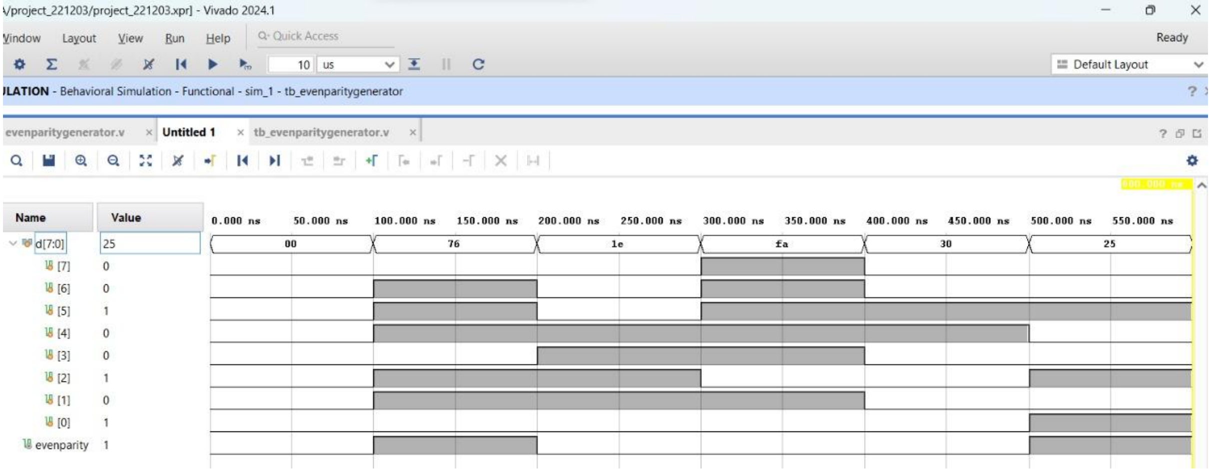
Verilog HDL Code:

```
module evenparitygenerator(d, evenparity);
input [7:0]d;
output reg evenparity;
always@(d)
        begin
                evenparity=^d;
        end
endmodule
```

Test bench for Parity Generator:

```
module tb_evenparitygenerator;
reg [7:0]d;
wire evenparity;
evenparitygenerator EVEN(d, evenparity);
initial
   begin
        d=8'b00000000;
        #100 d=8'b01110110;
        #100 d=8'b00011110;
        #100 d=8'b11111010;
        #100 d=8'b00110000;
        #100 d=8'b00100101;
        #100 $finish;
 end
initial begin
$monitor("Time: %0t |d: %b | evenparity: %b", $time,d,evenparity);
end
endmodule
```

SIMULATION WAVEFORM:



<u>HARDWARE OUTPUT</u>:

<u>I/O ports</u>:

Inputs:

d[7]: R2
d[6]: T1
d[5]: U1
d[4]: W2
d[3]: R3
d[2]: T2
d[1]: T3
d[0]: V2

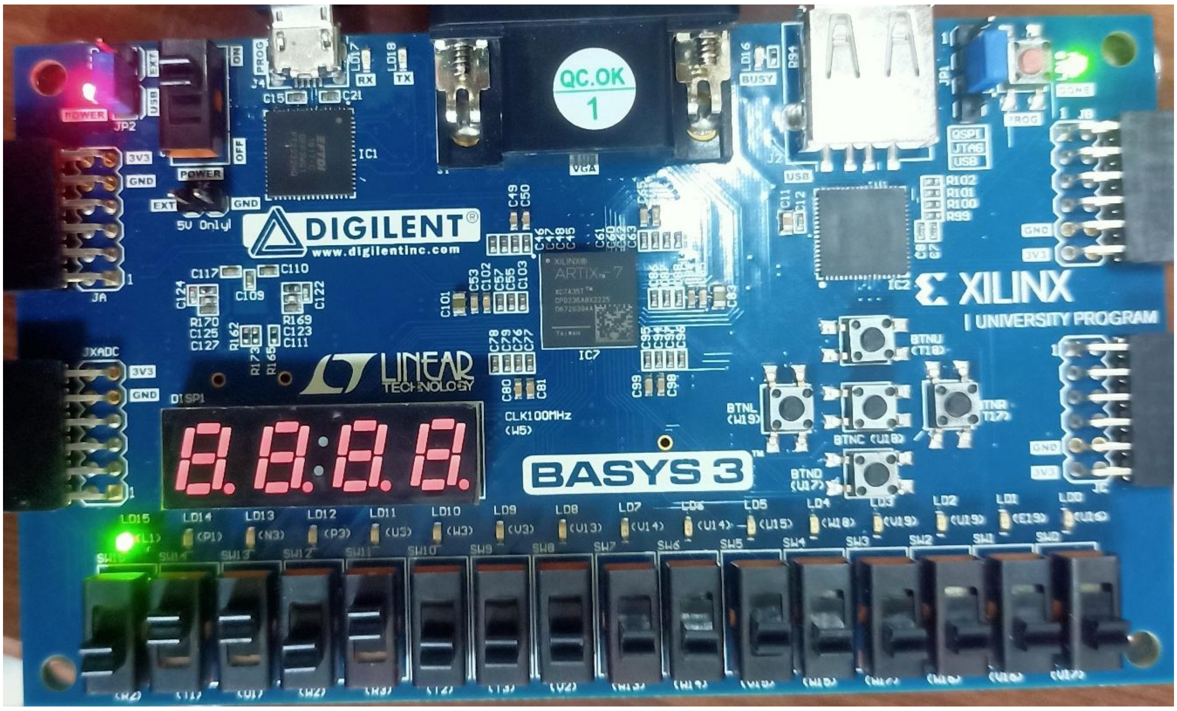
Outputs:

evenparity: L1

<u>For</u>

d[7]=0, d[6]= 1, d[5]=1, d[4]=0
d[3]=0, d[2]= 1, d[1]=1, d[0]=0

output: evenparity= 1



TRUTH TABLE:

Parity Generator: even

A	B	C	Paritybit
0	0	0	0
0	0	1	1
0	1	1	0

CIRCUIT DIAGRAM:

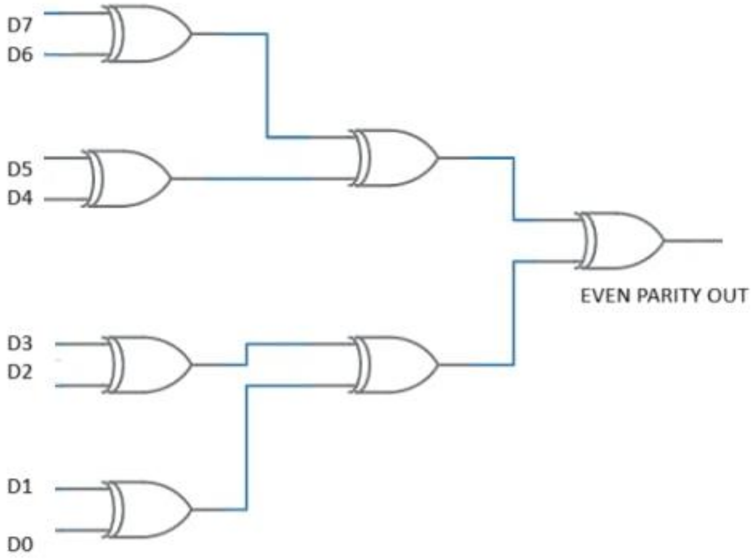


Fig. even parity generator

ALU:

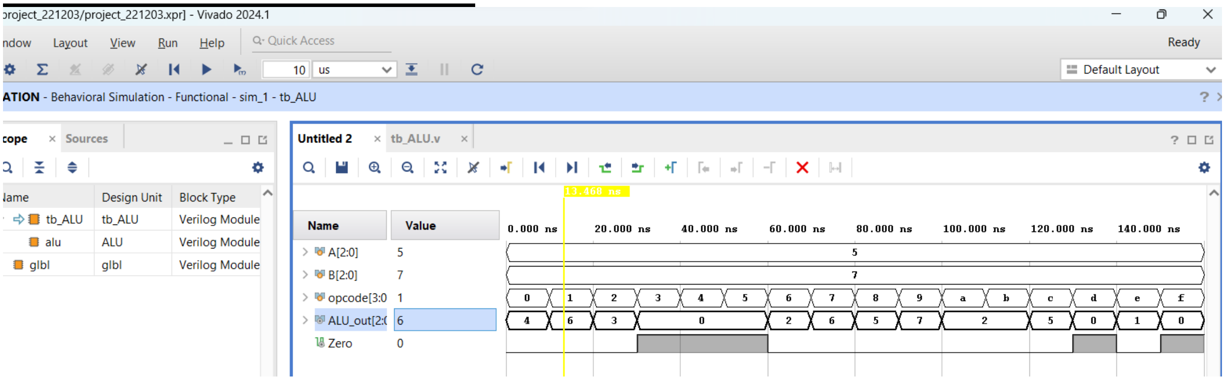
Verilog HDL Code:

```
module ALU (
   input [2:0] A,
   input [2:0] B,
   input [3:0] opcode,
   output reg [2:0] ALU_out,
   output reg Zero
);
 always @(*) begin
      case (opcode)
         4'b0000: ALU_out = A + B;
         4'b0001: ALU_out = A - B;
         4'b0010: ALU_out = A * B;
         4'b0011: ALU_out = A / B;
         4'b0100: ALU_out = A << B;
         4'b0101: ALU_out = A >> B;
         4'b0110: ALU_out = A << 1;
         4'b0111: ALU_out = {A[0], A[2:1]};
         4'b1000: ALU_out = A & B;
         4'b1001: ALU_out = A | B;
         4'b1010: ALU_out = A ^ B;
         4'b1011: ALU_out = ~(A & B);
         4'b1100: ALU_out = ~(A ^ B);
         4'b1101: ALU_out = (A > B) ? 1 : 0;
         4'b1110: ALU_out = (A < B) ? 1 : 0;
         4'b1111: ALU_out = (A == B) ? 1 : 0;
         default: ALU_out = 3'b0;
      endcase
      Zero = (ALU_out == 3'b0) ? 1 : 0;
end
endmodule
```

Test bench for ALU:

```
module tb_ALU;
   reg [2:0] A;
   reg [2:0] B;
   reg [3:0] opcode;
   wire [2:0] ALU_out;
   wire Zero;
   ALU  alu(
      .A(A),
      .B(B),
      .opcode(opcode),
      .ALU_out(ALU_out),
        .Zero(Zero)
    );
initial begin
        A = 3'b101; B = 3'b111;
        opcode = 4'b0000; #10
        opcode = 4'b0001; #10
        opcode = 4'b0010; #10
        opcode = 4'b0011; #10
        opcode = 4'b0100; #10
        opcode = 4'b0101; #10
        opcode = 4'b0110; #10
        opcode = 4'b0111; #10
        opcode = 4'b1000; #10
        opcode = 4'b1001; #10
        opcode = 4'b1010; #10
        opcode = 4'b1011; #10
        opcode = 4'b1100; #10
        opcode = 4'b1101; #10
        opcode = 4'b1110; #10
        opcode = 4'b1111; #10
        $stop;
        end
initial begin
        $monitor("Time: %0t | A: %b | B: %b | Opcode: %b | ALU_out: %b | Zero:
                                        %b", $time, A, B, opcode, ALU_out, Zero);
    end
endmodule
```

SIMULATION WAVEFORM:



roject_221203/project_221203.xpr] - Vivado 2024.1

ATION - Behavioral Simulation - Functional - sim_1 - tb_ALU

lame	Design Unit	Block Type
tb_ALU	tb_ALU	Verilog Module
alu	ALU	Verilog Module
glbl	glbl	Verilog Module

Name	Value
A[2:0]	5
B[2:0]	7
opcode[3:0	1
ALU_out[2:(	6
Zero	0

HARDWARE OUTPUT:

I/O ports:

Inputs:

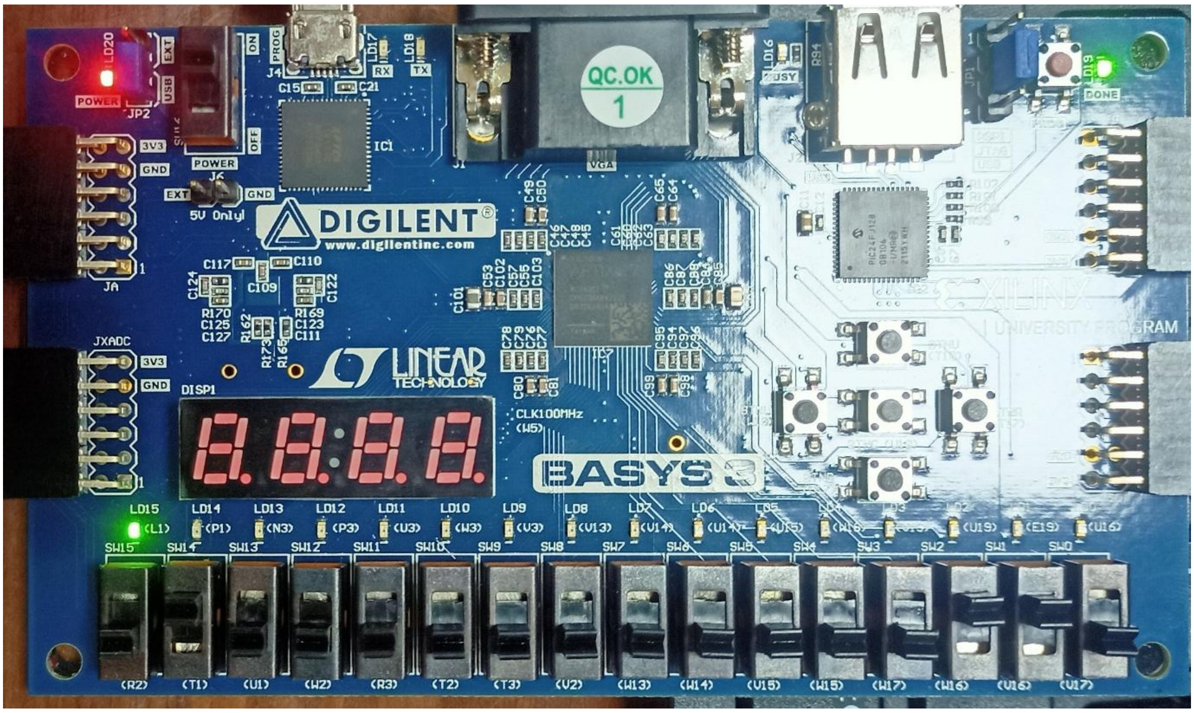
A[2]: R2
A[1]: T1
A[0]: U1
B[2]: W2
B[1]: R3
B[0]: T2

opcode [3]: W17
opcode [2]: W16
opcode [1]: V16
opcode [0]: V17

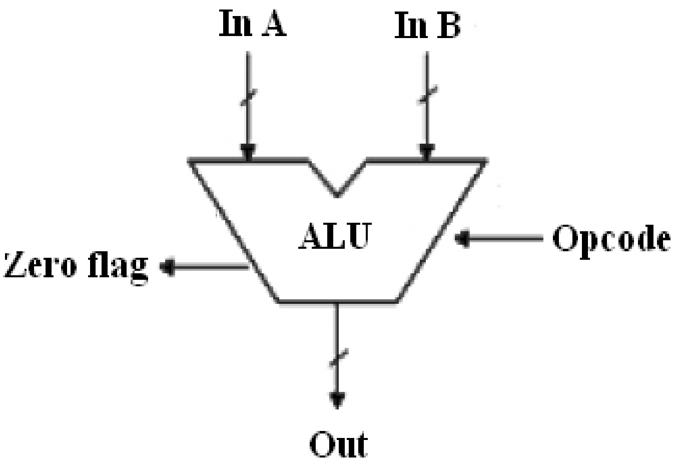
Outputs:

ALU_out [3]: L1
ALU_out [2]: P1
ALU_out [1]: N3
ALU_out [0]:
Zero: U16

For
opcode=0110 , A = 0100 , B =0000
Output:
ALU_out= 1000   (A<<1) i.e. A= 1000



BLOCK DIAGRAM:



RESULT:

The Comparators, Parity Generator and ALU has been successfully designed and implemented using Verilog HDL, and their functionalities have been thoroughly verified through simulation with Xilinx Vivado software on the Basys3 FPGA board.