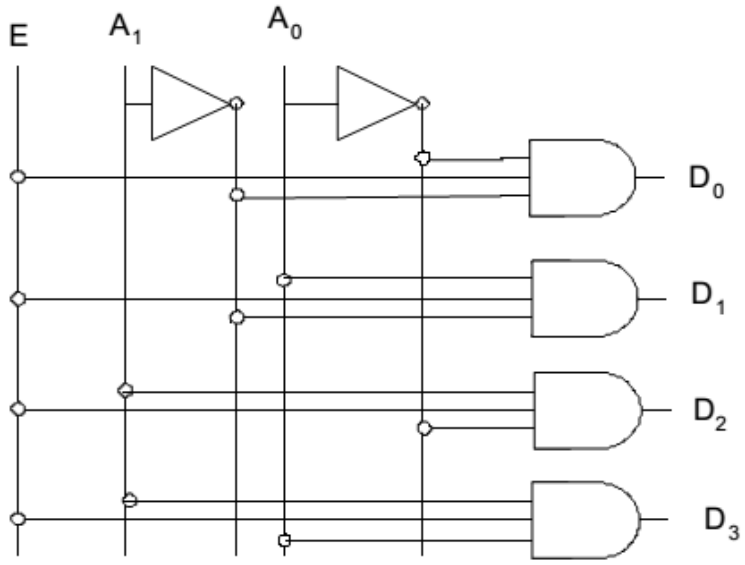
# Experiment 4

**Aim:** Write Verilog code to create a 2x4 Decoder using Behavioural Modeling and implement a 4x16 Decoder using 2x4 Decoder.

**Theory:**

A 2x4 decoder is a combinational digital circuit that has two inputs (A and B) and four outputs (Y0, Y1, Y2, and Y3). It is used to select one of the four output lines based on the input values. When the input values are 00, Y0 is active, when the input is 01, Y1 is active, and so on.




Implementation 2-to-4 decoder with enable

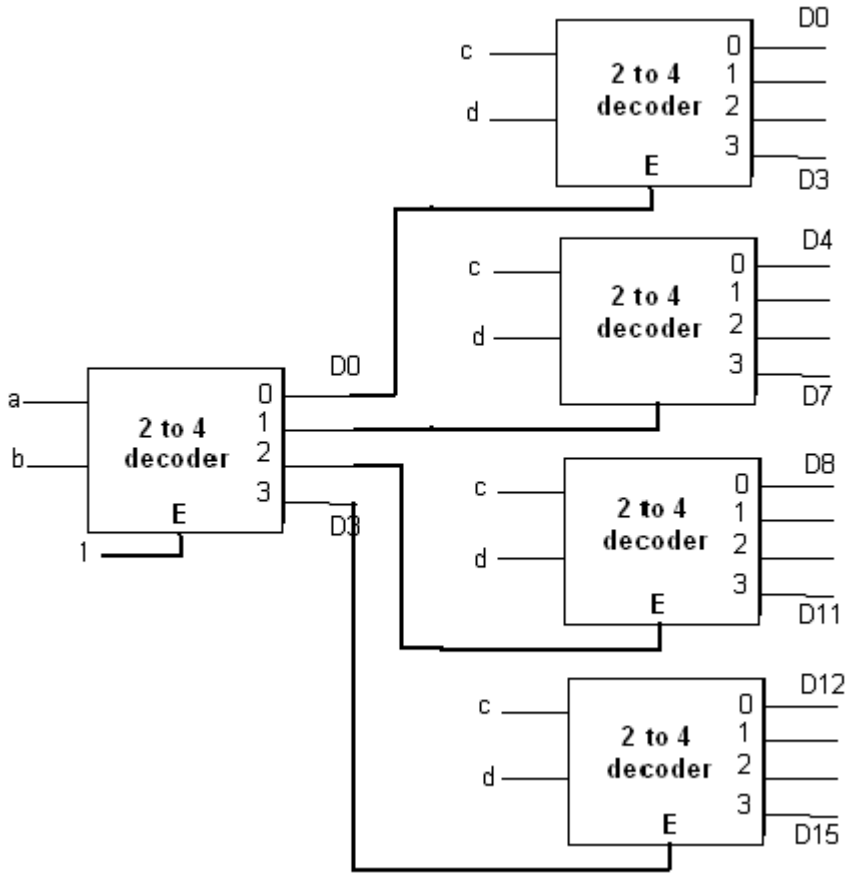
| Decimal value | Enable | Inputs | | Outputs | | | |
|---|---|---|---|---|---|---|---|
| | E | $A_1$ | $A_0$ | $D_0$ | $D_1$ | $D_2$ | $D_3$ |
| | 0 | X | X | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 2 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 3 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |

Truth table of 2-to-4 decoder with enable

Implementing a 4x16 Decoder Using 2x4 Decoders:

A 4x16 decoder is a digital circuit that has four inputs (D0, D1, D2, D3) and sixteen outputs (Y0 to Y15). It selects one of the sixteen output lines based on the four input lines. We can implement a 4x16 decoder using 2x4 decoders.

Truth table of 4x16 Decoder:




| D0 | D1 | D2 | D3 | Y0 | Y1 | Y2 | Y3 | Y4 | Y5 | Y6 | Y7 | Y8 | Y9 | Y10 | Y11 | Y12 | Y13 | Y14 | Y15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

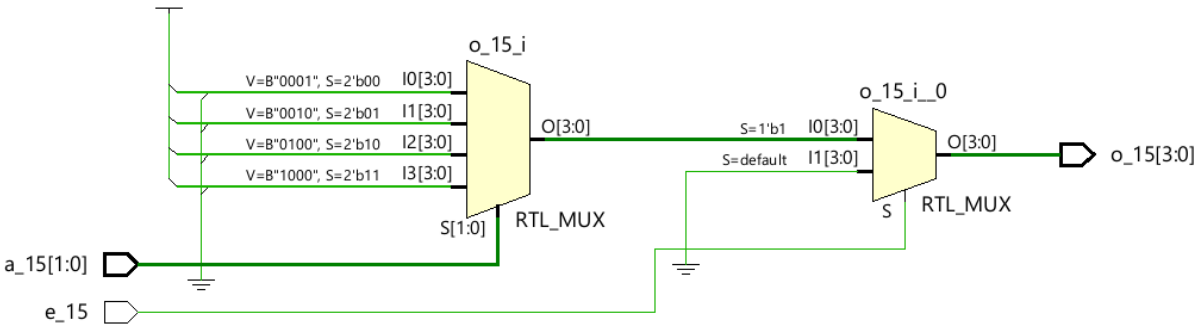
## Code:

## Behavioural

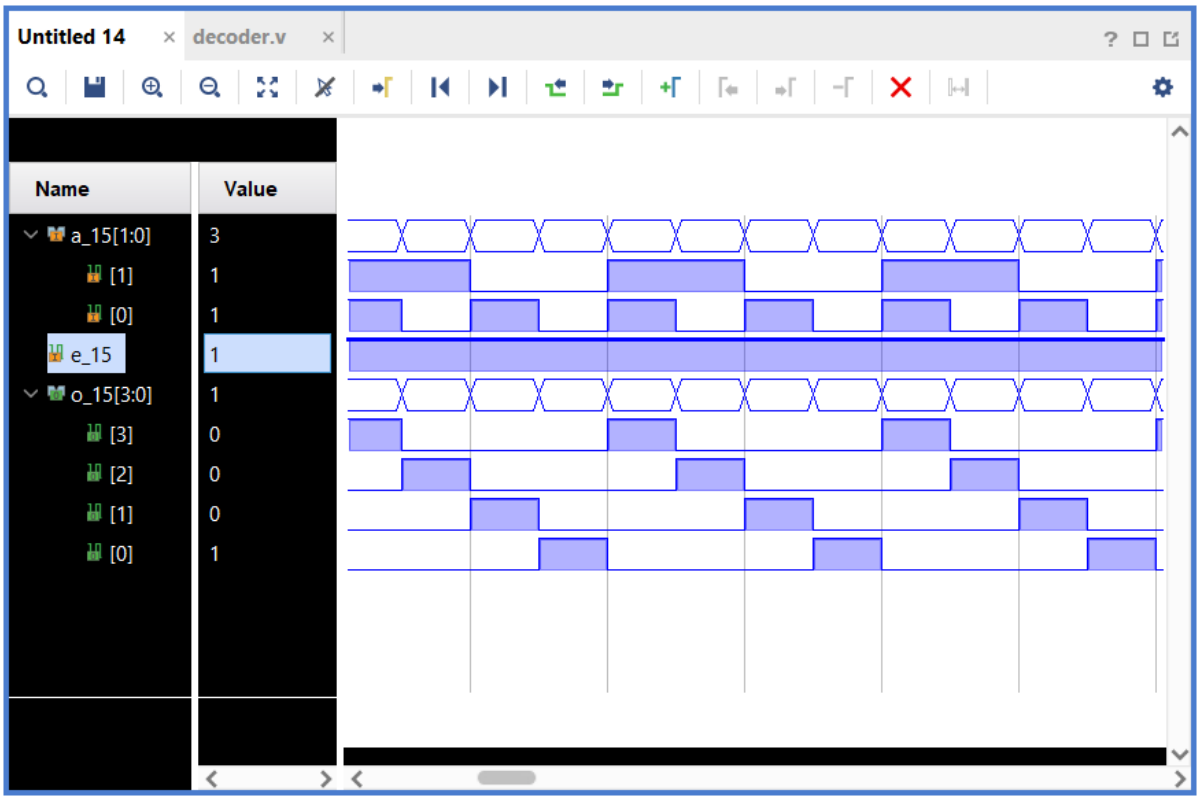
- **2x4 decoder:**

```
module decoder_2x4(en,a0,a1,y);
  input en,a0,a1;
  output reg [3:0]y;
  always @(en,a0,a1)
   begin
    if(en==1)
     begin
      if(a0==1'b0 & a1==1'b0) y=4'b0001;
      else if(a0==1'b0 & a1==1'b1) y=4'b0010;
      else if(a0==1'b1 & a1==1'b0) y=4'b0100;
      else if(a0==1'b1 & a1==1'b1) y=4'b1000;
      else y=4'bxxxx;
     end
    else
     y=4'b0000;
   end
endmodule
```

**RTL Schematic:**



## Function Verification:



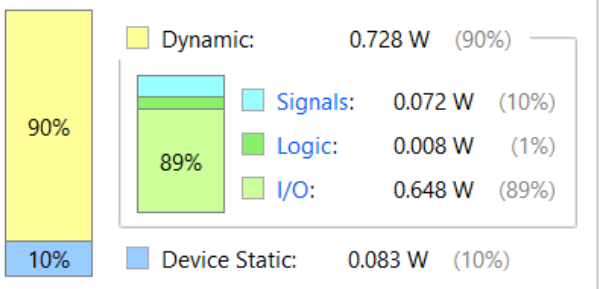

## Post power report:

**Summary**

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

| | |
|---|---|
| **Total On-Chip Power:** | **0.811 W** |
| **Design Power Budget:** | **Not Specified** |
| **Process:** | typical |
| **Power Budget Margin:** | **N/A** |
| **Junction Temperature:** | **26.5°C** |
| Thermal Margin: | 58.5°C (30.8 W) |
| Ambient Temperature: | 25.0 °C |
| Effective ϑJA: | 1.9°C/W |
| Power supplied to off-chip devices: | 0 W |

**On-Chip Power**



## Post synthesis timing summary:

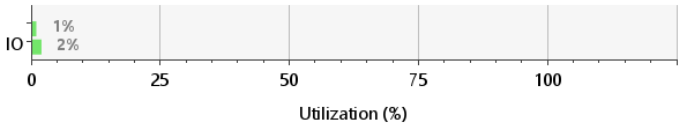
Unconstrained Paths - NONE - NONE - Setup

| Name | Slack | Levels | Routes | High Fanout | From | To | Total Delay | Logic Delay | Net Delay | Requirement | Source Clock |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Path 1 | ∞ | 3 | 4 | 4 | e_15 | o_15[2] | 4.526 | 3.359 | 1.167 | ∞ | input port clock |
| Path 2 | ∞ | 3 | 4 | 4 | e_15 | o_15[3] | 4.526 | 3.359 | 1.167 | ∞ | input port clock |
| Path 3 | ∞ | 3 | 4 | 4 | a_15[0] | o_15[0] | 4.512 | 3.345 | 1.167 | ∞ | input port clock |
| Path 4 | ∞ | 3 | 4 | 4 | a_15[0] | o_15[1] | 4.512 | 3.345 | 1.167 | ∞ | input port clock |

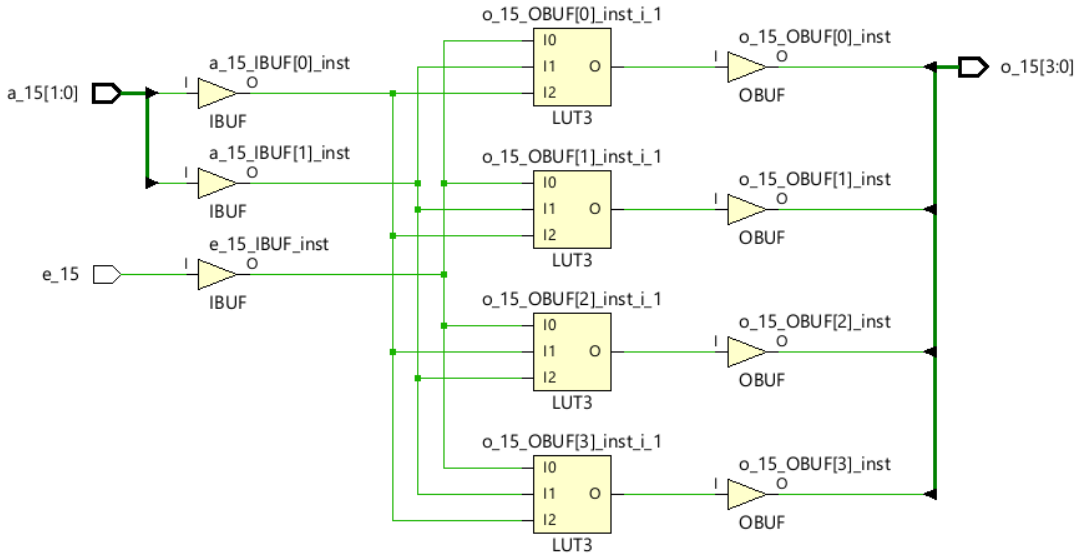
## Post utilization/ area summary:

**Summary**

| Resource | Utilization | Available | Utilization % |
|---|---|---|---|
| LUT | 2 | 41000 | 0.00 |
| IO | 7 | 300 | 2.33 |




## Post implementation schematic:



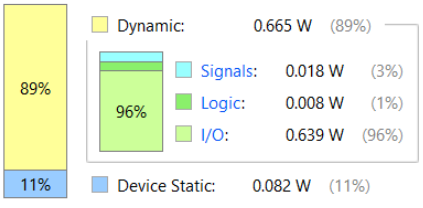

## Post implementation power report:

**Summary**

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

| | |
|---|---|
| **Total On-Chip Power:** | **0.747 W** |
| **Design Power Budget:** | **Not Specified** |
| **Process:** | typical |
| **Power Budget Margin:** | **N/A** |
| **Junction Temperature:** | **26.4°C** |
| Thermal Margin: | 58.6°C (30.9 W) |
| Ambient Temperature: | 25.0 °C |
| Effective θJA: | 1.9°C/W |
| Power supplied to off-chip devices: | 0 W |

**On-Chip Power**

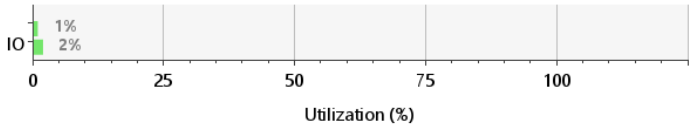


**Post implementation timing summary:**

| Name | Slack ^1 | Levels | Routes | High Fanout | From | To | Total Delay | Logic Delay | Net Delay | Requirement | Source Clock |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Path 1 | ∞ | 3 | 2 | 4 | a_15[0] | o_15[2] | 6.018 | 3.443 | 2.575 | ∞ | input port clock |
| Path 2 | ∞ | 3 | 2 | 4 | a_15[0] | o_15[1] | 5.752 | 3.272 | 2.480 | ∞ | input port clock |
| Path 3 | ∞ | 3 | 2 | 4 | e_15 | o_15[3] | 5.635 | 3.327 | 2.308 | ∞ | input port clock |
| Path 4 | ∞ | 3 | 2 | 4 | e_15 | o_15[0] | 5.604 | 3.419 | 2.185 | ∞ | input port clock |

**Post implementation utilization/ area summary:**

| Resource | Utilization | Available | Utilization % |
|---|---|---|---|
| LUT | 2 | 41000 | 0.00 |
| IO | 7 | 300 | 2.33 |




- **4x16 decoder:**

```
module decoder2x4(
   input[1:0] a,
   input e,
   output reg [3:0] o
   );
   always@(*)
      if (e)
         begin
         case(a)
            2'b00:begin
            o[0]=1'b1;
            o[3:1]=0;
         end
      2'b01:begin
         o[1]=1'b1;
         o[3:2]=0;
         o[0]=0;
         end
      2'b10:begin
         o[2]=1'b1;
         o[3]=0;
         o[1:0]=0;
         end
      2'b11:begin
         o[3]=1'b1;
         o[2:0]=0;
         end
      default:o=0;
```

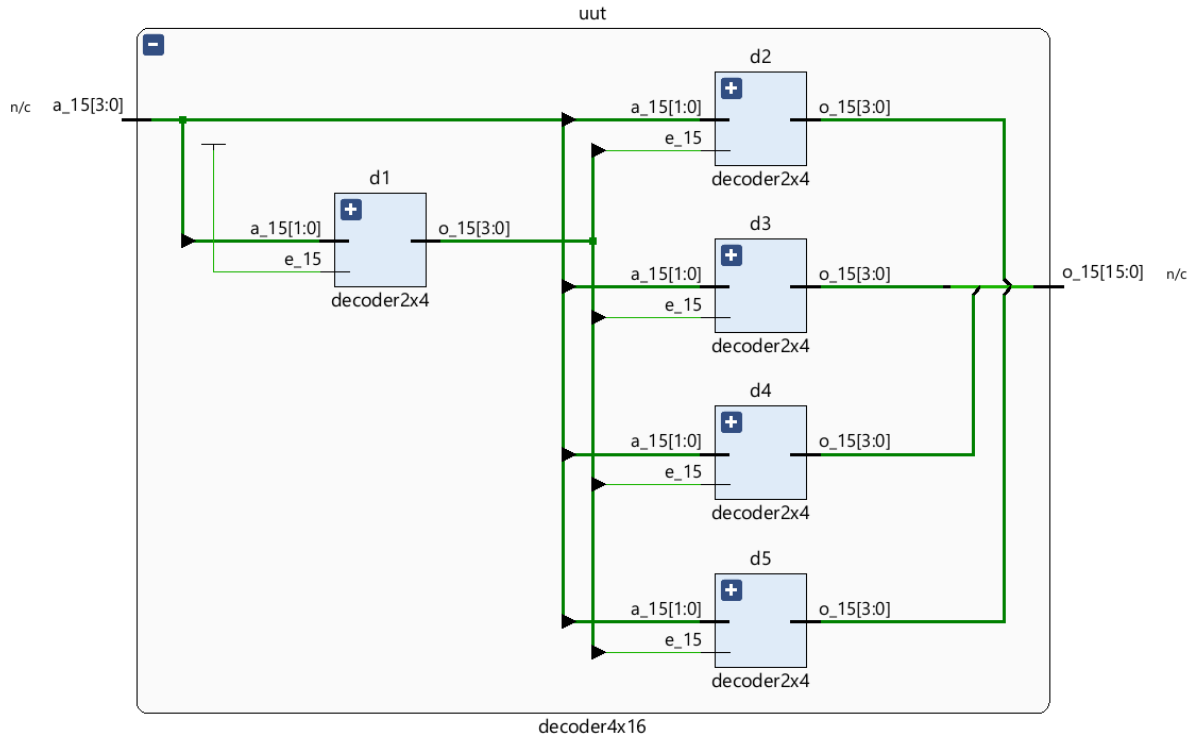
```
        endcase
        end
        else
           o=0;
endmodule

module decoder4x16(
   input[3:0] a,
   output [15:0] o
   );
   wire [3:0] t;
   decoder2x4 d1(.a(a[3:2]),.e(1'b1),.o(t));
   decoder2x4 d2(.a(a[1:0]),.e(t[0]),.o(o[3:0]));
   decoder2x4 d3(.a(a[1:0]),.e(t[1]),.o(o[7:4]));
   decoder2x4 d4(.a(a[1:0]),.e(t[2]),.o(o[11:8]));
   decoder2x4 d5(.a(a[1:0]),.e(t[3]),.o(o[15:12]));
endmodule
```

- **4x16 decoder Testbench:**

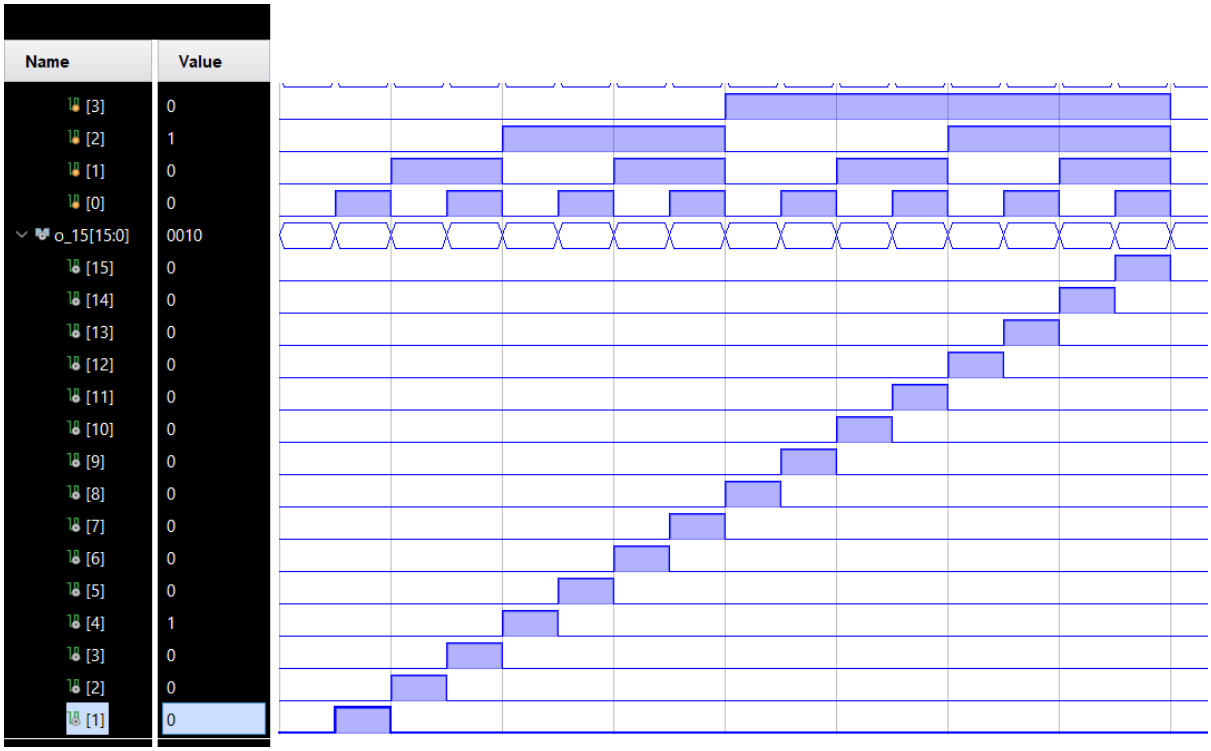
```
module testbench_decoder4x16;

 // Inputs
 reg [3:0] a;

 // Outputs
 wire [15:0] o;

 // Instantiate the decoder
 decoder4x16 uut (
   .a(a),
   .o(o)
 );

 // Clock signal
 reg clk = 0;
 always begin
   #5 clk = ~clk;
 end

 // Test vector
 initial begin
   $display("Testing 4x16 Decoder");
   $monitor("a = %b, o = %b", a, o);

   // Test all possible input combinations
```

```
    for (a = 0; a < 16; a = a + 1) begin
      #10;  // Wait for some time to observe the output
    end

    // Terminate the simulation
    $finish;
  end

endmodule

// Simulate the testbench
module tb_decoder4x16;
  reg clk;
  initial begin
    clk = 0;
    $dumpfile("decoder4x16.vcd");
    $dumpvars(0, tb_decoder4x16);
    $display("Starting simulation");
    // Simulate for 100 time units
    repeat (100) begin
      #5 clk = ~clk;
    end
    $finish;
  end
endmodule
```

**RTL Schematic:**



## Function Verification:



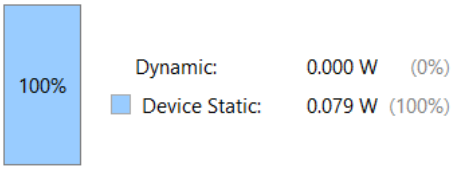

## Post power report:

**Summary**

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

| | |
|---|---|
| **Total On-Chip Power:** | **0.079 W** |
| **Design Power Budget:** | **Not Specified** |
| **Process:** | typical |
| **Power Budget Margin:** | **N/A** |
| **Junction Temperature:** | **25.1°C** |
| Thermal Margin: | 59.9°C (31.6 W) |
| Ambient Temperature: | 25.0 °C |
| Effective ϑJA: | 1.9°C/W |

**On-Chip Power**




## Conclusion:

- In this experiment, we written Verilog code for 2x4 decoder and using this decoder we have implemented 4x16 decoder and verified its Schematic and outputs, area utilization and its power consumption.