Worksheet of Experiment

Date of Experiment	3 sep, 2024
Name	Akshada Jotiram Ghorpade.
Registration Number	12302956.

Title of Experiment: Develop RTL designs in verilog to implement decoders and encoders and verify their functionality using cadence NCSIM.

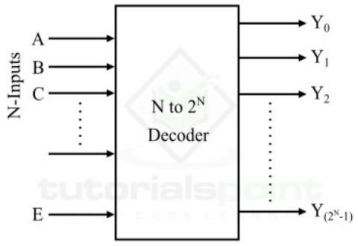
Name of Language: Verilog.

Name of Software: Cadence NCSIM, simvision.

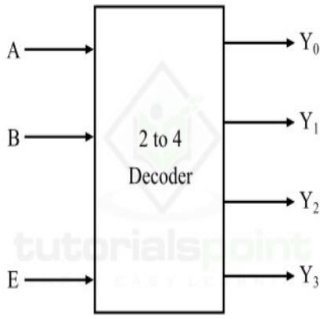
Theory of Experiment:

Decoder:

- A combinational logic circuit which converts N input lines into a maximum of 2_N output lines is called a **decoder**.
- Therefore, a decoder is a combination logic circuit that is capable of identifying or detecting a particular code. The operation that a decoder performs is referred to as decoding.



- **2:4 Decoder :**
  - The 2 to 4 decoder is one that has 2 input lines and 4 (2^2) output lines.
  - When this decoder is enabled with the help of enable input E, then its one of the four outputs will be active for each combination of inputs.



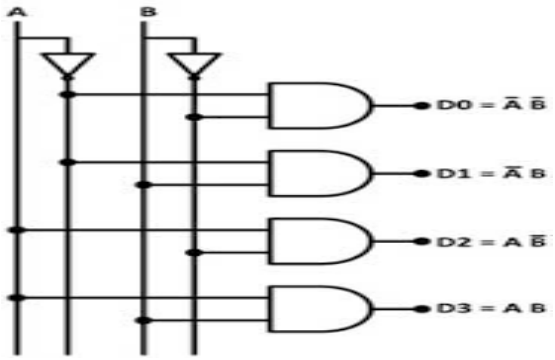
Truth table:

INPUT		OUTPUT			
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

Boolean expression:

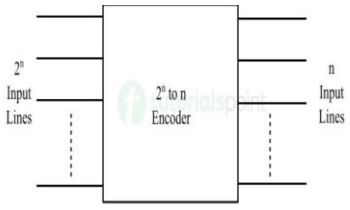
D0=A'·B'
D1=A'·B
D2=A·B'
D3=A·B

Circuit Diagram:



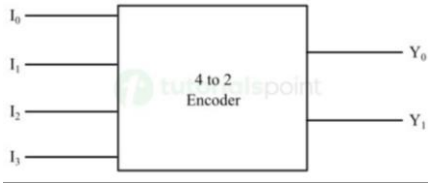
Encoder:

- An **encoder** is a digital combinational circuit that converts a human friendly information into a coded format for processing using machines.
- An encoder consists of a certain number of input and output lines. Where, an encoder can have maximum of "2^n" input lines whereas "n" output lines.



4:2 Encoder:

- A 4 to 2 Encoder is a type of encoder which has 4 (2^2) input lines and 2 output lines. It produces an output code (i.e., convert input information in a 2-bit format) depending on the combination of input lines.



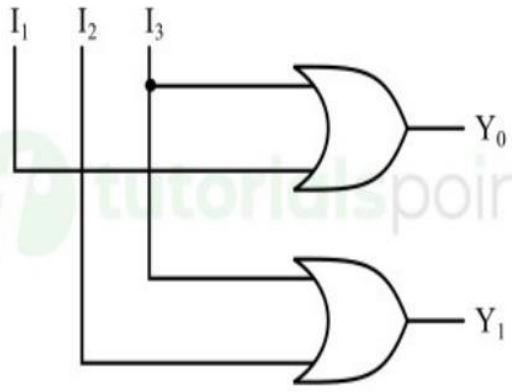
Truth table:

INPUT				OUTPUT	
I0	**I1**	**I2**	**I3**	**Y0**	**Y1**
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

Boolean expression:

$$Y_0 = I_2 + I_3$$
$$Y_1 = I_1 + I_3$$

Circuit Diagram:



Program and Stimulus (Testbench) program:

PROGRAM (Decoder):

```
module dec24(a, b, d0, d1, d2, d3);
input a,b;
output d0,d1,d2,d3;
assign d0=(~a&~b);
assign d1=(~a&b);
assign d2=(a&~b);
assign d3=(a&b);
endmodule
```

TESTBENCH (Decoder):

```
module dec24tb;
reg a,b;
wire d0,d1,d2,d3;
dec24 dut(a, b, d0, d1, d2, d3);
initial
begin
$monitor("a=%b, b=%b, do=%b, d1=%b, d2=%b, d3=%b", a, b, do, d1, d2, d3);
a=0; b=0;
#10 a=0; b=1;
#10 a=1; b=0;
#10 a=1; b=1;
#10 $stop;
end
endmodule
```

PROGRAM (Encoder):

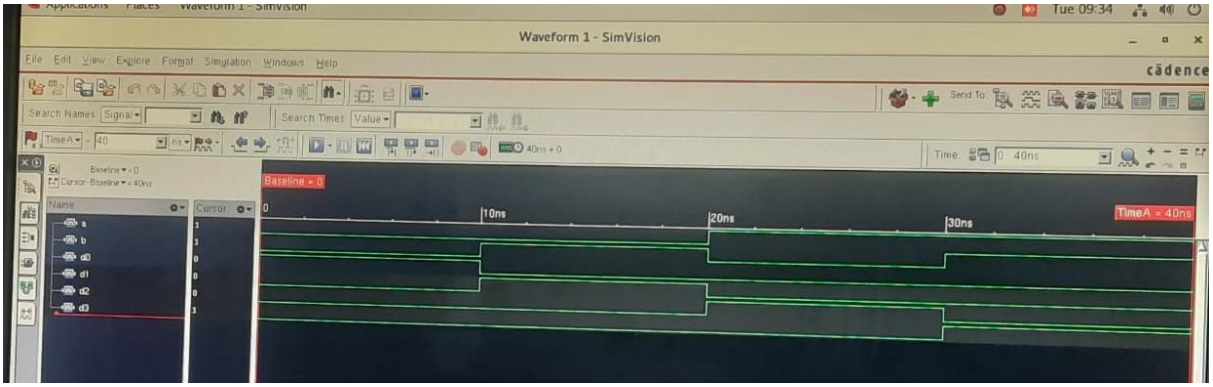
```
module enc42(I0,I1,I2,I3,A,B);
input I0,I1,I2,I3;
output A,B;
assign A=((~I0&~I1)&(I2^I3));
assign B=((I0&~I2)&(I1^I3));
endmodule
```

TESTBENCH (Encoder):

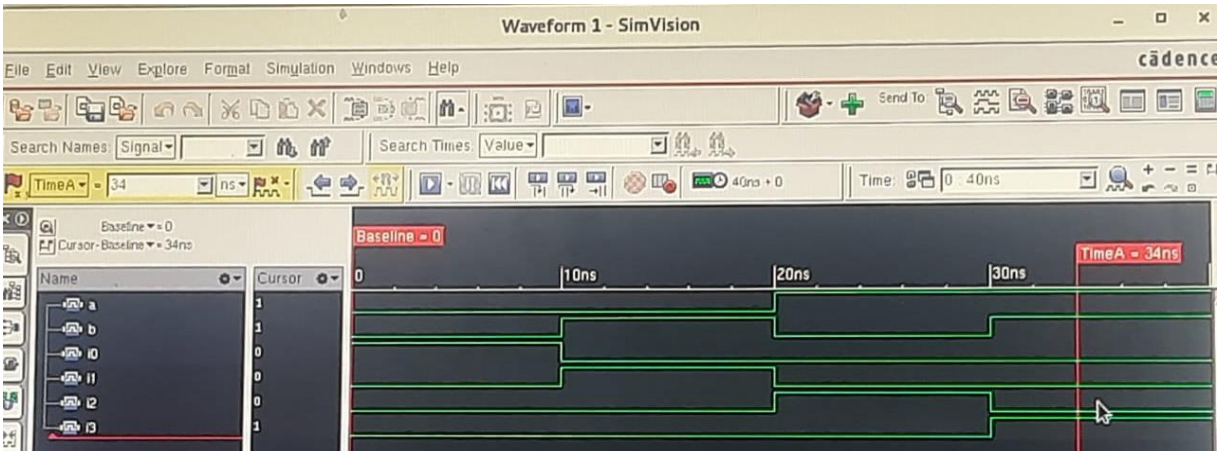
```
module enc42tb;
reg I0,I1,I2,I3;
wire A,B;
enc42 dut(I0,I1,I2,I3,A,B);
initial
begin
$monitor("I0=%b, I1=%b, I2=%b, I3=%b, A=%b, B=%b", I0,I1,I2,I3,A,B);
{I0,I1,I2,I3} = 4'b1000;
#10 {I0,I1,I2,I3} = 4'b0100;
#10 {I0,I1,I2,I3} = 4'b0010;
#10 {I0,I1,I2,I3} = 4'b0001;
#10 $stop;
end
endmodule
```

Simulation waveform:

Decoder

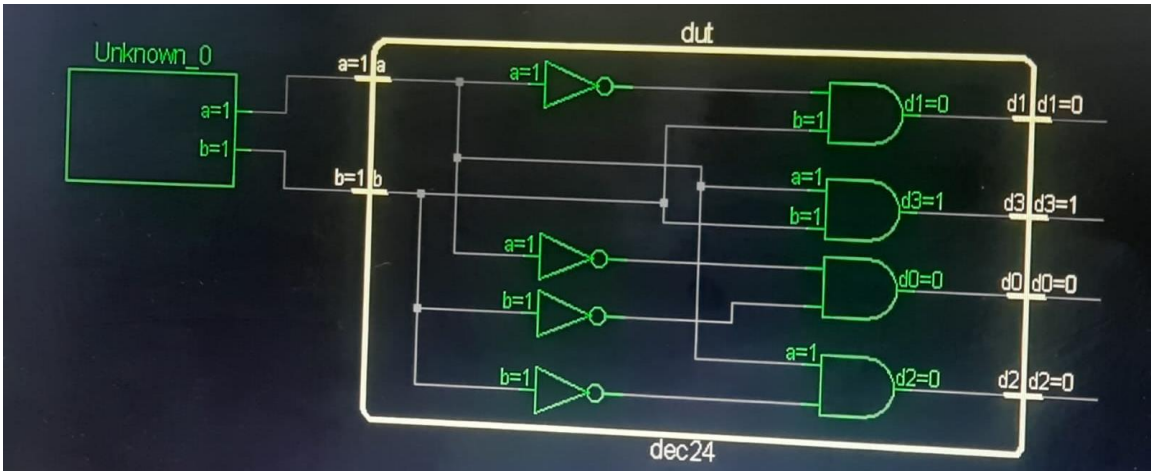


Encoder

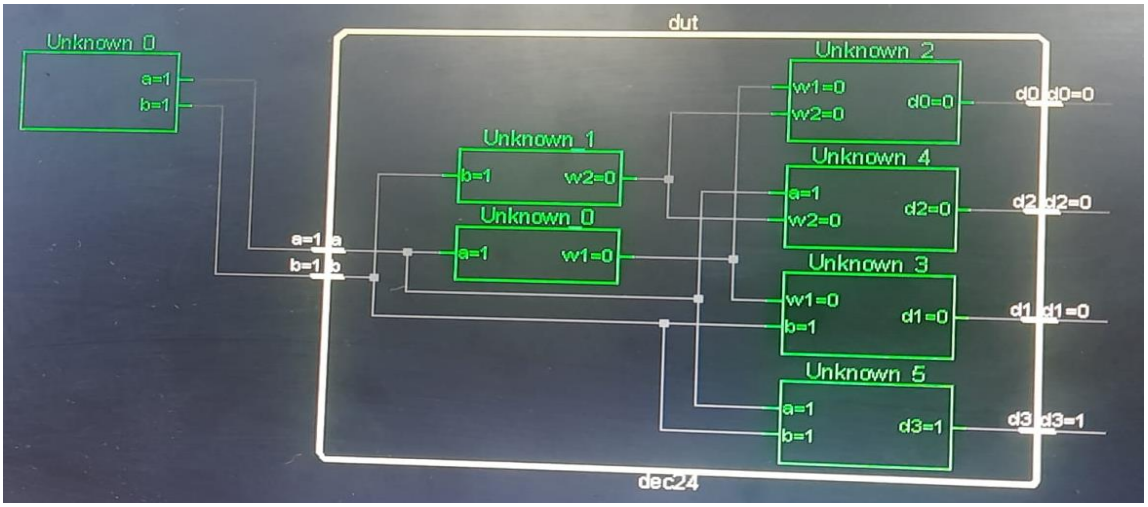


RTL Schematic:

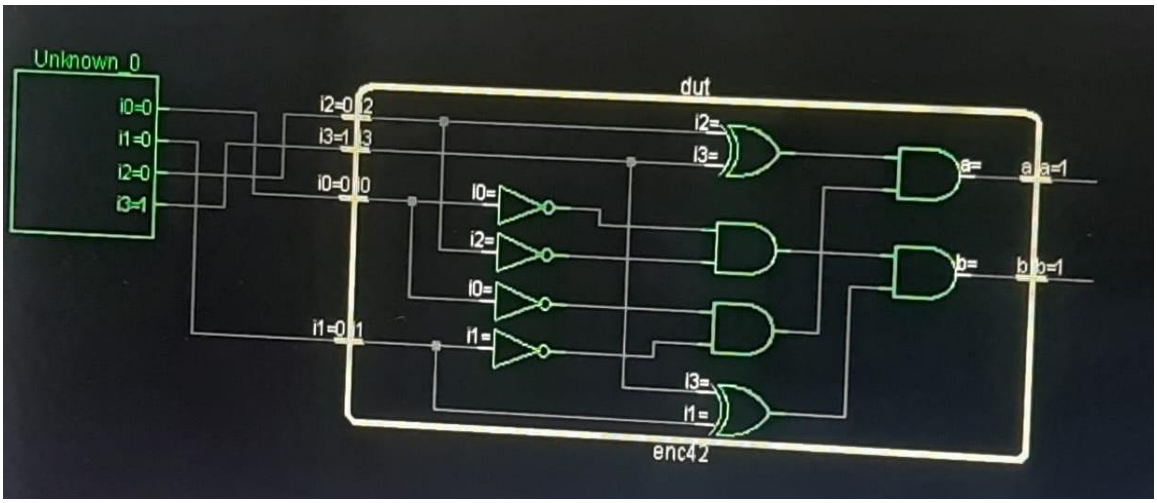
Decoder (Gate level)



Decoder (Dataflow level)

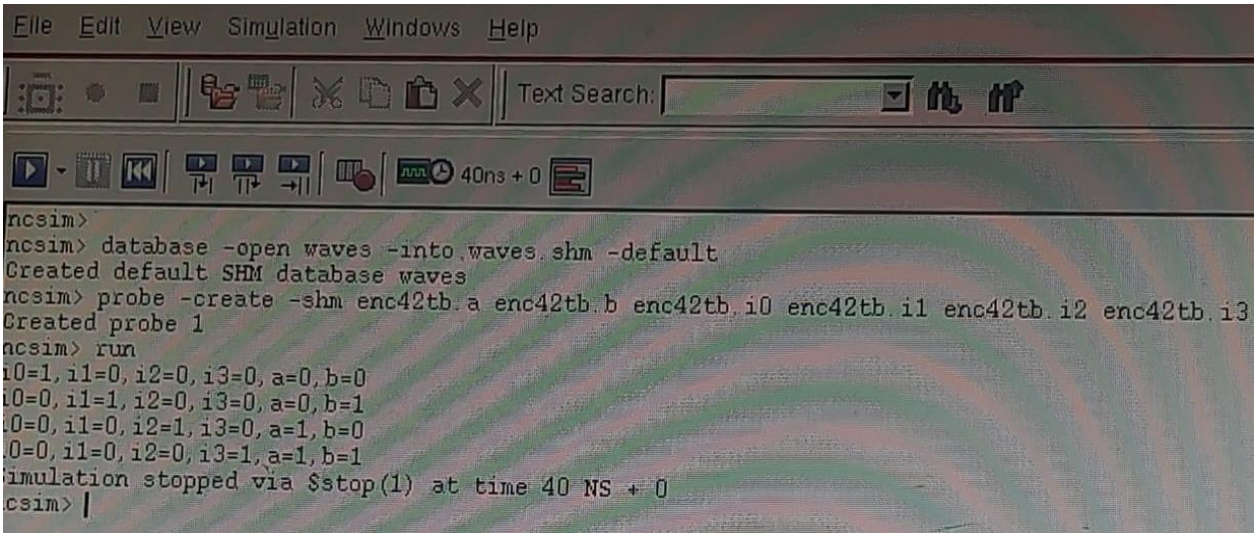


Encoder

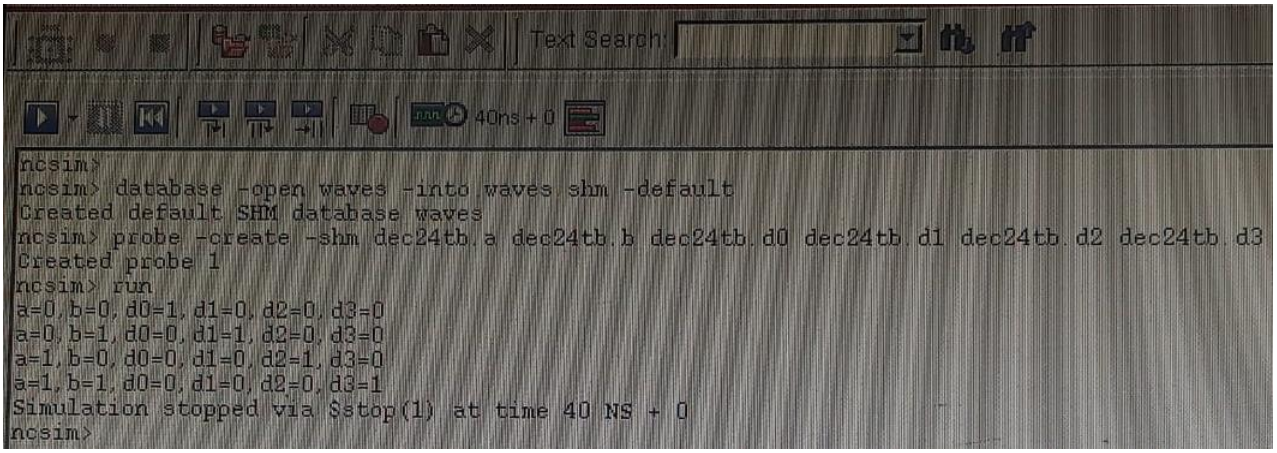


Console Window:

Encoder



Decoder



Learning Outcome:

1. Learn to write syntactically correct Verilog code for encoder and decoder.

2. Create testbenches to simulate and verify the functionality of encoder and decoder.

3. Apply techniques to optimize logic circuits for speed, area, and power efficiency.

4. Perform timing analysis to ensure circuits meet required timing constraints.

5. Analyze the performance of encoder and decoder and circuits in terms of timing, power, and area using simulation results.

Simulation waveform

RTL Schematic

Learning Outcome