

RTL CHALLENGE

DAY-12:- Verilog Code for Implementation of Encoder.

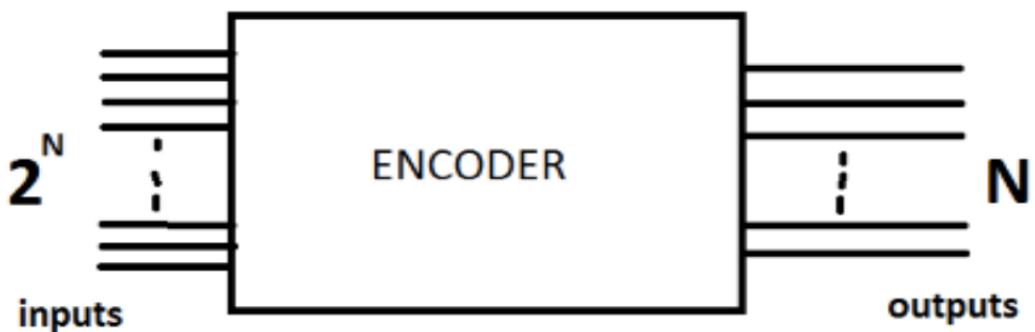
Software used:- Xilinx Vivado(2023.1)

Theory:

Encoder :-

In modern computer systems, an 'encoder' plays multifaceted roles across various domains. In data processing, it serves as a vital tool for transforming information from one format to another, ensuring compatibility and efficient transmission. In the realm of artificial intelligence and machine learning, particularly within deep learning frameworks, an encoder serves as a fundamental component of autoencoders, facilitating the compression of input data into a concise latent representation, crucial for tasks such as data compression, denoising, or feature learning. Moreover, in communication systems, encoders function as pivotal devices for converting data streams into suitable formats for transmission, ensuring reliable communication across diverse channels. Whether encoding data for efficient storage, compressing information for streamlined processing, or facilitating seamless communication, the encoder stands as a cornerstone in modern computing and information exchange.

Encoder circuit diagram:



DESIGN CODES & TEST BENCHES FOR ENCODERS:

Code for Encoder 2 to 1 :-

```
// Design Name: Encoder
// Module Name: Encoder2to1
// Project Name: RTL Schematic

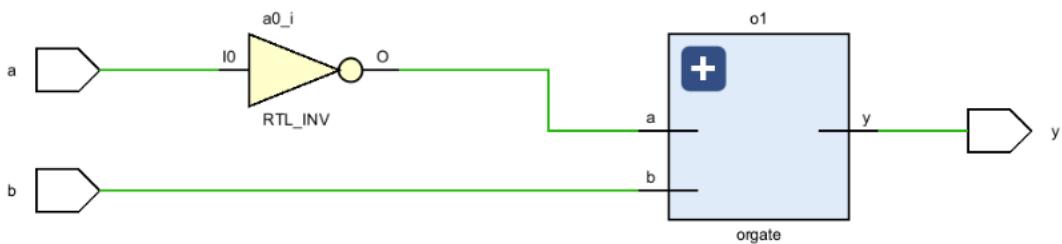
module Encoder2to1(
    input a,b,
    output y
);
    orgate o1 (.a(!a),.b(b),.y(y));
endmodule
```

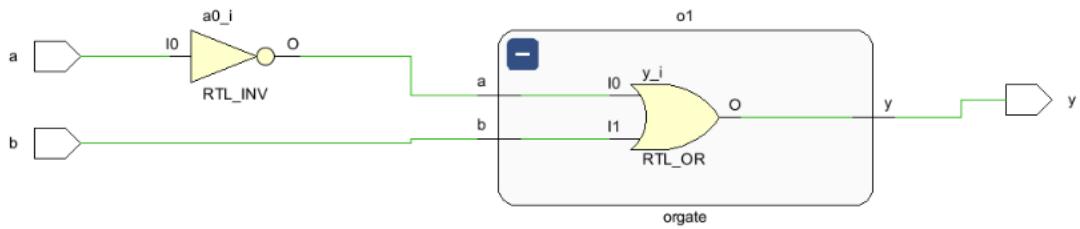
TEST BENCH:

```
// Design Name: Test bench
// Module Name: Encoder2to1_tb
// Project Name: RTL Schematic

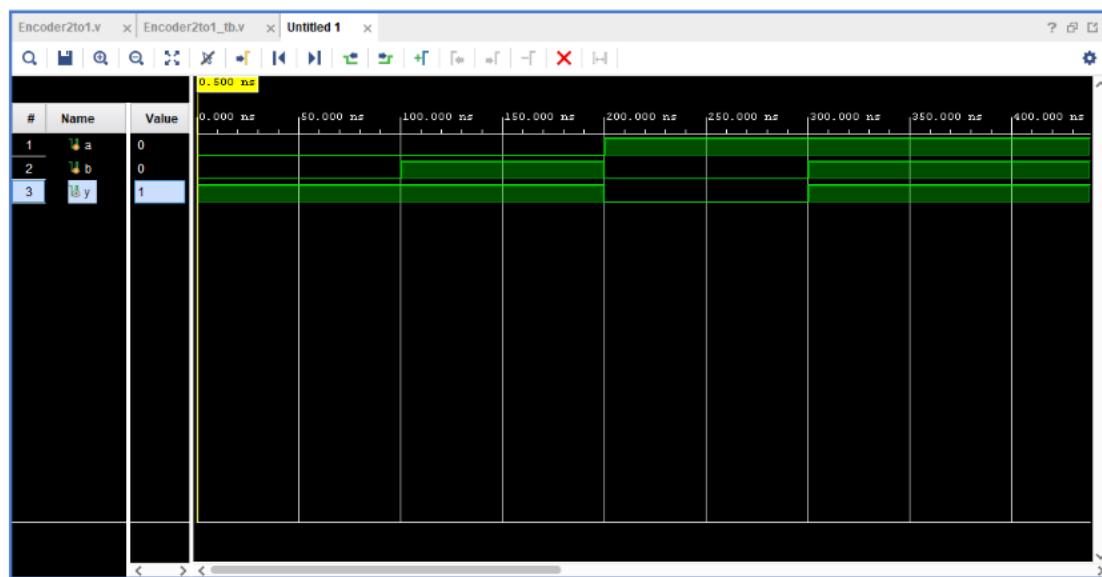
module Encoder2to1_tb();
reg a,b;
wire y;
Encoder2to1 uut (.a(a),.b(b),.y(y));
initial
begin
    a=0;b=0;#100;
    a=0;b=1;#100;
    a=1;b=0;#100;
    a=1;b=1;#100;
end
endmodule
```

RTL SCHEMATIC:





OUTPUT WAVE FORM:



Code for Encoder 4 to 2:-

```
// Design Name: RTL Schematic
// Module Name: Encoder4to2
// Project Name: Verilog

module Encoder4to2(
    input a,b,c,d,
    output y0,y1
);

    orgate o2 (.a(a),.b(b),.y(y0));
    orgate o3 (.a(c),.b(a),.y(y1));

endmodule
```

TEST BENCH:

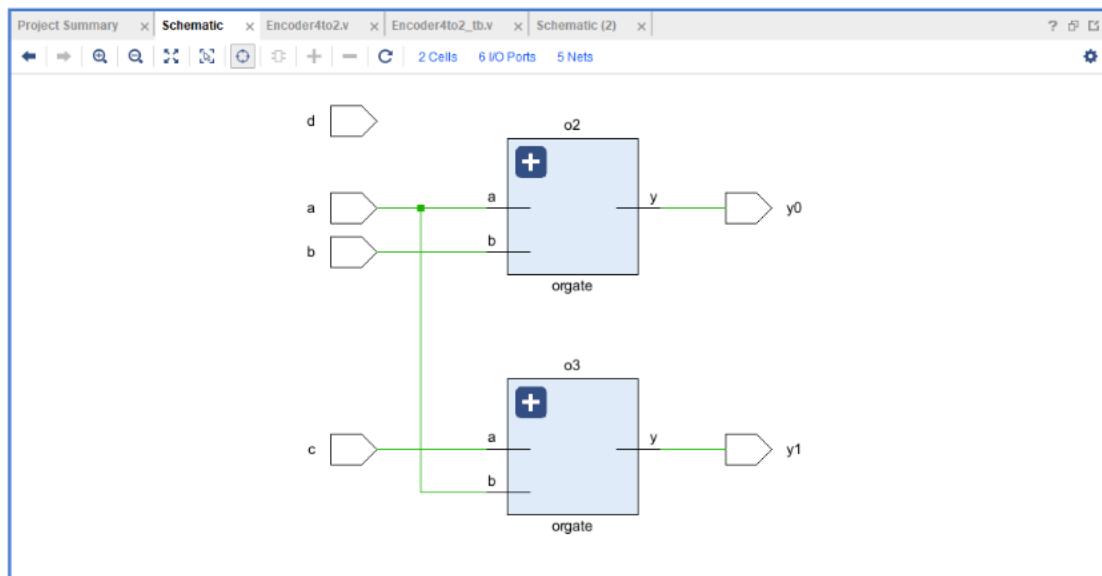
```
// Design Name: Test bench
// Module Name: Encoder4to2_tb
// Project Name: RTL Schematic

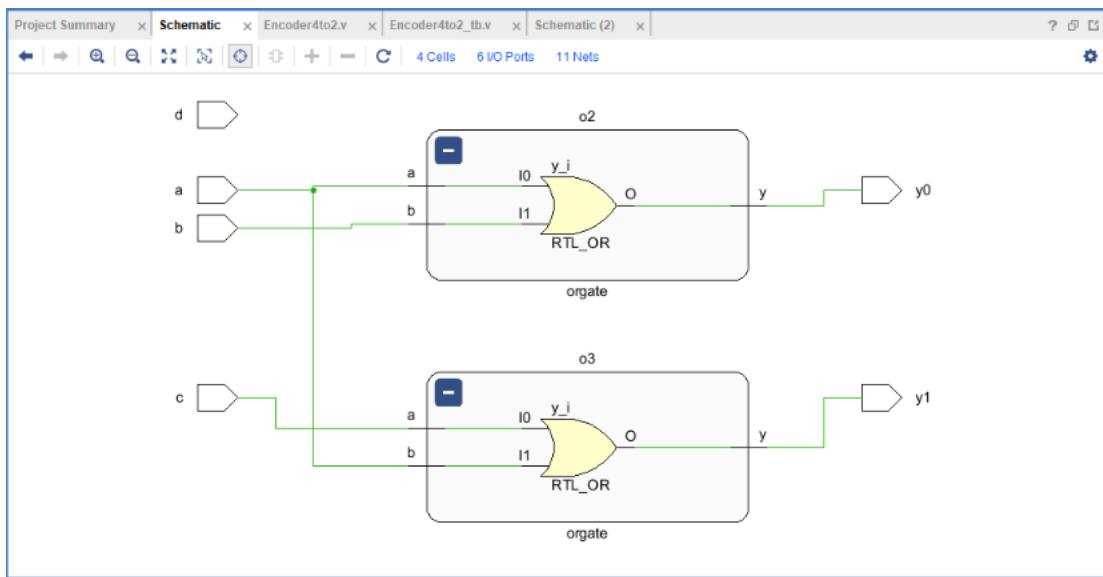
module Encoder4to2_tb();
reg a,b,c,d;
wire y0,y1;
Encoder4to2 uut (.a(a),.b(b),.c(c),.d(d),.y0(y0),.y1(y1));

initial
begin
    a=0;b=0;c=0;d=1;#100;
    a=0;b=0;c=1;d=0;#100;
    a=0;b=1;c=0;d=0;#100;
    a=1;b=0;c=0;d=0;#100;
end

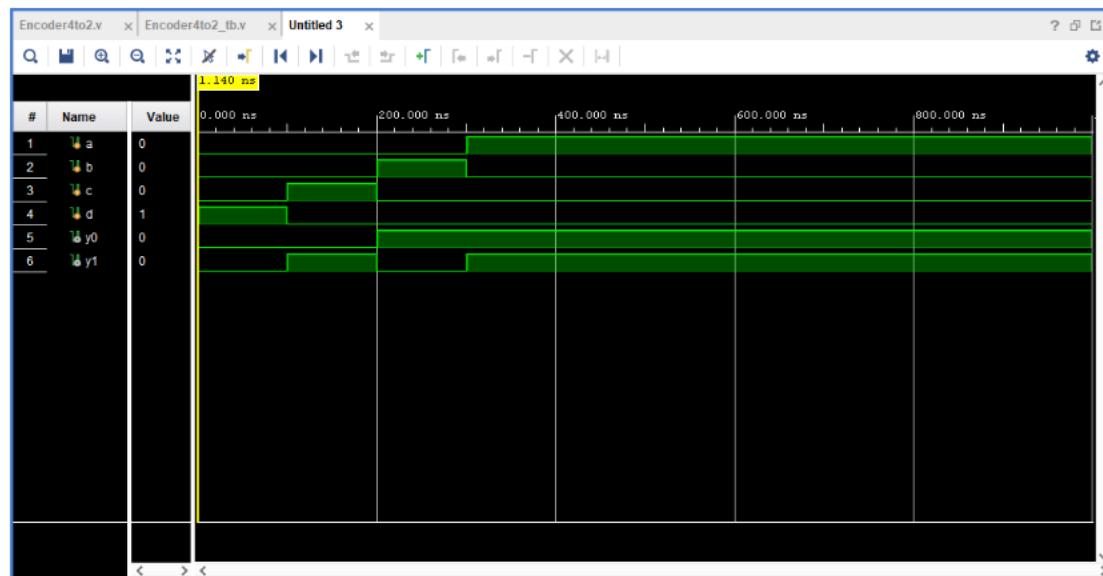
endmodule
```

RTL SCHEMATIC:





OUTPUT WAVEFORM:



Code for Encoder 8 to 3 :-

```

// Design Name: Encoder
// Module Name: Encoder8to3
// Project Name: RTL Schematic

module Encoder8to3(
    input a,b,c,d,e,f,g,h,
    output y0,y1,y2
);
orgatel o1 (.a(a),.b(c),.c(e),.d(g),.y(y0));
orgatel o2 (.a(b),.b(c),.c(f),.d(g),.y(y1));
orgatel o3 (.a(e),.b(d),.c(f),.d(g),.y(y2));

endmodule

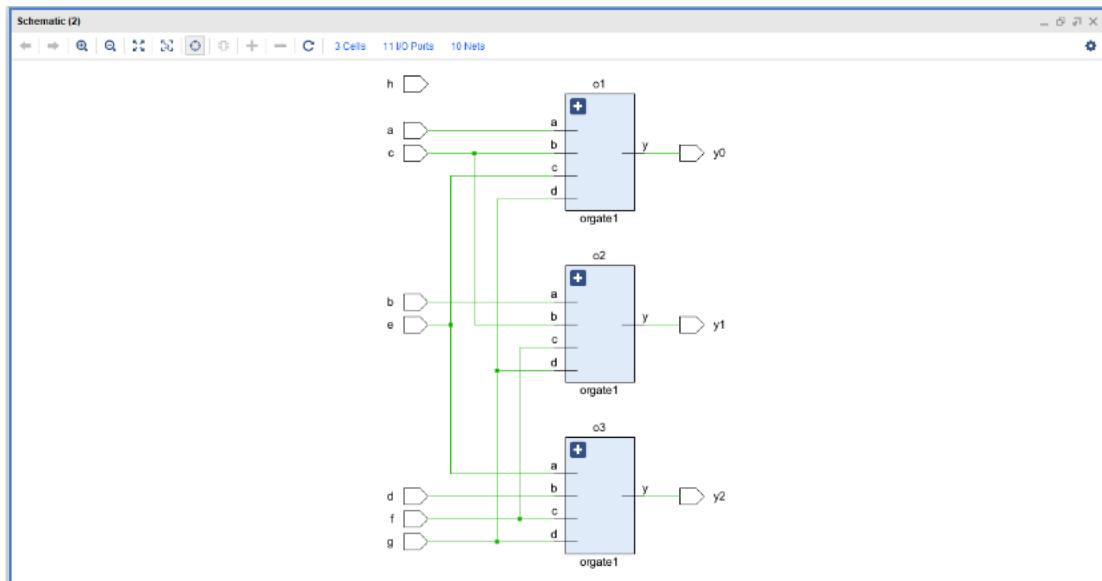
```

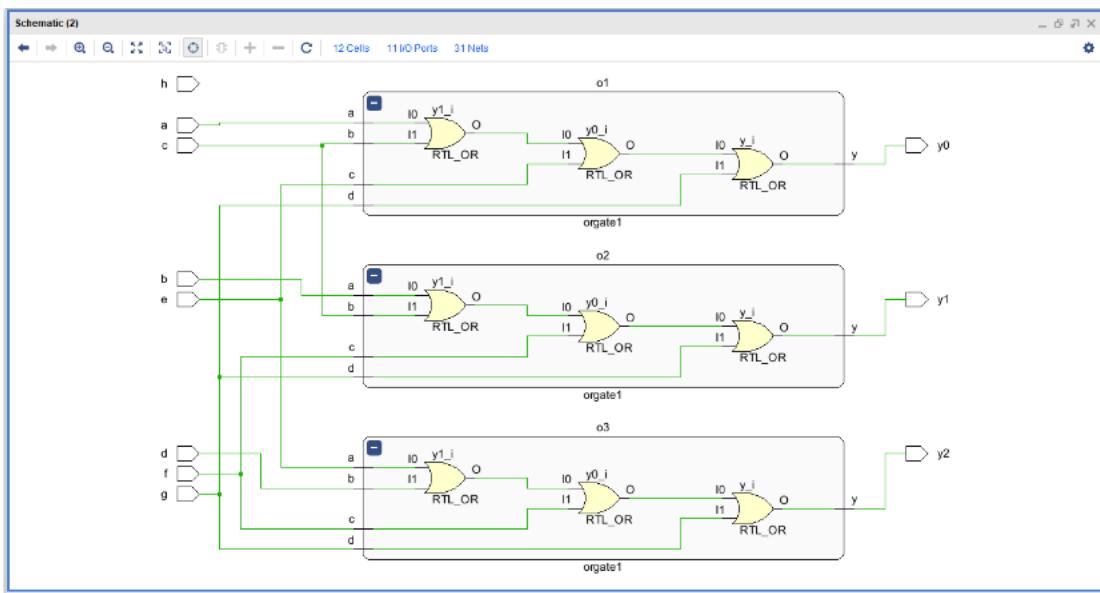
TEST BENCH:

```
// Design Name: TEST BENCH
// Module Name: Encoder8to3_tb
// Project Name: RTL

module Encoder8to3_tb();
reg a,b,c,d,e,f,g,h;
wire y0,y1,y2;
Encoder8to3 uut (.a(a),.b(b),.c(c),.d(d),.e(e),.f(f),.g(g),.h(h),.y0(y0),.y1(y1),.y2(y2));
initial
begin
    a=1;b=0;c=0;d=0;e=0;f=0;g=0;h=0;#100;
    a=0;b=1;c=0;d=0;e=0;f=0;g=0;h=0;#100;
    a=0;b=0;c=1;d=0;e=0;f=0;g=0;h=0;#100;
    a=0;b=0;c=0;d=1;e=0;f=0;g=0;h=0;#100;
    a=0;b=0;c=0;d=0;e=1;f=0;g=0;h=0;#100;
    a=0;b=0;c=0;d=0;e=0;f=1;g=0;h=0;#100;
    a=0;b=0;c=0;d=0;e=0;f=0;g=1;h=0;#100;
    a=0;b=0;c=0;d=0;e=0;f=0;g=0;h=1;#100;
end
endmodule
```

RTL SCHEMATIC:





OUTPUT WAVEFORMS:

