PROJECT ON VERIFICATION OF DIGITAL SYSTEMS

TOPIC : LAYERED TESTBENCH FOR PRIORITY ENCODER

NAME : MEGHA Y RAYANAGOUDRA [PES2UG22EC077]

ABHINANDANA N HOWLE [PES2UG22EC005]

OBJECTIVE :
THE OBJECTIVE OF A **PRIORITY ENCODER** IS TO CONVERT A BINARY INPUT WITH MULTIPLE BITS INTO A BINARY OUTPUT THAT REPRESENTS THE POSITION OF THE HIGHEST-ORDER BIT THAT IS SET TO 1. ADDITIONALLY, IT USUALLY PROVIDES A **VALID SIGNAL** TO INDICATE WHETHER ANY INPUT IS ACTIVE (NON-ZERO).

INTRODUCTION:

A **priority encoder** is a type of digital circuit that is used to convert a set of binary inputs into a binary output code that represents the position of the highest-order active input bit. It also provides an output signal that indicates whether any input is active or not.

In simple terms, it "encodes" the position of the highest set bit (1) from a set of inputs, with priority given to the higher-order bits. This makes it useful in applications where you need to know the highest active signal from a group of signals, such as in interrupt handling, data compression, and digital communication systems.

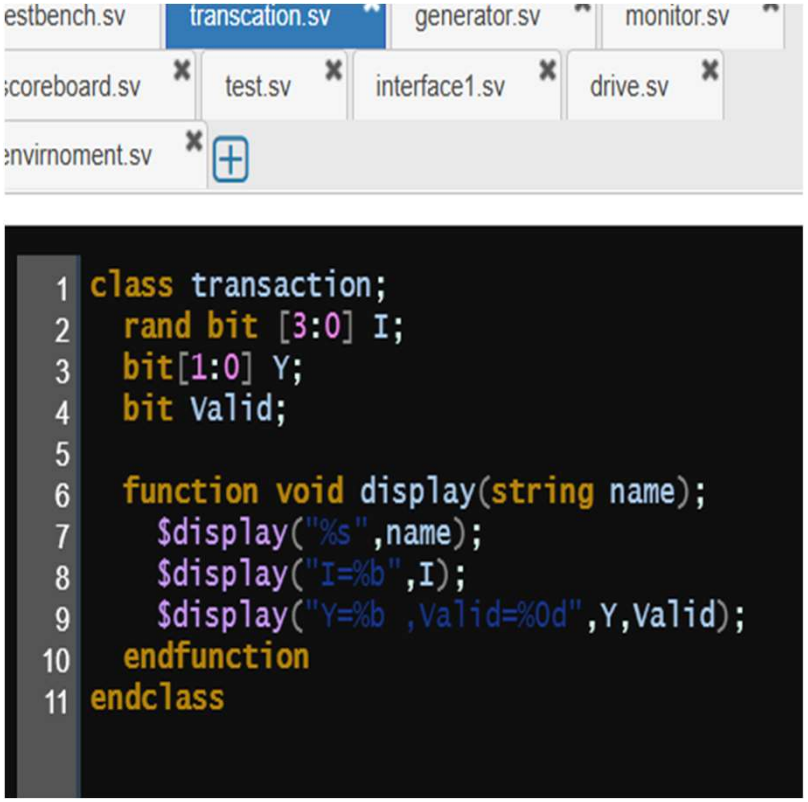
Truth table for 4 bit priority encoder :

Input				Output		
I3	I2	I1	I0	O1	O0	V
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	X	0	1	1
0	1	X	X	1	0	1
1	X	X	X	1	1	1

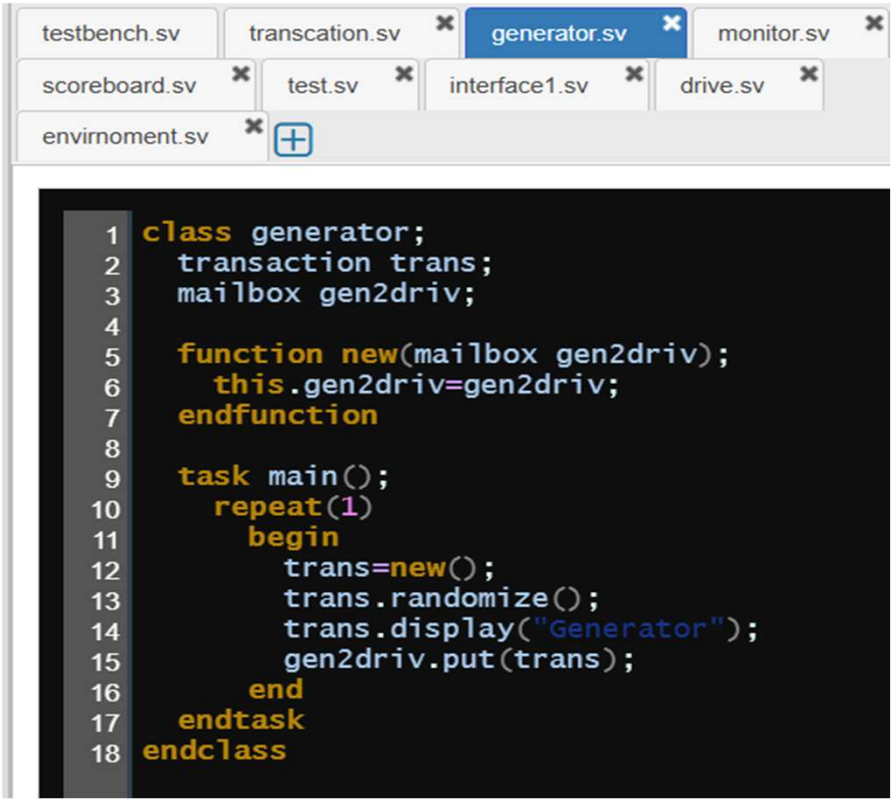
```
1  module priority_encoder (
2    input bit [3:0] I,   // 4-bit input
3    output bit [1:0] Y,   // 2-bit output (priority)
4    output bit Valid   // Valid flag
5  );
6      always @(*) begin
7          case (I)
8              4'b0000: begin Y = 2'b00; Valid = 0; end
9              4'b0001: begin Y = 2'b00; Valid = 1; end
10             4'b0010: begin Y = 2'b01; Valid = 1; end
11             4'b0011: begin Y = 2'b01; Valid = 1; end
12             4'b0100: begin Y = 2'b10; Valid = 1; end
13             4'b0101: begin Y = 2'b10; Valid = 1; end
14             4'b0110: begin Y = 2'b10; Valid = 1; end
15             4'b0111: begin Y = 2'b10; Valid = 1; end
16             4'b1000: begin Y = 2'b11; Valid = 1; end
17             4'b1001: begin Y = 2'b11; Valid = 1; end
18             4'b1010: begin Y = 2'b11; Valid = 1; end
19             4'b1011: begin Y = 2'b11; Valid = 1; end
20             4'b1100: begin Y = 2'b11; Valid = 1; end
21             4'b1101: begin Y = 2'b11; Valid = 1; end
22             4'b1110: begin Y = 2'b11; Valid = 1; end
23             4'b1111: begin Y = 2'b11; Valid = 1; end
24             default: begin Y = 2'b00; Valid = 0; end
25         endcase
26     end
27 endmodule
28
```

SYSTEM VERILOG CODE FOR PRIORITY ENCODER

TRANSCATION CODE



GENERATOR CODE



DRIVER CODE

scoreboard.sv | test.sv | interface1.sv | drive.sv | envirnoment.sv

```
class drive;
  virtual intf vif;
  mailbox gen2driv;
  function new(virtual intf vif,mailbox gen2driv);
    this.vif=vif;
    this.gen2driv=gen2driv;
  endfunction

    task main;
      repeat(1)
        begin
          transaction trans;
          gen2driv.get(trans);
          vif.I  <= trans.I;
          trans.Y=vif.Y;
          trans.Valid=vif.Valid;
          trans.display("Driver");
        end
    endtask
endclass
```

MONITOR CODE

testbench.sv | transcation.sv | generator.sv | monitor.sv | scoreboard.sv | test.sv | interface1.sv | drive.sv | envirnoment.sv

```
class monitor;
  virtual intf vif;
  mailbox mon2scb;

  function new(virtual intf vif,mailbox mon2scb);
    this.vif=vif;
    this.mon2scb=mon2scb;
  endfunction

  task main;
    repeat(1)
      #5;
    begin
      transaction trans;
      trans=new();
      trans.I=vif.I;
      trans.Y=vif.Y;
      trans.Valid=vif.Valid;
      mon2scb.put(trans);
      trans.display("Monitor");
    end
  endtask
endclass
```

SCOREBOARD CODE

```
class scoreboard;
  bit [3:0] I;
  bit [1:0] Y;
  bit Valid;
  mailbox mon2scb;

  function new(mailbox mon2scb);
    this.mon2scb=mon2scb;
  endfunction

  task main;
    transaction trans;
    repeat(1)
      begin
        mon2scb.get(trans);
        if(I==4'b0000) assert (Y==2'b00 && Valid==0);
        else if (I==4'b0001) assert (Y==2'b00 && Valid==1);
        else if (I==4'b0010) assert (Y==2'b01 &&Valid==1);
        else if (I==4'b0011) assert (Y==2'b01 && Valid==1);
        else if (I==4'b0100) assert (Y==2'b10 && Valid==1);
        else if (I==4'b0101) assert (Y==2'b10 && Valid==1);
        else if (I==4'b0110) assert (Y==2'b10 && Valid==1);
        else if (I==4'b0111) assert (Y==2'b10 && Valid==1);
        else if (I==4'b1000) assert (Y==2'b11 && Valid==1);
        else if (I==4'b1001) assert (Y==2'b11 && Valid==1);
        else if (I==4'b1010) assert (Y==2'b11 && Valid==1);
        else if (I==4'b1011) assert (Y==2'b11 && Valid==1);
        else if (I==4'b1100) assert (Y==2'b11 && Valid==1);
        else if (I==4'b1101) assert (Y==2'b11 && Valid==1);
        else if (I==4'b1110) assert (Y==2'b11 && Valid==1);
        else if (I==4'b1111) assert (Y==2'b11 && Valid==1);
        trans.display("Scoreboard");
      end
  endtask
endclass
```

INTERFACE CODE


```
1 interface intf();
2   logic [3:0]I;
3   logic [1:0]Y;
4   logic Valid;
5 endinterface
```

ENVIRONMENT CODE

```
1  `include "transcation.sv"
2  `include "generator.sv"
3  `include "drive.sv"
4  `include "monitor.sv"
5  `include "scoreboard.sv"
6
7  class environment;
8    generator gen;
9    drive driv;
10   monitor mon;
11   scoreboard scb;
12   mailbox m1;
13   mailbox m2;
14   virtual intf vif;
15   function new(virtual intf vif);
16     this.vif=vif;
17     m1=new();
18     m2=new();
19     gen=new(m1);
20     driv=new(vif,m1);
21     mon=new(vif,m2);
22     scb=new(m2);
23   endfunction
24
25   task test();
26     fork
27       gen.main();
28       driv.main();
29       mon.main();
30       scb.main();
31     join
32   endtask
33
34   task run;
35     test();
36     $finish;
37   endtask
38 endclass
```

TEST CODE

scoreboard.sv | test.sv | interface1.sv

```
1  `include "envirnoment.sv"
2  program test (intf i_intf);
3    environment env;
4    initial
5      begin
6        env=new(i_intf);
7        env.run();
8      end
9  endprogram
10
```

TESTBENCH CODE

testbench.sv | transcation.sv | generator.sv | monitor.sv

scoreboard.sv | test.sv | interface1.sv | drive.sv

envirnoment.sv

```
1  `include "interface1.sv"
2  `include "test.sv"
3  module tbench_top;
4    intf i_intf();
5    test t1(i_intf);
6    priority_encoder h1(
7      .I(i_intf.I),
8      .Y(i_intf.Y),
9      .Valid(i_intf.Valid)
10 );
11 endmodule
```

OUTPUT

```
# Loading work.priority_encoder(fast)
# Loading work.intf(fast)
#
# run -all
# Generator
# I=0101
# Y=00 ,Valid=0
# Driver
# I=0101
# Y=00 ,Valid=0
# Monitor
# I=0101
# Y=10 ,Valid=1
# Scoreboard
# I=0101
# Y=10 ,Valid=1
# ** Note: $finish    : envirnoment.sv(37)
#    Time: 5 ns  Iteration: 0  Instance: /tbench_top/t1
# End time: 01:34:43 on Dec 02,2024, Elapsed time: 0:00:01
# Errors: 0, Warnings: 0
# *** Summary ********************************************
#     qrun: Errors:   0, Warnings:   0
#     vlog: Errors:   0, Warnings:   1
#     vopt: Errors:   0, Warnings:   3
#     vsim: Errors:   0, Warnings:   0
#   Totals: Errors:   0, Warnings:   4
Done
```

EDA PLAYGROUND LINK

https://www.edaplayground.com/x/RDTE

THANK YOU