

Day 13 Topic: Verilog codes for 8 to 3 Priority Encoder Using Cadence (Xcelium)

◆ What is an Encoder?

An encoder is a combinational logic circuit that converts 2^n input lines into n output lines.

- Example: A 4-to-2 encoder has 4 inputs and 2 outputs.
 - Here, we are talking about an 8-to-3 encoder → 8 inputs → 3 outputs.
-

◆ What is a Priority Encoder?

The difference between a normal encoder and a priority encoder is:

- In a normal encoder, if multiple inputs are active (1), the output becomes ambiguous (not correct).
- In a priority encoder, we assign priority to inputs. The highest-priority input will decide the output, even if multiple inputs are active.

 Priority makes the circuit more practical.

◆ 8-to-3 Priority Encoder

- Inputs: D0, D1, D2, D3, D4, D5, D6, D7
- Outputs: Y2, Y1, Y0 (3-bit binary output)
- Extra output: Valid bit (V) → tells whether any input is active.

Priority Rule:

- D7 has the highest priority
- D0 has the lowest priority

That means:

- If D7 = 1, output = 111 (no matter what others are).
 - If D6 = 1 and D7 = 0, output = 110.
 - If D5 = 1 and higher ones are 0, output = 101.
 - ...and so on, until D0.
-

◆ Truth Table (Simplified)

Input Active Output (Y2 Y1 Y0) Valid (V)

D7	111	1
D6	110	1
D5	101	1
D4	100	1
D3	011	1
D2	010	1
D1	001	1
D0	000	1
None	---	0

◆ Output Equations

From the truth table, outputs can be written using OR logic:

- $Y_2 = D_4 + D_5 + D_6 + D_7$
- $Y_1 = D_2 + D_3 + D_6 + D_7$
- $Y_0 = D_1 + D_3 + D_5 + D_7$
- $V = D_0 + D_1 + D_2 + D_3 + D_4 + D_5 + D_6 + D_7$

Here + means OR.

◆ Example

If inputs are:

- $D_5 = 1, D_2 = 1$
Since D5 has higher priority than D2 → Output = 101

If inputs are:

- D0 = 1, D7 = 1
Since D7 has higher priority → Output = 111

◆ 1.1) Design Code

```

priority_encoder.v

// 8-to-3 Priority Encoder
module priority_encoder8x3(
    input [7:0] D,      // 8 inputs
    output reg [2:0] Y, // 3-bit output
    output reg Valid   // indicates if any input is active
);
    always @(*) begin
        casez(D) // z = don't care, helps for priority
            8'b1???????: begin Y = 3'b111; Valid = 1; end
            8'b01???????: begin Y = 3'b110; Valid = 1; end
            8'b001???????: begin Y = 3'b101; Valid = 1; end
            8'b0001???????: begin Y = 3'b100; Valid = 1; end
            8'b00001?????: begin Y = 3'b011; Valid = 1; end
            8'b000001???: begin Y = 3'b010; Valid = 1; end
            8'b0000001?: begin Y = 3'b001; Valid = 1; end
            8'b00000001: begin Y = 3'b000; Valid = 1; end
            default: begin Y = 3'b000; Valid = 0; end
        endcase
    end
endmodule

```

◆ 1.2) Test Bench Code

tb.v

```
module tb_priority_encoder8x3;
    reg [7:0] D;
    wire [2:0] Y;
    wire Valid;

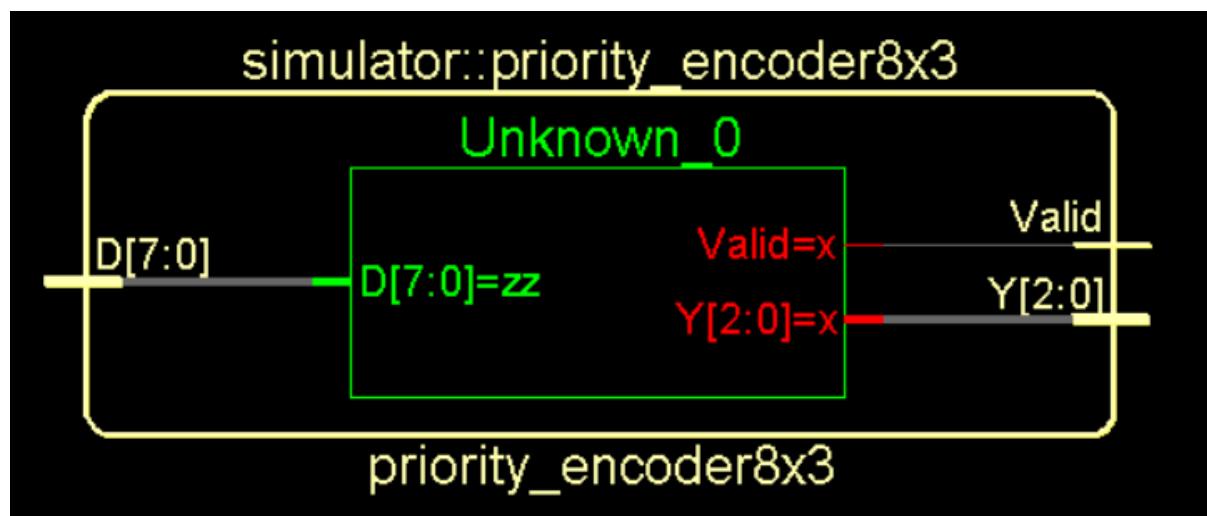
    // Instantiate DUT
    priority_encoder8x3 uut(.D(D), .Y(Y), .Valid(Valid));

    initial begin
        $display("D7..D0 | Y  Valid");
        $display("-----");

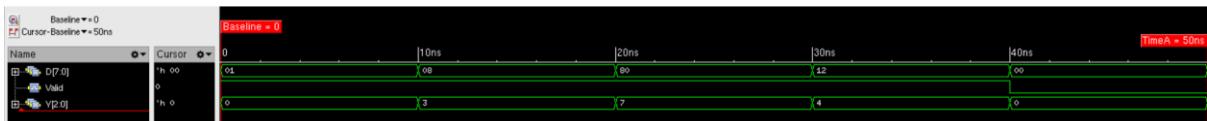
        D = 8'b00000001; #10; // Expect Y=000, Valid=1
        D = 8'b00001000; #10; // Expect Y=011, Valid=1
        D = 8'b10000000; #10; // Expect Y=111, Valid=1
        D = 8'b00010010; #10; // Expect Y=100, Valid=1 (D4 wins over D1)
        D = 8'b00000000; #10; // Expect Y=000, Valid=0

        $stop;
    end
endmodule
```

◆ 1.3) Schematic



◆ 1.4) Wave Forms



◆ Applications

- Used in interrupt handling in microprocessors (highest-priority interrupt is serviced first).
 - Used in communication systems to encode multiple signals.
 - Used in data compression and control systems.

