

Codificador de prioridad

Realizar un codificador de prioridad que emule el comportamiento del circuito integrado 107172, el cual se muestra en la siguiente tabla:

INPUTS									OUTPUTS				
EI	0	1	2	3	4	5	6	7	A2	A1	A0	GS	EO
H	X	X	X	X	X	X	X	X	H	H	H	H	H
L	H	H	H	H	H	H	H	H	H	H	H	H	L
L	X	X	X	X	X	X	X	L	L	L	L	L	H
L	X	X	X	X	X	X	L	H	L	L	H	L	H
L	X	X	X	X	X	L	H	H	L	H	L	L	H
L	X	X	X	X	L	H	H	H	L	H	H	L	H
L	X	X	X	L	H	H	H	H	H	L	L	L	H
L	X	X	L	H	H	H	H	H	H	L	H	L	H
L	X	L	H	H	H	H	H	H	H	H	L	L	H
L	L	H	H	H	H	H	H	H	H	H	H	L	H

Pinout and internal block diagram of the 107172 priority encoder IC. The pinout shows 16 pins: 1 (EI), 2 (A2), 3 (A1), 4 (A0), 5 (GND), 6 (VCC), 7 (E0), 8 (GS), 9 (A0), 10 (A1), 11 (A2), 12 (EI), 13 (GS), 14 (E0), 15 (VCC), 16 (A0). The internal diagram shows a 3-to-2 priority encoder with inputs EI, A2, A1, A0 and outputs E0, GS, A2, A1, A0.

Se realiza el programa mediante una serie de if else para asegurar la prioridad.

```

module Encoder
(
    input [7:0] sel,
    input en,
    output [2:0] A,
    output GS,
    output EO
);
    reg [4:0] out_reg;
    always @(sel, en)
    begin
        if (en)
            out_reg <= 5'b11111;
        else if (sel == 8'b11111111)
            out_reg <= 5'b11110;
        else if (!sel[7]) out_reg <= 5'b00001;
        else if (!sel[6]) out_reg <= 5'b00101;
        else if (!sel[5]) out_reg <= 5'b01001;
        else if (!sel[4]) out_reg <= 5'b01101;
        else if (!sel[3]) out_reg <= 5'b10001;
        else if (!sel[2]) out_reg <= 5'b10101;
        else if (!sel[1]) out_reg <= 5'b11001;
        else if (!sel[0]) out_reg <= 5'b11101;
        else out_reg <= 5'bxxxxx;
    end
end

```

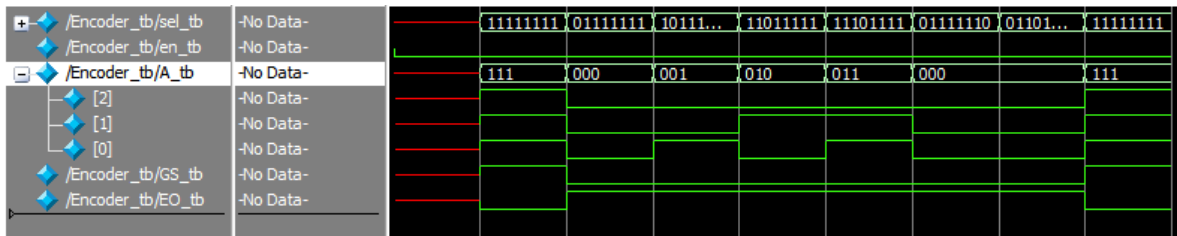
Se puede observar que, para facilitar la lectura del código, se hizo uso de un registro de salidas para posteriormente asignarlas donde corresponden.

```
assign A[2] = out_reg[4];
assign A[1] = out_reg[3];
assign A[0] = out_reg[2];
assign GS = out_reg[1];
assign EO = out_reg[0];

endmodule
```

Para poder probar se realiza una cama de pruebas con las siguientes características.

```
initial begin
    #0 en_tb = 0;
    #10 sel_tb = 8'b11111111;
    #10 sel_tb = 8'b11111110;
    #10 sel_tb = 8'b11111101;
    #10 sel_tb = 8'b11111011;
    #10 sel_tb = 8'b11111110;
    #10 sel_tb = 8'b11101110;
    #10 sel_tb = 8'b01111110;
end
```



Se puede observar que efectivamente el codificador respeta la prioridad ya que a pesar de elevar el bit menos significativo, se toma como prioridad el mas significativo ya que es el que tiene mayor prioridad.