**CDAC Feb 2015**

8 x 3 Priority Encoder

Name: Bhrigu Bhargava

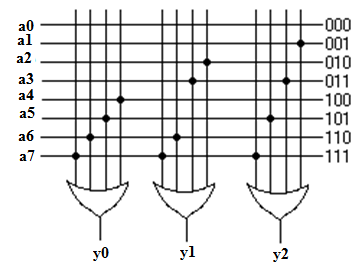
**8 x 3 Priority Encoder using Concurrent Statements**

# Design Approach:

Digital Encoder is a Digital Device opposite of the Digital Decoder. A simple encoder circuit can receive a single active input out of 2n input lines generate a binary code on n parallel output lines. For example a single bit 8 to 3 encoder takes in 8 bits input ( assume **a0,a1,a2,a3,a4,a5,a6,a7**) and 3 outputs (assume **y0,y1,y2**)

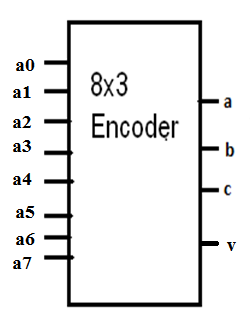
A regular encoder assumes that exactly one input is 1 (while all other inputs are 0). Consider this assumption. Normally, we have 8 inputs, which mean we have 28 or 256 rows. Only 8 rows satisfy the assumption. The truth table for an 8-3 binary encoder (8 inputs and 3 outputs) is shown below. It is assumed that only one input has a value of 1 at any given time, and then the output is invalid.

# Circuit Diagram:

****

**Fig 1:- Circuit Diagram of 8 x 3 Priority Encoder**

**Block Diagram:**

****

**Fig 2:- Block Diagram of 8 x 3 Priority Encoder**

**Truth Table:**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Input | | | | | | | | Output | | |
| Y0 | **Y1** | **Y2** | **Y3** | **Y4** | **Y5** | **Y6** | **Y7** | **a** | **b** | **c** |
| 1 | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| 0 | **1** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **1** |
| 0 | **0** | **1** | **0** | **0** | **0** | **0** | **0** | **0** | **1** | **0** |
| 0 | **0** | **0** | **1** | **0** | **0** | **0** | **0** | **0** | **1** | **1** |
| 0 | **0** | **0** | **0** | **1** | **0** | **0** | **0** | **1** | **0** | **0** |
| 0 | **0** | **0** | **0** | **0** | **1** | **0** | **0** | **1** | **0** | **1** |
| 0 | **0** | **0** | **0** | **0** | **0** | **1** | **0** | **1** | **1** | **0** |
| 0 | **0** | **0** | **0** | **0** | **0** | **0** | **1** | **1** | **1** | **1** |

**Source Code:**

module enco\_p (ip,v,op);

output [2:0] op;

input [7:0] ip;

output v;

wire [2:0] t1;

assign t1=(ip[0]===1)? 0:

(ip[1]===1)? 1:

(ip[2]===1)? 2:

(ip[3]===1) ? 3:

(ip[4]===1) ? 4:

(ip[5]===1) ? 5:

(ip[6]===1)? 6:

(ip[7]===1)? 7:3'bx;

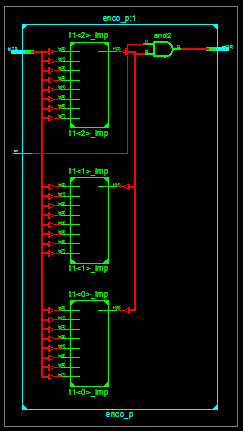
assign op=t1;

assign v=(t1===3'bx)?0:1;

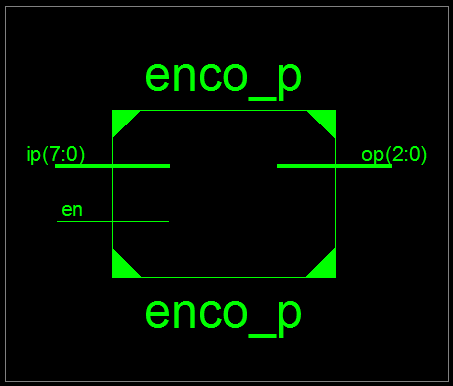
endmodule

**Synthesis:**

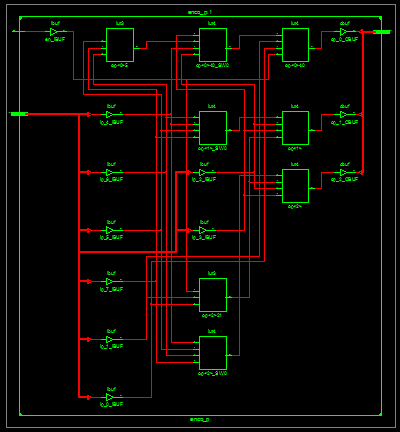
1. RTL Schematic



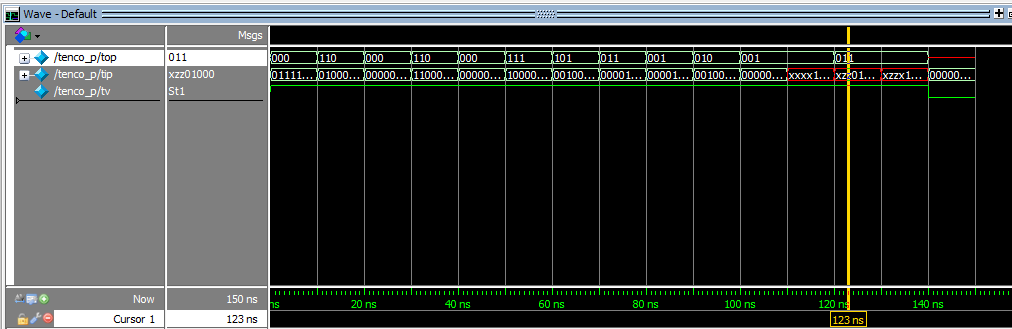
1. Block Diagram

****

1. Tech Schematic



1. Simulation Waveform Result

****

**Error:**

None.

**Verified by:**

Dharamveer Babusen Chindawat (150240133007)