

## DAY #8

# 30 DAYS OF VERILOG

## AIM – TO IMPLEMENT 1x2 DEMUX

A **1x2 demultiplexer (1-to-2 demux)** is a combinational circuit that has **one input line** and **two output lines**. It works in the opposite direction of a multiplexer (MUX).

1. **Inputs and Outputs:**

- **Input (A):** There's a single input line (A).
- **Outputs ( $Y_0$  and  $Y_1$ ):** There are two output lines ( $Y_0$  and  $Y_1$ ).

2. **Selection Line:**

- The demux has **one selection line ( $S_0$ )**.
- Depending on the value of the selection line, the input (A) will be connected to either  $Y_0$  or  $Y_1$ .

3. **Truth Table:**

- When  $S_0 = 0$ , the input is connected to  $Y_0$ .
- When  $S_0 = 1$ , the input is connected to  $Y_1$ .

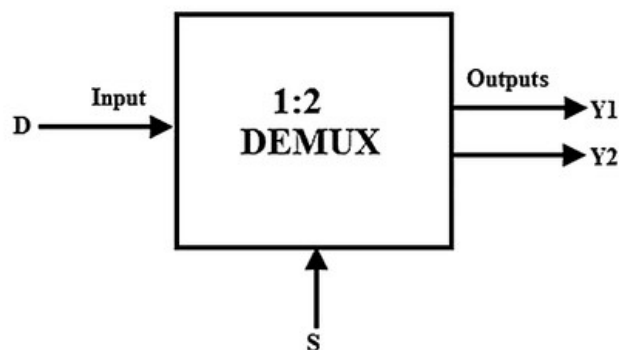
4. **Logical Expressions:**

- $Y_0 = S_0' \cdot A$
- $Y_1 = S_0 \cdot A$
- 

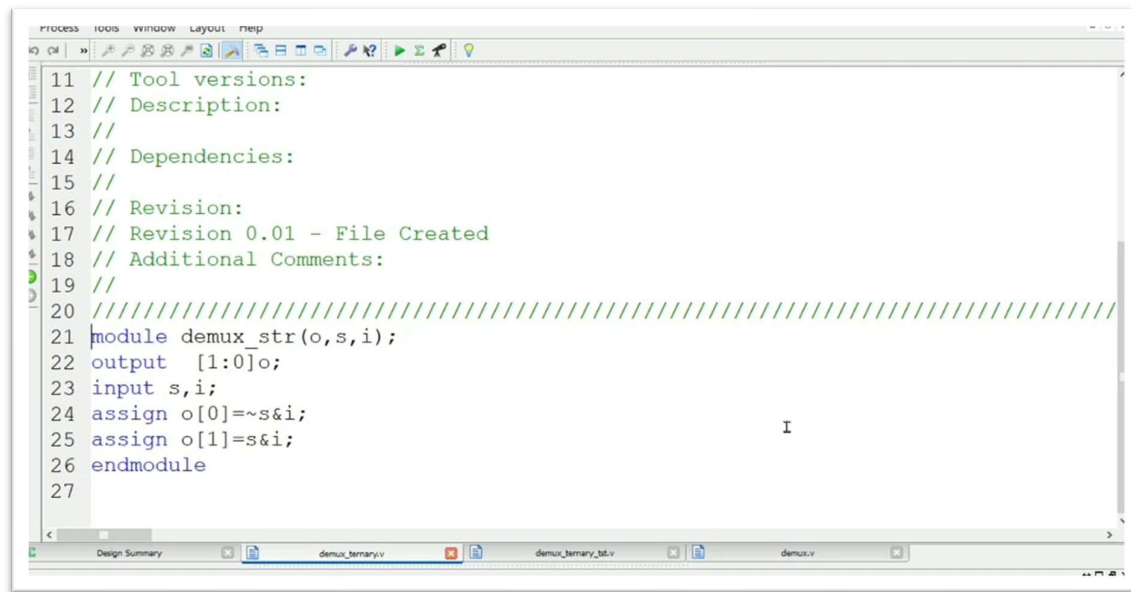
Truth Table –

Select	Input	Outputs	
S	D	$Y_0$	$Y_1$
0	0	0	0
0	1	0	1
1	0	0	0
1	1	1	0

Schematic –



## Verilog Code–



```
11 // Tool versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 ///////////////////////////////////////////////////////////////////
21 module demux_str(o,s,i);
22 output [1:0]o;
23 input s,i;
24 assign o[0]=~s&i;
25 assign o[1]=s&i;
26 endmodule
27
```

The screenshot shows a Verilog code editor with a menu bar (Process, Tools, Window, Layout, Help) and a toolbar. The code defines a module named `demux_str` with two inputs, `s` and `i`, and a two-bit output `o`. The output is calculated using two assign statements: `o[0] = ~s & i` and `o[1] = s & i`. The editor's status bar at the bottom shows the file name `demux_tamary.v`.

## Waveform –

