📘 Day 7 Topic: Verilog codes for DE-MUX (1:2, 1:4, 1:8) Using Cadence (Xcelium)

A **DEMUX** is the reverse of a MUX:

One input → routed to **one of many outputs** based on **select lines**.

1) MUX(1:2)

🔹 1.1) Design Code

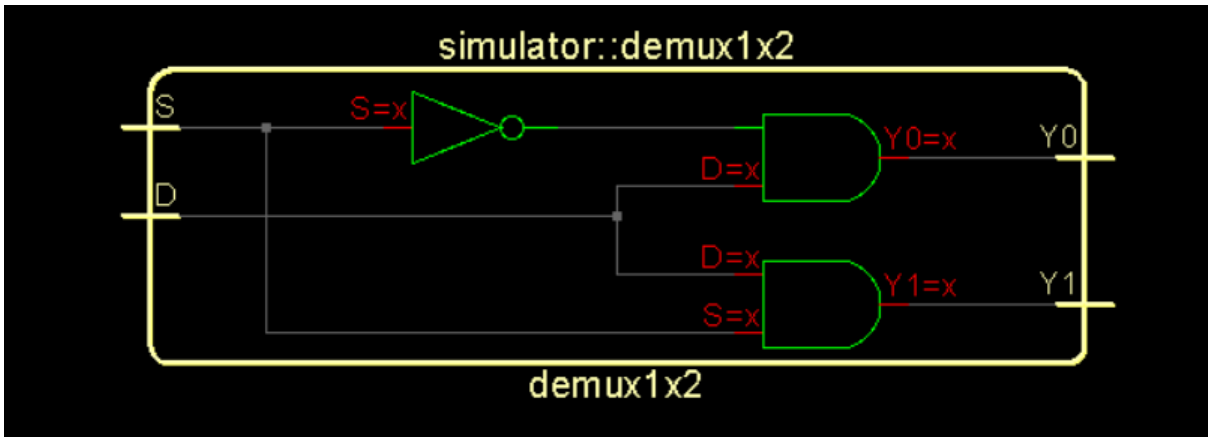
```
// 1x2 DEMUX
module demux1x2(input D, S, output Y0, Y1);
   assign Y0 = (~S) & D;  // If S=0, D goes to Y0
   assign Y1 = ( S) & D;  // If S=1, D goes to Y1
endmodule
```

🔹 1.2) Test Bench Code

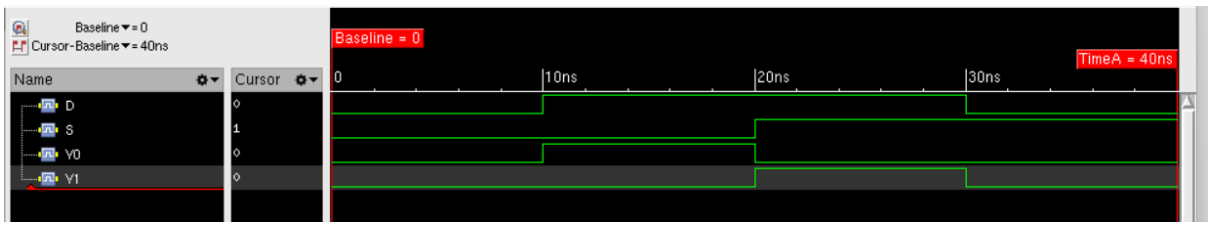
```
`timescale 1ns/1ps
module tb_demux1x2;
   reg D, S;
   wire Y0, Y1;

   demux1x2 uut(.D(D), .S(S), .Y0(Y0), .Y1(Y1));

   initial begin
      $display("S D | Y0 Y1");
      $display("-----------");

      S=0; D=0; #10; $display("%b %b | %b  %b",S,D,Y0,Y1); // 00 → Y0=0, Y1=0
      S=0; D=1; #10; $display("%b %b | %b  %b",S,D,Y0,Y1); // 01 → Y0=1, Y1=0
```

```
        S=1; D=1; #10; $display("%b %b | %b  %b",S,D,Y0,Y1); // 11 → Y0=0, Y1=1
        S=1; D=0; #10; $display("%b %b | %b  %b",S,D,Y0,Y1); // 10 → Y0=0, Y1=0

        $stop;
    end
endmodule
```

- **1.3) Schematic**



- **1.4) Wave Forms**



2) MUX (1:4)

2.1) Design Code

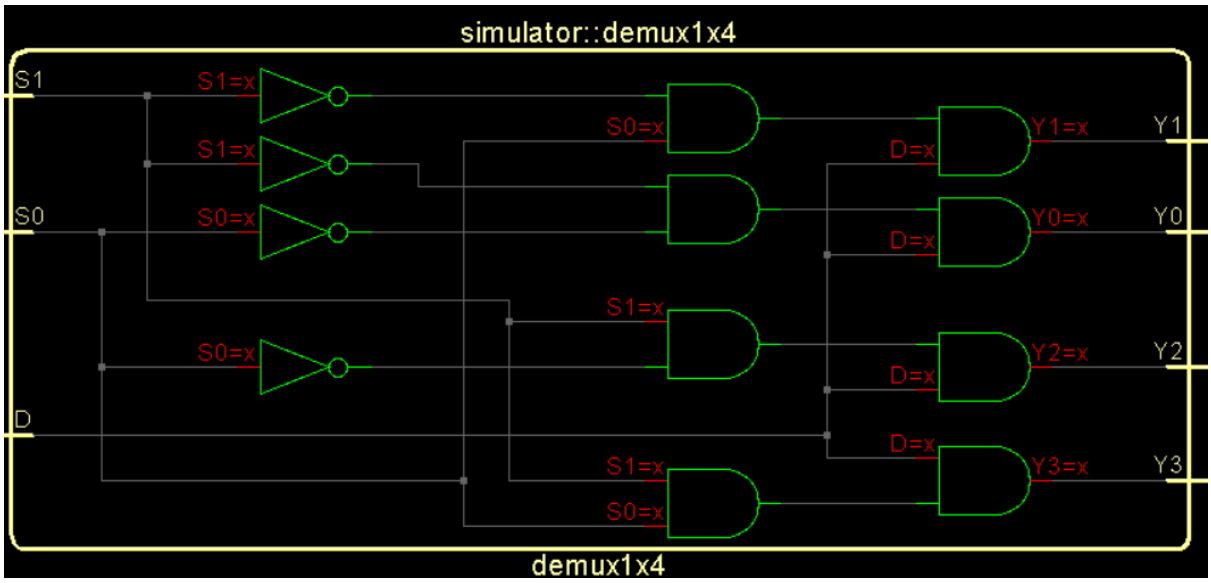
```
// 1x4 DEMUX
module demux1x4(input D, input S0, S1, output Y0, Y1, Y2, Y3);
   assign Y0 = (~S1 & ~S0) & D;
   assign Y1 = (~S1 &  S0) & D;
   assign Y2 = ( S1 & ~S0) & D;
   assign Y3 = ( S1 &  S0) & D;
endmodule
```

2.2) Test Bench Code

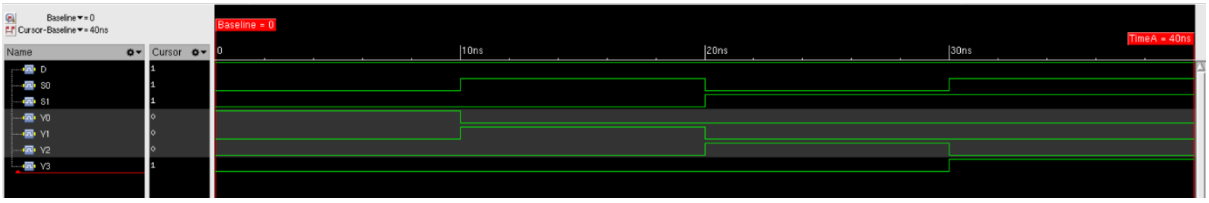
```
`timescale 1ns/1ps
module tb_demux1x4;
   reg D, S0, S1;
   wire Y0, Y1, Y2, Y3;

   demux1x4 uut(.D(D), .S0(S0), .S1(S1), .Y0(Y0), .Y1(Y1), .Y2(Y2), .Y3(Y3));

   initial begin
      $display("S1 S0 D | Y0 Y1 Y2 Y3");
      $display("---------------------");

      D=1;
      {S1,S0}=2'b00; #10; $display("%b  %b  %b | %b  %b  %b  %b",S1,S0,D,Y0,Y1,Y2,Y3);
      {S1,S0}=2'b01; #10; $display("%b  %b  %b | %b  %b  %b  %b",S1,S0,D,Y0,Y1,Y2,Y3);
      {S1,S0}=2'b10; #10; $display("%b  %b  %b | %b  %b  %b  %b",S1,S0,D,Y0,Y1,Y2,Y3);
      {S1,S0}=2'b11; #10; $display("%b  %b  %b | %b  %b  %b  %b",S1,S0,D,Y0,Y1,Y2,Y3);
```

```
    $stop;
  end
endmodule
```

- **2.3) Schematic**



- **2.4) Wave Form**



3) MUX (1:8)

- **3.1) Design Code**

```
// 1x8 DEMUX
module demux1x8(input D, input S0, S1, S2,
                output Y0, Y1, Y2, Y3, Y4, Y5, Y6, Y7);
  assign Y0 = (~S2 & ~S1 & ~S0) & D;
```

```
    assign Y1 = (~S2 & ~S1 &  S0) & D;
    assign Y2 = (~S2 &  S1 & ~S0) & D;
    assign Y3 = (~S2 &  S1 &  S0) & D;
    assign Y4 = ( S2 & ~S1 & ~S0) & D;
    assign Y5 = ( S2 & ~S1 &  S0) & D;
    assign Y6 = ( S2 &  S1 & ~S0) & D;
    assign Y7 = ( S2 &  S1 &  S0) & D;
endmodule
```

- **3.2) Test bench Code**

```
`timescale 1ns/1ps
module tb_demux1x8;
    reg D, S0, S1, S2;
    wire Y0,Y1,Y2,Y3,Y4,Y5,Y6,Y7;

    demux1x8 uut(.D(D), .S0(S0), .S1(S1), .S2(S2),
                 .Y0(Y0), .Y1(Y1), .Y2(Y2), .Y3(Y3),
                 .Y4(Y4), .Y5(Y5), .Y6(Y6), .Y7(Y7));

    initial begin
        $display("S2 S1 S0 D | Y0 Y1 Y2 Y3 Y4 Y5 Y6 Y7");
        $display("-----------------------------------");

        D=1;
        {S2,S1,S0}=3'b000; #10; $display("%b  %b  %b  %b | %b %b %b %b %b %b %b %b",S2,S1,S0,D,Y0,Y1,Y2,Y3,Y4,Y5,Y6,Y7);
        {S2,S1,S0}=3'b001; #10; $display("%b  %b  %b  %b | %b %b %b %b %b %b %b %b",S2,S1,S0,D,Y0,Y1,Y2,Y3,Y4,Y5,Y6,Y7);
        {S2,S1,S0}=3'b010; #10; $display("%b  %b  %b  %b | %b %b %b %b %b %b %b %b",S2,S1,S0,D,Y0,Y1,Y2,Y3,Y4,Y5,Y6,Y7);
        {S2,S1,S0}=3'b011; #10; $display("%b  %b  %b  %b | %b %b %b %b %b %b %b %b",S2,S1,S0,D,Y0,Y1,Y2,Y3,Y4,Y5,Y6,Y7);
```

{S2,S1,S0}=3'b100; #10; $display("%b %b %b %b | %b %b %b %b %b %b %b %b",S2,S1,S0,D,Y0,Y1,Y2,Y3,Y4,Y5,Y6,Y7);

{S2,S1,S0}=3'b101; #10; $display("%b %b %b %b | %b %b %b %b %b %b %b %b",S2,S1,S0,D,Y0,Y1,Y2,Y3,Y4,Y5,Y6,Y7);

{S2,S1,S0}=3'b110; #10; $display("%b %b %b %b | %b %b %b %b %b %b %b %b",S2,S1,S0,D,Y0,Y1,Y2,Y3,Y4,Y5,Y6,Y7);

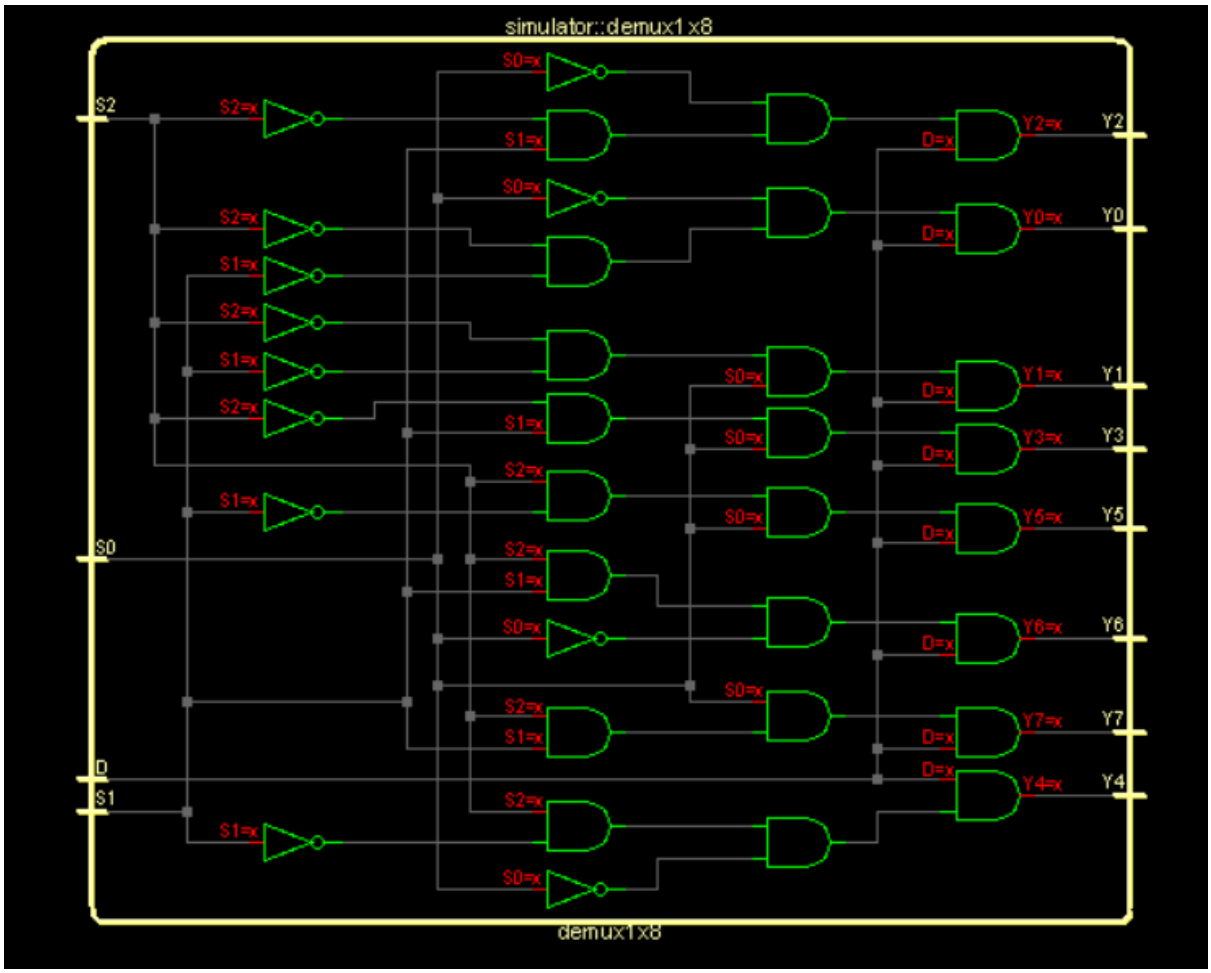
{S2,S1,S0}=3'b111; #10; $display("%b %b %b %b | %b %b %b %b %b %b %b %b",S2,S1,S0,D,Y0,Y1,Y2,Y3,Y4,Y5,Y6,Y7);

$stop;

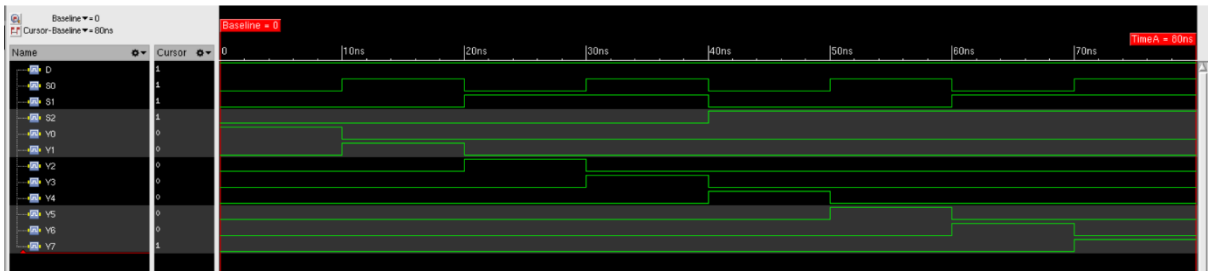
end

endmodule

- **3.3) Schematic**



3.4) Wave Form


Baseline = 0
Cursor-Baseline = 80ns
Name
Cursor
D
S0
S1
S2
Y0
Y1
Y2
Y3
Y4
Y5
Y6
Y7
Baseline = 0
TimeA = 80ns
0
10ns
20ns
30ns
40ns
50ns
60ns
70ns