

RTL CHALLENGE

DAY-11:- Verilog Code for Implementation of DeMultiplexer

Software used:- Xilinx Vivado(2023.1)

Theory:

Demultiplexer:-

A demultiplexer, often abbreviated as "demux," is a crucial electronic component in digital circuit design that performs the opposite function of a multiplexer. While a multiplexer selects one of several input signals and routes it to a single output line based on a control signal, a demultiplexer takes a single input signal and distributes it across multiple output lines based on a control signal. Demultiplexers are particularly useful for tasks such as data distribution, signal routing, and data decompression in digital systems. They are commonly employed in combination with multiplexers to enable bidirectional data transmission and efficient resource utilization within digital circuits. With their ability to efficiently distribute signals to multiple destinations, demultiplexers contribute significantly to the versatility and functionality of digital systems across various applications, including telecommunications, multimedia processing, and memory systems.

Code for Demux 1 to 2:-

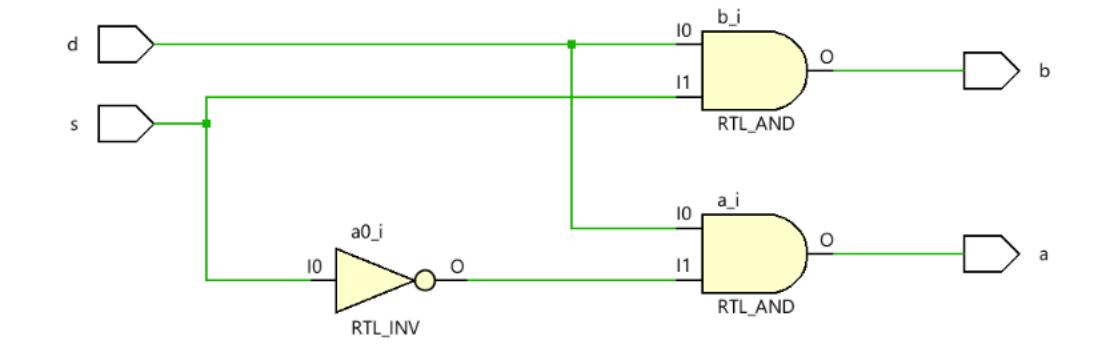
```
module Demux1to2 (
    input d,s,
    output a,b
) ;
and (a,d,!s) ;
and (b,d,s) ;

endmodule
```

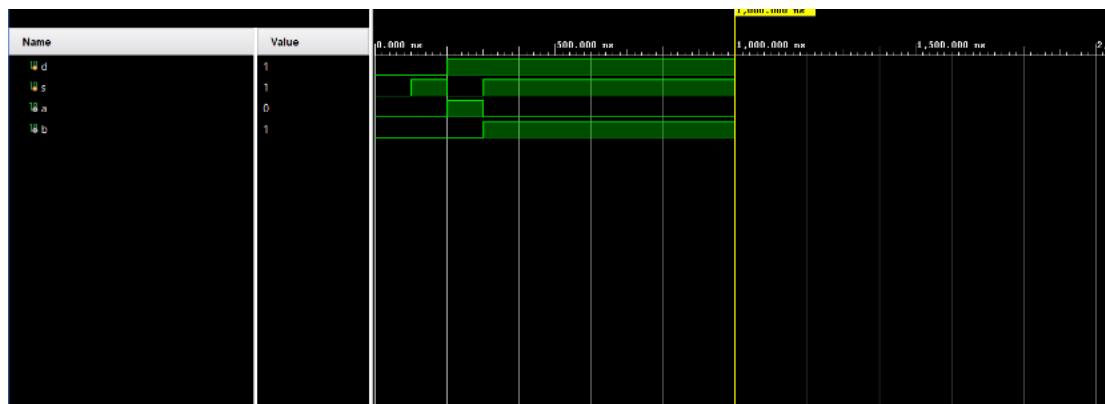
Test Bench:-

```
module Demux1to2_tb;
reg d,s;
wire a,b;
Demux1to2 uut(.d(d),.s(s),.a(a),.b(b));
initial
begin
    d=0;s=0;#100;
    d=0;s=1;#100;
    d=1;s=0;#100;
    d=1;s=1;#100;
end
endmodule
```

Schematic:-



Waveforms:-



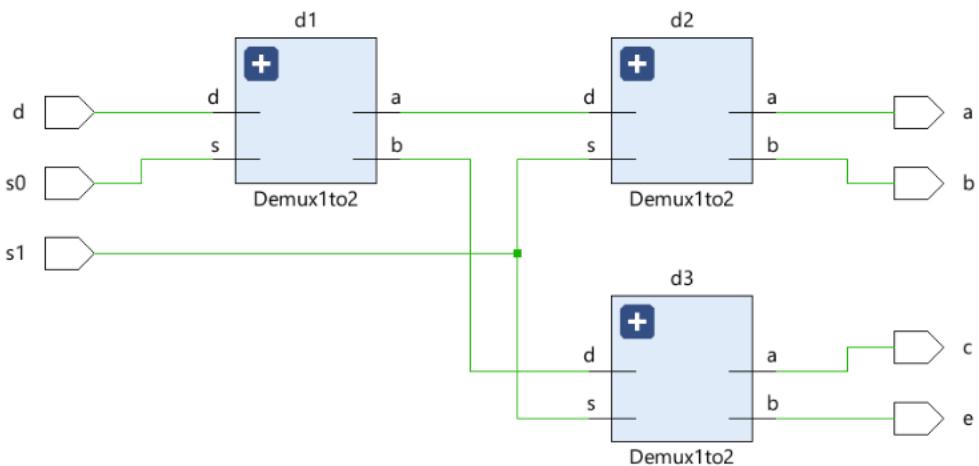
Code for Demux 1 to 4:-

```
module Demux1to4(
    input d,s0,s1,
    output a,b,c,e
);
    wire w1,w2;
    Demux1to2 d1 (.d(d),.s(s0),.a(w1),.b(w2));
    Demux1to2 d2 (.d(w1),.s(s1),.a(a),.b(b));
    Demux1to2 d3 (.d(w2),.s(s1),.a(c),.b(e));
endmodule
```

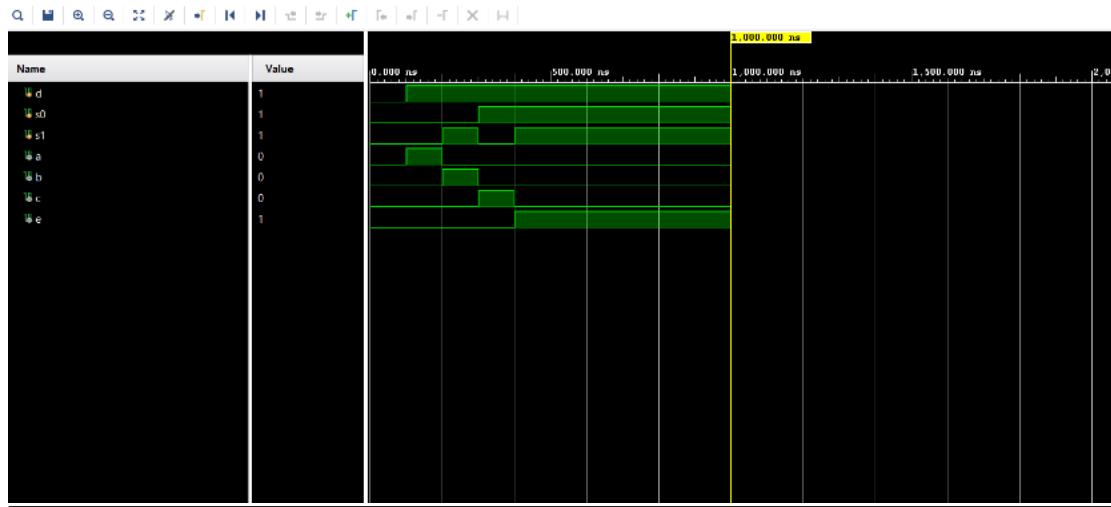
Test Bench:-

```
module Demux1to4_tb;
reg d,s0,s1;
wire a,b,c,e;
Demux1to4 uut(.d(d),.s0(s0),.s1(s1),.a(a),.b(b),.c(c),.e(e));
initial
begin
    d=0;s0=0;s1=0;#100;
    d=1;s0=0;s1=0;#100;
    d=1;s0=0;s1=1;#100;
    d=1;s0=1;s1=0;#100;
    d=1;s0=1;s1=1;#100;
end
endmodule
```

Schematic:-



Waveforms:-



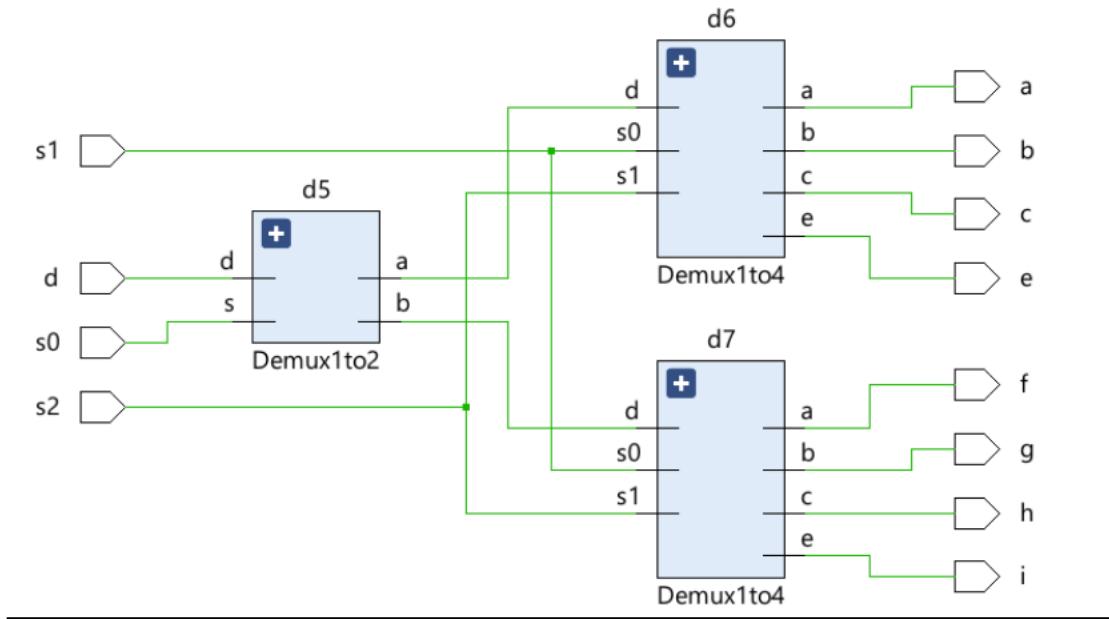
Code for Demux 8 to 1:-

```
module Demux1to8(
    input d,s0,s1,s2,
    output a,b,c,e,f,g,h,i
);
    Demux1to2 d5 (.d(d),.s(s0),.a(w1),.b(w2));
    Demux1to4 d6 (.d(w1),.s0(s1),.s1(s2),.a(a),.b(b),.c(c),.e(e));
    Demux1to4 d7 (.d(w2),.s0(s1),.s1(s2),.a(f),.b(g),.c(h),.e(i));
endmodule
```

Test Bench:-

```
module Demux1to8_tb;
reg d,s0,s1,s2;
wire a,b,c,e,f,g,h,i;
Demux1to8 uut(.d(d),.s0(s0),.s1(s1),.s2(s2),.a(a),.b(b),.c(c),.e(e),.f(f),.g(g),.h(h),.i(i));
initial
begin
    d=0;s0=0;s1=0;s2=0;#100;
    d=1;s0=0;s1=0;s2=0;#100;
    d=1;s0=0;s1=0;s2=1;#100;
    d=1;s0=0;s1=1;s2=0;#100;
    d=1;s0=0;s1=1;s2=1;#100;
    d=1;s0=1;s1=0;s2=0;#100;
    d=1;s0=1;s1=0;s2=1;#100;
    d=1;s0=1;s1=1;s2=0;#100;
    d=1;s0=1;s1=1;s2=1;#100;
end
endmodule
```

Schematic:-



Waveforms:-

