4-1 Mux

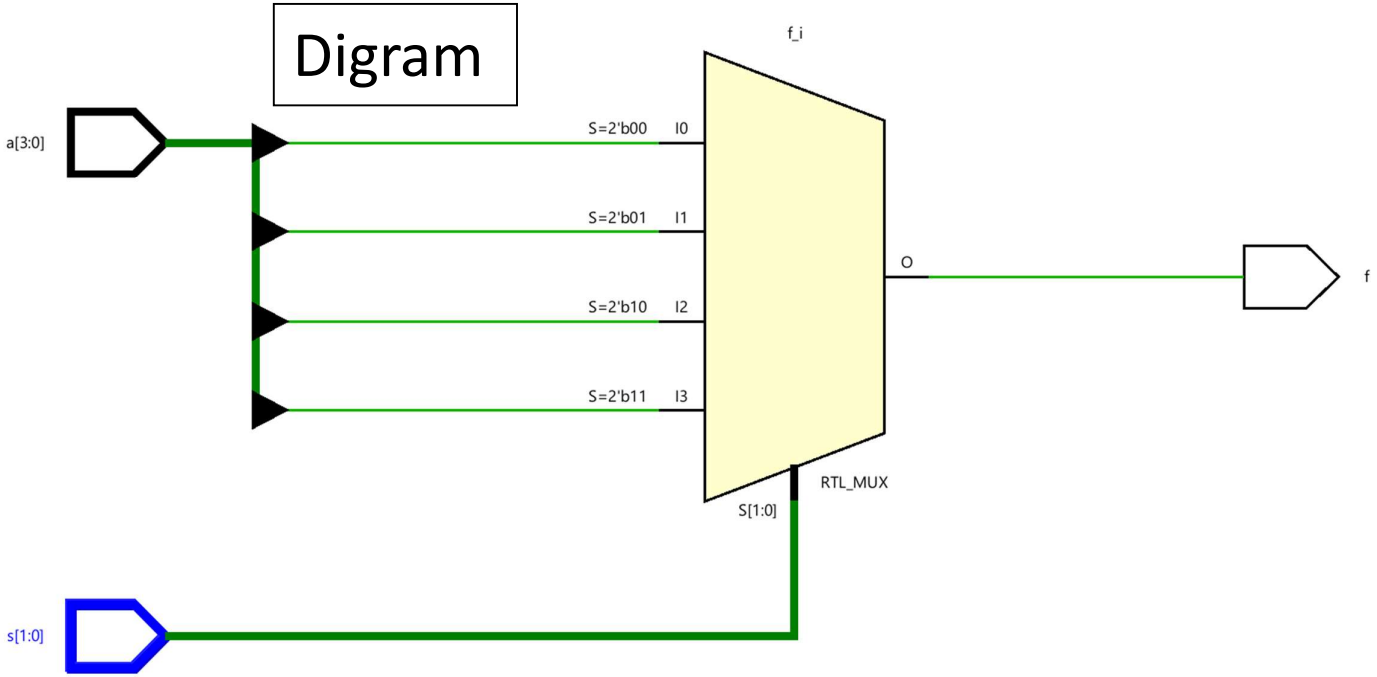
Project Summary | Design.v

```
1  // 4:1 MUX
2  module mux(s,a,f);
3    input [3:0]a;
4    input [1:0]s;
5    output reg f;
6    always @(s or a)
7      begin
8        case(s)
9          2'b00: f=a[0];
10         2'b01: f=a[1];
11         2'b10: f=a[2];
12         2'b11: f=a[3];
13         default: f=1'bx;
14       endcase
15     end
16 endmodule
```

Testbench.v

```
1  //4:1 MUX TB
2  module tb;
3    reg[3:0]a;
4    reg[1:0]s;
5    wire f;
6    mux u1(.f(f),.a(a),.s(s));
7    initial
8      begin
9      a= 4'b1011;
10           s=2'b00;
11       #10; s=2'b01;
12       #10; s=2'b10;
13       #10; s=2'b11;
14     end
15   initial
16     begin
17       $monitor("s=%0b,f=%0b",s,f);
18     end
19 endmodule
```

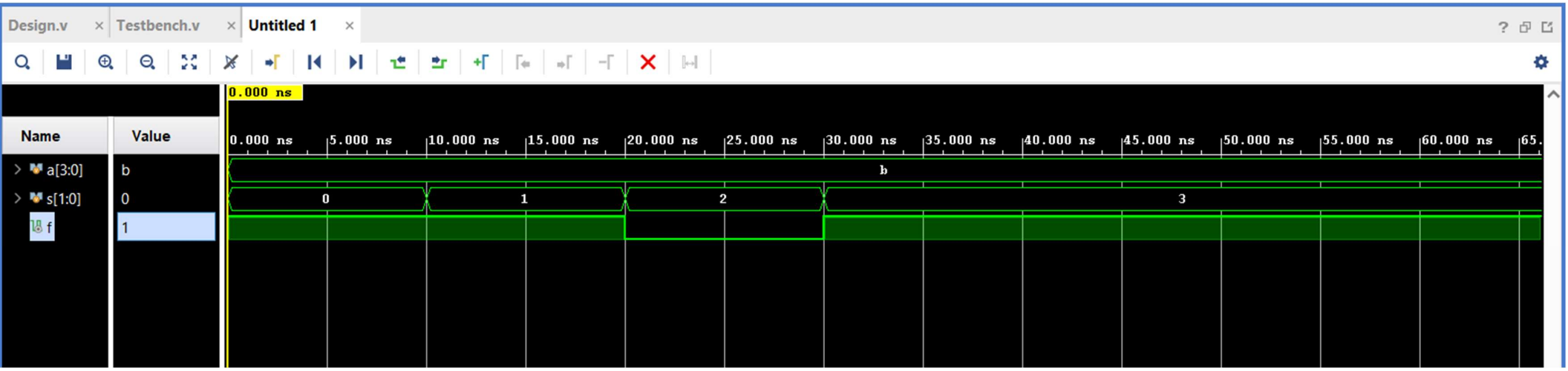
Digram



OUTPUT

```
# KERNEL: s=0,f=1
# KERNEL: s=1,f=1
# KERNEL: s=10,f=0
# KERNEL: s=11,f=1
```

Timing Digram



A **4-to-1 multiplexer (MUX)** is a combinational circuit that selects one of four input lines and routes it to a single output based on two select lines. It is a fundamental component in digital systems for data routing and signal management.

Functionality

- **Inputs:** Four data lines (D_0, D_1, D_2, D_3) and two select lines (S_0, S_1).
- **Output:** A single line (Y) that carries the selected input.
- **Enable Signal:** Some MUX designs include an enable (EN) pin to activate/deactivate the circuit.

Truth Table

S_1	S_0	Output (Y)
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

The Boolean expression is:
$Y=S1\bar{}S0\bar{}D0+S1\bar{}S0D1+S1S0\bar{}D2+S1S0D3Y=S1S0D0+S1S0D1+S1S0D2+S1S0D3$.

Implementations

1. Gate-Level Design

Uses AND, OR, and NOT gates to implement the Boolean expression:

- Four 3-input AND gates (one per input line).
- One 4-input OR gate for the output.

- NOT gates for select line inversions.

Key features:

- Uses case statements for clarity.
- Enable pin (EN) deactivates output when low.

Applications

1. **Data Routing:** Directs signals in communication systems.
2. **Memory Addressing:** Selects memory locations in microprocessors.
3. **Sensor Networks:** Aggregates data from multiple sensors.
4. **FPGA Design:** Used in reconfigurable logic blocks34.

Popular ICs

- **74LS153:** Dual 4-to-1 MUX with shared select lines.
- **SN74LVC1G157:** Single 4-to-1 MUX for low-voltage systems.