DAY-19
#100DAYSOFRTL

PROBLEM STATEMENT:--

1. Create a one-bit wide, 2-to-1 multiplexer. When sel=0, choose a. When sel=1, choose b.

Write your solution here

[Load a previous submission] Load

```
module top_module(
    input a, b, sel,
    output out );

    assign out=((sel==0)?a:b);


endmodule

```

Submit | Submit (new window)

Upload a source file...

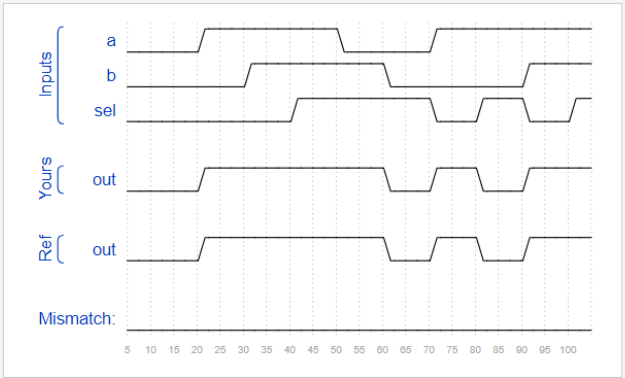
Status: Success!

You have solved 51 problems. See my progress...

Timing diagrams for selected test cases

These are timing diagrams from some of the test cases we used. They may help you debug your circuit. The diagrams show inputs to the circuit, outputs from your circuit, and the expected reference outputs. The "Mismatch" trace shows which cycles your outputs don't match the reference outputs (0 = correct, 1 = incorrect).

Sel chooses between a and b



2. Create a 100-bit wide, 2-to-1 multiplexer. When sel=0, choose a. When sel=1, choose b.

Write your solution here

[Load a previous submission] Load

```
module top_module(
    input [99:0] a, b,
    input sel,
    output [99:0] out );

    assign  out=((sel==0)?a:b);

endmodule
```

Submit | Submit (new window)

Upload a source file...

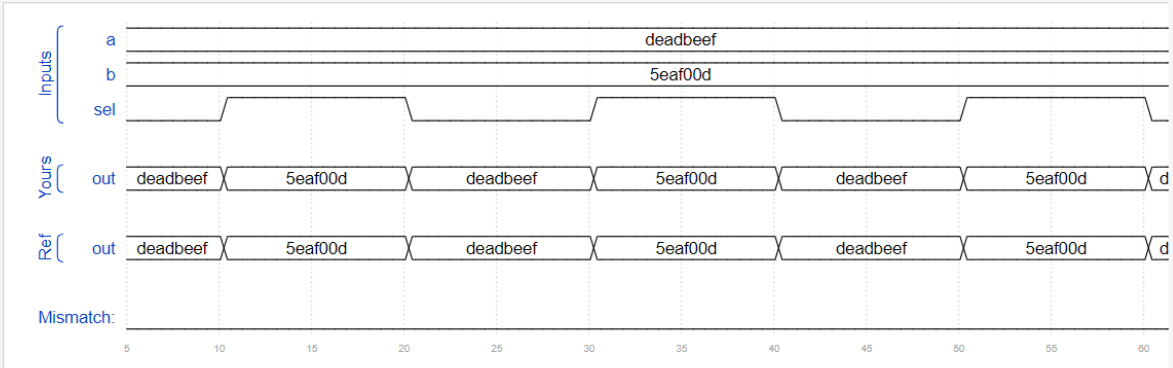
Status: Success!

You have solved 52 problems. See my progress...

Timing diagrams for selected test cases

These are timing diagrams from some of the test cases we used. They may help you debug your circuit. The diagrams show inputs to the circuit, outputs from your circuit, and the expected reference outputs. The "Mismatch" trace shows which cycles your outputs don't match the reference outputs (0 = correct, 1 = incorrect).

Beef or seafood?



3. Create a 16-bit wide, 9-to-1 multiplexer. sel=0 chooses a, sel=1 chooses b, etc. For the unused cases (sel=9 to 15), set all output bits to '1'.

```
module top_module(
    input [15:0] a, b, c, d, e, f, g, h, i,
    input [3:0] sel,
    output [15:0] out );

    always @(*) begin
    // Default output to all '1' for unused cases
    out = 16'hFFFF;

    // Efficient case statement for selecting inputs
    case (sel)
      4'b0000: out = a;
      4'b0001: out = b;
      4'b0010: out = c;
      4'b0011: out = d;
      4'b0100: out = e;
      4'b0101: out = f;
      4'b0110: out = g;
      4'b0111: out = h;
      4'b1000: out = i;
      default: out = out;

    endcase
        end

       endmodule

```

Status: Success!

You have solved 53 problems. See my progress...

Timing diagrams for selected test cases

These are timing diagrams from some of the test cases we used. They may help you debug your circuit. The diagrams show inputs to the circuit, outputs from your circuit, and the expected reference outputs. The "Mismatch" trace shows which cycles your outputs don't match the reference outputs (0 = correct, 1 = incorrect).