1. Simple Router

Description | Discussion | Solution

Easy

Prompt

Build a router circuit which forwards data from the input (`din`) to one of four outputs (`dout0`, `dout1`, `dout2`, or `dout3`), specified by the address input (`addr`).

The address is a two bit value whose decimal representation determines which output value to use. Append to `dout` the decimal representation of `addr` to get the output signal name `dout{address decimal value}`. For example, if `addr=b11` then the decimal representation of `addr` is `3`, so the output signal name is `dout3`.

The input has an enable signal (`din_en`), which allows the input to be forwarded to an output when enabled. If an output is not currently being driven to, then it should be set to `0`.

```
1  module model #(parameter
2    DATA_WIDTH = 32
3  ) (
4    input  [DATA_WIDTH-1:0] din,
5    input  din_en,
6    input  [1:0] addr,
7    output logic [DATA_WIDTH-1:0] dout0,
8    output logic [DATA_WIDTH-1:0] dout1,
9    output logic [DATA_WIDTH-1:0] dout2,
10   output logic [DATA_WIDTH-1:0] dout3
11 );
12
13 assign dout0 = (din_en && addr == 2'h0) ? din : '0;
14 assign dout1 = (din_en && addr == 2'h1) ? din : '0;
15 assign dout2 = (din_en && addr == 2'h2) ? din : '0;
16 assign dout3 = (din_en && addr == 2'h3) ? din : '0;
17
18 always_comb begin
19   dout0 = DATA_WIDTH'(32'hdeadbeef);
20   dout1 = DATA_WIDTH'(32'hdeadbeef);
21   dout2 = DATA_WIDTH'(32'hdeadbeef);
22   dout3 = DATA_WIDTH'(32'hdeadbeef);
23 end
24
25 endmodule
```

Testcase | Simulation

Success: 101 of 101 passed.

Reset | Run | Submit