

Structural Example

module majority (major, V1, V2, V3);

output major;

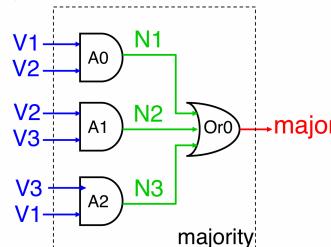
input V1, V2, V3;

wire N1, N2, N3;

and A0 (N1, V1, V2),
A1 (N2, V2, V3),
A2 (N3, V3, V1);

or Or0 (major, N1, N2, N3);

endmodule

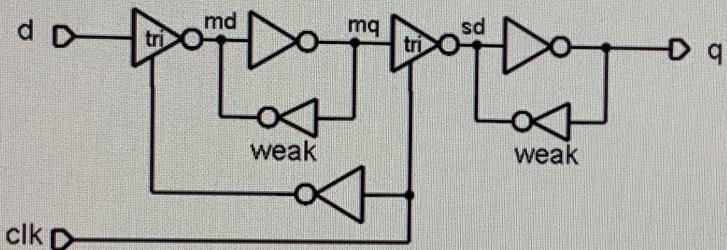


ECE 551 Exercise 02

Master-slave latch

This is the last problem of HW1 ^{pos}_{edge} (Flip flop)

Submit whatever you have for both the DUT and the testbench to the dropbox by end of class time.



- Code this circuit structurally in Verilog (in Modelsim environment) (attach the Verilog code for this module) (Note: tri-state gates might need a 1 time unit delay modeled for proper simulation, an inverting tri-state is verilog primitive `notif1`).
- Create a testbench, instantiate and simulate this circuit. Attach a print out of the waveforms, and the Verilog of the test bench
- Add an active high asynchronous reset input to this circuit. Code this new circuit in Verilog and update your test bench to test the new input. Again attach Verilog and waveforms
- Was the cost of adding the asynch reset high, medium, or low? Think in terms of number of transistors.

Test Bench

module mystery_tb();

reg D_stim;

Superset

reg clk;

wire q_out; // DUT will drive

System logic d_stim, clk, q_out;

// Instantiate DUT//

mystery ; DUT(.clk(clk), .d(d_stim), .q(q_out))

initial begin

clk = 0;

d_stim = 0;

#10; ^{generic} @ (posedge clk); #1;

clk = 1;

if (q_out != 1'60) begin

~~#1;~~

// is q_out == 0

~~#10~~

~~clk=0;~~

end

always

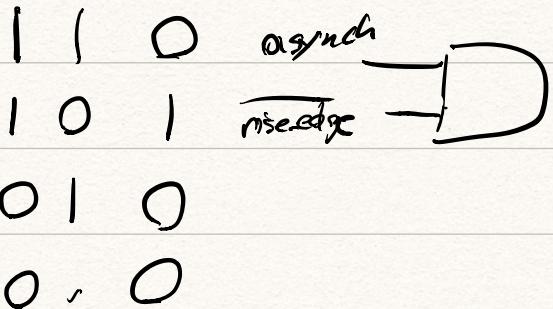
#5 clk = ~clk;

endmodule

Brute force \$display ("ERR:

q_out should be 0"
\$stop();
end

asynch nseedge



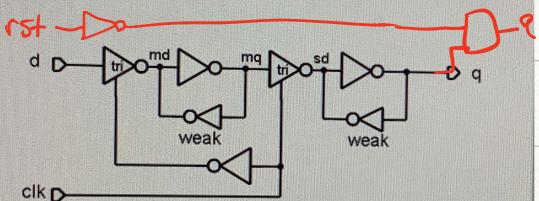
Asynch reset

ECE 551

Exercise02

This is the last problem of HW1

Submit whatever you have for both the DUT and the testbench to the dropbox by end of class time.



- a) Code this circuit structurally in Verilog (in Modelsim environment)(attach the Verilog code for this module)(Note: tri-state gates might need a 1 time unit delay modeled for proper simulation, an inverting tri-state is verilog primitive `notifl`).
- b) Create a testbench, instantiate and simulate this circuit. Attach a print out of the waveforms, and the Verilog of the test bench
- c) Add an active high asynchronous reset input to this circuit. Code this new circuit in Verilog and update your test bench to test the new input. Again attach Verilog and waveforms
- d) Was the cost of adding the asynch reset high, medium, or low? Think in terms of number of transistors.