1111 (fifteen).

"A" on input pattern 1010 (ten) and "F" on pattern.

Extend this Verilog code to also show the letter.

Review Problem 19
True if an odd number of inputs are true

EX. Debug a 9-input odd parity circuit

Rip up and retry?

Complex circuits require careful debugging

Debugging Complex Circuits
Debugging Complex Circuits (cont.)
Debugging Approach

- Test all behaviors.
  - All combinations of inputs for small circuits, subcircuits.

- Identify any incorrect behaviors.

- Examine inputs and outputs to find earliest place where value is wrong.
  - Typically, trace backwards from bad outputs, forward from inputs.
  - Look at values at intermediate points in circuit.

- DO NOT RIP UP, DEBUG!
Seat Belt Light

Circuit
Logic
Passenger
Pbelt
Dbelt

(Dbelt, Pbelt, Passenger) mapped into (light)

E.g., seat belt light: outputs are a pure function of the inputs; no feedback among inputs and outputs.

Combinational Logic

Feedback

Network
Logic

and combinational networks distinguish between sequential.
The presence of feedback network implemented from logic gates.

Readings: 3.1, 3.2.6, 3.2.8

Combinational vs. Sequential Logic
Must filter out temporary states

Circuit can temporarily go to incorrect states

Hazards/Glitches
Safe Sequential Circuits

- Clocked elements hide glitches/hazards
- Clock signal synchronizes operation
- Clocked elements on feedback, perhaps outputs
module D FF
begin
    input logic d, clk;
    output logic q;

    always @(posedge clk) begin
        $posedge q = d;
    end
endmodule

Basic D Flip Flop
D Flip Flop w/ Synchronous Reset

module D_FF (q, d, reset, ctk);
    input logic d, reset, ctk;
    output logic q;
always @(posedge ctk)
begin
    if (reset)
        q <= 0;
    else
        q <= d;
end
endmodule
module Testbench;

parameter ClockDelay = 100;

logic clk, reset, d, q;

endmodule

end

stop() ; // end the simulation

@posedge clk : d = 0; @posedge clk : d = 1; @posedge clk:
reset @posedge clk : d = 0; reset @posedge clk:
d = 0; reset @posedge clk:
d = 1; @posedge clk:

// set up the reset signal
int start; // begin

for ( forever @posedge (ClockDelay/2) clk : ~clk )

// set up the clock

D FF d out (q, d, reset, clk); // Instantiate the D FF

parameter ClockDelay = 100;

logic clk, reset, d, q;

endmodule stimulud:

Verilog Testbench