Review Problem

- Solve the following K-Map.

\[ f = \overline{C \overline{D}} + \overline{R \overline{D}} \]
Review Problem

- Convert the following circuit to NAND/NOR form

\[ \overline{A + \overline{B}} = \overline{\overline{A} \overline{B}} \]
Combinational vs. Sequential Logic

- **Readings:** 5-5.4.4

Network implemented from logic gates. The presence of feedback distinguishes between *sequential* and *combinational* networks.

**Combinational logic**
- no feedback among inputs and outputs
- outputs are a pure function of the inputs
- e.g., seat belt light:
  - (Dbelt, Pbelt, Passenger) mapped into (Light)

![Logic Circuit Diagram]
Hazards/Glitches

- Circuit can temporarily go to incorrect states

Must filter out temporary states
Safe Sequential Circuits

- Clocked elements on feedback, perhaps outputs
  - Clock signal synchronizes operation
  - Clocked elements hide glitches/hazards

![Logic Network Diagram]

Clock signal synchronizes operation and hides glitches/hazards in the logic network. The clock signal ensures that the data is processed at the right time, preventing any unwanted behavior or errors in the system.
Basic D Flip Flop

// Basic D flip-flop

module basic_D_FF (q, d, clk);
    output q;
    input d, clk;
    reg q; // Indicate that q is stateholding (POSITIVE EDGE TRIGGERED)
    always @(posedge clk)
        q <= d; // ALWAYS use <= to assign to clocked elements
    endmodule
D Flip Flop w/Synchronous Reset

// D flip-flop w/synchronous reset

module D_FF (q, d, reset, clk);
    output q;
    input d, reset, clk;
    reg q; // Indicate that q is stateholding

    always @(posedge clk)
        if (reset)
            q <= 0; // On reset, set to 0
        else
            q <= d; // Otherwise out = d

endmodule
module stimulus;
    reg clk, reset, d;
    wire q;

    parameter ClockDelay = 100;

    D_FF dut (.q, .d, .reset, .clk); // Instantiate the D FF

    initial clk <= 0;// Set up the clock
    always #(ClockDelay/2) clk <= ~clk; // Toggle clock every 50 time units

    initial // Set up the reset signal
    begin
        d <= 0; reset <= 1; @(posedge clk);
        reset <= 0; @(posedge clk);
        d <= 1; @(posedge clk);
        d <= 0; @(posedge clk);
        @(posedge clk);
        $stop(); // end the simulation
    end

endmodule
Testbench Waveforms
Flipflop Realities 1: Gating the Clock

NEVER put a logic gate between the clock and DFF’s CLK input.
Flipflop Realities 2: Clock Period, Applying Stimulus

- Clock Period?
- Apply Inputs when?
Flipflops require their inputs be stable for time period around clock edge.
Timing Definitions

- $T_{\text{setup}}$: Time $D$ must be stable BEFORE clock edge
  - Adds to critical path delay

- Clk-$\rightarrow$Q: Time from clock edge to Q changing
  - Adds to critical path delay

- $T_{\text{hold}}$: Time $D$ must be stable AFTER clock edge
  - Sets minimum path from Q of one DFF to D of another
Flipflop Realities 3: External Inputs

- External inputs aren’t synchronized to the clock
Dealing with Metastability

■ Single DFF

External

D flipflop

D                   Q

Clk

■ 2 DFFs in series

External

D flipflop

D                   Q

D flipflop

D                   Q

Clk

Clk

■ 2 DFFs in parallel

External

D flipflop

D                   Q

Clk

D flipflop

D                   Q

Clk