Lab 0: DFF and Counter

Kinner Parikh

September 6, 2022

CMPEN 331 - 001

1 Code

DFF Explanation: Set output(q) to input(ns) on the positive edge of the clock and when the clear is on, otherwise set the output to 000.

Counter Explanation: The counter follows the diagram, in that it increments when u is equal to 1 and decrements when u is equal to 0. However, in the case of u being 1 and ns being 101, we set ns to 000, and when u being 0 and ns being 000, we set ns to 101. And based on the current input q, we must change a, b, c, d, e, f, g to their corresponding defined values to represent the output on the 7-segment display.

Testbench Explanation: We first define all the components of the circuit: DFF object, counter object, registers and wires that connect up the modules. We initially set the clear and clock to 0 and 1 respectively, the counter input u to 1, then we wait 1ms and set the clear to 1 and 16ms later set the counter input u to 0. Then we have a loop where we wait for 1ms, then flip the clock.

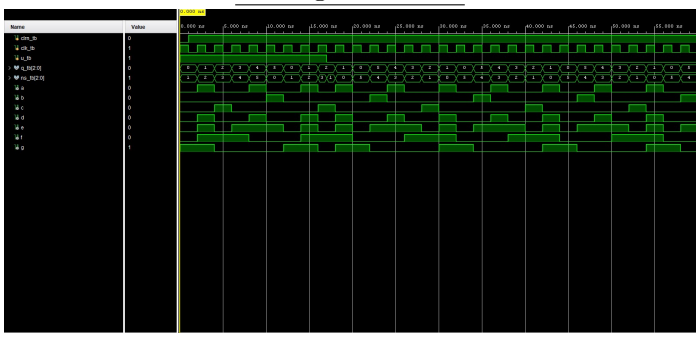
```
‘timescale 1ns / 1ps
// D-Flip Flop -> Author: provided
module dff3(input [2:0] ns, input clrn, input clk, output reg [2:0] q);
    always @ (posedge clk)
        begin
            if (clrn == 1) begin
                q <= ns;
            end
            else begin
                q <= 3’b000;
            end
        end
endmodule

// COUNTER -> Author: provided
module counter(input [2:0] q, input u, output reg [2:0] ns,
       output reg a, output reg b, output reg c, output reg d,
       output reg e, output reg f, output reg g );
   always @(*)
        begin
            if (u == 1) begin
                if (q == 3’b101) begin
                    ns <= 3’b000;
                end
                else begin
```

```
                ns <= q + 1;
            end
        end
        else begin
            if (q == 3'b000) begin
                ns <= 3'b101;
            end
            else begin
                ns <= q - 1;
            end
        end
    case(q)
        3'b000: begin
            g=1'b1;  f=1'b0;  e=1'b0;  d=1'b0;
            c=1'b0;  b=1'b0;  a=1'b0;
        end
        3'b001: begin
            g=1'b1;  f=1'b1;  e=1'b1;  d=1'b1;
            c=1'b0;  b=1'b0;  a=1'b1;
        end
        3'b010: begin
            g=1'b0;  f=1'b1;  e=1'b0;  d=1'b0;
            c=1'b1   b=1'b0;  a=1'b0;
        end
        3'b011: begin
            g=1'b0;  f=1'b1;  e=1'b1;  d=1'b0;
            c=1'b0;  b=1'b0;  a=1'b0;
        end
        3'b100: begin
            g=1'b0;  f=1'b0;  e=1'b1;  d=1'b1;
            c=1'b0;  b=1'b0;  a=1'b1;
        end
        3'b101: begin
            g=1'b0;  f=1'b0;  e=1'b1;  d=1'b0;
            c=1'b0;  b=1'b1;  a=1'b0;
        end
    endcase
  end
endmodule
```

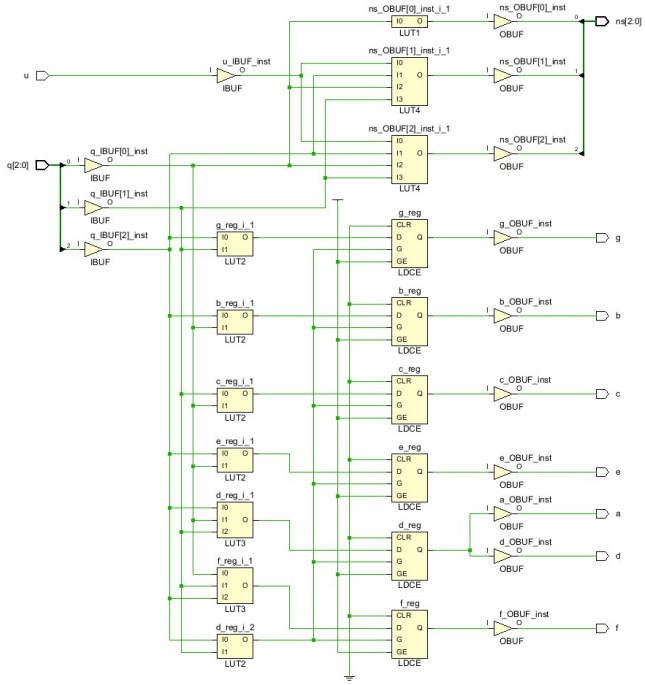
```
‘timescale 1ns / 1ps

// TESTBENCH -> Author: provided
module testbench();
    reg clrn_tb;
    reg clk_tb;
    reg u_tb;
    wire [2:0] q_tb;
    wire [2:0] ns_tb;
    wire a,b,c,d,e,f,g;
    dff3 dff3_tb(ns_tb, clrn_tb, clk_tb, q_tb);
    counter counter_tb(q_tb, u_tb, ns_tb, a, b, c, d, e, f, g);
    initial begin
        clrn_tb = 0;
        clk_tb = 1;
        u_tb = 1;
        #1 clrn_tb = 1;
        #16 u_tb = 0;
    end
    always begin
        #1;
        clk_tb = ~clk_tb;
    end
endmodule
```

2 Images

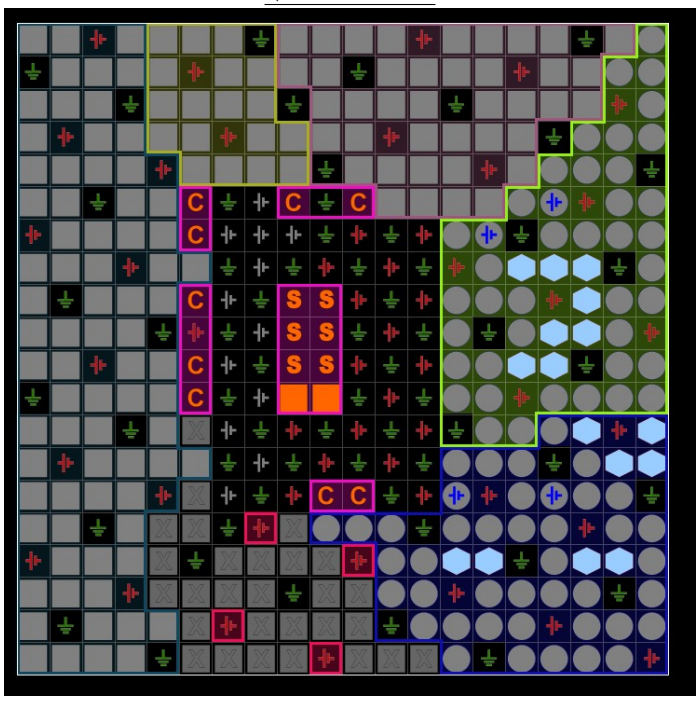
Timing Waveform



Schematic



I/O Planning



Floor Planning