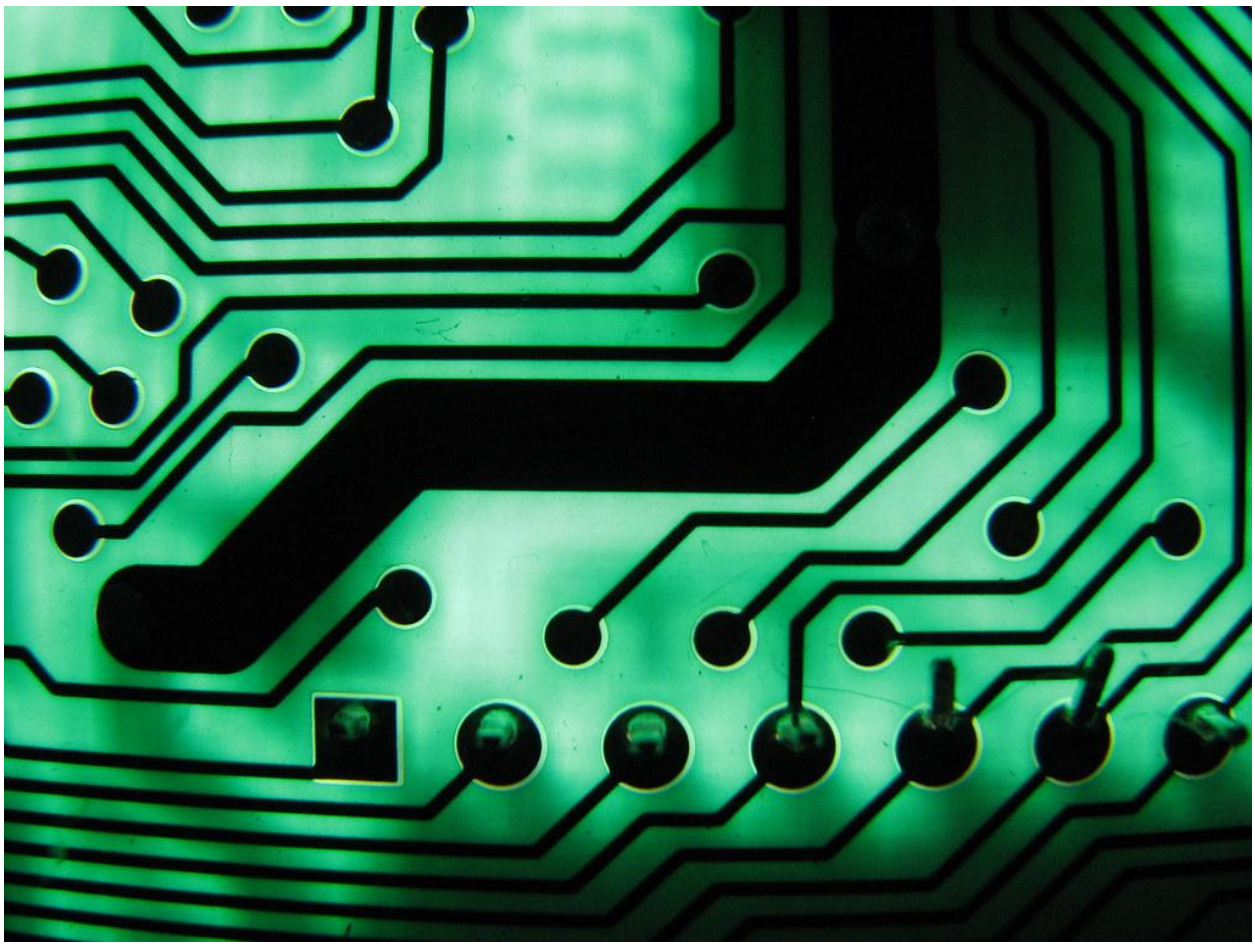
EEC 180 DIGITAL SYSTEMS II

Lab #4

Lab Report



Introduction

The purpose of this lab is to program an FPGA to do shift register implementation.

Tobin Joseph

Results

1 Prelab

Part 1.1 RE_DFF.v Draft

```
// Verilog code for rising edge D flip flop
module RE_DFF(D,clk,Q);
input D; // Data input
input clk; // clock input
output reg Q; // output Q

always @(posedge clk)
begin
      Q <= D;
end
endmodule
```

Figure 1.1.1 RE_DFF.v Draft Code.

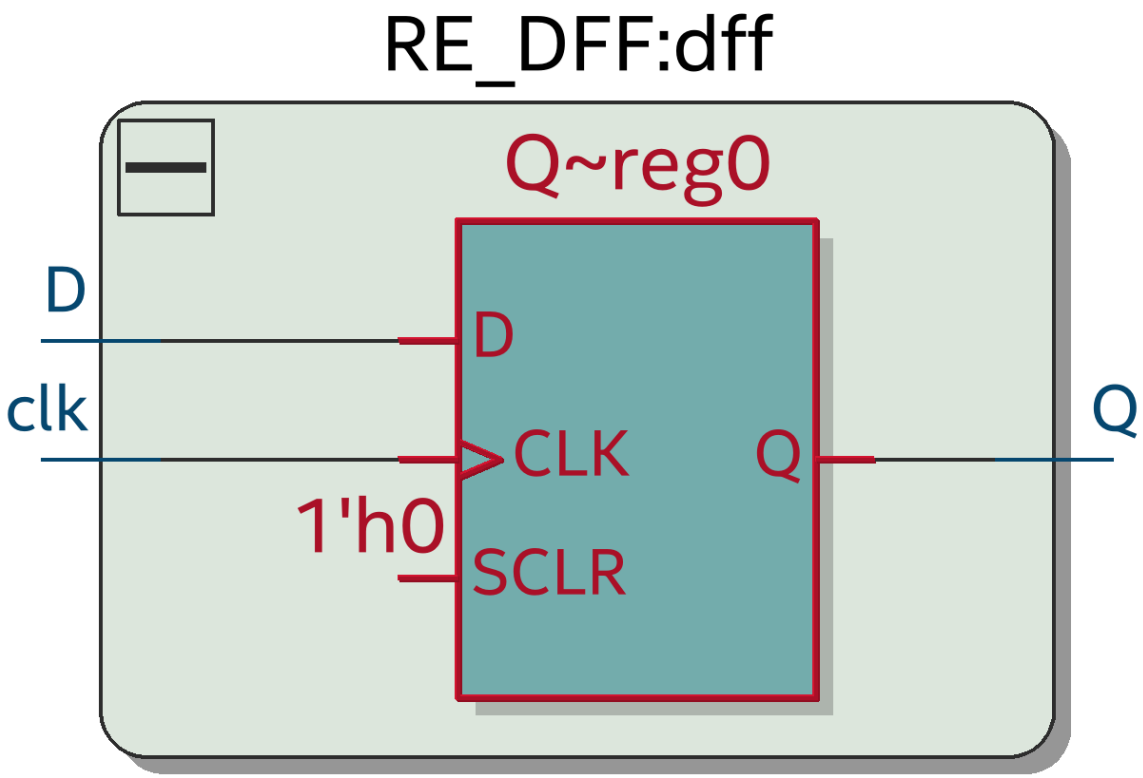


Figure 1.1.2 RE_DFF.V circuit.

Part 1.2 add3.v Draft

```
module mux2_1(Y, D0, D1, S);
output Y;
input D0, D1, S;
wire T1, T2, Sbar;

not (Sbar, S);
and (T1, D1, S), (T2, D0, Sbar);
or (Y, T1, T2);
endmodule
```

Figure 1.2.1 mux2_1.v Draft Code.

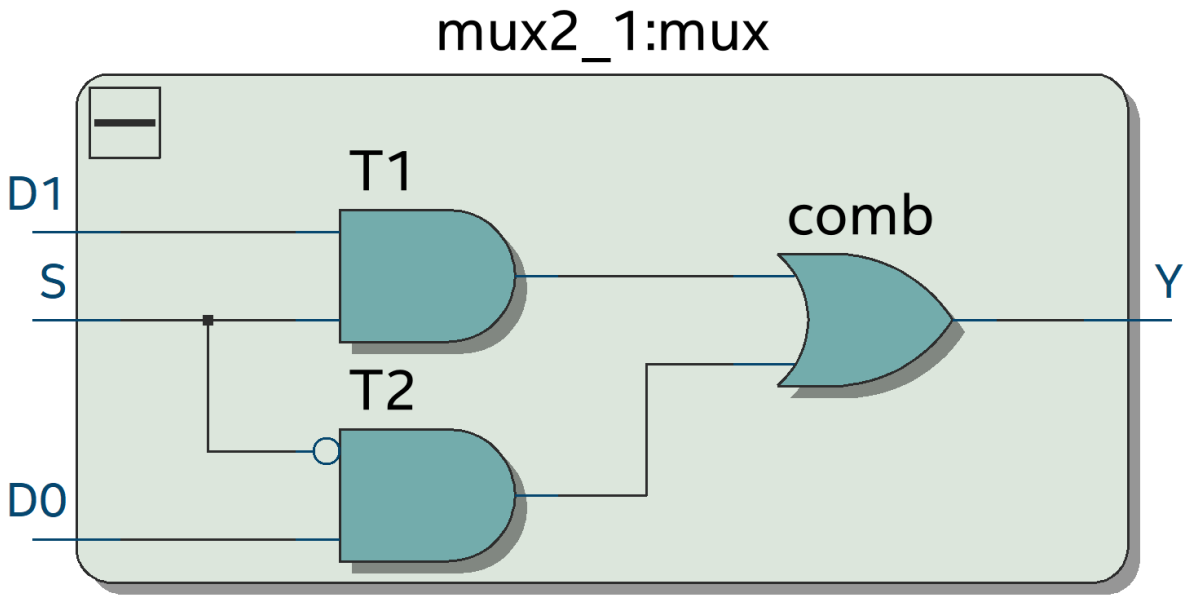


Figure 1.2.2 `mux2_1.v` Circuit.

Part 1.3 `m21_DFF.v` Draft

```
module m21_DFF(Q,D,SW,Load,clock);

output Q;

input wire D;

input wire SW;

input wire Load;

input wire clock;


wire selD;


mux2_1 mux(selD,D,SW,Load);

RE_DFF dff(selD,clock,Q);


endmodule
```

Figure 1.3.1 `m21_DFF.v` Draft Code.

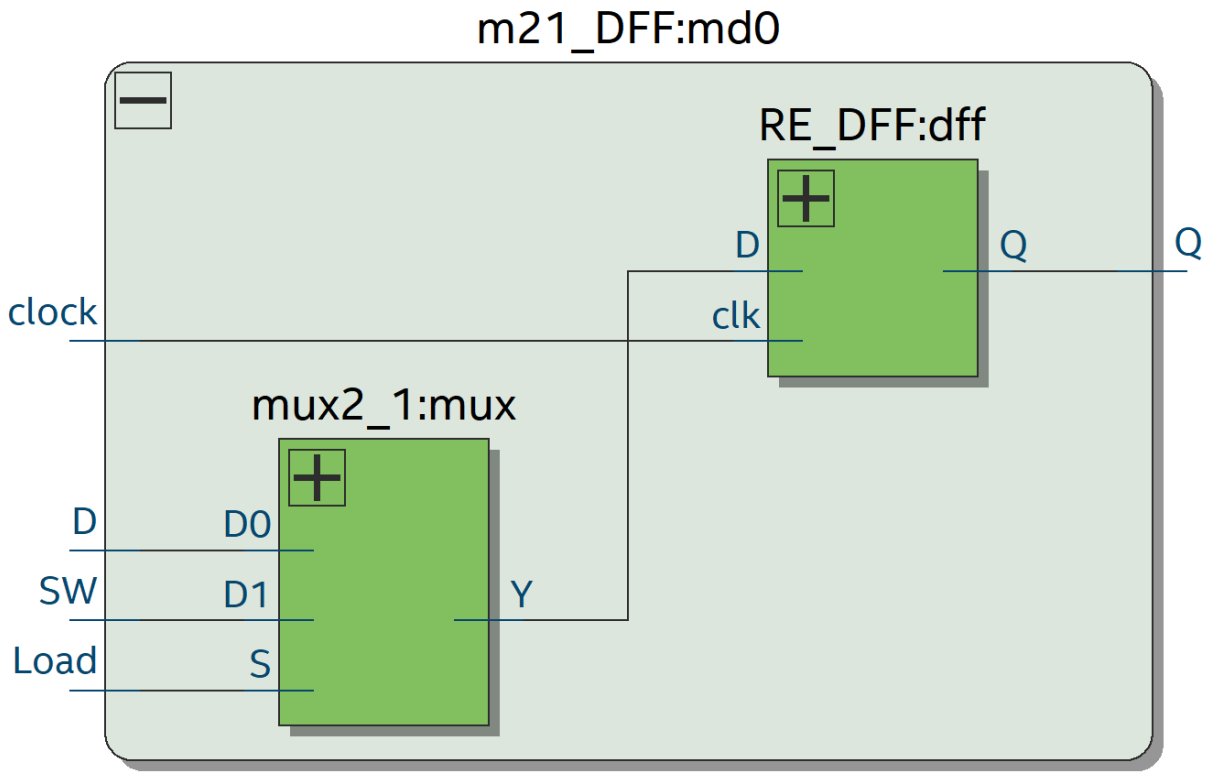


Figure 1.3.2 `m21_DFF.v` Block Diagram.

Part 1.4 `shift_DFF.v` Draft

```
module shift_DFF(LEDR,clock,SW,Load);


output wire [9:0] LEDR;

input wire clock;

input [9:0] SW;

inout wire Load;



// Only in the case when all LEDs are off,

// the circuit will inject a one into the shift register chain


wire D9;
```

```
assign D9 =
~(LEDR[9]|LEDR[8]|LEDR[7]|LEDR[6]|LEDR[5]|LEDR[4]|LEDR[3]|LEDR[2]|LEDR[1]|LED
R[0]);


m21_DFF md9(LEDR[9], D9, SW[9], Load, clock);

m21_DFF md8(.Q(LEDR[8]), .D(LEDR[9]), .SW(SW[8]), .Load(Load),
.clock(clock));

m21_DFF md7(.Q(LEDR[7]), .D(LEDR[8]), .SW(SW[7]), .Load(Load),
.clock(clock));

m21_DFF md6(.Q(LEDR[6]), .D(LEDR[7]), .SW(SW[6]), .Load(Load),
.clock(clock));

m21_DFF md5(.Q(LEDR[5]), .D(LEDR[6]), .SW(SW[5]), .Load(Load),
.clock(clock));

m21_DFF md4(.Q(LEDR[4]), .D(LEDR[5]), .SW(SW[4]), .Load(Load),
.clock(clock));

m21_DFF md3(.Q(LEDR[3]), .D(LEDR[4]), .SW(SW[3]), .Load(Load),
.clock(clock));

m21_DFF md2(.Q(LEDR[2]), .D(LEDR[3]), .SW(SW[2]), .Load(Load),
.clock(clock));

m21_DFF md1(.Q(LEDR[1]), .D(LEDR[2]), .SW(SW[1]), .Load(Load),
.clock(clock));

m21_DFF md0(.Q(LEDR[0]), .D(LEDR[1]), .SW(SW[0]), .Load(Load),
.clock(clock));




endmodule
```

Figure 1.4.1 `shift_DFF.v` Draft Code.

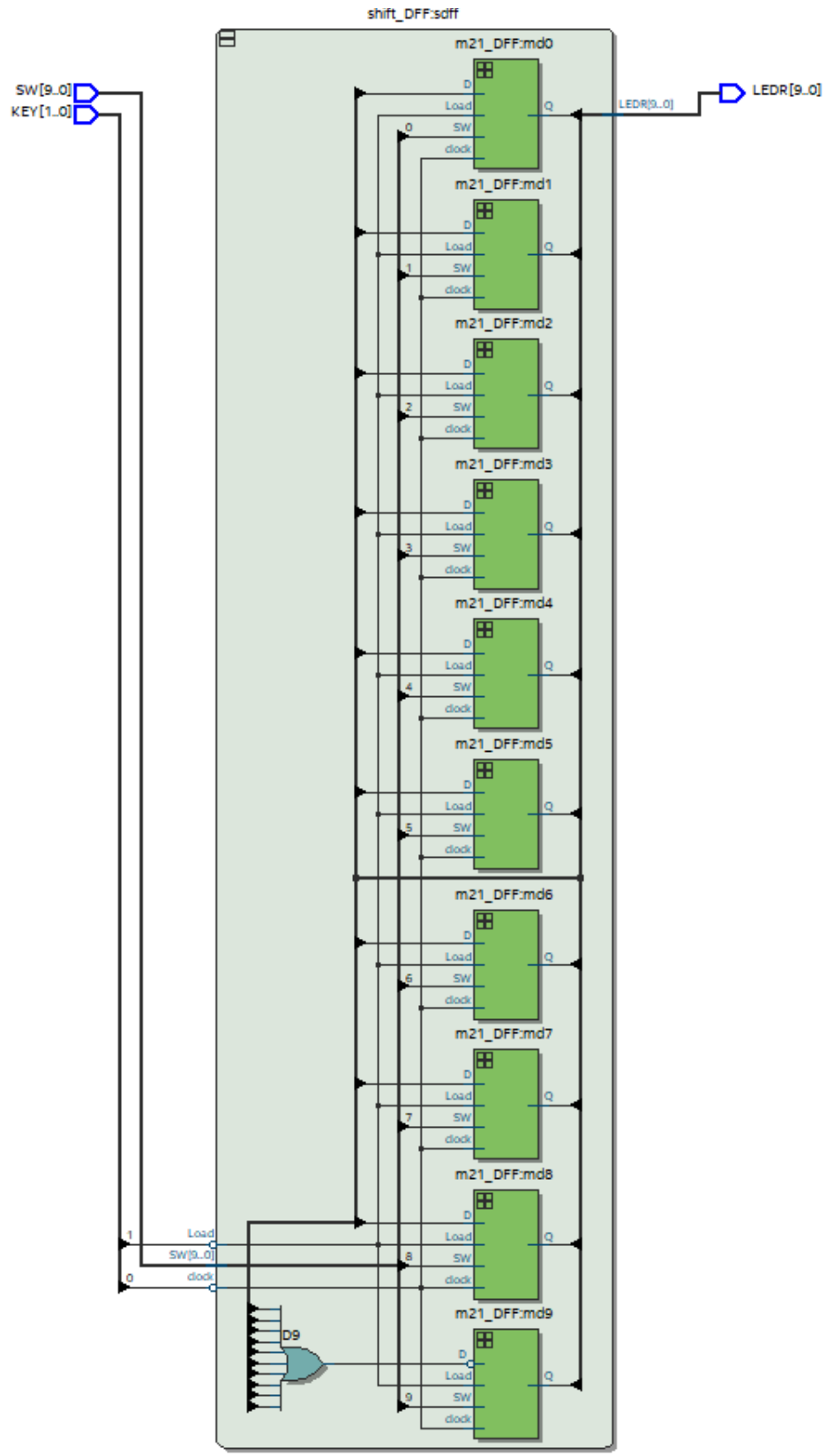


Figure 1.4.2 `shift_DFF.v` Block Diagram.

Part 1.5 tb_shift_DFF.v Draft

```
module tb_shift_DFF();

reg clk;
wire [9:0] LEDR;
reg [9:0] SW;
wire Load;

shift_DFF dff(LEDR,clk,SW,Load);

initial begin
  clk=0;
     forever #10 clk = ~clk;
end
initial begin
 Load = 1;
 SW = 10'b0000000000;
 #100;
 Load = 1;
 SW = 10'b1111111111;
 #100;
 Load <= 0;
 #100;
 Load = 0;
 #100;
 Load = 0;
```

```
#100;
Load = 0;
#100;
Load = 0;
#100;
Load = 0;
#100;
Load = 0;
#100;
Load = 0;
#100;
Load = 0;
#100;
Load = 0;
#100;
Load = 0;
#100;
Load = 0;
#100;
Load = 0;
#100;
Load = 0;
#100;
Load = 0;
#100;
Load = 0;
#100;
Load = 0;
#100;
```

```
 Load = 0;
 #100;
 Load = 0;
 #100;
 Load = 0;
 #100;
 Load = 0;

end
endmodule
```

Figure 1.5.2 tb_shift_DFF.v Draft Code.

Discussion

When we ...

Conclusion

We can see ...