Description

In this lab, we will implement a D flip-flop behaviorally using SystemVerilog and use several D flip-flops to implement a clock divider that blinks the LEDs on the Basys3 board. After completing this lab, I should be able to understand the functionality of a D flip-flop, be able to describe the Dff in SystemVerilog, and be able to understand the simple clock divider circuit.

Clock Divider

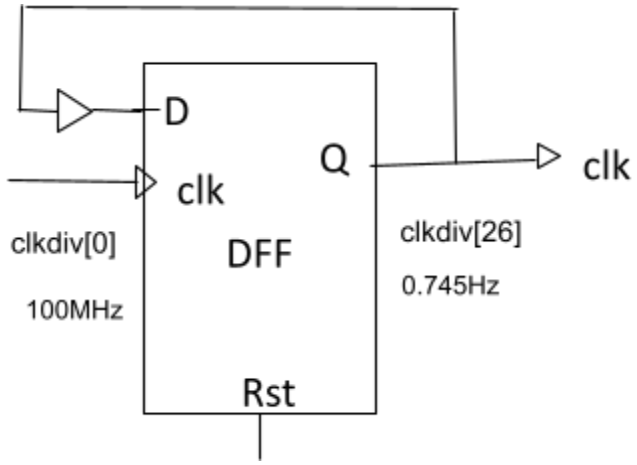

Figure 1.1: Black box diagram of the Clock Divider showing input and output ports.

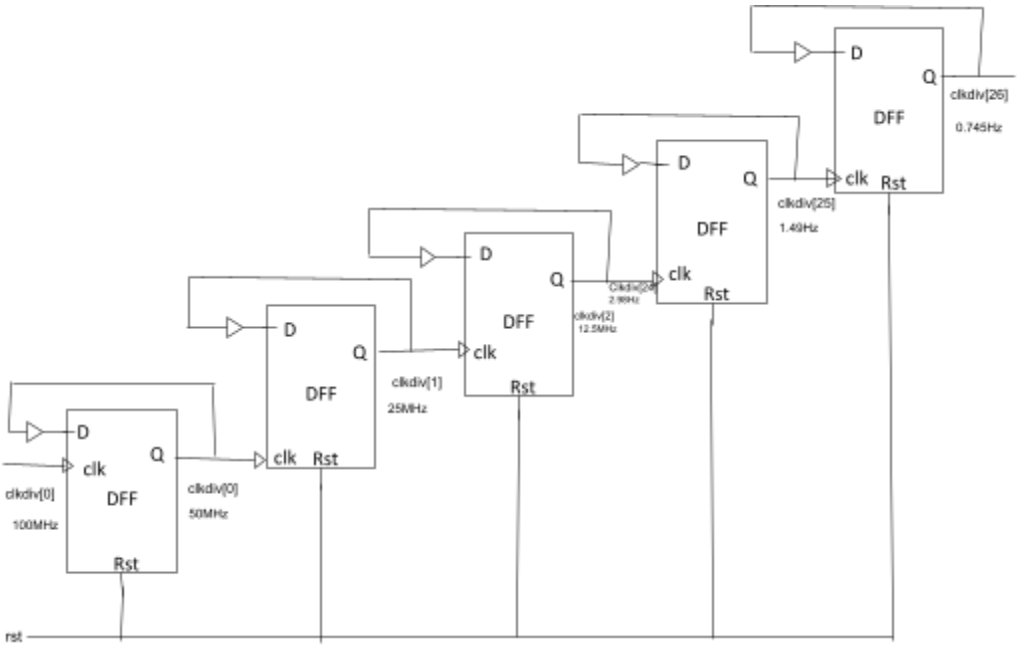

Figure 1.2: Structural Model for the clock divider. Shows the 5 consecutive D flip-flops.

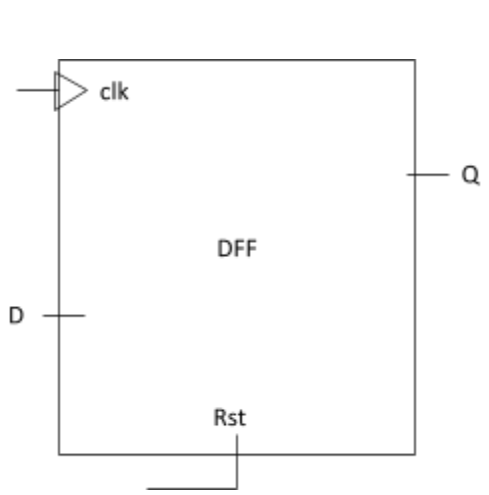

Figure 1.3: Black Box diagram for the DFF showing the inputs and outputs.

Circuit Schematics

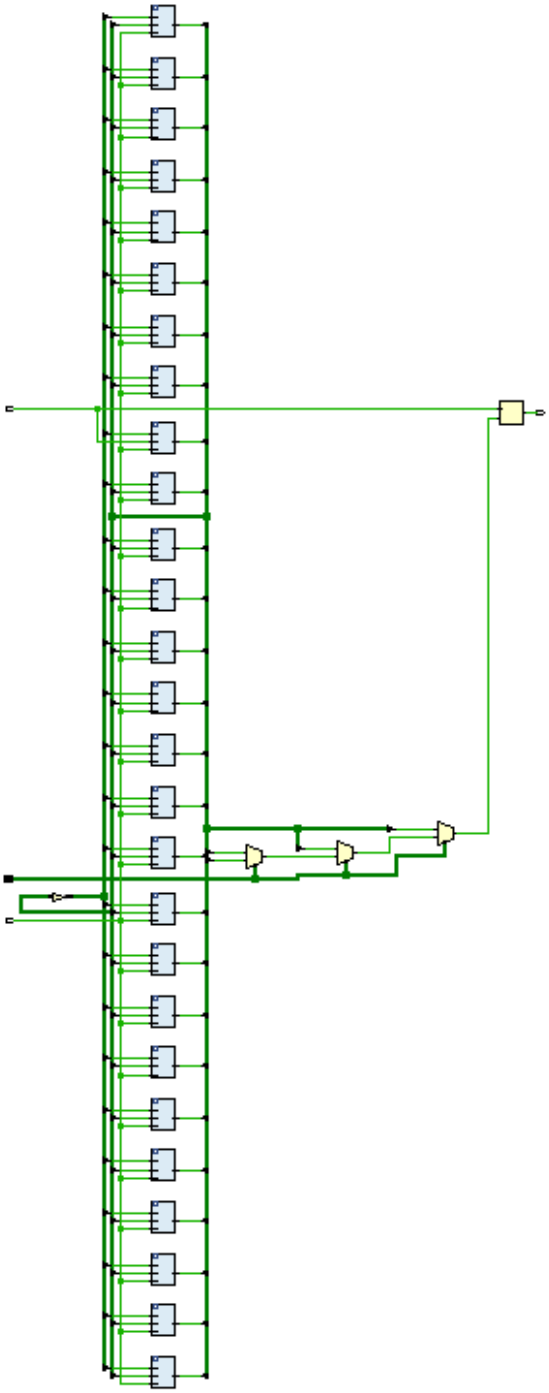

Figure 1.4: Screenshot of the Clock Divider circuit schematic after running the synthesis

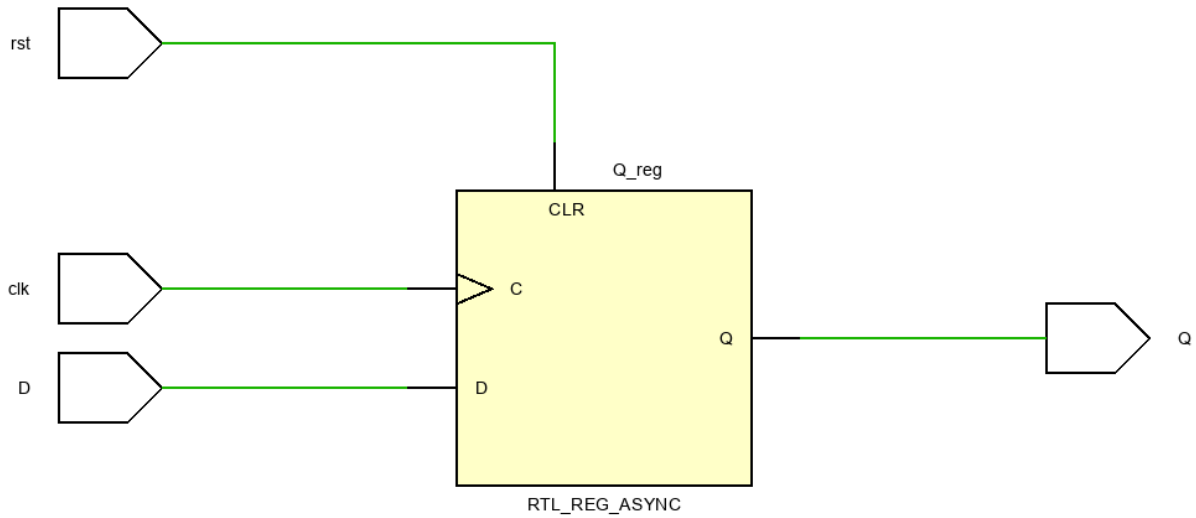

Figure 1.5: Screenshot of the D Flip Flop circuit schematic obtained from the synthesis

Simulation Results

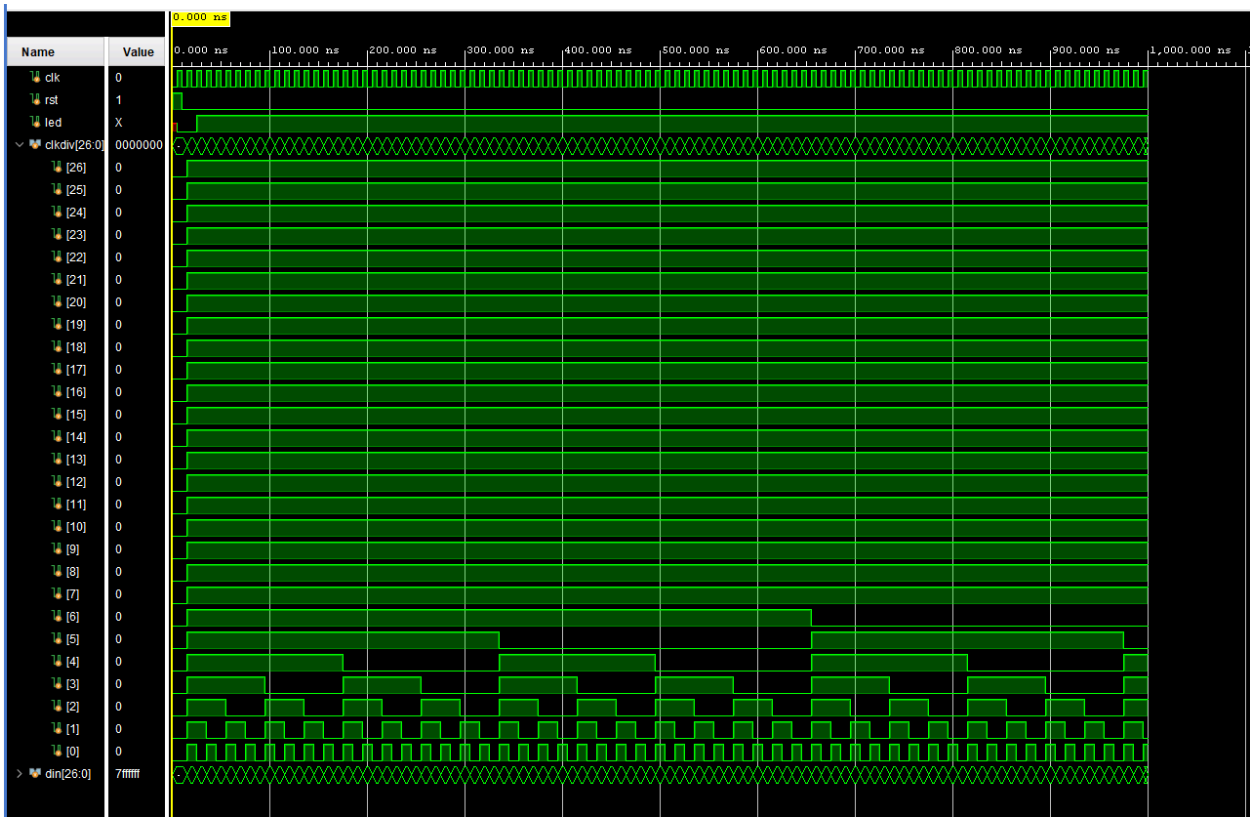

Figure 1.6: Screenshot of the timing diagram capturing all test cases. Shows instances where the clock is successfully divided in half by each stage.

The simulation results as expected. The divider works as it should and produces the expected output values for the given input combinations.

Source Code

Clock Divider Design Code:

```
module clk_divider(
   input clk,
   input rst,
   output logic led,
   input [1:0]SW
   );
   logic [26:0] din;
   logic [26:0] clkdiv;
   Dff Dff_inst0 (.clk(clk),.rst(rst),.D(din[0]),.Q(clkdiv[0]));
   genvar i;
   generate
   for (i = 1; i < 27; i=i+1)
      begin : Dff_gen_label
      Dff Dff_inst (.clk(clkdiv[i-1]),.rst(rst),.D(din[i]),.Q(clkdiv[i]));
   end
     endgenerate;
     assign din = ~clkdiv;

   always @ (posedge clk)
   begin
   if (SW==0)
      begin
         led <= clkdiv[26];
      end
   else if (SW==1)
     begin
         led <= clkdiv[25];
     end
   else if (SW==2)
      begin
         led <= clkdiv[24];
      end
   else
      begin
         led <= clkdiv[23];
      end
  end
endmodule
```

Clock Divider Sim Code:

```
module clk_dividerSim();

// Inputs
logic clk;
logic rst;
// Outputs
logic led;
// Instantiate the Unit Under Test (UUT)
clk_divider uut (.clk(clk),.rst(rst),.led(led));
always
#5 clk = ~clk;
initial begin
// Initialize Inputs
   clk = 0;
   rst = 1;
   #10 rst = 0;
// Wait 100 ns for global reset to finish
   #100;
   end
endmodule
```

DFF Design Code:

```
module Dff(
   input D,
   input clk,
   input rst,
   output logic Q
   );

   always @ (posedge(clk), posedge(rst))
   begin
   if (rst == 1)
   Q <= 1'b0;
   else
   Q <= D;
   end

endmodule
```