ENGG 125.03 Laboratory Activity 5: Shift Register

Activity Sheet; Submitted: 10 May 2021

Mallari, Kevin Luis T.
BS Computer Engineering
Electronics, Computer, and Communications Engineering
Ateneo de Manila University
Philippines

***Abstract*—The laboratory activity is about shift registers. A shift register is a type of digital circuit that uses a cascade of flip-flops in order to shift an input into the value of the output. It uses a clock signal in doing so. A shift register can either shift left or right, depending on the mode the register is set to. There are shift registers that can only shift right, or only shift left. However, the activity requires simulating the 74194 circuit, which is a bidirectional shift register. This means that it can do both typings of shifting.**

Keywords - Shift Register, ModelSim, Quartus, 74194

I. Computations and Solutions

Shift registers shift a data input into the value of an output. The placement of the input into the output depends on whether the shift register shifts left or right. For example, if 1 is shifted left into 0000, then the new value would be 0001 since it moves the previous value to the left. On the other hand, if 1 is shifted right into 0000, then the new value would be 1000 since the initial value moves to the right.

II. HDL Code

This section shows and explains the code made for the activity.

```
module shift_register_74194(d, clk, en,
rst, ql, qr);

parameter N = 4; // number of bits the
output will be

input d, clk, en, rst;
output reg [N-1:0] qr, ql;

initial ql = 0;
always @(posedge clk)
  if (!rst) ql <= 0;
  else
  if (en)
    ql <= {ql[N-2:0], d};

initial qr = 0;
always @(posedge clk)
  if(!rst) qr <= 0;
  else
    if (en)
      qr <= {d, qr[N-1:1]};

endmodule
module testbench();
parameter N = 4;
reg d, clk, en, rst;
wire [N-1:0] qr, ql;

shift_register_74194 device_under_test(d,
clk, en, rst, ql, qr);

always #10 clk = ~clk;

initial begin
  d = 1; clk = 0; en = 0; rst = 1;
end

initial begin
  #1 rst = 0;
  #1 en = 0;
  #20 en = 1;
  #20 rst = 1;
  #100 rst = 0;
  #110 rst = 1;
  #110 d = 0;
  #50 en = 0;
  repeat(20) @ (posedge clk)
    d <= ~d;

  repeat(20) @ (posedge clk);

  #1000 $stop; // stop the simulation
```

```
end

endmodule
```

Fig. 1. Code for the Shift Register

The parameter N is used to change the number of bits the output has easily. In this case, it is set to 4 since the shift register 74194 outputs a 4-bit result. There are four inputs in the code. These are the data, clock, enable, and reset. The data determines if the shifted number would be 0 or 1. The clock is the clock input in the integrated circuit. The enable input tells the code whether to continue or stop shifting. The reset input resets the output back to 0.

There are two (2) outputs in the code. One of the outputs is for the result of a shift right, and the other is the result of the shift left. The initial values of these outputs were set to 0.

The always keyword was used so that changes in the value of the outputs will only happen on the positive edge of the clock. A conditional statement was used with the reset input as the condition. If the reset input is 0, then the value of the outputs will be 0. Otherwise, it checks if the enable input is on. If it is on, then it will shift the value of d to the output. Otherwise, it retains the current value of the output by doing nothing.

The code for the test bench first instantiates the initial values of the inputs. The repeat keyword was used for the data input so that it inverts the value every 20 clock counts.

III. Output Circuit

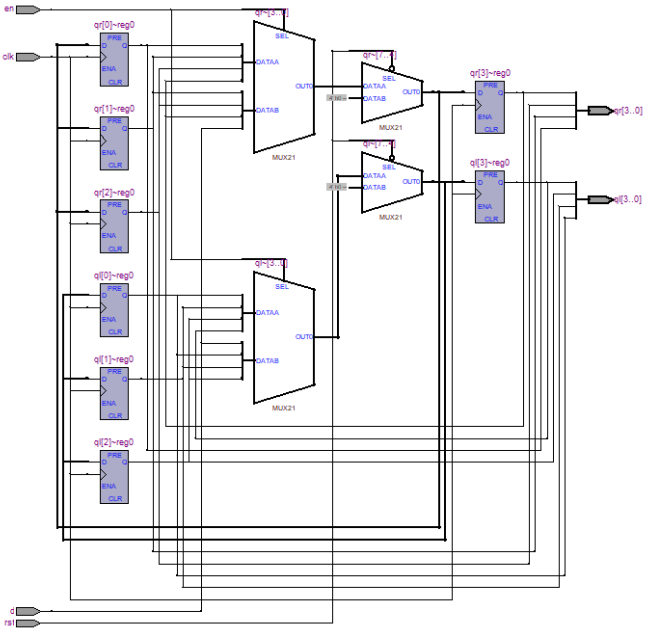

Fig. 2. Circuit for the Shift Register

The circuit in figure 2 is the output circuit of the entire code. Multiplexers and registers were added to the circuit in their appropriate places.

IV. Screenshot(s) of Simulations and VWF file

The screenshots of the simulations are presented below.

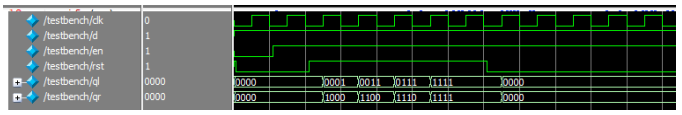

Fig. 3. Simulation for the Shift Register

As seen in the screenshot, the value of the data input (d) is simply 1. Therefore, 1 is being shifted in both the outputs until they eventually have a value of 1111. When the reset input is set to have a value of 0, then the output resets back to 0.

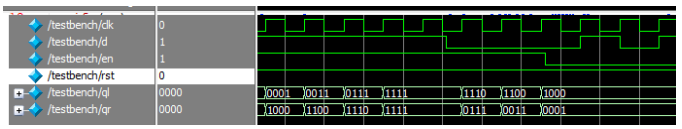

Fig.4. Simulation 2 for the Shift Register

Upon going deeper in the simulation, the value of the data input (d) is now set to 0. The values of the output change and shift 0 to its appropriate place. However, as seen in figure 4, when the enable input is set to 0, then nothing happens to the output and remains as is, even though the value of data alternates every 20 time counts.