# CS 207 Digital Logic - Spring 2020
# Lab 7

James Yu

Apr. 1, 2020

## Objective

1. Implement latches and flip-flops.

## Lab Exercise Submission

1. You should name all source code as instructed. Mis-named files will not be recognized.
2. You should submit all source code files with an extension "`.v`".
3. You should submit all source code directly into the sakai system below. **Do not compress them into one folder.**

   `https://sakai.sustech.edu.cn/portal/site/ebf68254-68c5-4cfe-9d42-b26758f854ee`

4. Lab exercises should be submitted before the deadline, typically one week after the lab session. No late submission policy applies to lab exercises.

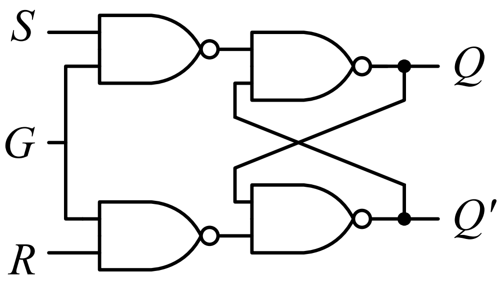
## 1 SR latch with clock control

Implement an SR latch with clock control signal as follows:

srlatch.v

```
module srlatch(output Q, output nQ, input S, input R, input G);

wire Rd,Sd;
nand na0 (Rd, R, G);
nand na1 (Sd, S, G);
nand na2 (Q, Sd, nQ);
```



```
nand na3 (nQ, Rd, Q);

endmodule
```

Or, we can have a straight-forward behavioral modeling:

srlatch.v

```
module srlatch(output Q, output nQ, input S, input R, input G);
reg Q;
reg nQ;
always @(G) begin
    if (G) begin
        case ({R,S})
        1: {Q,nQ} <= 2'b10;
        2: {Q,nQ} <= 2'b01;
        3: {Q,nQ} <= 2'bxx;
        endcase
    end
end
endmodule
```

To test the code, we have the following testbench:

sr_tb.v

```
module sr_tb;
reg CLK;
reg S, R;
wire Q, nQ;
initial CLK <= 1'b0;

always @(CLK)
begin
    #1 CLK<=!CLK;
end

srlatch mysr(Q, nQ, S, R, CLK);
```

```verilog

initial begin
    $dumpfile("sr.vcd");
    $dumpvars(0,sr_tb);
end

initial begin
    $monitor("%3t: Q = %b, nQ = %b", $time, Q, nQ);
    #5 S <= 1'b0; R <= 1'b0;
    #5 S <= 1'b1; R <= 1'b0;
    #5 S <= 1'b0; R <= 1'b0;
    #5 S <= 1'b0; R <= 1'b1;
    #5 S <= 1'b0; R <= 1'b0;
    #5 S <= 1'b1; R <= 1'b0;
    #5 S <= 1'b0; R <= 1'b0;
    #5 $finish;
end
endmodule
```

Try the code and observe the waveform.

# 2 Exercise

## 2.1 Exercise 1

Implement a D-latch **with clock control signal** based on the previous circuit of SR latch. The module should follow the design below:

dlatch.v

```verilog
module dlatch(Q, nQ, D, G);
// Module Code
endmodule
```

and save in file `dlatch.v`.

**Assignment**
Save the source code in **`dlatch.v`**. Upload this file to Sakai under **Assignments → Lab Exercise 7**.

## 2.2 Exercise 2

Implement a D flip-flop, a JK flip-flop, and a T flip-flop which are triggered by positive edges of a clock signal. The module should follow the design below:

dff.v

```
module dff(Q, D, CLK);
// Module Code
endmodule
```

jkff.v

```
module jkff(Q, J, K, CLK);
// Module Code
endmodule
```

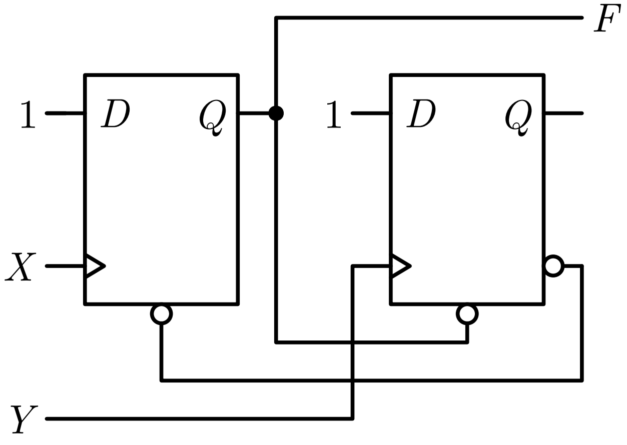
tff.v

```
module tff(Q, T, CLK);
// Module Code
endmodule
```

**Assignment**
Save the source code in **dff.v**, **jkff.v**, and **tff.v**. Upload these files to Sakai under **Assignments → Lab Exercise 7**.

## 2.3 Exercise 3

Implement the following circuit with your previous D flip-flop:



Note that the bubbled input at the bottom of the two FFs are reset signal, which resets the state to 0 upon negative edge input, i.e., from 1 to 0.

The module should follow the design below:

dff2.v

```
module dff2(F, X, Y);
// Module Code
endmodule
```

and save in file dff2.v.

### ⚠ Assignment

Save the source code in **dff2.v**. Upload this file to Sakai under **Assignments → Lab Exercise 7**.

### ⚠ Assignment

When you finished Lab Exercise 7, there should be five .v files in the Sakai system: **dlatch.v**, **dff.v**, **jkff.v**, **tff.v**, and **dff2.v**.