ECE 526L Lab4

In this experiment, you will model an edge triggered flip-flop using a hierarchical modeling approach.

1. For all your models, use the following delay data:

Single-input gates: intrinsic delay of 3 ns.

Two-input gates: intrinsic delay of 4 ns.

Three-input gates: intrinsic delay of 5 ns.

Capacitive loading of 0.5 ns for a fanout of one.

Capacitive loading of 0.8 ns for a fanout of two.

Capacitive loading of 1.0 ns for a fanout of three.

2.0 ns loading delay for a primary output.

2. Using primitive gates, write a Verilog module for the SR Latch shown in figure 1. Use a header like the one shown below but use a suitable timescale for the delays indicated previously. Do not lose data due to rounding.

```
`timescale 1ns/1ns module SR_Latch2(Q, Qnot, s0, s1, r0, r1);

output Q, Qnot;

input s0, s1, r0, r1;

.

.

.

endmodule
```

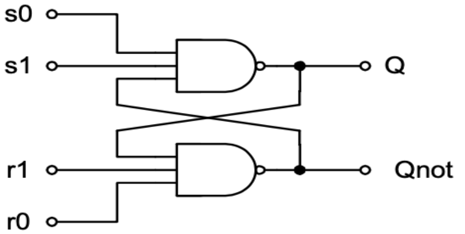
Save your module in a Verilog file with the same name as the module.

3. Using the SR_Latch2 module and primitive gates, write a Verilog module for the positive edge triggered flip-flop shown in figure 2. Use the following module header:

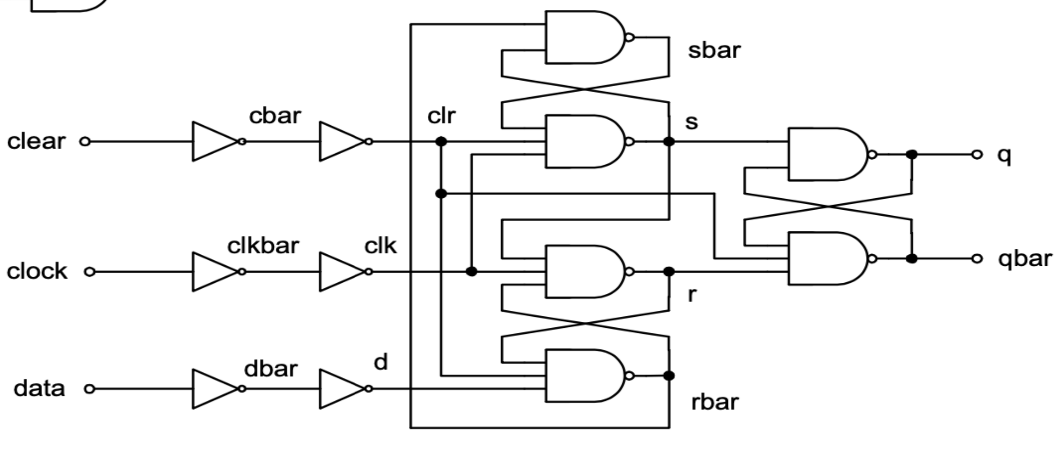
```
module dff(q, qbar, clock, data, clear);
output q, qbar;
input clock, data, clear;
.
.
.
endmodule
```

Save your module in a suitably-named .v file.

Note that the instances of the SR latch have different delays, depending on where they are located in the overall design. You must find a solution for setting proper delays while still using three instances of the original design.
There are several ways to make this work. Creating different designs, one for each delay characteristic, is not an acceptable solution. (Hint: can you use a 3 input NAND for only 2 inputs?)



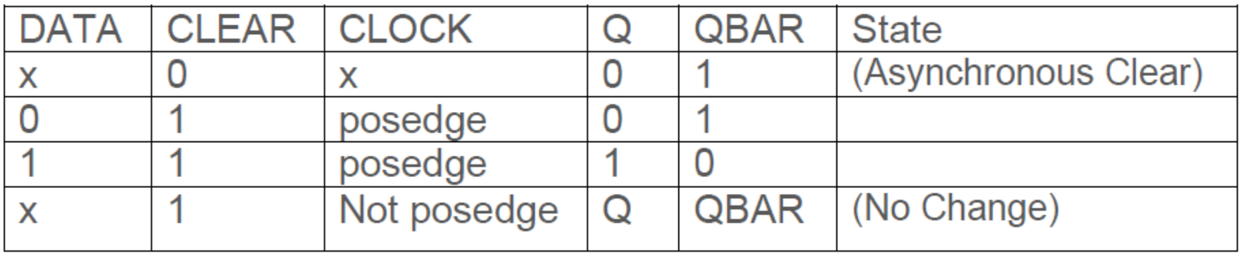
SR Latch
Figure 1



Positve Edge-Triggered D Flip-Flop with Clear
Figure 2

4. Write a testbench module to verify the functionality of your flip-flop. In this module, use each of the output system tasks, $monitor, $display, $write and $strobe. Do not simply display the same data using each task. Instead, take advantage of the differences between them to show that you understand those differences and can use them to good effect.

5. The D flip flop should function as shown in the table below. However, using this table as a template to write your vectors will not work correctly, due to the persistence of X values in Verilog and that indeterminate is not a specific value. Instead, write a small set of vectors that checks each JKFF function.



DATA	CLEAR	CLOCK	Q	QBAR	State
x	0	x	0	1	(Asynchronous Clear)
0	1	posedge	0	1	
1	1	posedge	1	0	
x	1	Not posedge	Q	QBAR	(No Change)

The inputs must be stable before the clock edge. You can cause rising edges on the clock line by first setting it to 0 and then to 1. Experiment with a symmetric clock (50% duty cycle)

Be sure to allow sufficient time between input changes for the effects to propagate through the circuit. This is a sequential circuit and must be tested accordingly.

Lab Questions:

1- What’s the critical path (longest delay) of this design?

2- What is the maximum operating frequency for your circuit?