Lab #3

Latch, Flip-Flop, and Prescaler

Objective

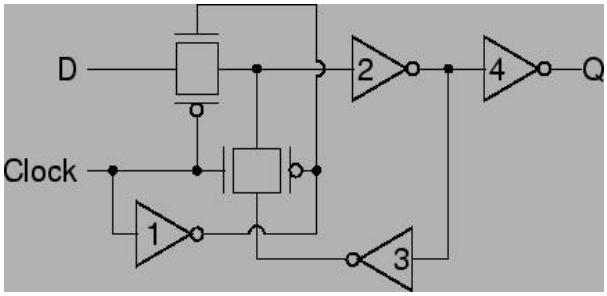
Build a latch and a master-slave D-type flip-flop. Then design and build a high speed divide-by-five prescaler with the DFF you made.

Design guidelines

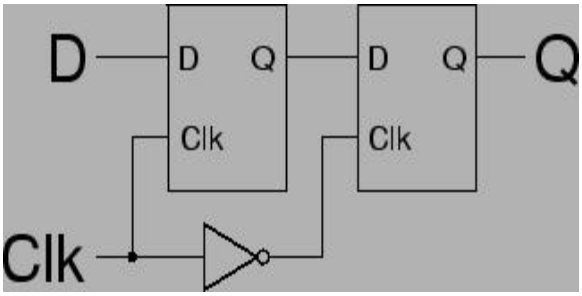
- For any measurement you make, the device has to have proper drive and load models; use 60ps for both rise and fall time on initial input signals.
- Refer to **Wolf 5.2** for information on latches and flip-flops;
- Use latch and DFF designs given in this handout;
- The prescaler shall have a 40% duty cycle;
- Size is **NOT** important in this lab, but power consumption is;
- Follow the second lab's design guidelines;
- You will be graded against the rest of the class for the result of prescaler's (power consumption * Clk-to-Q delay), smaller the value better the grade.

Structure of latches and flip-flops

The following figure is the structures of a latch and DFF you need to use for this lab.



Latch

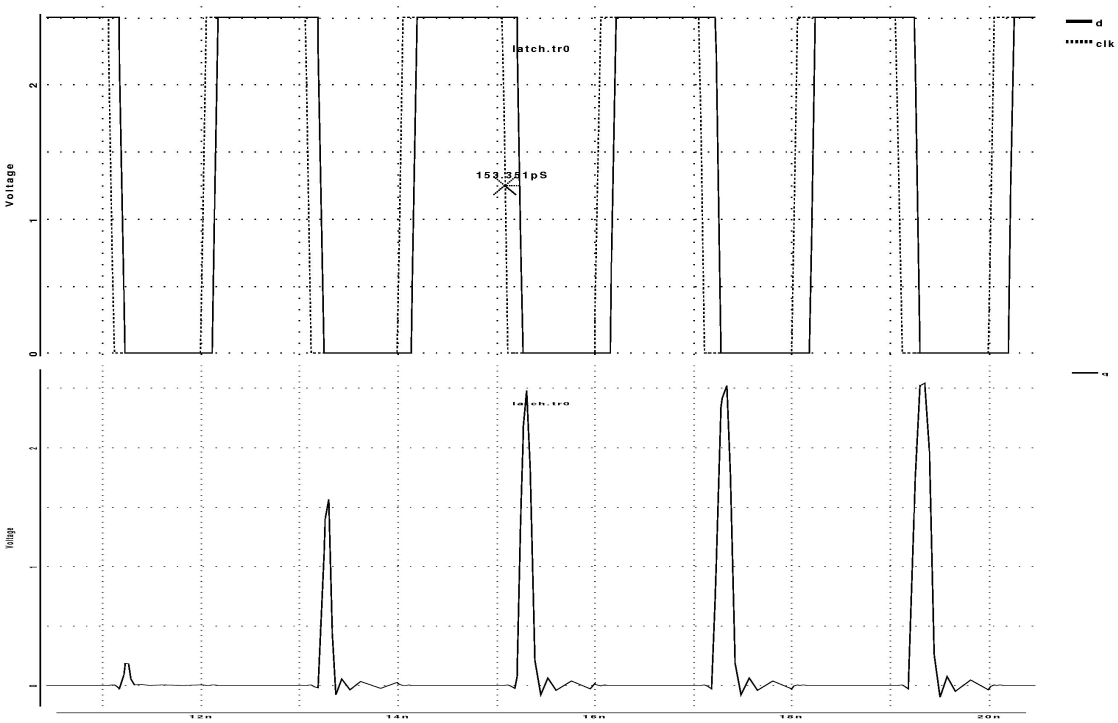


D-type master-slave flip-flop

Notice the difference between these designs and the ones shown in **Wolf**. The DFF is built from two latches and therefore called master-slave FF. In the latch's structure, inverter #4 is the driver to boost up output signal's power for external circuit. Since the DFF is considered as a single circuit, you do not need to use inv#4 in the master latch. However, you still need to include the driver for the slave latch to drive the external circuit. (Hint: to maximize the performance of the DFF, you may want to use different size of latches.)

Testing latches and FFs

Two things affect a latch's performance: D-to-Q delay when it is transparent and clock setup and hold time when it is not transparent. It is straight forward to measure the D-to-Q delay. However, since clk setup and hold time depend on both inputs, it is not so easy to measure them in traditional way. What we can do is feed in two inputs with square waves that have slightly different frequencies, i.e. one is 1GHz and another is 1.01GHz. How much difference it needs depends on the resolution you want (normally res. of 100 to 200 is good enough). The idea behind this is after every clock cycle, one signal will slightly drift away from the other and therefore, simulates two input signals with different switch time. The example given before (1GHz vs. 1.01GHz) can cover signals difference from 0 to 1GHz in 10MHz increment (0, 10MHz, 20MHz, 30MHz...). Once setup the input signals, we can monitor the output until the latch fails. The value right before that is what we want (why?). The following figure is an example on measuring the setup time delay:



In this example, the clk setup time delay is 153.35ps. This is the shortest time D-input signal has to wait until the latch considers it as valid signal. Measurement of clk hold time is similar to this one, except for measuring right before the rising edge of the clk signal.

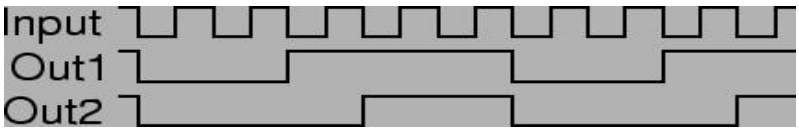
Performance measurement of a FF is comparable to a latch. However, D-to-Q delay is meaningless to a FF. What is important is clk-to-Q delay, besides the clk setup/hold time. You can apply similar techniques you used on latch measurement to measure these values.

To measure the power consumption of a device, we just need to measure the supply voltage and current for the device: Power = Voltage X Current

High-speed prescaler

When we want to sample an extremely high frequency clock signal, higher than traditional counter can handle, we need to use prescaler to scale down the clock frequency first. Logically, a prescaler is no more than a counter. If you still remember, the BCD counter you built in ECE152A lab is an example for a divide-by-10 prescaler. A divide-by-5 prescaler output a signal with the frequency five times lower than the input one. The following figure shows two of the possible outputs for divide-by-5 prescalers. Out1 is the prescaler that has a 60% duty cycle and Out2 is the one that has a 40% duty cycle

:



One key aspect of a prescaler is that it must be self resetting. When the circuit powers up, the registers will fall into random states. It has to be reset to one of the counting states before the prescaler can function properly. This could be done with external reset logic. However, adding extra gates can slow down the circuit. Another solution is to carefully design a state machine so that all unused states are pointing to one of the counting states. You need to use the second solution for your high-speed prescaler design.

The divide-by-5 prescaler can be built with only three DFFs and one logic gate. Bear in mind that performance and power are the greatest factors for this lab. To maximize the speed, the DFF's and logic gates among the synchronous counter have to be optimized together. Also, to operate at high speed, it is important to reduce the effective capacitance of internal and external nodes, which leads to the reduction of power consumption as well as of the propagation delay.

Project procedures

1. Build a quasi-static latch and test it in the test environment you used on lab#2 (same as following);
2. Use the latch to build a D-type master-slave flip-flop and test it;
3. Design a 40% duty cycle, divide-by-5 prescaler with the DFF you made;
4. Test the prescaler and measure the maximum frequency the prescaler can take;
5. Convert the latch, DFF, and prescaler designs into *MAX* layouts; use information you collected from *SUE* as reference;
6. Extract the designs; test and simulate them with HSpice and take measurements;

Data collecting

- T_{setup} and T_{hold} for latch, DFF (*SUE, MAX*);
- Propagation delay for latch (D-Q), DFF (clk-Q), and prescaler (*SUE, MAX*);
- Maximum input frequency the prescaler can handle (*SUE, MAX*);
- Power consumption for latch, DFF, and prescaler (*SUE, MAX*);
- Area of the prescaler layout (boundary box in *MAX*).

Design work to turn in

- Schematic for latch, DFF, and prescaler (*SUE*);
- Prescaler state diagram;
- Layout plot for the latch, DFF, and prescaler (*MAX*);
- The setup of your test environment (*SUE, MAX*);
- Your own test codes (hspice, verilog, etc., if you have any);
- Proof of measurements (*SUE*, *MAX*);
- Technical report with the explanation of your observation; also explain why the values are different between *SUE* measurement and *MAX*'s;
- Result of prescaler's power consumption * propagation delay, unit in Watts*Seconds and maximum frequency the prescaler can handle, unit in MHz (*SUE, MAX*)

Due day

In two weeks (November 18/20)

Note

Due to low resolution, screen-capture layout printouts are no longer accepted. Go to http://bears.ece.ucsb.edu/class/ece124a/newlabfaq.html for information on how to print out your layout designs.