Assignment

To design a register cell.

Description

In this CAD assignment, you will design, layout and simulate a 1-bit register (flip-flop) cell. Your full-custom datapath will require a couple of multi-bit registers and you should design this 1-bit register as a bit-sliced component for this purpose. You probably won't use this cell in your final datapath, but this exercise will introduce you to complex leaf-cell design as well as datapathstyle layout concepts. Note that once again you must work on your own for this assignment.

Implementation

It is often desirable to be able to reset asynchronously the state of the register to a known value (usually zero). A typical gate implementation of a positive edge-triggered master-slave D flip-flop is shown in Figure 1.

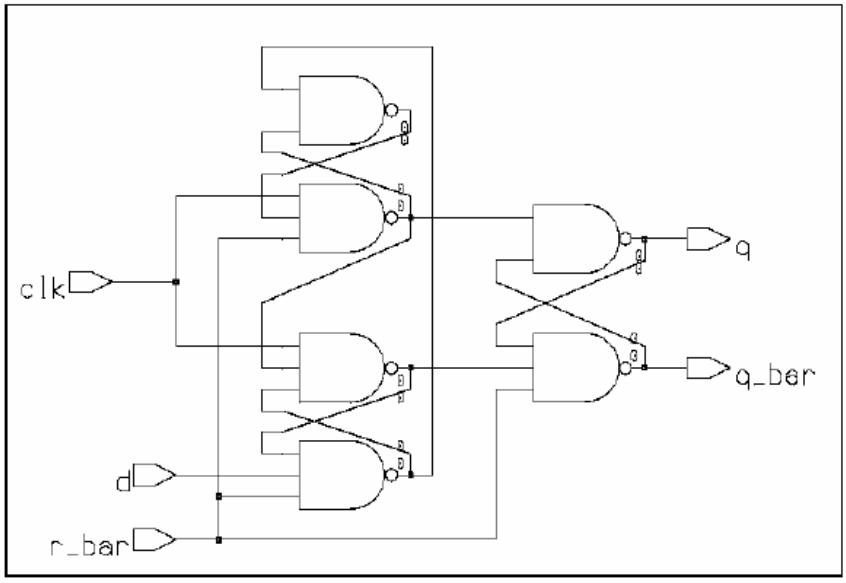

Figure 1: Positive Edge Triggered Flip-Flop Using Logic Gates

An example for a static CMOS transmission gate-based implementation is shown in Figure 2. Use of this design, or similar designs more adapted to CMOS technology, would result in many fewer transistors than the flip-flop of Figure 1, and it is not recommended that Figure 1 be used. Note that a conceptually simpler starting point would be a flip-flop composed of two latches (connected in master/slave configuration) like the master latch in Figure 2 (which is still not being held up as the best design). The flip-flop could be further compacted by using a dynamic or pseudo-dynamic style with twophase or single-phase clocking, but we do not support dynamic design styles for EECS 427 (pseudo-dynamic is acceptable).

Layout Considerations

Compared to CAD Assignment 1 you should pay more attention to the layout aspects of this design. Consider this a bit-sliced component with a target bit-slice width (pitch) no less than 75 lambda. Bit-slice widths larger than 120 lambda will likely result in odd aspect ratios for your complete datapath. If the bit-slice pitch is too small (e.g. 40 lambda), you may find in the future that you don't have enough space to route data buses over top of the datapth. Your layout must be a flat layout (i.e. no hierarchy). This will result in a more compact layout. On this and all future CAD assignments, we will be looking more closely at your layout density. Note that the clock drivers and negation need not be included in your cell design.

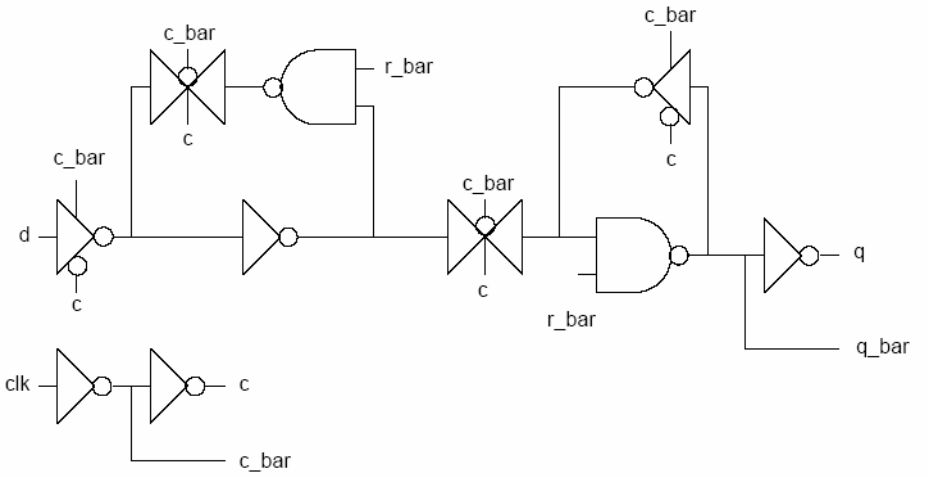

Figure 2: Positive Edge Triggered Flip-flop Using CMOS Transmission Gates and Clocked Inverters

Procedure

Schematic design

Make a directory called **cad2** and do all your work in that directory. Choose the type of flip-flop that you wish to implement, but be sure to include an asynchronous reset to zero. You do not have to be concerned about speed performance of your designs yet. This will come into play in CAD 3.

Create a transistor level schematic of the D flip-flop. **Be sure to note the direction of the arrows on the transistors in your schematic** (specifically for transmission gates), as NC Verilog will not correctly simulate your schematic if they are placed with the wrong orientation.

Digital Simulation

Run NCVerilog on the D flip-flop cell to exhaustively simulate the various input, state, and output combinations.

Layout and Verification
When implementing the layout of your cell, you should keep in mind the various layout considerations. While doing DRC and LVS, follow the same procedure as in CAD 1. Be sure to match the ordering of the inputs in your schematic and layout. Save the LVS reports as **lvs_<comp_name>.rep**. Extract the parasitic capacitances and back annotate using the "Build Analog" method. Add 25 fF to the output node to account for the load that the flip-flop may drive.

Analog Simulation
Run spice on the D flip-flop and obtain the rise and fall CLK-Q delay times as well as the rise and fall setup and hold times for the D flip-flop. (Exactness is not required for the calculation of setup time, but you should try to get a relatively accurate estimate). Use the cursors to find your delay estimates. Save an image of your waveforms with the cursors in place that helped you calculate the delay.

Requirements

You should have the following in your **cad2** directory.
• Schematic of your D flip-flop.
• NC Verilog waveform files showing functional verification of your 1-bit register.
Briefly describe each simulation in your README file.
• Layout of the D flip-flop.
• Spice .png images showing the following: maximum rise and fall
CLK-Q delays and setup and hold times for each. Rise/fall delays are measured from the 50% point of CLK to 50% of the Q output transition. You do not have to be exact about the setup time – just get within a few picoseconds.
• LVS reports
• analog_extracted View
• A README file with the names and paths of all of the requested files. Include a few paragraphs describing your choice of design and verification rationale.

Deadline

You need to turn CAD2 in by **Saturday, Jan. 20, 2007, 7pm**.