Lab #6: Sequential circuit element characterization
ECEn 451
Dr. David A. Penry

In this lab you will use the Cadence tools to characterize the sequential circuit element you designed in Lab 5.

IMPORTANT NOTE: The parametric simulations you will run take up an enormous amount of disk space and old simulation results are not consistently deleted when you run a new simulation. You may find it necessary to clean out other files from your CAEDM account and to periodically remove simulation files. If Cadence appears to do nothing when you tell it to run a simulation, it is likely because you have run out of disk quota. Simulation files can usually be found in a sub-directory called "simulation" which is created either in your homedirectory or in the directory from which you invoke virtuoso.

Part I: Environment setup

Copy the test environment schematic you set up in lab 5 to a new cell. This can be done in the library manager by right-clicking on the cell name for the test environment you used in lab 5 and selecting "Copy". Enter the new cell name in the "To" portion of the form which pops up and press "OK".

Launch and configure the Analog Design Environment as in Lab 4 and simulate this schematic for 400ns to double-check that everything is still working.

Print the schematic of your circuit.

Part II: Characterizing T_{ccq}

The T_{ccq} parameter is simply the clock-to-q delay when the input edge arrives "very early". The configuration from Part I should have this property, so simply use the calculator's delay function (as in lab 2) to calculate the rising and falling clock-to-q delays. These should be calculated for both rising and falling output. Record these numbers in the table at the end of this document, as you will need them later.

Part III: Characterizing the rising output edge

Part IIIa: characterizing Tsetup and Tpcq

As we discussed in class, the setup time and propagation delay are not just simple measured parameters. Instead, we define them to be the values of the D->CLK delay and CLK->Q delay when the D->Q delay is minimized. To make this measurement, you need to set up a parametric simulation as in Lab 4. Set it up in the following fashion:

1. Change the properties of the pulse generator driving input D in the following fashion:
    1. Change "Delay time" to read "indelay". Indelay will be the parameter you sweep with.
    2. Change the period to 500 ns.
    3. Change the pulse width to 400 ns. These last two changes will ensure that there is only one rising edge in the simulation.
2. In the ADE window, in the Analysis->Choose box, choose the "Conservative" accuracy defaults. This increases the simulator's accuracy; this analysis requires very good accuracy.
3. Add a calculation of rising D->Q delay to the outputs; this is done using Output-s>Setup and using the calculator to set up the expression (very much like in lab 4). The expression you want is:

```
 delay(VT("/Din") 2.5 1 "rising" VT("/Qout") 2.5 1 "rising" 0 0 nil
nil))
```

Name the new output DtoQ for reference. Don't forget to use the Add button in the Setting Outputs dialog box to add the output to the list.

Now, find the point at which D->Q is minimized by doing the following:

1. Run a parametric simulationwhich "sweeps" the value of the "indelay" variable. Set it up and run it from the Tools->Parametric Analysis. You should use a fairly large sweep range (say 0ns to 20ns) and 10 steps. **Make sure you put in the n suffix on the range or Cadence may hang.**

   In the waveform window, you'll see the plots of the DtoQ output expressions vs. indelay and the waveforms. The DtoQ expression will have a "sawtooth" form: it decreases as indelay increases and then suddenly jumps. That jump occurs when the data arrives too late to be captured in the first clock cycle.

2. Now perform another parametric analysis with a smaller sweep range (around 3ns total) centered around the indelay values which gave the minimum DtoQ. After running the simulations, you'll see a "zoom in" on that portion of the curve. Take a look at the waveform window and zoom in on the first clock edge. You'll see that the Q outputs appear later and later as the data arrives later.

3. Continue to run parametric analysis with ever-decreasing range sizes until you find the value of indelay which produces the minimum value of DtoQ with at least 5ps accuracy. You'll know you have this much accuracy when the range of the sweep is 50ps or less. Record this value of indelay.

4. **At this point, print out the plot of the DtoQ vs. indelay graph and the waveform window zoomed in on the first clock edge so that you can see the output waveforms getting later.**

5. Now, close the parametric analysis and go back to the main ADE window. Double click on the "Value" box for the "indelay" variable and enter the value of indelay which minimized DtoQ delay in the "Value" box of the window that pops up. Click "OK". The point of this step is to tell the "normal" simulation to use the value of indelay you found in step 3.

6. Run the simulation from the ADE window. (i.e. click on the green arrow)

7. Use the calculator to find the D rising to CLK rising delay. This delay is Tsetup, rising. Enter it in the table at the end of this document. **NOTE: it is entirely possible that you might have negative setup time for your cell because the pulse generator delays the clock inside of the cell. If the data edge arrives after the clock edge, setup is negative**.

8. Use the calculator to find the CLK rising to Q rising delay This delay is Tpcq,rising. Enter it in the table at the end of this document.

Part IIIb: Characterizing hold time

For hold times, we want to have the input arriving so late that the output doesn't change by more than 5%. The procedure is very similar to that for finding setup time, but the trick we're going to use is to measure the delay between D and Q differently. We'll start the measurement when D passes 50%, but end it when Q passes 5%. If Q

moves by more than 5%, we'll get a measurement of no more than a few nanoseconds. If Q doesn't get above 5% after the first clock edge, we'll get a measurement which is a clock cycle too large. The input delay time where we switch from "small" to "large" output delay is the point where we should measure hold time.

1. Change the DtoQ expression to measure rising D to rising Q output delay, but set the threshold level for the output to 0.25V (5% of Vdd). The expression would look like:

```
 delay(VT("/Din") 2.5 1 "rising" VT("/Qout") 0.25 1 "rising" 0 0 nil
nil))
```

2. Run parametric analysis as before, but now, instead of looking for the minimum DtoQ delay, you're looking for the point at which it suddenly jumps to being very high (around 40ns). What you'll see in the waveform window if you zoom in on the first clock edge is that the output starts to go up and then falls. We're trying to find the point where the "hump" is less than 250mV to within 5ps resolution for indelay.

3. Once you find the value of indelay within 5ps,
   print out the plot of the DtoQ vs. indelay graph and the waveform window zoomed in on the first clock edge so that you can see the "hump" decreasing.

4. Take the value of indelay, plug it into the "normal" simulation by double-clicking on the indelay Design Variable in the ADE window, and measure CLK to D in the calculator. (**Note: in other words, D's time minus CLK's time!)** That CLKtoD measurement will be Thold, rising. Enter it in the table at the end of this document.

Part IVa

Repeat part IIIa for a falling output edge. To ensure that you have only one falling edge, change the pulse generator so that pluse width = "(40n + indelay)". **NOTE: the parenthesis matter!**

When looking at the waveforms, you'll zoom to the second rising clock edge instead of the first, because the output isn't supposed to be low until after the second edge.

Note that you will need to modify the DtoQ equation to use "falling" and "rising". Be careful when you measure DtoCLK and CLKtoQ that you measure with respect to the second rising clock edge. Also, don't forget to change the DtoQ expression back to 2.5V for Qout, so you have:

```
delay(VT("/Din") 2.5 1 "falling" VT("/Qout") 2.5 1 "falling" 0 0 nil nil))
```

Part IVb

Repeat part IVa for a falling output edge. To ensure that you have only one falling edge, change the pulse generator as you did in Part IVa. (In other words, use the same schematic as Part IVa). Change the DtoQ expression to 4.75V for Qout, so you have:

```
delay(VT("/Din") 2.5 1 "falling" VT("/Qout") 4.75 1 "falling" 0 0 nil nil))
```

Again, when looking at the waveforms, you'll zoom to the second rising clock edge instead of the first.

Deliverables

Turn in the following:

1. Part I: The complete transistor-level schematics of the circuit you designed in Lab 5.
2. The table at the end of this document listing T_{ccq}, T_{pcq}, T_{setup}, and T_{hold}, for both rising and falling outputs as computed in parts II-IV.
3. Part IIIa:
    1. A plot showing DtoQ delay vs. indelay once you have reached a sweep size of 50ps or less.
    2. The accompanying waveforms showing the outputs arriving ever-later.
4. Part IIIb:
    1. A plots showin DtoQ delay vs. indelay once you have reached a sweep size of 50ps or less
    2. The accompanying waveforms showing the output humps.
5. Part IVa:
    1. A plot showing DtoQ delay vs. indelay once you have reached a sweep size of 50ps or less.
    2. The accompanying waveforms showing the outputs arriving ever-later.
6. Part IVb:
    1. A plots showin DtoQ delay vs. indelay once you have reached a sweep size of 50ps or less
    2. The accompanying waveforms showing the output humps.

Clearly label all of the output plots with which part they come from and whether they are for rising or falling edges.

You must provide hard-copy; electronic submissions will not be accepted.

Be aware that if you choose to work in a team that you are both responsible for knowing how to use all the tools. There will be exam questions which cover tool usage.

Characterization Data Table (to turn in)

Output direction	***Tccq***	***Tpcq***	***Tsetup***	***Thold***
Rising				
Falling				