MCC092 Integrated Circuit Design: Lab 4

Clock Tree Analysis Using Command Line Simulation

Version 2017-08-23

2015-2017 (65 nm CMOS): Lena Peterson, Per Larsson-Edefors
and Stavros Giannakopoulos
Chalmers University of Technology

Contents

1 Lab Purpose

The main purpose of this fourth lab is that you should understand the impact of capacitance and resistance in on-chip wires on the delay, should have pondered the difference in waveform shape between purely capacitive loads and RC loads, and should also know how one can drive wires effectively using repeaters. In this lab you will investigate the impact of the wire models and also discover how accurate your quite basic hand calculations are.

A secondary purpose is that you should know how to run a simulation program (in this case, text editors and Spectre with waveform viewer) from the command line. You should know the basics about how to write the necessary simulator input file. This knowledge is important because sometimes the graphical user interface (GUI) of the simulator limits you more than the simulator itself. Also, sometimes it is more convenient to run simulations in batch, and it is easier to keep track of the runs if they can be automated, which is hard to achieve using the GUIs in the Cadence tools.

2 Pre-Lab Assignments

You will find the pre-lab assignment in a separate document available from the course home page in PingPong. You can find it most easily under the assignment called **Pre-lab preparation for lab 4**. This is also where you are to submit your solution to the pre-lab assignment. Note that even though you work in pairs in the lab you are to submit you own individual solution.

At the beginning of the lab you will get your pre-lab solution back with feedback from the teachers. Before you go ahead with the in-lab tasks, you should correct any mistakes you have made in the pre-lab assignment.

3 Lab Goal

In this lab you will carry out the instructions in this memo and show the results to the lab teaching assistants (TAs). If you do not finish on time, you will have to complete the in-lab tasks on your own and show the lab TAs your printed results at a later time (for example, at your next scheduled in-lab session). You should show the lab TAs these results during, or after, the in-lab session:

- The result of your simulation of the H tree with $R_{SH} = 0.000000001\ \Omega/\square$ (this approximates an r_{wire} of almost 0), driven by your tapered buffer.
- The result of your simulation of the H tree with $R_{SH} = 0.01\ \Omega/\square$, driven by your tapered buffer.
- Your Spectre input file with two H trees, with and without inverters, i.e., repeaters.
- The result of your simulation of the two H trees with and without inverters, i.e., repeaters.

4 Introduction

In this lab you will use the Cadence circuit simulator Spectre as a stand-alone tool and view the results in the waveform viewer ViVa.

If you have time at the end of the lab, we strongly encourage you to perform one or more of the additional tasks listed in Section 10. They will give you more insights and also permit you to experiment with Spectre and its netlist interface more freely. You can also do these tasks later, but then without the benefit of a lab TA to assist you.

5 Using the Command Line for Text Editing and Simulation

Enter the Spectre code given in pre-lab task 1 in a Linux text editor, e.g., gedit, emacs, or vi, and save the file as inverter.scs.

- Run Spectre in command-line mode:

  ```
  > /nfs4/soleil.ce.chalmers.se/4/MCC092/Y2017/spectre inverter.scs &
  ```

- Assuming that the simulation was concluded successfully (how would you know?), then start ViVa. Note that it will take some time for its windows to appear, due to it starting up complex software.

  ```
  > /nfs4/soleil.ce.chalmers.se/4/MCC092/Y2017/viva inverter.raw &
  ```

- Once the Virtuoso Visualization and Analysis (ViVa) window opens, click on **Browser→Results→Open results...** and a new window will appear named **Select Waveform Database**. There, navigate in the directory where you have your spectre files with the left subwindow, and select the file `inverter.raw` and click **Open**.

- In the **Browser** subwindow click the little + sign on the left of your file and click on `Tran` (the transient simulation you just run). Then select the node name for the voltage waveform that you want to plot, e.g., in and out, and double click on it. (If you get any error message that the voltage name is not a proper waveform, the psf soft link is wrong and you need to change it. See the "How to..." document for details on this issue.)

- Study the waveforms. Do they make sense? When you are happy with the results, show them to the lab TAs.

6 Delay of Zero-Resistance H-Tree with Tapered Buffer

Your first more complex task is to simulate an H-tree with the tapered buffer that you designed in pre-lab task 3 and measure the delay from the input to the tapered buffer to the output in the tapered buffer. You are to use the input file you designed but change it so that the sheet resistance is (close to) 0 $\Omega/\square$.

- If you have a lab partner, then switch the input files you prepared for task 7 in the pre-lab, and read through you lab partner's file.
- Discuss with your lab partner which one of the two files you think is better (criteria could be, e.g., easier to read, easier to modify, or having more well-chosen circuit-node names).
- Based on your discussion, decide which of your two input files to use in this task.
- Copy the selected input file to the directory where you are to run this lab and change directory to that file. (Here we assume that myfile is located in your home directory to start with):

```
> cp myfile MCC092/cadence
> cd MCC092/cadence
```

- Rename the selected file:

```
>mv myfile htreeonlyC.scs
```

- In the text editor, change the sheet resistance to a very low (but non-zero) value, since 0 is not possible for the simulator to handle[1]. We recommend that you use a value of $R_{SH} = 0.00000001\ \Omega/\square$ for the parameter RSH in the input file. If you have used the template as intended you should only have to change one parameter in your input file.
- Run Spectre:

```
> /nfs4/soleil.ce.chalmers.se/4/MCC092/Y2017/spectre htreeonlyC.scs &
```

- Study the log message produced by Spectre (check especially the final line where it is stated that Spectre is completing with XX errors, YY warnings and ZZ notices). If you get any errors from Spectre you have to correct them using the text editor, save the file and rerun the simulation.
- Once you have managed to simulate the netlist file successfully to the end start ViVa:

```
> /nfs4/soleil.ce.chalmers.se/4/MCC092/Y2017/viva htreeonlyC.raw &
```

- Once the Virtuoso Visualization and Analysis (ViVa) window opens, click on **Browser→Results→Open results...** and a new window will appear named **Select Waveform Database**. There, navigate in the directory where you have your spectre files with the left subwindow, and select the file `htreeonlyC.raw` and click **Open**.
- In the **Browser** subwindow click the little + sign on the left of your file and click on `Tran` (the transient simulation you just run). Then select as many voltage and or current nodes you need to plot bu double clicking them.

[1]The problem here is that there are two nodes involved when defining the resistor in our netlist. Zero resistance would mean that the two nodes are electrically identical which causes confusion for the simulator algorithm. Implementing zero resistance rather has to be done by shorting the two nodes using only one of the node names. While this is easy to do, this would however impact the netlist structure; and this we don't want to do.

- Inspect the results in the **ViVa** window. If the results do not look as you expect, check your input file for netlist errors and rerun the simulation with the updated input file. Note that, thanks to the & added at the end of the ViVa command, the waveform viewer is run in the background, which means that you do not have to close ViVa each time you rerun the simulation. In fact, we will notice that the command line is now available for you to use for, e.g., the text editor and Spectre.
- Again, use iteratively the text editor, Spectre and the waveform viewing in ViVa in order to fix the input netlist file so the results are correct.
- Measure the delay between two successive stages in the tapered buffer and the delay through the entire tapered buffer.
- If the delay does not correspond at all to what you expect from the pre-lab's task 4, then you need to carefully study the input netlist file for errors, fix them and resimulate.
- Show your results to the lab TAs.
- Before moving on to the next task close all curves in ViVa.

7 Delay of H-Tree with Both R and C

In this section, you will re-introduce the resistances in the H-tree and investigate the resulting delays and waveforms.

- Copy the input file from the previous task to another file, e.g., using:

```
> cp htreeonlyC.scs htreeRC.scs
```

- Open the new file `htreeRCinv.scs` in a text editor and make the following changes:
  - Change the RSH parameter to 0.01 Ω/□ which is a realistic value[2].
  - Save the file.
- Run a Spectre simulation with the new input file:

```
> /nfs4/soleil.ce.chalmers.se/4/MCC092/Y2017/spectre htreeRC.scs &
```

- Fix any errors in the input netlist file using the text editor and rerun until the Spectre simulation concludes successfully.
- Open the file `htreeRC.raw` in ViVa by selecting again **Browser→Results→Open results...**. Notice that in the **Browser** subwindow there are two files loaded.
- Plot the relevant waveforms.

[2]Thanks to the use of the approximation of 0 Ω/□ in the previous assignment we can use the same netlist structure, but only change the resistance value. Using identical netlists saves time and reduces the risk of making mistakes.

- Measure the delay from the input of the H-tree, i.e., the output of the last stage in the tapered buffer, to one of the H-tree leaves and compare with your results in pre-lab task 4. Note that now when the delay in the tree is longer, you may have to extend the width and period of the input square wave signal, and consequently also the simulation time, to make sure that the output settles before the input changes again. Write down the delays because you will need them later if you have time to do the extra tasks.
- Also investigate the delays between the different levels of the tree and compare with your results of pre-lab task 4.
- Reflect on the shape of the transitions. In a clock tree the delay in itself is not so important, rather it is the difference in delay that matters. How does the shape of the output transition influence the sensitivity?
- Show your results to the lab TAs.
- Close all plots in ViVa before moving onto the next task.

8 Delay of H-Tree with Repeaters and Tapered Buffer

Now we want two H-trees in the same input file to make it easier to compare the H-tree with repeaters in it, with the one without. Therefore you need to create a file with two H-trees in it[3] and modify one of them by adding the repeaters.

- Copy the file from the last task to a new file:

```
> cp htreeRC.scs htreeRCrep.scs
```

- Open the new file `htreeRCinv.scs` in a text editor and make the following changes:
  - Copy the description of your entire H-tree so that you have two H-trees in the file.
  - In the second H-tree, change all node names by adding the same character to all node names. Do not forget to change also the node names in the tapered buffer.
  - In the second H-tree, modify all the instance names by adding one character at the end to all of them.
  - In the second H-tree, add the four 500X inverters between level 2 and 3 of the H-tree, as shown in the pre-lab. Choose good names for the four extra circuit nodes you have introduced.
  - In the second H-tree, also add the two 1000X inverters.
  - Now there are many circuit nodes in the circuit; to simplify life a bit you can add a "save" statement with just the circuit nodes you want to investigate after the simulation.
  - Save the modified file.

[3] You can also work with two different subcircuits in two included files, but since we need two different versions of the H-tree netlist there is no way around the fact that two different netlists need to be coded.

- Simulate the modified file with Spectre:

```
> /nfs4/soleil.ce.chalmers.se/4/MCC092/Y2017/spectre htreeRCrep.scs &
```

- If there are any errors in the input file fix them in the text editor and resimulate.
- Plot relevant waveforms in ViVa.
- Measure the delay with repeaters from the output of the tapered buffer to one of the leaves. Compare to the calculated delay of pre-lab task 7.
- Measure the delay in level 1, 2 and 3 and compare to your result from pre-lab task 7.
- Plot the leaf waveforms for the two H-trees (with and without repeaters).
- Show your results to the lab TAs.
- Close down all plots in ViVa before you move on to any other task.

9 Clock Skew Caused by Unbalanced H-Tree Load

In a clock distribution network, it is more important that the clock edges arrive in a controlled manner at the leaf nodes, than that the delay from root to leaf node is as short as possible. The difference in arrival time at the leaf nodes is called clock skew.

The purpose of this experiment it to measure the clock skew in the H-tree when the capacitive load is unevenly distributed among the leaves of the tree. The effect is most easily seen if the capacitive load is increased quite a bit in one leaf and decreased at a leaf in the opposite corner of the tree. One wants to do this experiment both for the tree with and for the tree without repeaters, so start by copying you input file from the previous task to a new file and then modify that file.

- Copy the file from the last task to a new file:

```
> cp htreeRCrep.scs htreeRCrep-skew.scs
```

- Open the new file `htreeRCrep-skew.scs` in a text editor and make the following changes:
  - To one of the four leaves of each tree add the capacitance corresponding to 200 1X inverters.
  - To the opposite leaf in the two H-trees subtract the same capacitance so that the total leaf capacitance of each tree remains the same.
- Simulate the modified file with Spectre:

```
> /nfs4/soleil.ce.chalmers.se/4/MCC092/Y2017/spectre htreeRCrep-skew.scs &
```

- If there are any errors in the input file fix them in the text editor and resimulate.

- Plot relevant waveforms in ViVa.
- In each of the two trees measure the clock skews for the two nodes for which you changed the capacitance.
- In each of the two trees measure the clock skews for the two nodes for which you which did not change the capacitance.
- Show your results to the lab TAs.
- Close down all plots in ViVa before you move on to the extra tasks.

To think about: What happens to the delay to the two leaves where you did not change the capacitive load? Which one of the two trees has a higher clock skew?

10 Extra for Those Who Have Time

Here are some extra tasks that you can try if you are done with the tasks above before the end of the lab time. They are listed here in priority order:

Impact of Wire Model

Throughout this lab we have used a model with only one π link for each wire segment, but we know from theory that with three π links (where each link has one third of the resistance and capacitance of the single π link) we get within 3% of the true delay. So it would be interesting to see the difference in delay when we use one π link and three π links. A three-stage π-link subcircuit could look like this:

```
subckt piwire3 (n1 n4)
       parameters cap=10f res=0.00000001
       cap1 (n1 0) capacitor c=cap/6
       res1 (n1 n2) resistor r=res/3
       cap2 (n2 0) capacitor c=cap/3
       res2 (n2 n3) resistor r=res/3
       cap3 (n3 0) capacitor c=cap/3
       res3 (n3 n4) resistor r=res/3
       cap4 (n4 0) capacitor c=cap/6
ends piwire
```

The easiest way to do this experiment is to make a copy of the file with the most RC delay in it and try it there:

```
> cp htreeRC.scs htreeRC-3pi.scs
```

Use a text editor to copy the subcircuit above into the new file and change all subckt references from piwire to piwire3. Simulate and measure the delay. How much difference do you find from your previously simulated delays? It is worth noticing that this difference in delay shows

up only when using circuit simulation which solves the differential equations properly. In hand calculations, where we use the simpler Elmore delay model,there will be no difference in delay.

Supply Current

Rerun any of the previous experiments to see the current drawn from the power supply and compare. In the case where you have two H-trees in the same file it is a good idea to add another voltage source so that you can distinguish the current drawn by the two trees. Add this command in the Spectre input file to save the current in the raw file:

```
save vvdd:currents
```

Here, we have assumed that vvdd is the voltage source from which the supply current is drawn.

What is the maximum current drawn from the power supply and at what times do peaks appear? What happens at those times?

Redesign of Tapered Buffer for H-Tree with Repeaters

For the case when we had added the repeaters, the tapered buffer is overdesigned since the tapered buffer is now driving less capacitance than it was designed for. Redesign your tapered buffer for this case and measure the delay in the two cases from the input to the tapered buffer to the leaves.

The impact of fast and slow corners

There are transistor models available also for the fast (FF) and slow (SS) corners. If you change the model file name at the beginning of one of your the Spectre input file to end in SS or FF, that model will be used instead, and as a result you can see the difference in delay. The wire models we use in this lab will not be impacted by the choice of corner, so in which case would you expect to see the biggest difference in delay?

11 The End

In this lab exercise you have investigated the impact of capacitance and resistance in a clock tree. In a clock tree is more important that the delay is the same to all leaves (that is, no clock skew) than the delay is short. For the same delay, an RC-dominated waveform is slower to rise around the tripping point than is the waveform from a well-designed inverter output, so the RC waveform is more sensitive to differences in delay. When we use repeaters we introduce uncertainty (variation) in the inverters, but we also decouple the network so that changes in load has no effect at other leaves in the tree.