ENGN2912E: Low Power VLSI System Design

Homework Assignment #2

Due **Wednesday, Oct.4, 2017, in class**

This homework is to be done alone.

This homework assignment is meant to give you a better understanding of MTCMOS and how it is used in VLSI System Designs. While many of you may have some familiarity with SPICE, I am not assuming you are acquainted with the use of Cadence Virtuoso, so please go over the schematic and simulation tutorial as suggested in class. The tutorial is available on the course webpage. Also included on the webpage is a link to the reference paper I mentioned in class [1].

As discussed in class, static power due to sub-threshold leakage is highly dependent on the sub-threshold voltage of the devices used to implement the logic gates. Using high V_{th} devices (VTH) can lead to a dramatic improvement in the leakage current, but has the undesirable effect of increasing the propagation delay of the gate. The idea of MTCMOS is to allow gates to operate at low V_{th} delay values, when in "active" mode, but dissipate low leakage power when in "standby" mode.

For this assignment, you will be using a 45nm bulk CMOS technology library for your simulations. The SPICE parameter values are as follows:

- 45nm process technology
- Vdd = 1.0V
- Low-Vth_{-N} = 0.22, Low Vth_{-P} = − 0.17 (approx.)
- High-Vth_{-N} = 0.35, High Vth_{-P} = −0.33 (approx.)

Complete the following without a load capacitor (in Cadence, connect an output pin to NAND3 output):

1. In Cadence, design a simple 3-input NAND gate in CMOS logic. Use low-V_{th} (VTL) devices. Assume a minimum sized transistor with $W_n/L_n = W_p/L_p = 3$, where $L_n = L_p = 50$nm (allowing for $L_{eff} = 45$nm) and $\mu_n/\mu_p = 2$. Note that while you can use the standard gate library for a guide, you will need to design your NAND gate from scratch, at the transistor level.

2. Do a theoretical analysis on paper first, describe and explain the input patterns that produce the worst-case rise/fall delays.

   Next, use Cadence to simulate and demonstrate correct logical and electrical behavior with input patterns that produce worst-case rise/fall delays from your paper analysis. Report the worst-case rise/fall delays.
   (Hint: you may need to choose ***bit*** as the input source function in Setup Analog Stimuli.)

   There are two ways you can get the delay values, please choose the one you feel is easier:

   (1) (Recommended) You can see visually input and outputs changing, but to get values, set up a marker at 0.5V (half of Vdd) and it will report the cross of the 50% point.

To create the horizontal marker, select the signal you want (there will be a yellow bar on the far left side to show which signal is currently selected), go to Marker→Create Marker→Horizontal and set it to 0.5 and "on". You will see a horizontal marker line at 0.5V across the signal you selected.

- For a short and simple transient response like this, you can move your mouse to the cross points, the time stamps will show automatically. Manually record these time stamps and compute the propagation delays.
- For a long and complicated transient signals, you can export the cross point axes to a csv file (go to Marker→Export→Horizontal Markers), and compute using other softwares like Matlab.

(2) You can also do something similar as was shown in the tutorial:

```
cross(getData("/vout" ?result "tran") 0.5 1 "falling") - cross(getData("/vin" ?result "tran") 0.5 1 "rising")
```

Where /vin and /vout are the input and output net names in your schematic.

3. Describe and explain the input pattern that dissipates the highest *dynamic* current (i.e. the one that has the highest drain current I_d when the output switches). Then use Cadence to obtain the average dynamic power in your circuit.
   In this case, average current is measured during the rise time and fall time (0.1*Vdd to 0.9*Vdd).

   Continue with the following steps in Cadence:
   - Plot vout, select and zoom in a rise/fall voltage edge, record the rise/fall time stamps (there are two time stamps: 10%*vdd and 90%*vdd).
   - Plot the current of the supply voltage source
   - In ADE, go to Tools -> Calculator, type the following expression:
     `average(clipX(i("/V0/PLUS" ?result "tran") time1 time2))`

     Note that you need change `time1` and `time2` in the above expression to the rise/fall time stamps you recorded, otherwise you will get errors. The result will not be plotted but can be displayed in Calculator.

4. Describe and explain the input pattern that dissipates the highest *static* power and the input pattern that dissipates the least static power. Then use Cadence to obtain the values. Remember, you don't want to switch inputs to measure current. Also, you will have to run the simulation over a longer time period than for dynamic power.

5. Repeat steps 1-4 for a 3-input NAND gate implemented with high- V_{th} (VTH) devices.

6. Starting with the low-V_{th} NAND gate, add sleep transistors to the pull up and pull down networks using high-V_{th} (VTH) devices for the sleep transistors. Note that insertion of the sleep transistors will require careful sizing so that delay is minimally affected and leakage is minimized.

7. With the gate in active mode (i.e., sleep transistors are on), measure the following:
   a. Propagation delay for all input patterns (from the 50% Vdd point of the input signal to the 50% Vdd point of the output signal). You can refer to the tutorial for the propagation delay expression

b. Worst case dynamic power
c. Leakage current using input assignments that minimize leakage

8. Repeat part 7 with the gate in standby mode (i.e., with the sleep transistors in the off state). Note that parts (a) and (b) do not necessarily make sense when operated in standby mode, but the waveforms and power values are interesting to observe.

9. Create a circuit consisting of a cluster of ten 3-input NAND gates, all sharing the same NMOS and PMOS sleep transistors. Resize the sleep transistors to maintain performance (i.e., propagation delay) assuming
   a. only 1 of the inputs switches,
   b. 30% of the inputs switch, and
   c. all of the inputs switch at once.

   Simulations should show the output switching as a result of the input(s) changing. NOTE: you can tie multiple inputs together to manage the simulation in an easier way.

10. **BONUS:** Implement the MTCMOS latch design discussed in class (also shown in Figure 6 of the reference paper [1]). Run SPICE simulations showing propagation delay, dynamic power, and leakage power (in active and sleep mode). (Note: For those of you who are very familiar with SPICE, I especially encourage you to do this bonus part.)

Please hand in the following:

a) Schematic of design. Include device sizes for each transistor. You can choose which parameters are shown in the schematic from the properties window (select component and press 'q'). You will have to export the schematic as an image: go to **File > Export Image …** then select a white background in the Appearance section and save it to the desired location.
b) Timing diagrams of all simulations (different inputs changing can be done on the same simulation run). Make sure to apply a white background before saving as an image: go to **Graph > Properties** and select a white background. You can save the plot going to **File > Save Image …**
c) Summary of delay and total current (and power) estimate information from SPICE. Include a table summarizing all results.
d) A full discussion on the significance of your results. Just reporting results in a table with no discussion regarding their significance is not very useful to me.

You may find the following papers helpful in completing this assignment:

[1] S. Mutah, T. Douseki, Y. Matsuya, T. Aoki, S. Shigematsu, and J. Yamada, "1-V Power Supply High-Speed Digital Circuit Technology with Multi-Threshold Voltage CMOS," *IEEE Journal of Solid-State Circuits*, vol. 30, no. 8, pp. 847–853, 1995

[2] M. Anis, S Areibi, M. Elmasry, "Design and Optimization of Multithreshold CMOS (MTCMOS) Circuits," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Volume 22, Issue 10, Oct. 2003, pp. 1324–1342