

**HOMEWORK 1.****Due: Wednesday, February 15, 2017.****This is an individual assignment!****The goal of this assignment is to get familiar with the class technology. It could be fairly long – knowledge of some scripting language (like Python) could be useful.****1. Models**

Use the SPICE model to characterize the class 32nm LP CMOS process; parameter files correspond to devices n105 and p105 in the process directory (~ee241/synopsys-32nm/hspice/saed32nm.lib – In your SPICE files, you can include the line `.lib '/home/ff/ee241/synopsys-32nm/hspice/saed32nm.lib' TT` ). The nominal supply voltage for this process is 1.05V. Note that the devices need to be instantiated as subcircuits in SPICE (Use an X prefix, not M).

- a) Determine the threshold voltage  $V_{Th}$ , for the NMOS and PMOS devices (for  $V_{BS} = 0$ ,  $L = 32\text{nm}$  and  $W = 1\mu\text{m}$ ), by extrapolating from the  $I_D$ - $V_{GS}$  curve at low  $V_{DS}$ . Explain your circuit setup. How does this result compare to values reported in the model file and the DC OP analysis?
- b) For the model used in class  $I_{DSat} = \frac{W}{L} \frac{\mu_{eff} C_{ox} E_C L}{2} \frac{(V_{GS} - V_{Th})^2}{(V_{GS} - V_{Th}) + E_C L}$ , find the values of  $E_C L$  that best fit the NMOS and PMOS characteristics. Use the  $V_{Th}$  value from part a). Note that you may want to modify the  $I_{DSat}$  equation to account for finite output resistance.
- c) We will try to extract the parameters  $V_{Th}$  and  $\alpha$  for the alpha-power-law model  $I_D = K(V_{GS} - V_{Th})^\alpha$  from SPICE simulations. Use Matlab to determine  $K$ ,  $V_{Th}$  and  $\alpha$  (hint: use the `lsqcurvefit` function). Determine the parameters for both NMOS and PMOS transistors.
- d) By setting  $\alpha = 1$ , find the best  $V_{Th}$ 's that correspond to linear dependence of current on  $V_{GS}$ .

## 2. Transistor sizing

In this problem, we will explore optimal transistor sizing.

- a) Using SPICE and 32nm LP model with 1.05V supply, find the required width of the PMOS transistor that minimizes the propagation delay  $(t_{pHL} + t_{pLH})/2$  for the CMOS inverter. NMOS transistor width is fixed at 100nm.
- b) Find the intrinsic delay of this inverter,  $p$ . It is set by the ratio of diffusion to gate capacitances, but it is better to find it from the delay measurements.
- c) Optimally size the CMOS NAND2 gate. Find the required width ( $W$ ) for the NMOS transistors in the pull-down such that the equivalent resistance of the pull-down network is the same as the equivalent resistance of the pull down transistor in an inverter from part a). Use hand analysis with parameters from Problem 1. (In case you haven't been able to solve Problem 1, you can use  $E_C L = 0.8V$ ,  $V_{DD} = 1V$ , and  $V_{Th} = 0.5V$ ). Compare with SPICE and discuss any discrepancies.
- d) Find the logical effort and the intrinsic delay of the NAND2 gate from part c).
- e) Repeat part c) for a NAND3 gate. Compare with SPICE and discuss any discrepancies. Find the logical effort of this gate.
- f) Optimally size NOR2 gate by using SPICE to match the inverter from part a). Find the logical effort.

## 3. Design a 5-32 decoder using a typical VLSI flow

- a) Work through GCD: VLSI's Hello World (on the course website / Github), a brief tutorial that will describe how industry tools translate RTL code to final layout. Try to understand the high-level concepts of what is happening and don't worry too much about the details for now. Turn in:

-The first 45 lines of the post-place-and-route Verilog netlist

- b) In your /scratch/username directory on hpse-(9-15).eecs.berkeley.edu, type:  

```
source ~ee241/tutorials/ee241.bashrc
git clone ~ee241/tutorials/decoder
```

  
Edit src/decoder.v, and write a verilog description of a 5 to 32 decoder. Open src/decoder\_tb.v (a test bench we wrote for you) and try to understand the file. Next, edit build/constraints.tcl and build/Makefrag and describe the following constraints: 30fF load capacitance on each output pin, inputs driven by a minimum sized inverter, and target maximum frequency. Then synthesis, place-and-route, and test the design (dc-syn/, icc-par/, vcs-sim-gl-par/), and turn in:

- A screenshot of the layout
- The critical path in ns
- A post P&R DVE simulation waveform showing correct functionality and the critical path