EE 577A – VLSI System Design Spring 2019
Pierluigi Nuzzo TA: Hongfei
Lab 3

Score:___/100

Student ID:________________________ **Name: ________________________**

Assigned: March 29th
Due: April 5th at 11:59pm (Submit via the provided DEN link). No Late submissions.

Notes:

- This lab introduces you to FinFET circuit design as well as simulation with HSPICE.
- Your lab report must be **one pdf file**, which includes all the materials. Other file formats will not be graded.
- In this lab, please use viterbi–scf1.

FinFET Circuit Design

Double-Gate FinFET Basics

Fin-type field-effect transistors (FinFETs) are promising substitutes for bulk CMOS at the nanoscale. FinFETs are double-gate devices. The two gates of a FinFET can either be shorted for higher performance or independently controlled for lower leakage or reduced transistor count. This gives rise to a rich design space. This lab provides an introduction to various interesting FinFET logic design styles, and novel circuit designs. Figure 1 shows the shorted-gate (SG) FinFETs where the two gates are connected together, leading to a three-terminal device, and the independent-gate (IG) FinFETs, the top part of which is etched out, giving way to two independent gates. Because the two independent gates can be controlled separately, IG-mode FinFETs offer more design options.

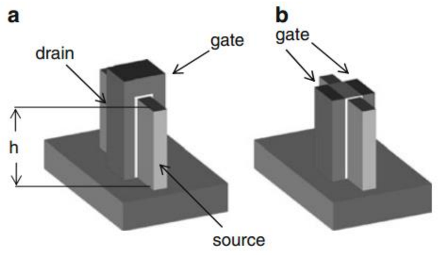

Fig 1. (a) SG-mode FinFET; (b) IG-mode FinFET

In general, three modes of FinFET logic gates are logically obvious: (1) SG-mode, in which FinFET gates are tied together; (2) low-power (LP)-mode, in which the back-gate bias is tied to a reverse-bias voltage to reduce subthreshold leakage; and (3) IG-mode, in which independent signals drive the two device gates.

Figure 2 shows the implementation of a two-input NAND gate in each of the above modes. A hybrid IG/LP-mode NAND gate, which employs a combination of LP and IG modes is also presented. Similarly, other Boolean functions can be implemented in CMOS styles in each of the above-mentioned modes. (Vhi and Vlow are the corresponding reverse-bias voltages.)

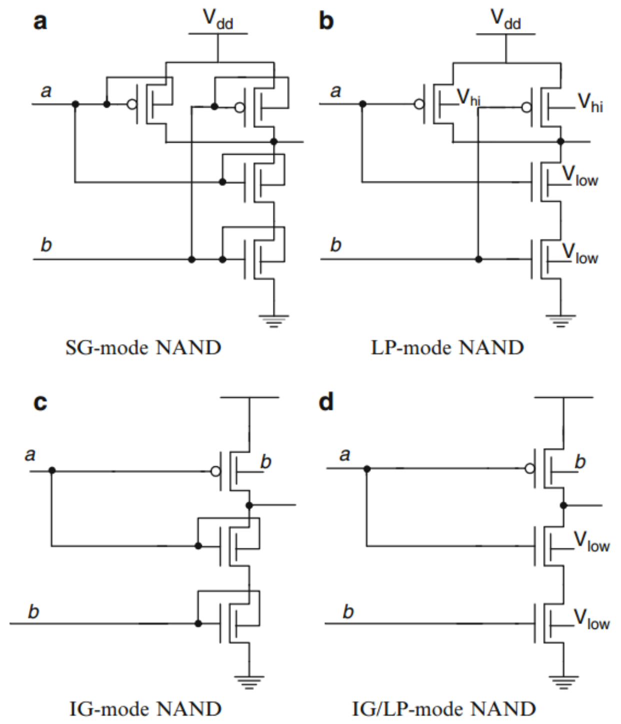

Fig 2. Different FinFET-based NAND gate designs

In this lab, you need to build basic gates for a given 7nm FinFET process, and compare the performance and power consumption of IG, LP, SG and IG/LP gate designs.

What you are given

- hp7nfet.pm: n-type finfet model file from 7nm PTM high performance FinFET library
- hp7pfet.pm: p-type finfet model file from 7nm PTM high performance FinFET library

Basic Setup for Hspice Simulation

- VDD=0.7V
- Input slope=10ps
- Gate length=11nm (it is still 7nm technology)
  .param lg=11n
- Upload the model files in the simulation folder and include the following lines:
  .include "hp7nfet.pm"
  .include "hp7pfet.pm"
- Define parameters for the number of fins:
  .param n_fin = 1
  .param p_fin = 1
- Example of drawing a single inverter:
  mn1 Z A Gnd Gnd nfet L=lg NFIN=n_fin
- mp1 Z A Vdd Vdd pfet L=lg NFIN=p_fin
- The back gate pin is the Body pin of the FinFET model
- Rise/Fall Delay: 0.5VDD of input to 0.5VDD of output
- Rise/Fall Time: 0.2VDD/0.8VDD of output to 0.8VDD/0.2VDD of output

Tasks:

- Find the best number of fins for SG-mode 1X INV, NAND, NOR gates such that the rise and fall delays are balanced. Report the rise delay, fall delay, rise time, fall time, number of fins for n type and p type finfets for each gate, respectively.

- Find the rise delay, fall delay, rise time and fall time for 2X, 4X INV, NAND, NOR gates, respectively.
- Based on the 4 different 1X NAND gate designs in Figure 2, draw 4 1X NOR gate designs using the same number of fins. Compare the following items:
  1. rise delay, fall delay, rise time, fall time
  2. leakage power, switching power
  3. switching energy
- Design a 1-bit Full-Adder using any mode of FinFET.
  1. Perform functional test for all possible combinations of inputs (A, B, C) = (0,0,0)~(1,1,1) and report the output.
  2. Report Rise/Fall Delay.

Hints

You may write multiple hspice files and a perl or python script to assist the simulation process. It is possible that the back gate won't affect the performance significantly.

What you need to submit

One PDF report which contains:

1) For each task, you need to have screenshot of ONE hspice file (though you have multiple hspice files)
2) Perl/python script screenshot if you have any
3) Results of each task

Reference:

Mishra, Prateek, Anish Muttreja, and Niraj K. Jha. "Finfet circuit design." Nanoelectronic Circuit Design. Springer New York, 2011. 23-54.
PTM library available [online]: http://ptm.asu.edu/

PTM FinFET DRAM

Hspice Simulation

1) Implement a one-bit 1T DRAM cell in Hspice initially with Fin#=1 and C=1fF using the PTM FinFET CMOS technology and the circuit setup shown in Figure 1.

2) Do transient analysis for a time duration T. T should be selected long enough to see V_C discharging to $V_{DD}/2$ (due to leakage). The time it takes for V_C to discharge from V_{DD} to $V_{DD}/2$ is referred to as Refresh Time Limit (R.T.L.). Use the trig-targ command in Hspice to calculate the R.T.L. Also use the AVG command to find the average capacitance current (I_{avg}). Finally, use the AVG command to measure the average power dissipated by the capacitance (P_{avg}).

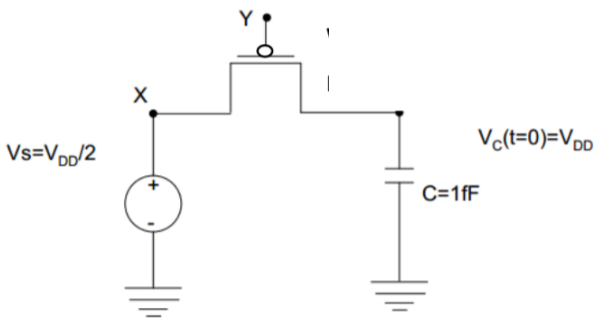

Figure 1: Circuit setup

Python scripting and Matlab analysis

3) Sweep Fin# between 1 and 5 and sweep C logarithmically between 1fF and 50pF for 11 different values (1f, (50,000)^(0.1)f, (50,000)^(0.2)f, …, (50,000)f). Calculate the R.T.L., I_{avg}, P_{avg} for each combination of C-W values and print the results in a text file named dram.txt. Note that the larger is C, the longer is T needed to measure R.T.L.
4) Write a Matlab m-file, analysis.m, to read the dram.txt file and plot the 3D graphs of
   i. R.T.L. vs. (C, Fin#)
   ii. I_{avg} vs. (C, Fin#)
   iii. P_{avg} vs. (C, Fin#)
5) In the eval.m calculate the following statistics: min, max, standard deviation, and average of
   i. R.T.L.
   ii. I_{avg}
   iii. P_{avg}

Implementation Details:

- Hspice does not have a GUI and we need other software to visualize the output waveforms. In this lab, we use Wave Viewer.
- Run the hspice file using this command (filename.sp is the name of the hspice file you created): hspice filename.sp
- You should see no error after running hspice (warnings are fine).
- A lot of output files will be created by hspice. We are interested in *.tr0 (transient analysis output) and *.mt0 (measurements output).
- You can download Matlab to your laptop from ITS: http://www.usc.edu/its/software/
- Useful commands for Matlab:

```
Matlab  –nojvm
help max, min…
Comment %
Example:
wset=load('/home/scf-16/yue/EE577a/dram.txt');
c=[1:1:10]; %x-array: one-dimensional
d=[1:1:10]; %y-array: one-dimensional
h=c'*d; %z-matrix: two-dimensional
surf(c,d,h) % Plot 3D graph
```

Report

1) The spice deck
2) Perl/Python script
3) dram.txt

4) analysis.m
5) 3D graphs of R.T.L., I_{avg} and P_{avg}
6) Calculated statistic values