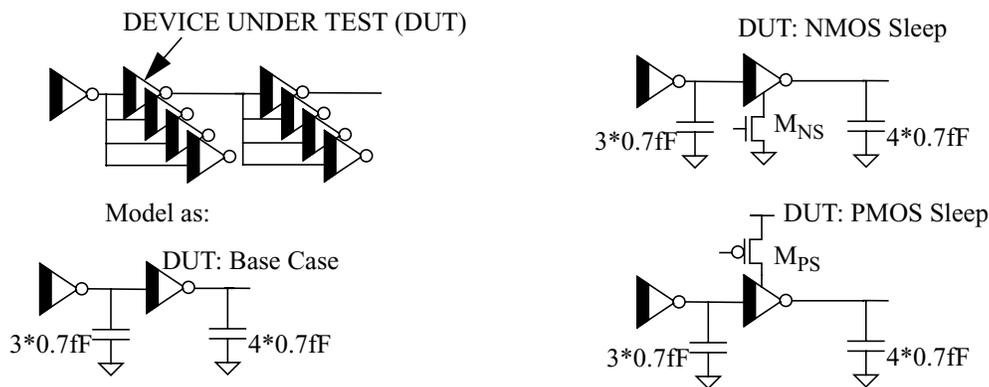# 6.374: Analysis and Design of Digital Integrated Circuits
Problem Set # 4 Solutions

Problem 1: MTCMOS Sleep Device Sizing

Figure 1: A driver circuit schematic

This problem examines sizing for MTCMOS sleep devices in a cutting edge technology. The devices in the problem are 0.07μm FETs modeled by a Predictive BSIM model. We generated the predicted models using Berkeley Predictive Technology Model (*http://www-device.eecs.berkeley.edu/~ptm/introduction.html*). You may find this site useful for generating models to use in your project.

Please use the following assumptions in this problem:
- Every inverter has a fanout of 4. The gate capacitance of an inverter can be approximated as 0.7fF/FET.
- The inverter has PMOS W/L = 0.22um/0.07um and NMOS W/L = 0.1um/0.07um.
- V_{DD}=1.2V.
- AS=AD=6*λ*W and PS=PD=10*λ + W.
- The minimum grid size is 0.0025um. In other words, the width of your FETs should be N*0.0025u, N=integer.
- To get the model files, type the following four lines:
```
.include 'nfet007hvt.l'
.include 'nfet007lvt.l'
.include 'pfet007hvt.l'
.include 'pfet007lvt.l'
```
- To instantiate a FET, use the following names: NMOShvt, NMOSlvt, PMOShvt, PMOSlvt.
- When finding leakage, assume V_{GS}=0. Measure leakage for the input that gives the worst-case improvement.
- HINT: Use subcircuits to make your life easier. The documentation on the webpage shows how.
- HINT: To simulate settling leakage currents, add the following lines to your spice deck:
```
.options accurate
.options method=gear
```

a) For the circuits in Figure 1, use HSPICE to find a size for M_{NS} and another for M_{PS} such that the inverters only see 10% reduction in propagation delay. Repeat your simulation to find sizes to achieve only 5% degradation in t_p. Turn
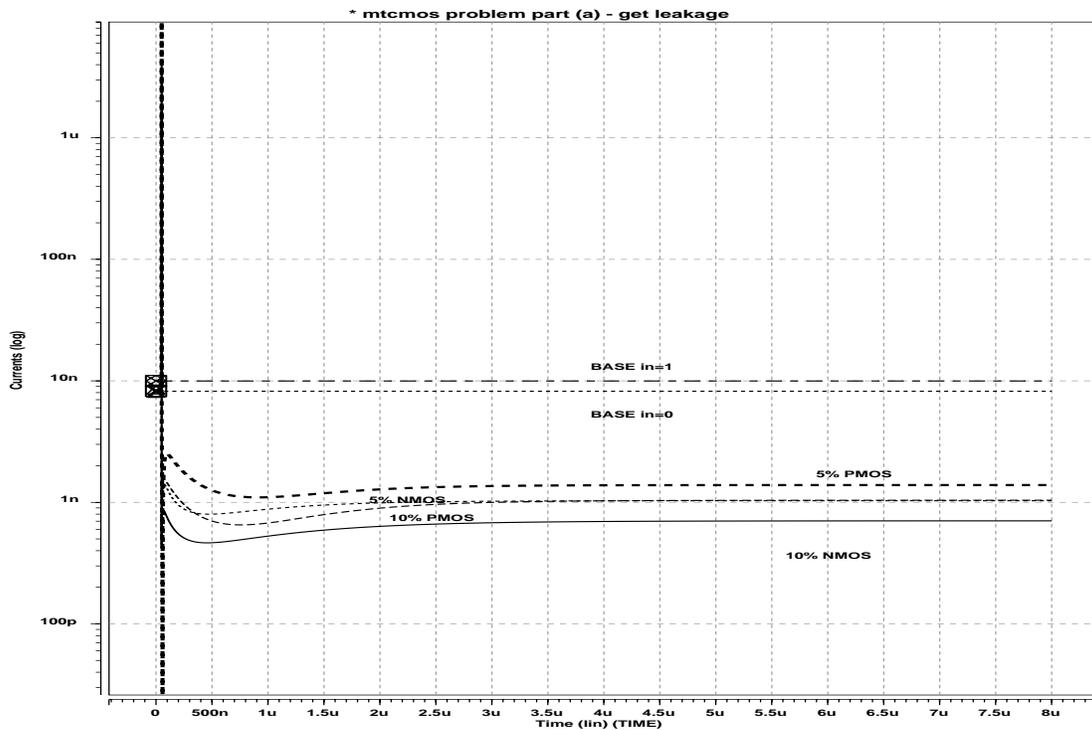
in ONLY the following:

i) Fill out Table 1.

ii) Transient simulation plot showing I_D settle to the sleep value when the inverter enters sleep mode. Use the drain current in the NMOS device in the inverter for your measurements. Show the current settling for all four cases on one plot (PMOS 5%, PMOS 10%, NMOS 5%, NMOS 10%). Use a SEMILOG plot (Y-axis log).
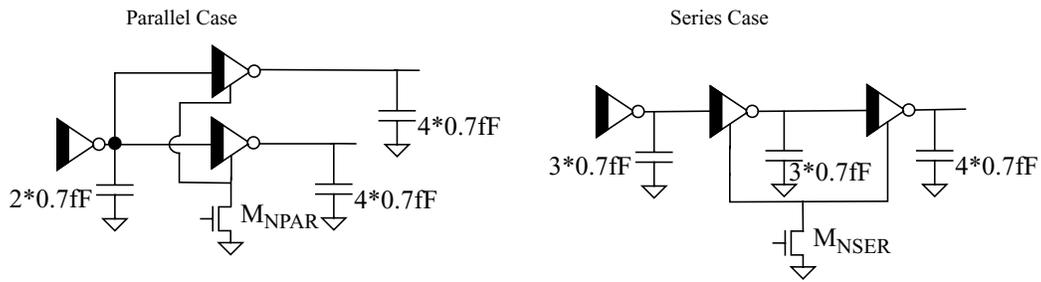
SOLUTION:

Table 1: Results from Part (a)

Delay Penalty	M_{NS} (um)	M_{PS} (um)	I_{DBASE} (nA)	I_{DNS} (nA)	I_{DPS} (nA)	I_{DBASE}/I_{DNS} (X)	I_{DBASE}/I_{DPS} (X)
5%	0.44	1.4375	9.9718(in=1) 8.2115(in=0)	1.032(in=1)	1.389(in=0)	9.66	5.91
10%	0.2325	0.725	9.9718(in=1) 8.2115(in=0)	0.706(in=1)	1.043(in=0)	14.12	7.87

These numbers depend reasonably strongly on the measured propagation delay (which varies with the transient step size). Do not worry if your numbers are different as long as they are in the right vicinity.

**Figure 2:** Sizing Sleep Devices for Parallel and Series Inverters

b) Refer to Figure 2. Use HSPICE to size M_{NS} for both cases to achieve 5% and 10% delay degradation over the *appropriate* base case. Turn in ONLY the following:
i) Fill out Table 2.
SOLUTION:

Table 2: Results from Part (b)

Delay Penalty	RSeries M_{NPAR} (um)	Parallel M_{NSER} (um)	I_{DBASE} (nA)	I_{DSER} (nA)	I_{DPAR} (nA)	I_{DBASE}/I_{DSER} (X)	I_{DBASE}/I_{DPER} (X)
5%	0.97	0.2975	18.1834(ser,in=0,1)	0.826(in=0)	2.184(in=0)	22.01	7.52
10%	0.485	0.165	19.9436(par,in=1) 16.423(par,in=0)	0.5774(in=0)	1.444(in=0)	31.49	11.37
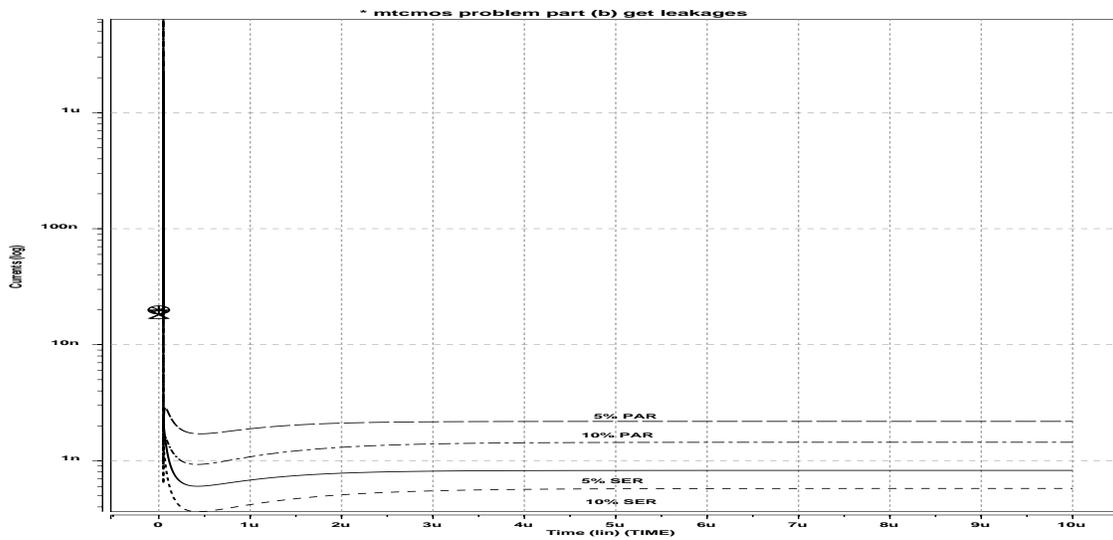
c) Refer to Figure 3. You may use any number of sleep devices with or without sharing. You also may insert a 2-input gate into the dotted box. Use HSPICE to minimize standby leakage current while maintaining no more than a 10% delay penalty. Turn in ONLY the following:

i) A schematic of the circuit including the sleep devices and their sizes.

ii) The total width of the sleep devices.

iii) The total leakage savings in standby mode relative to active mode leakage (in X, for worst-case).

iv) A transient simulation showing the total leakage current settling to its steady-state value when the circuit enters sleep. Use a SEMILOG plot (Y-axis).
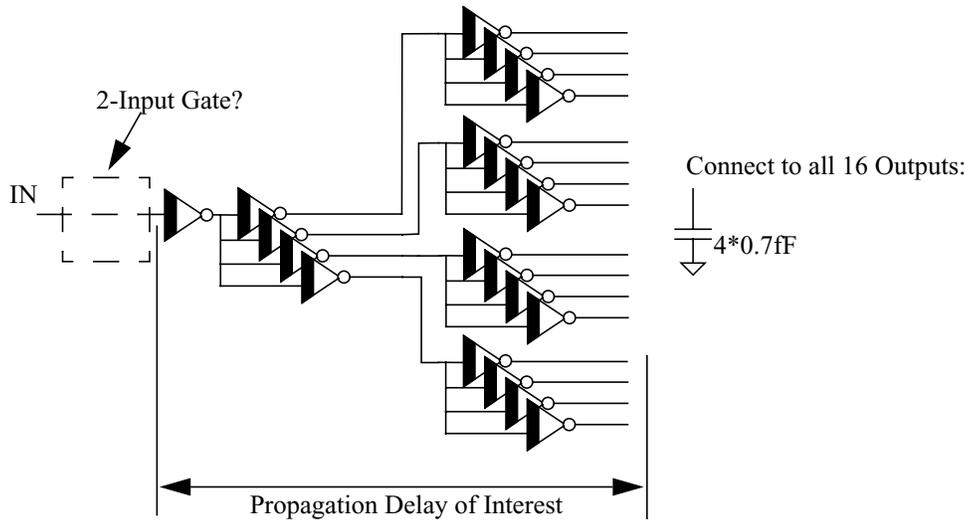
Figure 3: Sizing Sleep Device(s) for an Inverter Tree

SOLUTION:

There are many possible solutions to this problem. A few good options are:

- Use 1 PMOS device for the input stage and the output stage. Use one NMOS device for the middle stage. Use a NOR gate at the input with SLEEP as the second input to take advantage of the stack effect. Using this approach, I got 196X leakage savings without much optimizing. The total width was 7.05um so savings were about 28X/um.
- Use 1 PMOS device for the entire thing. Less sleep device area, but leakage dominated by the stage which has no stack effect.
- Same as the top two, but with NMOS and PMOS switched. Advantage: less area. Disadvantage: lower leakage savings.

Since we are supposed to minimize leakage, we DEFINITELY should use the 2-input gate to generate a known input vector during sleep mode. This lets us use the stack effect to our advantage.

Problem 2: Adder Design

You are to design, layout, and simulate an **16-bit ripple carry adder** with the following specifications:

Table 3: Input/Output Signals

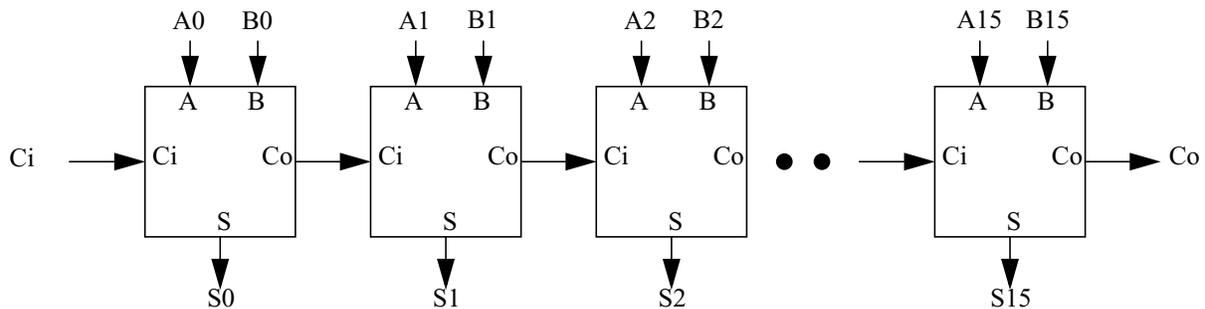| Input	Description	Output	Description
A[15:0]	15-bit operand	S[15:0]	15-bit output sum vector
B[15:0]	15-bit operand	Co	Carry out
Ci	Carry in		

Figure 4: 16-bit ripple carry adder

Your inputs have to go through two minimum length inverters (W_p/W_n=1.125/0.375) before driving the adder inputs in your netlist (do **NOT** layout these inverters). All outputs have a capacitive load of 30fF.

a) Layout the adder in MAGIC using **ANY** static CMOS logic style with rail to rail outputs.

Your goal is to **minimize energy/addition** meeting a specific delay constraint, by lowering V_{dd} and/or by your choice of static circuit style. The worst case critical path delay must be $t_p \leq 4ns$. Observe good layout techniques that reduce parasitic capacitances and resistances. Turn in **ONLY**:

i) a schematic of your 1-bit adder implementation

ii) the layout of the 1-bit adder and its area

iii) the layout of the full 16-bit adder and its area

After problem set 2 you have become experts in layout, so your layout should be compact and not excessively spread out. This time there will be points off for unnecessary wasted area (money).

You must print a **color** version of your layout.

Solution

There is no wrong or right answer.

We have examined many different adder designs with different trade-offs.

b) Extract and verify that your adder works. You have to use the input vector file located in: *input.vec*. Use SimWave to view your nanosim waveforms. (see FAQ for details). Does your adder work? (Yes/No).

c) Determine the critical path in the circuit (tell us the input vectors that give the worst case delay), and simulate the extracted circuit in HSPICE with the input pattern displaying the critical path. Turn in a plot showing the worst case propagation delay of your adder.

Solution

The worst case delay for a ripple carry adder, happens when at every one-bit full adder, the carry out bit makes a transition. Assuming equal rise and fall times, a transition that could give the maximum delay for an adder is A=0x0000, B=0xffff and C=0->1.

d) Using the input vector file located in: *input.vec*, report the dissipated power at V_{dd}= 2.5V, when the input frequency is 100Mhz. What is the average energy per addition?

Solution

The average energy per addition depends on the way that you implemented the adder.

e) Using the input vector file located in: *input.vec*, report the dissipated power at your desired power supply voltage, when the input frequency is 100Mhz. Your design must meet the delay constraint ($t_p \leq 4ns$), at this operating voltage (see part(c)). What is the average energy per addition?

Solution

Assuming you met the timing constraint and there was a delay margin that you could exploit, you could lower the supply voltage to reduce the power consumtion of your circuit.

Layout Requirements

All A,B inputs should come in from the top of the cell in metal 1 or 2. All S outputs should come out from the bottom of the cell in metal 1 or 2.

Useful hints and suggestions :
• To find out your plot area in MAGIC use macro 'f' followed by 'b'. This selects your cell and 'boxes' it. You should get the area of the box in terms of your grid size.
• Remember device sizes in Magic are multiples of lamda