University of California
College of Engineering
Department of Electrical Engineering
and Computer Sciences

E. Alon

Tuesday, December 15, 2009
5:00-8:00pm

EECS 141: FALL 2009—FINAL EXAM

For all problems, you can assume that all transistors have a channel length of 100nm and the following parameters (unless otherwise mentioned):

NMOS:
V_{Tn} = 0.2V, μ_n = 400 cm^2/(V·s), C_{ox} = 1.125 μF/cm^2, v_{sat} = 1e7 cm/s, L=100nm, γ=λ=0
PMOS:
$|V_{Tp}|$ = 0.2V, μ_p = 200 cm^2/(V·s), C_{ox} = 1.125 μF/cm^2, v_{sat} = 1e7 cm/s, L=100nm, γ =λ=0

NAME	Last First

GRAD/UNDERGRAD	

Problem 1: _____/ 26

Problem 2: _____/ 19

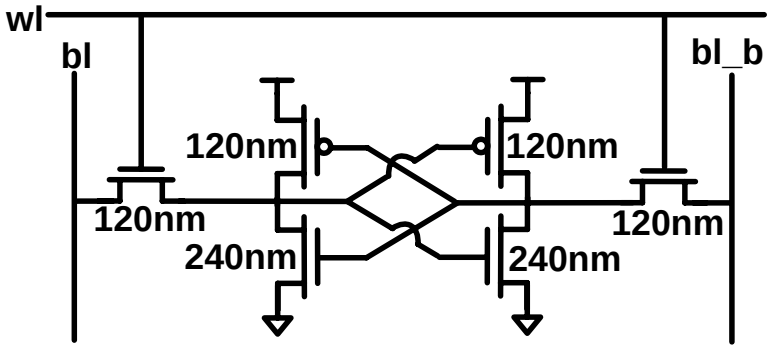
Problem 3: _____/ 22

Problem 4: _____/ 26

Total: _____/ 93

PROBLEM 1: SRAM Design (26 pts)

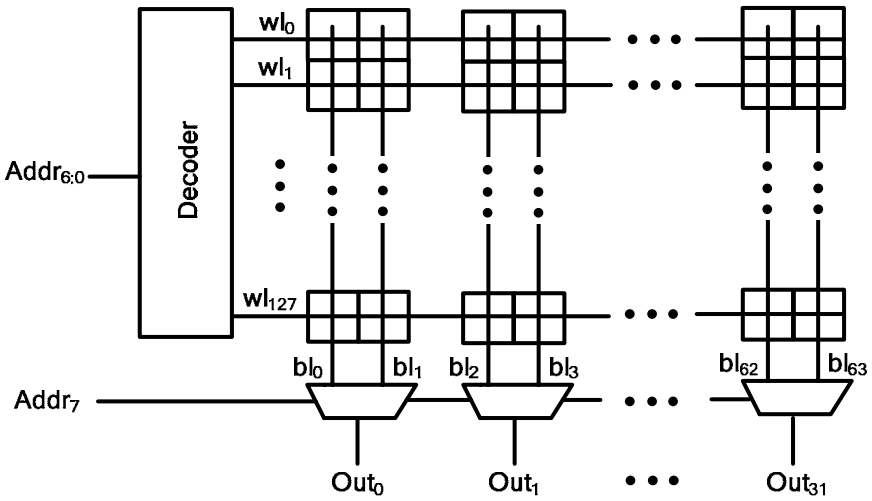
For this problem we will be looking at a 256x32 SRAM (i.e., each wordline drives 32 cells, and each bitline has 256 cells on it), with each cell sized and implemented as shown below. You can assume that $C_G = C_D = 2fF/\mu m$, $R_{sqn} = 10k\Omega/\square$, $R_{sqp} = 20k\Omega/\square$, that the transistors are quadratic (i.e., long-channel), and that you can ignore all wire parasitics.



a) (6 pts) Assuming you use a static decoder with only NAND2's and inverters, and that the input capacitance on each address input must be less than 2fF, what is the minimum delay of the decoder for this 256x32 memory? You should provide your answer in units of $t_{FO4} = (4+\gamma)t_{inv}$, you can assume that the parasitic delay of all of the NAND gates is γt_{inv} (i.e., the same as an inverter), and you can ignore the fact that you can't build a fractional number of stages.

b) (4 pts) For this same 256x32 SRAM, what is the delay (still in units of t_{FO4}) from the wordline rising to the bitline crossing Vdd/2?

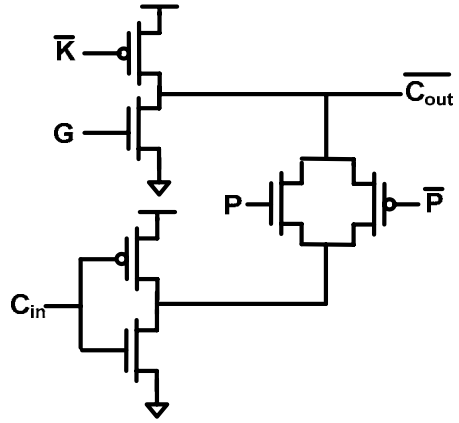
c) (8 pts) Now let's examine the effect of repartitioning the SRAM into a 128x64 array with 2-input MUXes selecting the appropriate final output, as shown below. Using your answers from parts a) and b), ignoring the capacitive loading of the MUX on the bitlines, and assuming that the delay of the MUX is 2 t_{FO4}, now what is the total delay of the SRAM from Addr to Out?



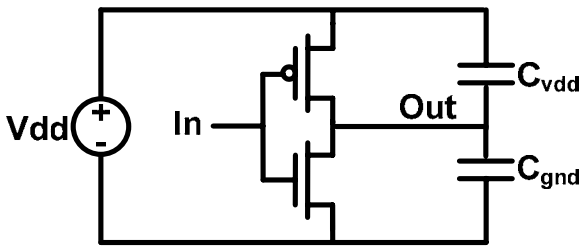
d) (8 pts) Now assuming that the SRAM is partitioned into a $256/N_{part}$ x $32*N_{part}$ array, what value of N_{part} results in the minimum total delay for the SRAM? You should assume that the overall logical effort of the decoder is independent of N_{part}, and that the delay of an N-input MUX is N t_{FO4}.

PROBLEM 2: Miscellaneous (19 points)

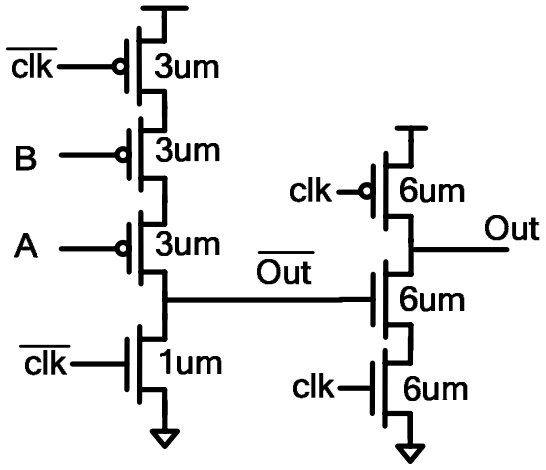
a) (5 pts) Assuming $R_{sqp} = 2*R_{sqn}$ and quadratic devices, size the gate shown below so that the worst-case pull-up and pull-down resistances are equal.



b) (6 pts) As a function of Vdd, C_{vdd}, and C_{gnd}, how much energy is pulled out of the supply voltage Vdd when In transitions from low to high? How about when In transitions from high to low? Note that you can ignore all capacitors associated with the transistors except those explicitly drawn in the figure.

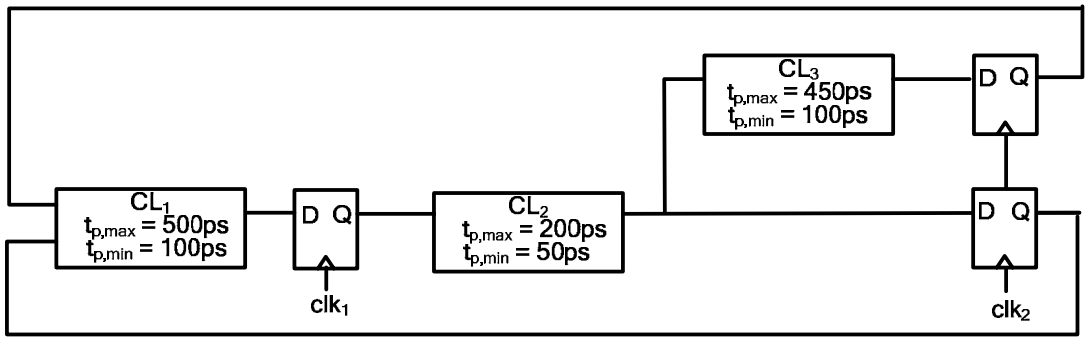


c) (8 pts) One of your colleagues (who didn't take EE141) shows you the circuit below and complains that "something is wrong with HSPICE" because their simulations show that the circuit functions correctly for Vdd = 0.6V, but that the output isn't always correct when Vdd=1.2V. Assuming that C_G=2fF/μm, C_D=1.5fF/μm, and that V_{TN}=|V_{TP}|=0.3V, explain why the circuit really doesn't work at Vdd=1.2V, and calculate the maximum supply voltage for which the circuit will function correctly. (Hint: Using the simple switch model, what are the VTCs of the dynamic gates shown below?)

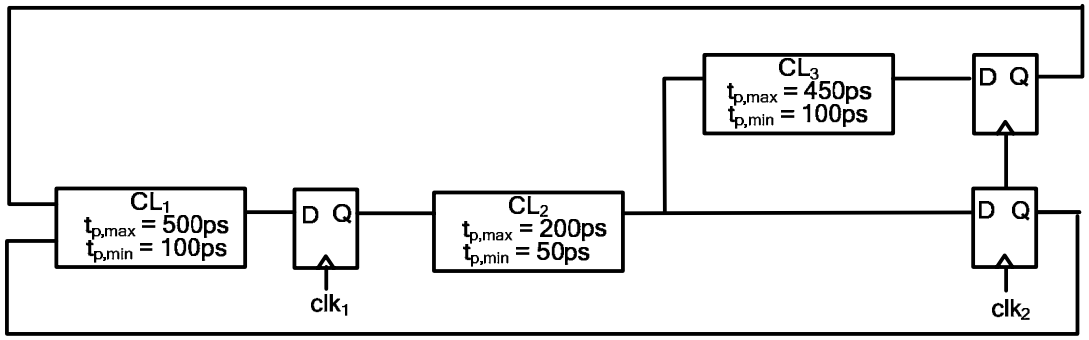


PROBLEM 3: Timing and Clock Distribution (22 points)

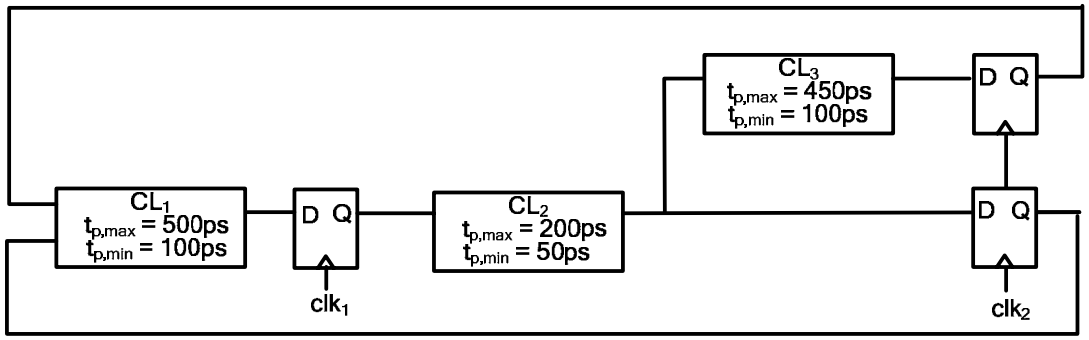
In this problem we will be examining the pipeline shown below. The minimum and maximum delays through the logic are annotated on the figure, and the flip-flops have the following properties: $t_{clk\text{-}q}$ = 50ps, t_{setup} = 50ps, and t_{hold} = 50ps. You can assume that the clock has no jitter.



a) (6 pts) Assuming there is no skew between clk_1 and clk_2, what is the minimum clock cycle time for this pipeline? Are there any minimum delay (hold time) violations?



b) (6 pts) Now let's assume that clk_1 and clk_2 can be skewed relative to each other by up to +/-60ps. Now what is the minimum clock cycle time? Are there any minimum delay violations?

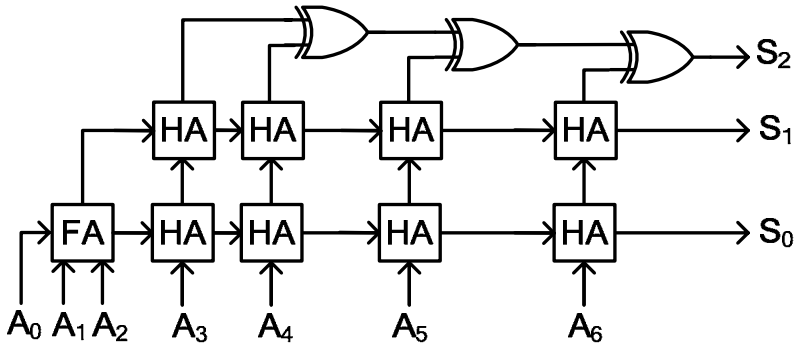


c) (10 pts) Continuing to assume that there can be +/-60ps of random skew between clk_1 and clk_2, can you introduce any intentional skew and/or delay into the circuit in order to reduce the minimum cycle time without causing any hold time violations? If so, you should indicate where in the pipeline you want to add skew or delay, calculate what the new cycle time is, and prove that you will not have any hold time violations. If not, you should explain why this it isn't possible to improve the cycle time without introducing a hold time violation.

PROBLEM 4: Arithmetic (26 pts)

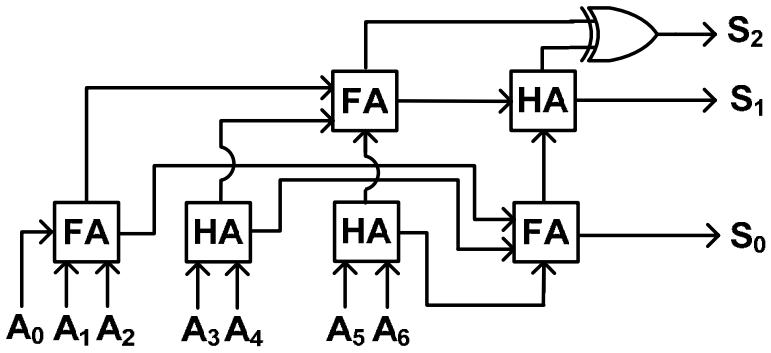
In this problem we will look at designing a circuit that adds together 7 1-bit binary numbers $A_{6:0}$ into one 3-bit output $S_{2:0}$ (whose value ranges from 0 to 7).

a) (4 pts) Shown below is a simple implementation of this circuit that uses only half adders (HA), full adders (FA), and XOR gates. Assuming that the sum and carry delays of the half and full adders are equal to each other and to the delay of the XORs, how many gate delays are there on the critical path for this circuit? (Please mark this path on the schematic.)



b) (4 pts) If each FA requires $10\mu m^2$ of area, each HA requires $6\mu m^2$, and each XOR requires $4\mu m^2$, how much total area is occupied by the circuit from part a)?

c) (8 pts) Ace says she has a better implementation for this circuit and shows you the schematic below. Under the same assumptions as part a) and b), how many gate delays are on the critical path of this circuit, and how much area does it consume? Once again, please highlight the critical path on the schematic.



d) (10 pts) Still using only full adders, half adders, and XORs, and without increasing the delay, draw an implementation for this circuit that uses even less area than the one from part c). How much area does this implementation require?