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

E. Alon **Thursday, December 20, 2007.**
12:30-3:30pm

EECS 141: FALL 2007—FINAL EXAM

For all problems, you can assume the following transistor parameters (unless otherwise mentioned):

NMOS:

$V_{Tn} = 0.4$, $k'_n = 115\ \mu A/V^2$, $V_{VSAT} = 0.6V$, $\lambda = 0$, $\gamma = 0.4\ V^{1/2}$, $2\Phi_F = -0.6V$

PMOS:

$V_{Tp} = -0.4V$, $k'_p = -30\ \mu A/V^2$, $V_{VSAT} = -1V$, $\lambda = 0$, $\gamma = -0.4\ V^{1/2}$, $2\Phi_F = 0.6V$

NAME	Last Solutions First

GRAD/UNDERGRAD	

Problem 1: _____/ 16

Problem 2: _____/ 21

Problem 3: _____/ 22

Problem 4: _____/ 16

Problem 5: _____/ 14

Total: _____/ 89

PROBLEM 1: SRAM Design (16 pts)

In this problem, we will be looking at a 256 x 256 SRAM, where each 6T cell is 4µm wide and 3µm tall. All of the devices in the SRAM cell are minimum length (L = 0.25µm), and you can assume that both the NMOS access transistors and the NMOS pull-down transistors are 0.5µm wide. You can also assume that when V_{DD} = 2.5V, the transistors have C_G = 2fF/µm, C_D = 1fF/µm, and $R_{sqn} = R_{sqp}/2$ = 15kΩ/□.

a) (2 pts) For V_{DD} = 2.5V, what is the total capacitance loading each bitline in this memory? You can assume that the bitline wire has a capacitance/length of 0.15fF/µm.

$$C_{diff,cell} = W \cdot C_D = 0.5\mu m \cdot 1fF/\mu m = 0.5fF$$

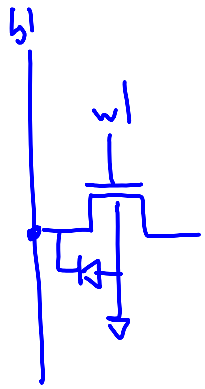
$$C_{wire,cell} = H \cdot C_w = 3\mu m \cdot 0.15fF/\mu m = 0.45fF$$

$$C_{total,cell} = 0.95fF$$

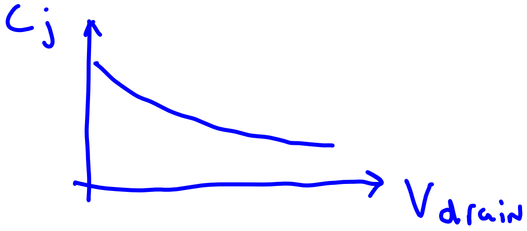
$$C_{bl} = 256 \cdot C_{total,cell}$$

$$\boxed{C_{bl} = 243.2fF}$$

b) (3 pts) If the supply voltage of the SRAM was reduced to 1.5V, would the effective capacitance on the bitlines increase, decrease, or stay the same? Why?

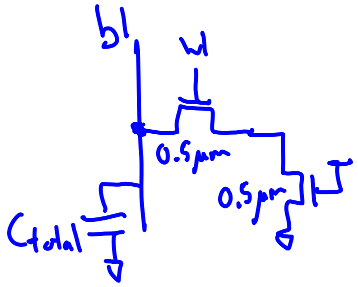


The effective bit line capacitance would increase because the reverse bias on the drain-to-substrate junction decreased from 2.5V to 1.5V. This decrease in reverse bias increases the linearized diffusion capacitance.



c) **(4 pts)** Given your answer to part a) (i.e., V_{DD} = 2.5V), and assuming that the delay of the final stage in the decoder is 100ps, what is the delay (in ns and including the slope effect) from the wordline rising to the bitline falling?

$$t_p = t_{p,step} + \frac{V_T^*}{V_{DD}} \cdot t_{p,in} \qquad V_T^* = V_T + \frac{V_{VSAT}}{2} = 0.7V$$



$$t_{p,step} = \ln(2) \cdot \frac{2 \cdot 0.25\mu m}{0.5\mu m} \cdot R_{sq,N} \cdot C_{total}$$

$$= \ln(2) \cdot 15k\Omega \cdot 243.2fF \approx 2.53ns$$

$$t_p = 2.53ns + \frac{0.7V}{2.5V} \cdot 100ps \qquad \boxed{t_p \approx 2.56ns}$$

d) **(4 pts)** If the memory operates at 100MHz, how much dynamic power is dissipated due to switching the 256 bitlines? You can assume that the bitlines swing all the way from 0V to V_{DD}.

Every cycle, either bl or bl_b will be pulled down when the memory is accessed. So:

$$P_{dyn} = 256 \cdot C_{bl} \cdot V_{DD}^2 \cdot f_{clk}$$

$$P_{dyn} = 256 \cdot 243.2fF \cdot (2.5V)^2 \cdot 100MHz$$

$$\boxed{P_{dyn} = 38.9mW}$$

e) **(3 pts)** Now we'll look at whether we can save some power by adding sense amps to this design. If the sense amps would dissipate 10% of the power you calculated in part d), but the bitlines now swing only between 2V and V_{DD} (=2.5V), will adding sense amps reduce the power of the SRAM? If so, by how much?

Recall that $P = Q \cdot V_{DD} \cdot f$, and $Q = \Delta V \cdot C$

So: $P_{bl,new} = \frac{2.5V - 2V}{2.5V} \cdot P_{bl,old} = 0.2 \cdot P_{bl,old}$

$$P_{total,new} = P_{bl,new} + P_{sense-amp}$$

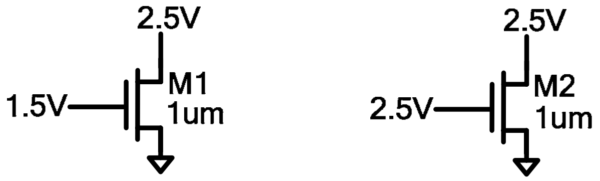
$$P_{total,new} = 0.2 P_{bl,old} + 0.1 P_{bl,old}$$

$$P_{total,new} = 0.3 P_{bl,old}$$

Power is reduced by 70% (11.7mW instead of 38.9mW)

PROBLEM 2: Logical Effort and Transistor Models (21 pts)

a) **(3 pts)** Using the unified transistor model, what is the ratio of drain currents for the two 0.25µm long transistors shown below? I.e., what is $I_{DS}(M1)/I_{DS}(M2)$?

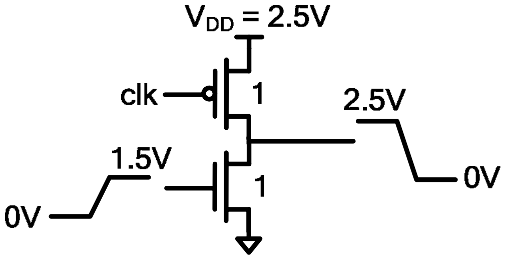


Both devices are in velocity saturation, where

$$I_{DS} = \frac{W}{L} k_N (V_{GS} - V_{TN} - V_{VSAT}/2) V_{VSAT}$$

So: $\dfrac{I_{DS}(M1)}{I_{DS}(M2)} = \dfrac{(1.5-0.4-0.3)}{(2.5-0.4-0.3)}$ $\boxed{\dfrac{I_{DS}(M1)}{I_{DS}(M2)} = \dfrac{4}{9}}$

b) **(4 pts)** Shown below is a dynamic (unfooted) inverter where the input to the inverter swings only between 0V and 1.5V, while the power supply for the dynamic inverter is 2.5V. During evaluation, what is the logical effort of this dynamic inverter relative to a standard inverter with $W_p = 2W_n$ and VDD = 2.5V? (Hint: your answer to part a) is directly related to this problem.)

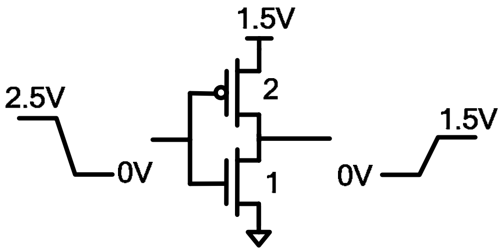


Since the pull-down device has less current than it would with $V_{GS} = 2.5V$, its effective resistance increases by $I_{DS}(M2)/I_{DS}(M1)$.

So: $R_{dinv} = \frac{9}{4}$ $C_{dinv} = 1$

$$LE_{dinv} = \frac{9/4 \cdot 1}{1 \cdot 3}$$ $\boxed{LE_{dinv} = \frac{3}{4}}$

c) **(6 pts)** Now we'll look at the static inverter (shown below) that would drive the dynamic inverter we just looked at. Once again comparing to a standard inverter with a supply voltage of 2.5V, what is the logical effort of this inverter for the rising transition at the output? (Hint: What is the effective resistance of the PMOS transistor?)



Remember that $R \propto \frac{V_{DD}}{I_{DSAT}}$

So, the relative resistance of the PMOS device is set by:

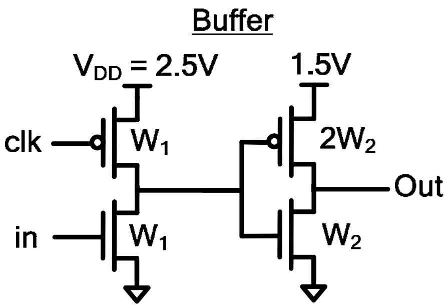
$$R_{PMOS} = \frac{1.5V}{I_{DSATP}(1.5V)} \bigg/ \frac{2.5V}{I_{DSATP}(2.5V)}$$

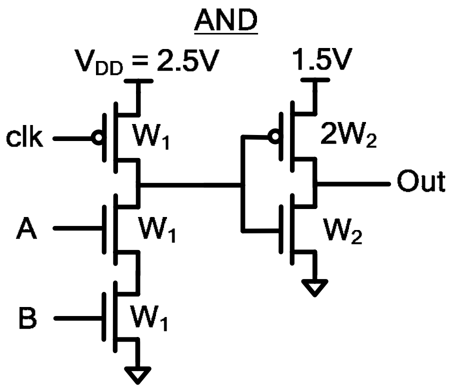
$$R_{PMOS} = \frac{1.5V}{2.5V} \cdot \frac{(2.5V - 0.4V - 0.5V)}{(1.5V - 0.4V - 0.5V)} = \frac{1.5}{2.5} \cdot \frac{1.6}{0.6} = 1.6$$

Since the input capacitance is the same, the LE is just the ratio of resistance:

$$\boxed{LE_{INV} = 1.6}$$

d) (8 pts) Let's now look at building chains of logic using such "alternating supply" domino gates (as shown below). Assuming your answer to part b) was LE_{dinv} = 0.75 and to part c) was LE_{inv} = 1.5, what is the EF/stage you should target to minimize the delay of a chain of these gates?





Just like normal domino gates, we first have to find the effective LE of our fastest buffer. In this case:

$$LE_{eff} = \sqrt{LE_{dinv} LE_{inv}} = \sqrt{\frac{3}{4} \cdot \frac{3}{2}} = \frac{3}{2\sqrt{2}}$$

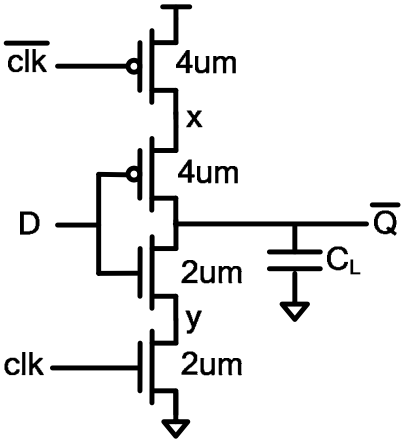
So, the optimum EF/stage is:

$$EF/stage = LE_{eff} \cdot 4$$

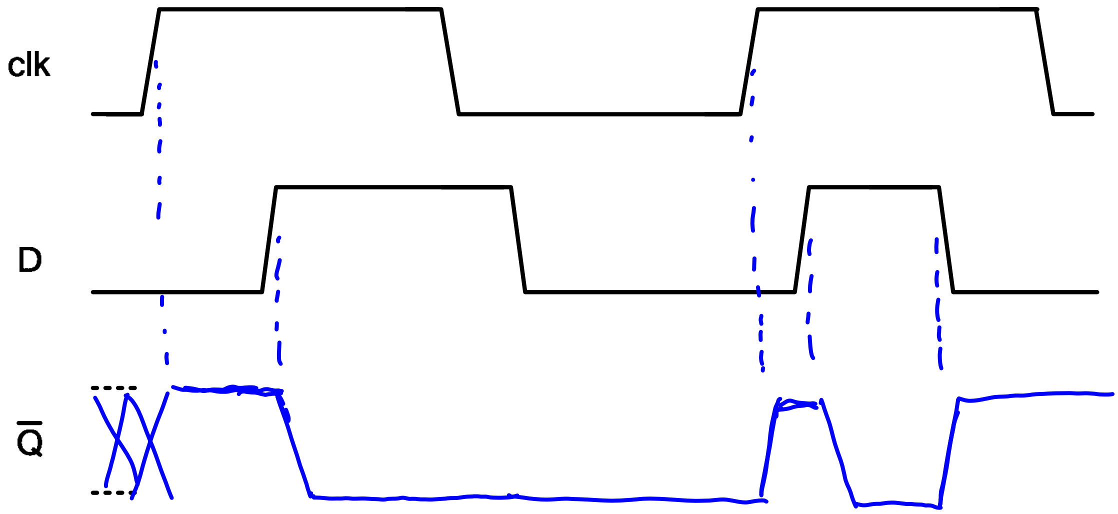
$$\boxed{EF/stage = 3\sqrt{2} \approx 4.243}$$

PROBLEM 3: Sequential Elements (22 points)

In this problem we will be examining the latch shown below, which has been implemented out of a tri-state inverter. Throughout this problem, you can assume that $V_{DD} = 2.5V$, $C_G = 2fF/\mu m$, $C_D = 2fF/\mu m$, and $R_{sqn} = R_{sqp}/2 = 15k\Omega/\square$.

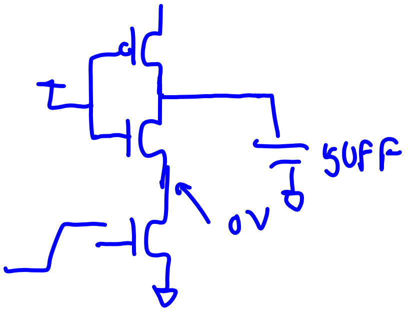


a) **(2 pts)** Assuming the latch is ideal (i.e., has no delay, zero setup/hold time, etc.), fill in the waveform for $\overline{Q}$ given the clock and data inputs shown below.

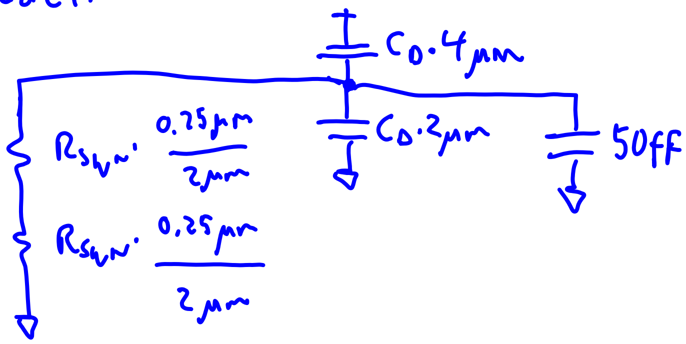


b) (6 pts) For C_L = 50fF and assuming $\overline{Q}$ is initially charged to V_{DD} and y is initially 0V, what is the $t_{clk\text{-}q}$ of this latch when D=1? You should assume that clk is a ramp.

D=1 :



RC model:


$C_D \cdot 4\mu m$
$C_D \cdot 2\mu m$
50fF
$R_{sq,n} \cdot \frac{0.25\mu m}{2\mu m}$
$R_{sq,n} \cdot \frac{0.25\mu m}{2\mu m}$

$$t_{clk-q} = 2R_{sq,n}\,\frac{0.25\mu m}{2\mu m}\cdot\left(C_D\cdot(2\mu m+4\mu m)+50fF\right)$$

$$\boxed{t_{clk-q} = 232.5ps}$$

c) (4 pts) One of your fellow designers comes to you one day and says that when she used this latch in her circuit with C_L = 5fF, the latch failed to function correctly. More specifically, if D transitioned, the inverter receiving the output of the latch would transition even when clk was low. However, you have verified that the latch functions correctly when C_L = 50fF. What is the cause of the error when the latch has a small C_L? (Hint: Think about the second transition of D)

The cause of this error is charge sharing. For example, on the second transition of D, x was previously charged to 2.5V while $\bar{Q}$ is at 0V (and should remain there). However, when D goes low and connects x to $\bar{Q}$, the voltage would rise well above $V_{DD}/2$ (because $C_x > 6\mu m \cdot C_D + 5fF$)

d) (6 pts) What is the minimum load capacitance required for the latch to function correctly? You can assume that the inverter receiving the output of the latch has an ideal VTC with a switch point $V_{sw} = V_{DD}/2$, and that none of the transistors' source/drain regions have been shared.

First, we need to find the worst case for charge sharing.

$$C_x = 4\mu m \cdot C_D + 4\mu m \cdot C_D + 4\mu m \cdot C_G = 24fF$$

$$C_y = 2\mu m \cdot C_D + 2\mu m \cdot C_D + 2\mu m \cdot C_G = 12fF$$

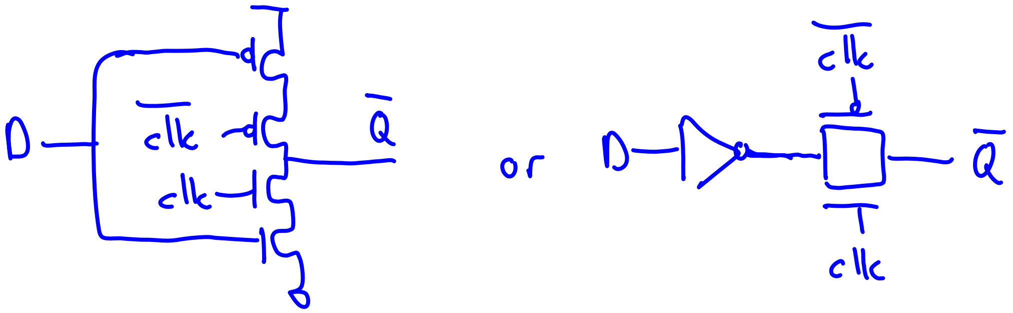
C_x is larger, so:

$$\frac{C_x \cdot V_{DD}}{C_x + C_{tot}} < \frac{V_{DD}}{2} \longrightarrow C_{tot} > 24fF$$

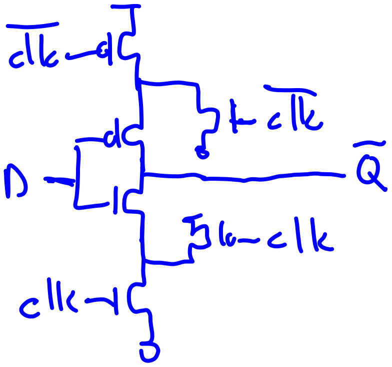
Capacitance from transistors is 12fF, so $\boxed{C_L > 12fF}$

e) (4 pts) Other than artificially increasing C_L, how can you modify the latch to fix the problem you identified in part c)? You should explain your fix and draw a new transistor-level schematic of the latch (no sizing necessary). (Note that there is more than one possible fix – you will receive bonus credit for up to two additional fixes you identify and draw.)

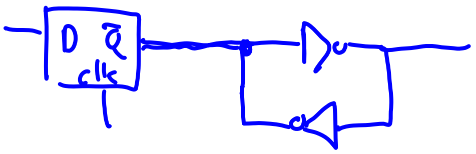
Fix #1: Re-order the latch so that never get charge sharing when clk is low:



Fix #2: Pre-charge/discharge the intermediate nodes.

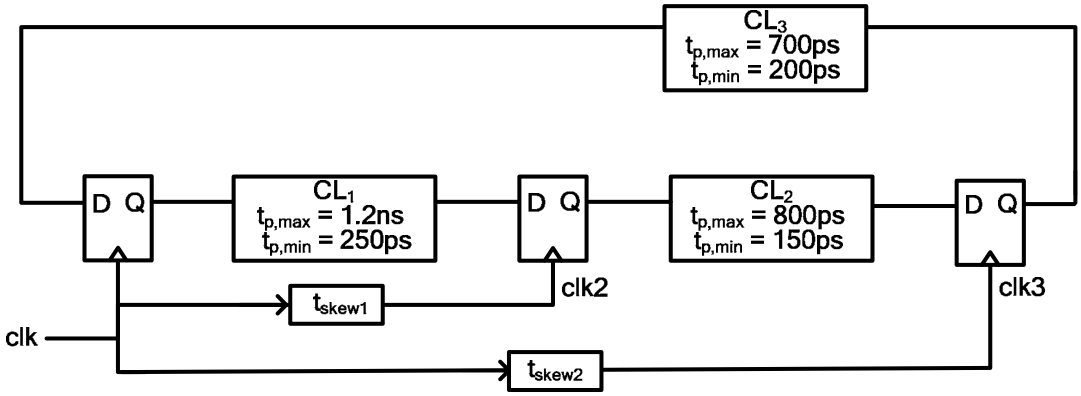


Fix #3: Keeper. Note that this fix is the least fool-proof, since the keeper could get flipped by the charge-sharing.



PROBLEM 4: Timing (16 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: $t_{clk\text{-}q}$ = 150ps, t_{setup} = 50ps, and t_{hold} = 100ps. You can assume that the clock has no jitter, but t_{skew1} and t_{skew2} can be either positive or negative.

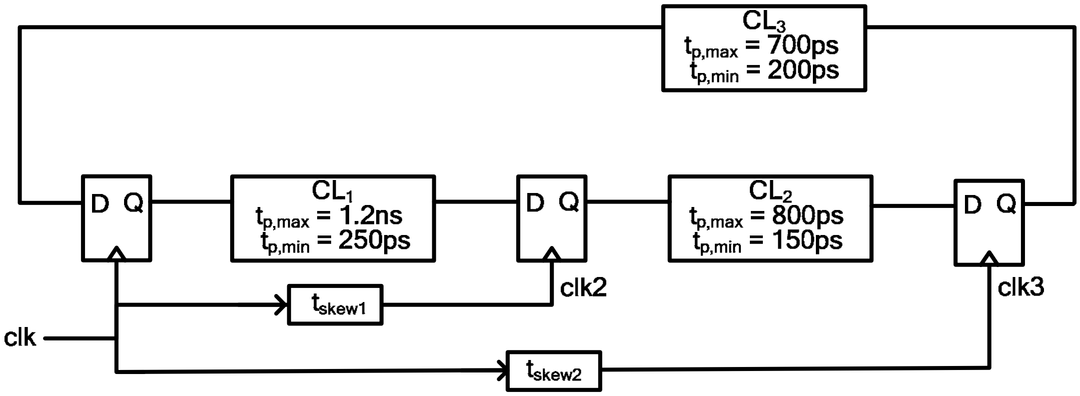


a) **(4 pts)** What is the minimum clock cycle time if t_{skew1} = -100ps and t_{skew2} = 50ps?

Worst-case set by CL_1:

$$T_{cyc,min} + t_{skew_1} = t_{clk,q} + 1.2ns + t_{setup}$$

$$\boxed{T_{cyc,min} = 1.5ns}$$

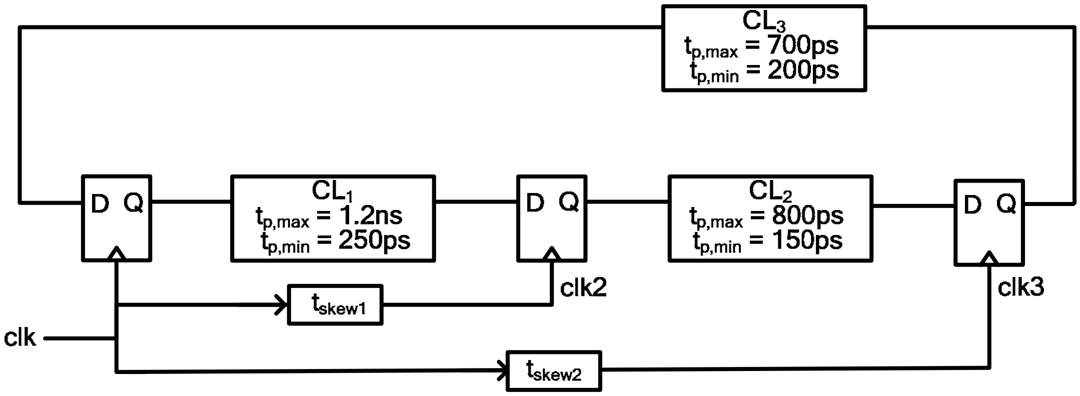


b) (4 pts) Assuming t_{skew2} is fixed at 50ps, how negative can t_{skew1} be before this pipeline (repeated above for your convenience) fails a hold-time constraint?

Negative tskew1 gives more time for signal to race through CL₂. So:

$t_{skew1} + t_{clk\text{-}q} + 150ps > t_{hold} + t_{skew2}$

$$\boxed{t_{skew1} > -150ps}$$



c) **(8 pts)** If you could intentionally set the values of t_{skew1} and t_{skew2}, what values would you choose in order to minimize the cycle time of this pipeline (again repeated above)? What would be the cycle time in this case?

First, let's look at total logic delay.

$$t_{p,max_1} + t_{p,max_2} + t_{p,max_3} = 2.7ns$$

Best we can do is get $T_{cyc} = t_{clk-q} + \frac{2.7ns}{3} + t_{setup}$

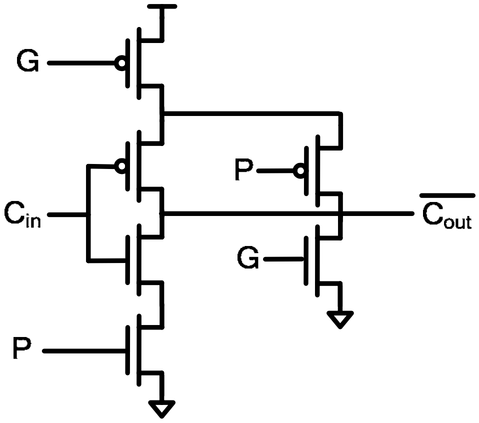
$$\boxed{T_{cyc} = 1.1ns}$$

Know that path through CL_1 takes 1.4ns, so we'll set $\boxed{t_{skew_1} = 300ps}$ to give CL_1 the extra time it needs.

Now, know that CL_2 takes $800ps + t_{clk-q} + t_{setup} = 1ns$. We started with a 1.1ns cycle, but took away 300ps because of t_{skew_1}. So, we need to add back just enough time for CL_2 to evaluate, and thus $\boxed{t_{skew_2} = 200ps}$.

PROBLEM 5: Arithmetic Blocks (14 points)

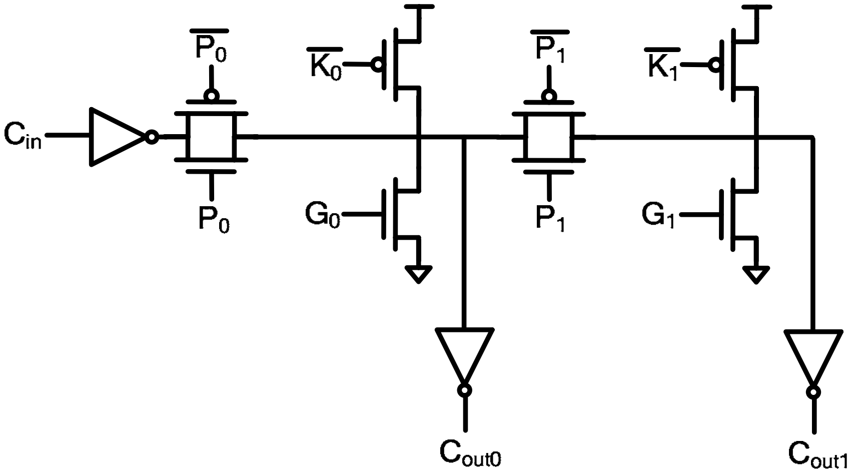
a) **(3 pts)** Shown below is a static CMOS implementation of a gate that computes the carry-out at a particular bit position. If the "P" signal that was fed into this gate was calculated using A + B (instead of A ⊕ B), would the output of this gate still be correct? Why or why not?



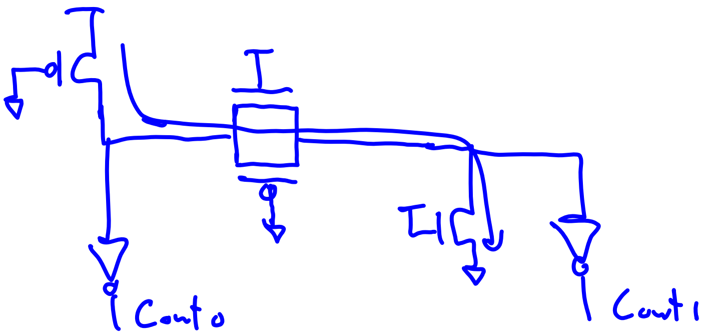
The only difference between A+B and $A \oplus B$ is when $A=B=1$. In this case, the OR version sets $P=1$, while the XOR version sets $P=0$.

This gate implements $\overline{C_{out}} = \overline{G + P \cdot C_{in}}$, and when $A=B=1$, $G=1$. So, in this case $C_{out}=1$ no matter what P is, and thus the output of the gate will be correct with either version of P.

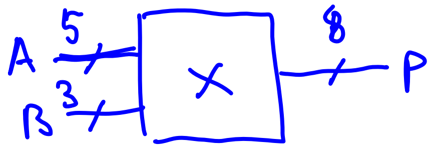
b) (5 pts) Now let's look at a the gate shown below that computes the carry-out for two bit positions, but this time implemented with a Manchester carry chain. Is this gate guaranteed to function correctly if the P signals that are fed to this gate are calculated as A + B? Why or why not?



Because we have used bidirectional pass-gates, this time the gate will not function correctly for either version of P. Consider for example $A_1 = B_1 = 1$ and $A_0 = B_0 = 0$. If the OR version of P is used, we will have $P_0 = 0$, $K_0 = 1$, $P_1 = 1$, $G_1 = 1$, leading a drive fight between the K_0 device and the G_1 device.

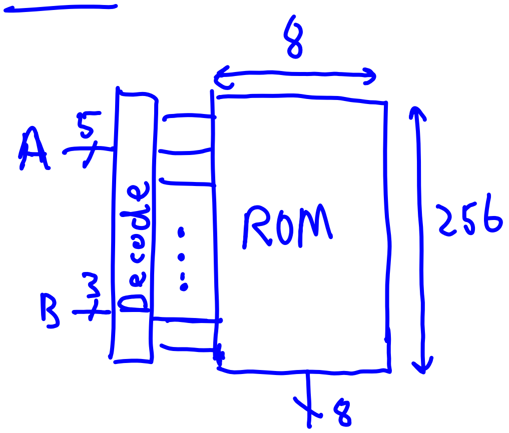


c) (6 pts) One of the designers at your company (who didn't take EE141) comes to you one day and says that he never understood the logic for multipliers, and proposes to build a multiplier out of a ROM instead of out of full-add and half-add cells. How large of a ROM would you need to implement a 5-bit by 3-bit multiplier? (You can quote your answer in terms of the number of rows and number of columns this ROM would need to contain.) Do you think this a good idea? Why or why not?



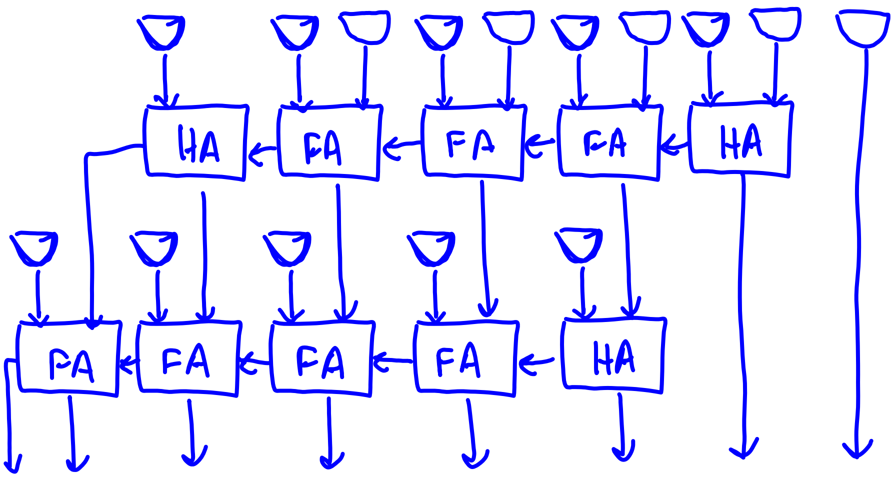
There are $2^{(5+3)}$ possible inputs to the multiplier, and the output is also (5+3) bits wide.

ROM:



So, we would need a 256×8 ROM to implement the multiplier. Therefore, this really isn't a good idea, since implementing the multiplier the "normal" way would only require 15 AND gates and 10 adder cells. An adder-based design could thus achieve much lower area, power, and delay.

"Normal":


HA
FA
FA
FA
HA
FA
FA
FA
FA
HA