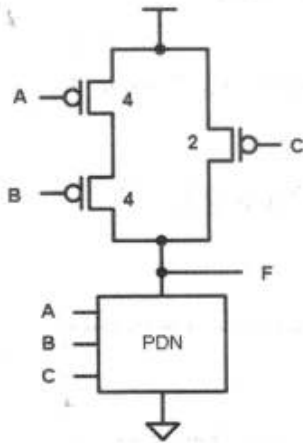


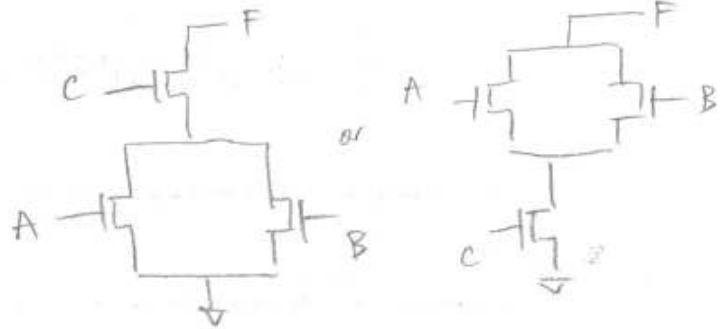
Problem 1: Combinational Logic Gates (13 pts)

Given the PMOS pull-up network below:

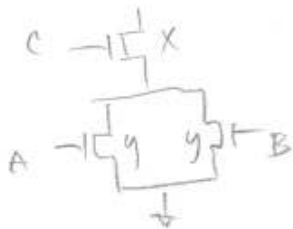


- a. Draw the NMOS pull-down network to complete the static CMOS gate. (2pt)

PDN:



- b. Size the pull-down transistors such that its worst-case drive strength is the same as a unit inverter (PMOS to NMOS ratio of 2/1) and that **all the input capacitances are the same**. (4 pt).



(0.5 pt)

$$\frac{1}{x} + \frac{1}{y} = 1 \quad (\text{pull-down strength same as inverter}) \quad (1 \text{ pt})$$

$$x + 2 = y + 4 \quad (\text{same input caps}) \quad (1 \text{ pt})$$

$$\Rightarrow x = y + 2$$

$$\therefore \frac{1}{y+2} = \frac{1}{y} = 1 \Rightarrow \frac{1}{y+2} = 1 - \frac{1}{y} = \frac{y-1}{y}$$

$$\Rightarrow y+2 = \frac{y}{y-1} \Rightarrow (y+2)(y-1) = y = y^2 + y - 2 = y$$

$$\Rightarrow y^2 = 2 \Rightarrow y = \sqrt{2}$$

$$x = \sqrt{2} + 2$$

Pull-down network:

$$M_A = \sqrt{2} = 1.4 \quad (0.5 \text{ pt})$$

$$M_B = \sqrt{2} = 1.4 \quad (0.5 \text{ pt})$$

$$M_C = \sqrt{2} + 2 = 3.4 \quad (0.5 \text{ pt})$$

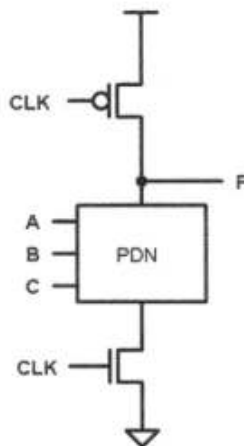
c. What is the logical effort of this gate for each input? (1.5 pt)

$$LE_A = \frac{\sqrt{2} + 4}{3} = \frac{C_A}{C_{inv}}$$

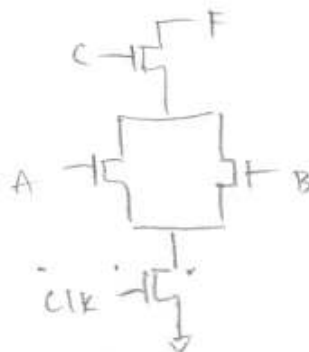
$$LE_B = LE_A = LE_C \quad \text{since } C_A = C_B = C_C$$

$$\begin{aligned} LE_A &= \frac{\sqrt{2} + 4}{3} = 1.8 & (0.5 \text{ pts}) \\ LE_B &= 1.8 & (0.5 \text{ pts}) \\ LE_C &= 1.8 & (0.5 \text{ pts}) \end{aligned}$$

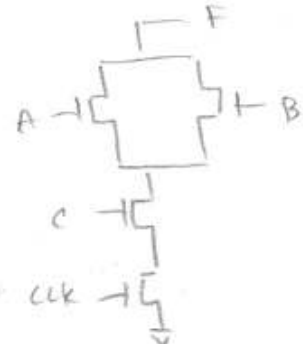
Assume now that we want to implement the same function in a dynamic CMOS gate as shown below.



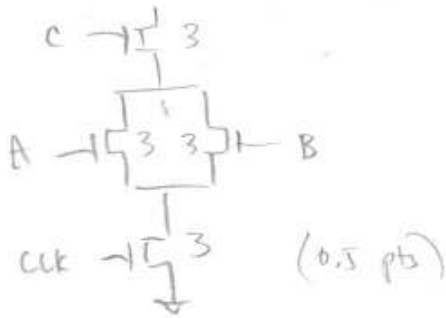
d. Draw the NMOS pull-down network such that it has the same logic function as the static CMOS gate in Part 1. (2 pt)



or



- e. Size the transistors such that the worst-case drive strength for all inputs is the same as a unit 2/1 inverter. (2pt)



Pull-down network:

$$\begin{aligned} M_A &= 3 & (0.5 \text{ pts}) \\ M_B &= 3 & (0.5 \text{ pts}) \\ M_C &= 3 & (0.5 \text{ pts}) \end{aligned}$$

- f. What is the logical effort of this gate? (1.5pt)

$$LE_A = LE_B = LE_C \text{ since } C_A = C_B = C_C$$

$$LE_A = \frac{3}{3} = \frac{C_A}{C_{inv}} = 1$$

$$\begin{aligned} LE_A &= 1 & (0.5 \text{ pts}) \\ LE_B &= 1 & (0.5 \text{ pts}) \\ LE_C &= 1 & (0.5 \text{ pts}) \end{aligned}$$

Problem 2: Optimizing Complex Logic Networks for Speed (17 pts)

Throughout this problem you can assume the following: $\gamma = 1$ and the gate capacitance of a minimum sized NMOS = 1fF. X , Y , Z stands for the input capacitance of that particular gate and *min* means that this gate is sized such that it has equal driving strength to a minimum sized inverter with a NMOS to PMOS ratio of 1/2 (as assumed in class all the time)

One day your boss shows up on your office and shows you the logic network shown in Fig. 1. It turns out that it is the most critical part in the new super-processor your company is working on and he asks you to analyze the network and optimize it for speed.

The two critical paths are the ones from In1 to Out and from In2 to Out.

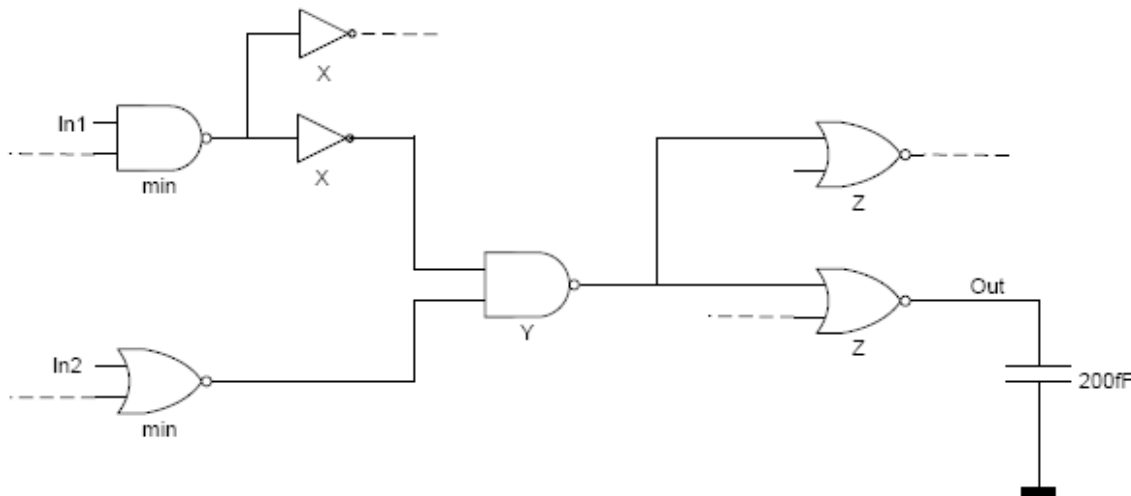


Figure 1: Logic network in need for speed

- a) First you start analyzing the path from In1 to Out. In order to do so you calculate the path electrical fan-out F , the path logical effort LE , the path branching effort B , the total path effort PE as well as the optimal stage effort SE (You are assuming that you can optimize this path independently from the path In2 to Out). Finally, you estimate the total intrinsic delays along the path (Σp_i) and the minimum delay $t_{pin1-out,min}$ from In1 to Out in terms of t_{inv} . (5 pt)

$$C_{in1} = 4 \text{ fF}$$

$$(1 \text{ pt}) F = C_{out}/C_{in1} = 200 / 4$$

$$(0.5 \text{ pt}) LE = 4/3 \times 1 \times 4/3 \times 5/3$$

$$(0.5 \text{ pt}) B = 2 \times 2$$

$$(0.5 \text{ pt}) PE = F \times LE \times B \\ = 200 / 4 \times 80/27 \times 4 = 592.6$$

$$(1 \text{ pt}) SE = PE^{1/4} = 4.93$$

$$(0.5 \text{ pt}) \Sigma p_i = 2 + 1 + 2 + 2 = 7$$

$$(1 \text{ pt}) t_{pin1-out,min} = t_{inv} (\Sigma p_i + 4 \times SE) = 26.72$$

- b) You repeat the same for the path from In2 to Out. Again, you assume you can independently optimize that path for speed. (5 pt)

$$C_{in2} = 5 \text{ fF}$$

$$(1 \text{ pt}) F = C_{out}/C_{in2} = 200 / 5$$

$$(0.5 \text{ pt}) LE = 5/3 \times 4/3 \times 5/3$$

$$(0.5 \text{ pt}) B = 2$$

$$(0.5 \text{ pt}) PE = F \times LE \times B \\ = 200/5 \times 100/27 \times 2 = 296.3$$

$$(1 \text{ pt}) SE = PE^{1/3} = 6.67$$

$$(0.5 \text{ pt}) \Sigma p_i = 2 + 2 + 2 = 6$$

$$(1 \text{ pt}) t_{pin2-out,min} = t_{inv} (\Sigma p_i + 3 \times SE) = 26$$

- c) Based on this information, you decide to optimize the delay of the critical (that is, the slowest) path by carefully sizing the transistors. (If the path between In2 and Out is the slower one, optimize Y and Z first and then optimize X to minimize the delay from In1 to Out without changing Z and Y). After you are done you also write down the actual delay of the path you just optimized. (4 pt)

Since In1 → Out has the higher minimum delay this path has to be optimized.

One way of solving this problem is to start from the output and move towards the input.

$$Z = C_{out}/SE \times g \times b = 200/4.93 \times 5/3 \times 1 = 67.61 \text{ fF}$$

$$Y = Z/SE \times g \times b = 67.61/4.93 \times 4/3 \times 2 = 36.57 \text{ fF}$$

$$X = Z/SE \times g \times b = 36.57/4.93 \times 1 \times 1 = 7.42 \text{ fF}$$

Just to check:

$$C_{in1} = X/SE \times g \times b = 7.42/4.93 \times 4/3 \times 2 = 4 \text{ fF}$$

$$(1 \text{ pt}) X = 7.42 \text{ fF}$$

$$(1 \text{ pt}) Y = 36.57 \text{ fF}$$

$$(1 \text{ pt}) Z = 67.61 \text{ fF}$$

$$(1 \text{ pt}) t_{pinx-out} = 26.72 t_{inv}$$

$$t_{pin1-out} = t_{inv} (\Sigma p_i + 4 \times SE) = 26.72 t_{inv}$$

- d) Based on the obtained transistor sizes, recalculate the delay of the second (faster) path. Is it still lower than the other one? If not, explain how this is possible. (3 pt)

(1 pt for delay calculation, 2 pts for reasoning)

$$\begin{aligned}
 t_{\text{pin2-out}} &= t_{\text{p1stnor}} + t_{\text{prest}} \\
 &= t_{\text{inv}} (2 + f_{\text{1stnor}} \times g_{\text{nor}}) + t_{\text{inv}} (p_{\text{nans}} + p_{\text{nor}} + 2 \times SE_a) \\
 &= t_{\text{inv}} (2 + Y/C_{\text{in2}} \times 5/3) + t_{\text{inv}} (2 + 2 + 2 \times SE_a) \\
 &= t_{\text{inv}} (6 + 7.314 \times 5/3 + 2 \times 4.93) \\
 &= 28.05 t_{\text{inv}}
 \end{aligned}$$

The delay from In2 to Out is now not only longer than its minimum $t_{\text{pin2-out,min}}$, but also longer than the optimized delay from In1 to Out. The reason for that is that in the given circuit it is not possible to optimize the two paths separately to achieve the ideal minimum delay for both at the same time, since they have gates in common. The optimum solution for In1 to Out is not the optimum solution for In2 to Out at the same time for this example.

(Note: In reality one would have to try to jointly minimize the delay of both paths to achieve the minimum overall worst case delay. In that case it turns out that both delays would be the same. Anyway, this is a much harder problem and beyond the scope of this exam.)

$$t_{\text{pin2-out}} = 28.05 t_{\text{inv}}$$

PROBLEM 3: Dynamic Logic (12 pts)

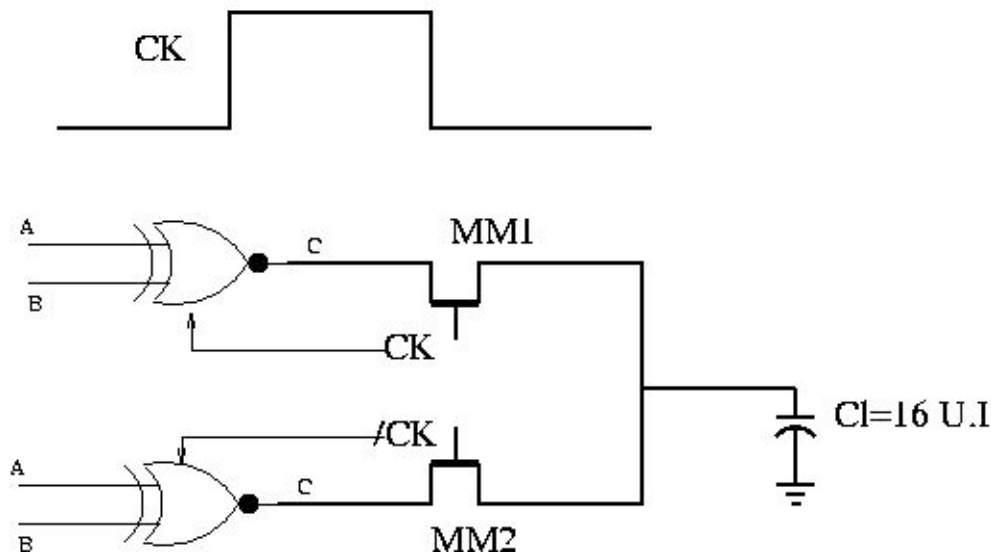
- a. We would like to implement the functionality $y = \sim(A \text{ XOR } B)$ (that is, XNOR) using **np-CMOS dynamic logic** so that the maximum performance is obtained. The A and B inputs are available from unity-strength (minimum size) buffers. The output C is loaded with a capacitance equal to 16 minimum-size inverters. For all gates in your design, you may assume that the parasitic delay $p = 0$. You may also assume that $\gamma = 1$, and a PMOS/NMOS driving strength ratio of 2.
- Draw the circuit topology that would minimize delay, and annotate the relevant transistors sizes. (8 points). Explain your choices.

Unfortunately, there is no good solution to this problem. The NEXOR function requires a combination of both the non-inverted and the inverted versions of the input signals. This is impossible to accomplish in a pure version of np-CMOS where all the inputs either have to show only a 1->0 or 1->1 transition during evaluation for an n-block, and a 0->1 or 0->0 transition for a p-block. Any derived signal by necessity will reverse the transition and cannot be combined with signals from the previous stage.

The only way to accomplish this is to add a final CMOS combining stage – but this will be suboptimal from a performance perspective.

Most answers provided some correct functionality, but violated the composition rules of dynamic CMOS

b. A clever engineer has figured out a way to further enhance the performance of this circuit. He proposes a ping-pong approach, as shown below. The logic is replicated in two paths, activated on opposite clock phases. Transistors MM1 and MM2 are used to connect the load C_L to the top gate during its evaluation period ($CLK = 1$) and to the bottom gate during its evaluation period ($CLK = 0$).



- Explain why this may be an interesting idea and why it may help to increase performance (2 pts)

This "ping-pong" approach has the advantage of hiding the pre-charge cycle. Both circuits interchange pre-charge and evaluation phases. The circuit can hence be substantially faster.

- Also identify a potential (major) problem with the proposed circuit. Can you propose a solution for it? (2 pts)

Hint: think about the potential pitfalls of dynamic circuits.

The problem is charge redistribution. If the output evaluates to "0", connecting C_L to the precharged output of the other gate when the clock switches (for the next evaluation) will result in a major drop in voltage (C_L is actually quite large). Having a substantial level restoring device could work, but it complemented with a buffer isolating the load capacitance from the dynamic nodes.