6.004 Tutorial Problems
L07 – CMOS Logic

Note: A subset of essential problems are marked with a red star (★). We especially encourage you to try these out before recitation.

Problem 1. ★

For each of the functions F and G, if the function can be implemented using a **single CMOS gate**, please draw the corresponding single CMOS gate. If it cannot be implemented using a single CMOS gate, then write NONE. **For full credit, use a minimum number of FETs.**

A	B	C	F	G
0	0	0	1	1
0	0	1	1	1
0	1	0	0	1
0	1	1	1	0
1	0	0	1	1
1	0	1	0	0
1	1	0	0	1
1	1	1	1	0

Draw CMOS implementation of F(A,B,C) below or write NONE if F cannot be implemented as single CMOS gate.	**Draw CMOS implementation of G(A,B,C) below or write NONE if G cannot be implemented as single CMOS gate.**

Problem 2. ★

(A) A single CMOS gate, consisting of an output node connected to a single pFET-based pullup circuit and a single nFET-based pulldown circuit (as described in lecture), computes F(A, B, C, D). It is observed that F(1, 0, 1, 0) = 1. What can you say about the following values?

(circle one) F(0, 0, 1, 0) = : 0 … 1 … (can't say)

(circle one) F(1, 1, 1, 0) = : 0 … 1 … (can't say)

(circle one) F(1, 1, 1, 1) = : 0 … 1 … (can't say)

(B) The Boolean function F(A,B,C) can be implemented using a *single* CMOS gate operating as a combinational device that obeys the static discipline. It's known that F(1,1,0) = 1 and F(0,1,1) = 0. What can be determined about the value of F in the following cases? Please circle one of "0", "1" or "Can't tell".

(circle one) F(1,0,0) = 0 … 1 … Can't tell

(circle one) F(1,0,1) = 0 … 1 … Can't tell

(circle one) F(1,1,1) = 0 … 1 … Can't tell

(C) A single CMOS gate, consisting of an output node connected to a single pullup circuit containing one or more PFETs and a single pulldown circuit containing one or more NFETs (as described in lecture), computes $F(A,B)$. *F* has the property that for all A, $F(A,0) = \overline{F(A,1)}$. What can you say about the value of $F(1,0)$?

(circle one) F(1,0) = 1 … 0 … can't tell

Problem 3.

A minority gate has three inputs (call them A, B, C) and one output (call it Y). The output will be 0 if two or more of the inputs are 1, and 1 if two or more of the inputs are 0.

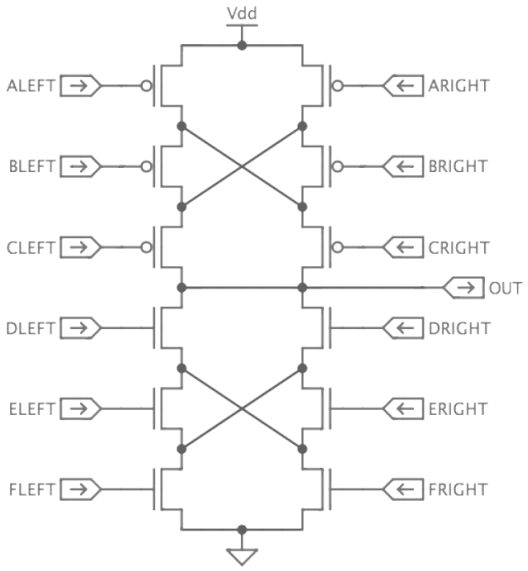
In the space below, draw the *pulldown* circuit for a single CMOS gate that implements the minority function, using the minimum number of NFETs. You needn't draw the *pullup* circuit.

If you're convinced that the function cannot be implemented as a single CMOS gate, give a brief, convincing explanation.

Can it be implemented as single CMOS gate? Circle one: YES can't tell NO

Problem 4.

In his bid for the Lemelson Prize, Ben Bitdiddle has invented the "flexible gate," a single CMOS gate that implements different functions depending how its inputs are wired up. The FlexGate® (see figure at right) uses 6 pFETs in its pullup circuit and 6 nFETs in its pulldown circuit.



Each of the FlexGate's twelve inputs can connected to an input signal (X, Y, …), GND (logical "0"), or VDD (logical "1"). To show off its versatility, Ben has asked you to show how to connect the inputs so the FlexGate computes several different functions whose Boolean equations are given below. Associated with each equation is a table with 12 entries; in each cell of the table please write an input name, GND, or VDD as appropriate. Note that there may be several possible implementations for each of the three functions – any correct answer will be acceptable. (Note: there should be an entry in each cell, i.e., a connection should be specified for each input!)

If the desired function cannot be implemented, please draw a big "X" through the table.

Fill in the tables below or mark with "X"

$$OUT = \overline{X \times Y}$$

input	LEFT	RIGHT
A		
B		
C		
D		
E		
F		

$$OUT = \overline{X + Y + Z}$$

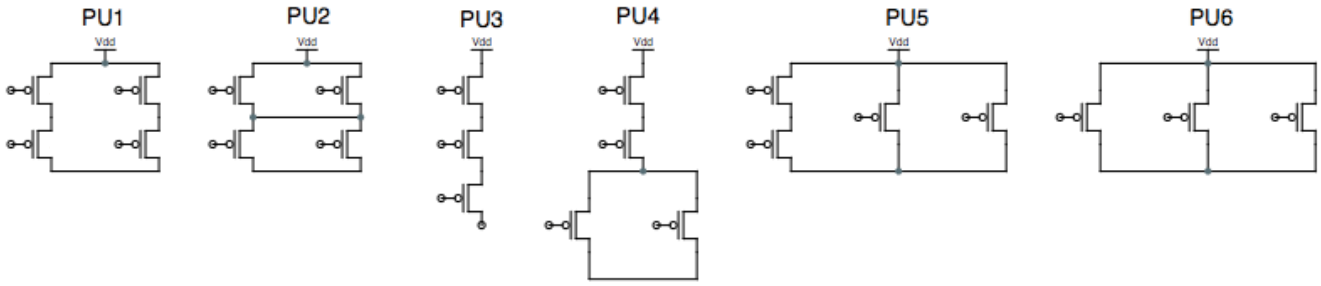
input	LEFT	RIGHT
A		
B		
C		
D		
E		
F		

$$OUT = \overline{X + Y \times Z}$$

input	LEFT	RIGHT
A		
B		
C		
D		
E		
F		

Problem 5. ★

You are trying to select pullups for several 3- and 4-input CMOS gate designs. You can choose between seven different pullups, given names PU1 through PU6, diagrammed below:



You can connect one of the inputs (A, B, C, or D) or a constant (GND, or VDD) to each pFET, allowing these pullups to be used in various ways to build gates with various numbers of inputs. Your goal, however, is to select the pullup that uses the minimum number of active transistors for each of the gates you are designing (i.e., a pFET that is never on is *not active*).

For each of the following 3- and 4-input Boolean functions, choose the appropriate pullup design, i.e., the one which, properly connected, implements that gate's pullup using the **minimum number of active** transistors. This may require applying De Morgan's Law or minimizing the logic equation first. If none of the above pullups meets this goal, write "NONE".

(A) $F(A,B,C,D) = \overline{AB + CD}$ **Choice or NONE:** _____

(B) $F(A,B,C,D) = \overline{(A+D)(B+C)}$ **Choice or NONE:** _____

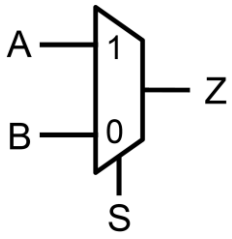
(C) $F(A,B,C) = A \cdot \overline{BC}$ **Choice or NONE:** _____

(D) $F(A,B,C,D) = \overline{A} + \overline{B} + \overline{C} \cdot \overline{D}$ **Choice or NONE:** _____

(E) $F(A,B,C,D) = \overline{(AB + C)D + (AB + C)\overline{D}}$ **Choice or NONE:** _____

Problem 6. ★

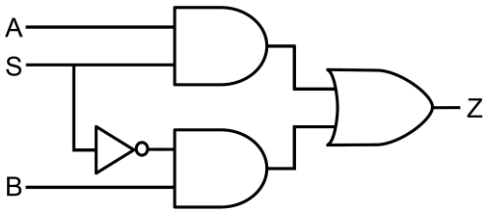
Muxes are used often so it is important to optimize them. In this problem you will design several variants of a 1-bit, 2-to-1 mux (shown to the right) using CMOS gates, and will compare their costs in number of transistors.



$$\mathbf{Z = A \cdot S + B \cdot \overline{S}}$$

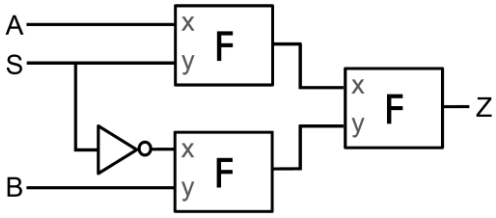
Note: Remember that a CMOS gate consists of an output node connected to a *single* pFET-based pullup circuit and a *single* nFET-based pulldown circuit. Gates obtained by combining multiple CMOS gates are not a CMOS gate.

(A) Consider the implementation shown below, which uses two AND gates and an OR gate. Because a single CMOS gate cannot implement AND or OR, each AND gate is implemented with a CMOS NAND gate followed by a CMOS inverter, and the OR gate is implemented with a CMOS NOR gate followed by a CMOS inverter. How many transistors does this implementation have?



Number of transistors in mux: _____

(B) Consider the implementation shown below, which uses three instances of gate F. Find the Boolean expression for F. If F can be built using a single CMOS gate, draw its CMOS implementation. Otherwise, give a convincing explanation for why F cannot be implemented as a CMOS gate. How many transistors does this implementation have?

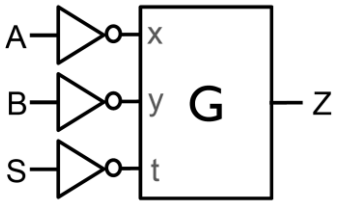


F(x,y) = _____

Draw CMOS gate that implements F or explain why it cannot be built.

Number of transistors in mux (if F can be built as a CMOS gate): _____

(C) Consider the implementation shown below, which uses gate G. Find the Boolean expression for G. If G can be built using a single CMOS gate, draw its CMOS implementation. Otherwise, give a convincing explanation for why G cannot be implemented as a CMOS gate. How many transistors does this implementation have?

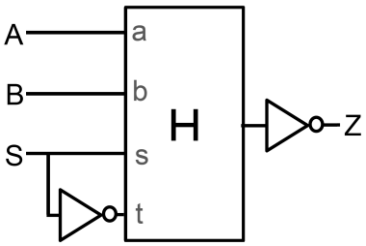


G(x,y,t) = _____

Draw CMOS gate that implements G or explain why it cannot be built.

Number of transistors in mux (if G can be built as a CMOS gate): _____

(D) Consider the implementation shown below, which uses gate H. Find the Boolean expression for H. If H can be built using a single CMOS gate, draw its CMOS implementation. Otherwise, give a convincing explanation for why H cannot be implemented as a CMOS gate. How many transistors does this implementation have?



H(a,b,s,t) = _____

Draw CMOS gate that implements H or explain why it cannot be built.

Number of transistors in mux (if H can be built as a CMOS gate): _____