BIRZEIT UNIVERSITY

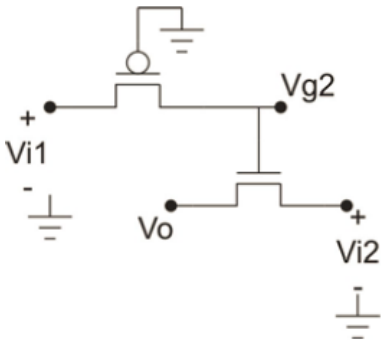
Electrical and Computer Systems Engineering Department
ENCS333
Homework 4

Due November 5th

Problem 1

Using the rules given in lecture notes, find **Vg2** and **Vo** required for both transistors to be ON in the two-transistor circuit shown below for each of the listed input voltage combinations. Assume **VDD = 2.5V**, **Vtn = 0.5V**, and **|Vtp| = 0.5V**.

(a) Vi1 = 0V, Vi2 = 0V
(b) Vi1 = 2V, Vi2 = 2V
(c) Vi1 = 2.5V, Vi2 = 2.5V
(d) Vi1 = 0V, Vi2 = 1V



solution

(a) Vi1 = 0V, Vi2 = 0V

Vi1-Vg1 = 0-0=0 < |Vtp|, so **Vg2** = Vg1+|Vtp| = 0+0.5 = **0.5V**

Vg2-Vi2 = 0.5-0 = 0.5V = Vtn, so **Vo** = Vi2 = **0V**

(b) Vi1 = 2V, Vi2 = 2V

Vi1-Vg1 = 2-0 =2 > |Vtp|, so **Vg2** = Vi1 = **2V**

Vg2-Vi2 = 2-2=0 < Vtn, so **Vo** = Vg2-Vtn = **2-0.5 = 1.5V**

(c) Vi1 = 2.5V, Vi2 = 2.5V

Vi1-Vg1 = 2.5-0 = 2.5 > |Vtp|, so **Vg2** = Vi1 = **2.5V**

Vg2-Vi2 = 2.5-2.5 = 0V < Vtn, so **Vo** = Vg2-Vtn = 2.5-0.5 = **2V**

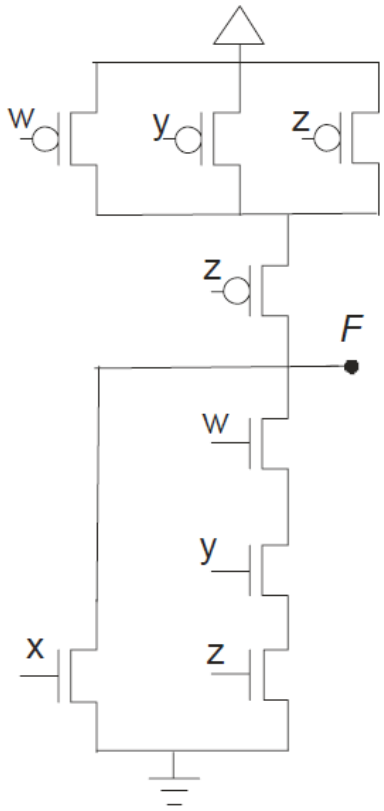
(d) Vi1 = 0V, Vi2 = 1V

Vi1-Vg1 = 0-0 = 0 < |Vtp|, so **Vg2** = Vg1+|Vtp| = 0+0.5 = **0.5V**

Vg2-Vi2 = 0.5-1 = -0.50V < Vtn, so **Vo** = Vg2-Vtn = 0.5-0.5= **0V**

Design a transistor-level CMOS logic circuit to implement the function $F = \overline{(x + yz) \cdot (w + x)}$ using the least number of transistors.

HINT: Consider that you may need to expand the equation in order to reduce it using the logic properties shown in the Chapter 2 lecture notes.



solution

This function can be easily reduced using the property proven in HW1, which says (a+b)(a+c) = a+bc. Here, x=a, yz=b, and w=c. A brief derivation shows

$$F = \overline{(x + yz) \cdot (x + w)} = \overline{xx + xyz + xw + wyz} = \overline{x + xyz + xw + wyz}$$

$$= \overline{x(1 + yz + w) + wyz} = \overline{x + wyz}$$

The function has been reduced to 4 transistors and 3 operations. The schematic to implement $F = \overline{x + wyz}$ is shown on the right.

Construct the CMOS logic circuit that implements $Y = a + \overline{\bar{a} + b} + c\bar{b}$ using the fewest possible transistors. You are allowed to use inverted inputs (e.g., *a'*) rather than adding inverters to create these signals.

solution

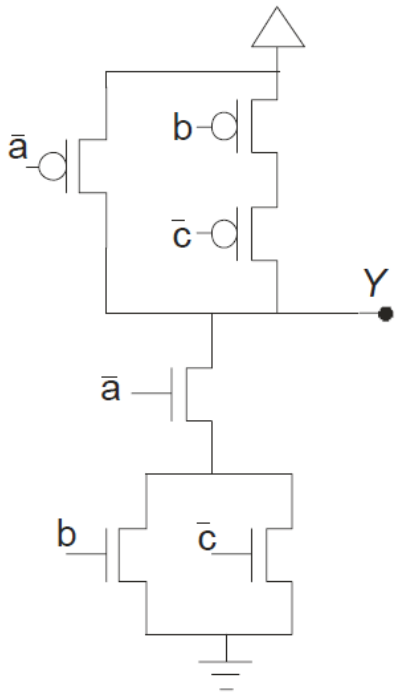
First let's simply the function to remove all inverted operations

$Y = a + (a\bar{b}) + c\bar{b}$

$Y = a + c\bar{b}$ (because A+AX = A)

Thus

$Y_p = \bar{a} + \bar{c}b$, and $Y_n = \bar{a} \cdot \overline{c\bar{b}} = \bar{a} \cdot (\bar{c} + b)$, which looks correct because all AND/OR operations are complemented between Yp and Yn, and both have the same inputs. Substituting parallel connections for OR operations and series connections for ANDs, we get:



Problem 4

The symbol for a 2:1 multiplexer and the truth table for a 4:1 multiplexer are shown below.

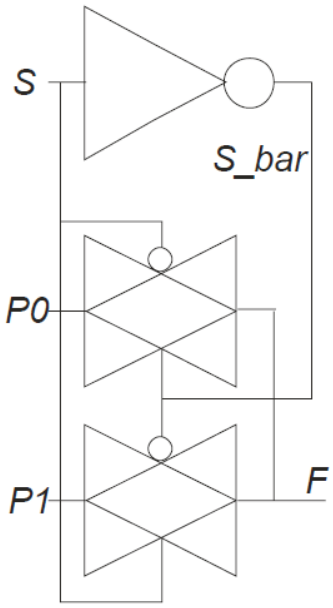
a) Construct a schematic for a 2:1 MUX using two CMOS transmission gates and an inverter.

b) Using the 2:1 MUX symbol below, construct a gate-level schematic for a 4:1 multiplexer using only 2:1 multiplexers. The inputs should be s0, s1, and P3:0 (P0 – P3) and the output is F.

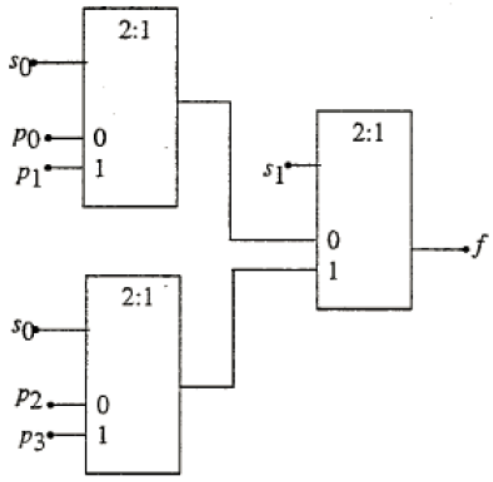
c) How many transistors are needed to form the transmission gate based 4:1 MUX in (b)?

d) Are there any redundant transistors that could be eliminated if you constructed the 4:1 MUX at the transistor level? If so, how many and which ones?

solution



(a)



(b)

c) 2:1 MUX has 6 transistors (2x2 (transmission gates) + 2 (inverter)). Thus the 4:1 MUX has 3x6 = **18 transistors**

d) Yes, the 2:1 MUX with an included INV creates a redundant inversion of s0. One INV could be eliminated from the first-stage 2:1 MUXs.

This would save **2 transistors**.

A CMOS metal layer with resistivity, ρ, $3x10^{-6}$ Ω-cm is 0.6μm thick. It is used to draw a signal trace that is 100μm long and 0.75μm wide.

a) Calculate the sheet resistance, Rs, of this metal layer.

b) How many "squares", n, are in the signal trace?

c) Use the results in (a) and (b) to determine the resistance of the trace.

solution

a) $Rs = \rho/t = 3x10^{-6}/(0.6x10^{-4}) = 0.05\ \Omega$

b) $n = L/W = 100/0.75 = 133.3$ squares

c) $R = (Rs)(n) = 0.05(133.3) = 6.67\ \Omega$

Draw the schematic for the CMOS circuit that implements the function ***F*** described by the truth table below. Use the least possible number of transistors. Explain your procedure and show the reduced function equation used to design the schematic.

Inputs			Output
X	Y	Z	F
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Draw the schematic for the CMOS circuit that implements the function ***F*** described by the truth table below. Use the least possible number of transistors. Explain your procedure and show the reduced function equation used to design the schematic.

Inputs			Output
X	Y	Z	F
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

solution (corrected)

There are two basic approaches: we can construct a K-map or write a sum of products expression and reduce it. Let's try the sum of products option.

We can write the sum of products for either the high terms (F) or the low terms (F'). Because there are fewer low/zero terms, let's try those.

F' = (X'YZ' + X'YZ + XYZ') = X'Y (Z'+Z) + XYZ'

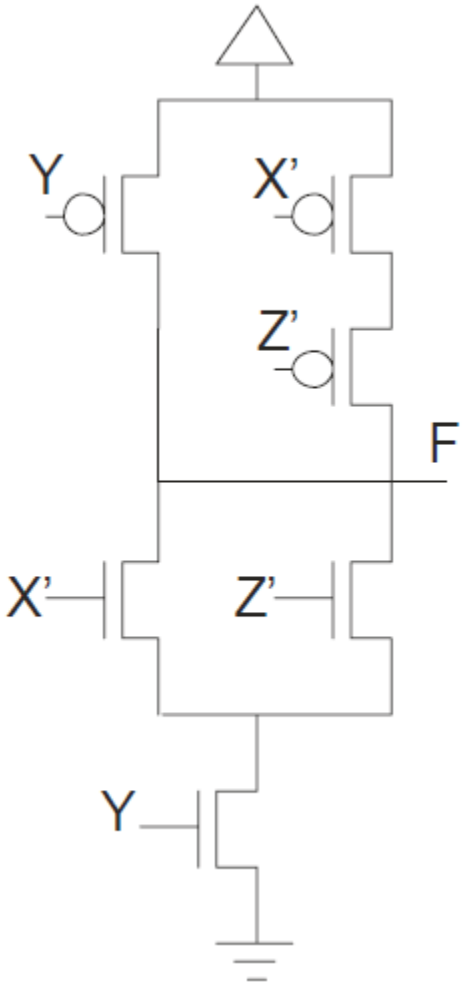
F' = X'Y + XYZ' = Y (X'+XZ') ⏲ **F' = Y (X'+Z')**

Thus **F = Y' + XZ**

Alternatively, by observation we can see that F = 1 when Y' OR XYZ is true. Thus, F = Y' + XYZ. Using the property covered in HW1 we can reduce this to

F = Y' +XZ

which matches our sum of products results. Note that we should get the same results if we started with the high terms sum of products



Sketch a color-coded stick diagram for the circuit that implements the function *fabcd*=+·+. Organize the layout so that the transistors can be implemented on a continuous strip of active (i.e., do not break the active).

solution:

An interesting feature of this function is that you have to organize the transistors within the schematic correctly in order to achieve the 'one continuous strip of

active' layout goals. The schematic below shows one possible implementation. Notice the b || c pMOS devices must be directly attached to either the ground or output nodes in order to draw a loop that does not violate the rules. Two possible loops (with X at the start and an arrow at the end) for this schematic implementation are shown (solid-orange line and dotted-blue line). The solid-orange loop is used to set the order of poly traces in the stick diagram below to a, d, b, c. However, there are many possible layouts for this function.