COMS12200 problem set #3

Within this problem slot, the idea is that you attempt to solve the set of pencil-and-paper, exam-style questions presented below; in doing so, you can (optionally) use an interactive system to anonymously register your solutions. More concretely, optionally start by installing the Socrative client, e.g.,

- for Chrome

  http://chrome.google.com/webstore/detail/socrative-student/nblhpecglllndfihipmpdoikimcmgkha

- for Android

  http://play.google.com/store/apps/details?id=com.socrative.student

- for iOS

  http://itunes.apple.com/gb/app/socrative-student/id477618130

or using the web-based application at

http://www.socrative.com,

then entering the 9-character "room name" which should be displayed top-center on the projector screen. Then, we will alternate as follows:

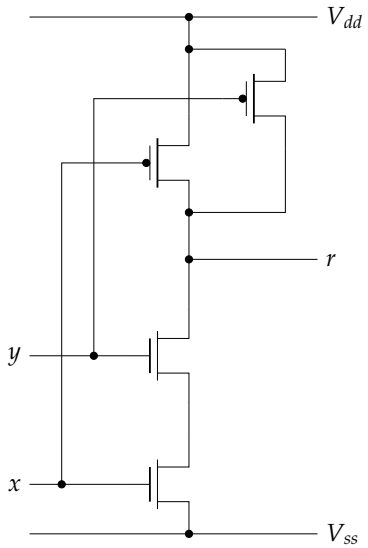
1. solve the current question, and optionally register your solution using Socrative,
2. wait until everyone is finished (or say ∼ 5 minutes elapse), at which point we will discuss the questions and solutions using any collated Socrative results as a starting point.

Q1. From the following list

`A:` has N-type semiconductor terminals and P-type body
`B:` has P-type semiconductor terminals and N-type body
`C:` is paired with another N-MOSFET to form a CMOS cell
`D:` has a threshold voltage above which the transistor is deemed active

identify **each** statement that correctly describes an N-MOSFET.

Q2. Consider the following implementation of a 2-input NAND gate:

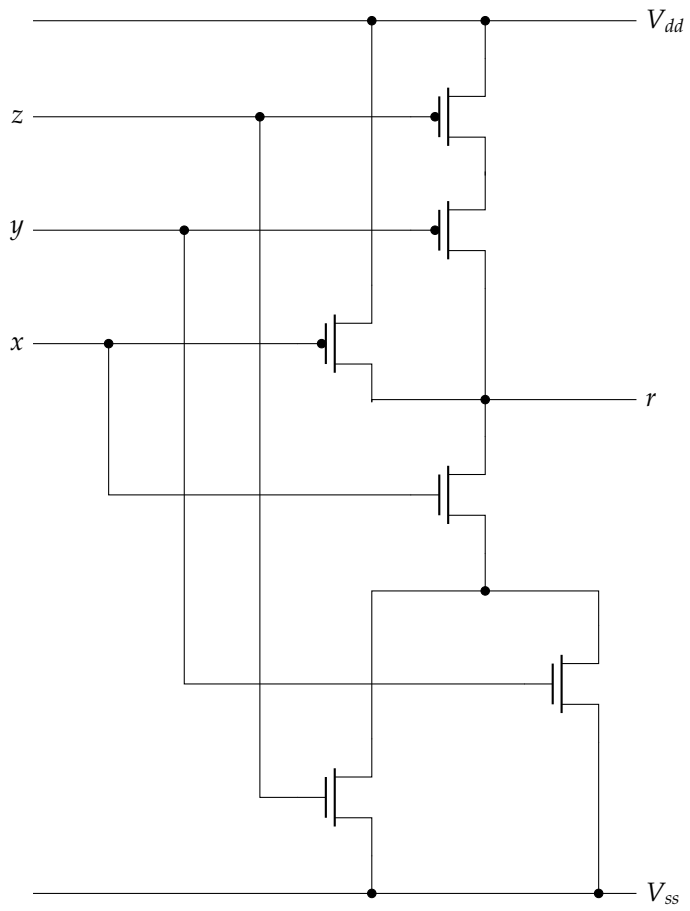


From the following list

`A:` two inputs x and y, and one output r
`B:` a pull-up network of P-MOSFET transistors
`C:` a pull-down network of BJT transistors
`D:` two power rails supplying different voltage levels
`E:` a flux capacitor

identify **each** component evident in the implementation?

Q3. Consider the following organisation of MOSFET transistors



which implements a 3-input Boolean function $r = f(x, y, z)$. *Which* function, from the following, do you think it matches?

`A:` $r = x \wedge y \wedge z$
`B:` $r = x$
`C:` $r = \neg(x \wedge (y \vee z))$
`D:` $r = x \wedge (y \vee z)$
`E:` $r = x \vee y \vee z$

Q4. Recall that a 2-input XOR operator can be described via the following truth table:

XOR		
x	y	r
0	0	0
0	1	1
1	0	1
1	1	0

An implementation of this operator is realised by combining logic gate instances, e.g., for NOT, NAND, AND, NOR, and OR, while attempting to minimise the total number of underlying MOSFET-based transistors. How many such transistors do you think it uses?

A: 14
B: 16
C: 18
D: 20
E: 22

Q5. A buffer can be described as a "pass through" logic gate: although it performs no computation (i.e., the output r matches the input x, so $r = x$), it does impose a delay (often roughly the same as a NOT gate). It may be termed a non-inverting buffer (cf. an *inverting* buffer, or NOT gate) because of this.

You are asked to implement a buffer, using an unconstrained organisation of N- and P-MOSFET transistors alone. Assuming you attempt to minimise the number used, how many transistors do you need?

A: 0
B: 2
C: 4
D: 6
E: 8

Q6. Consider a 16-bit register, constructed from CMOS-based D-type latches. Based on high-level reasoning about this component alone, if the initial value stored is $DEAD_{(16)}$ then overwriting it with which of the following

A: $BEEF_{(16)}$
B: $F00D_{(16)}$
C: $1234_{(16)}$
D: $FFFF_{(16)}$
E: $0000_{(16)}$

might you expect to consume more power?