COMS12200 problem set #4

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. Recalling that **?** denotes don't-care, the following truth table

f			
x	y	z	r
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	**?**
1	1	1	1

describes a 3-input, 1-output Boolean function f st. $r = f(x, y, z)$. Which of the following Boolean expressions

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

correctly realises f?

Q2. A m-output, 1-bit demultiplexer connects a 1-bit input x to one of m separate 1-bit outputs (say r_i for $0 \leq i < m$). The output is selected using an l-bit control signal c (or, equivalently, c is a collection of l separate 1-bit control signals). If $m = 5$, what value of l is required?

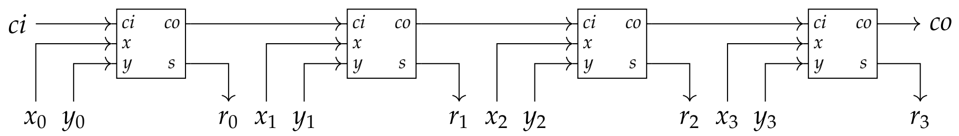
A: 0
B: 1
C: 2
D: 3

E: 4

Q3. Imagine you want to design an 8-input, 8-bit multiplexer. Rather than do so from scratch, you intend to form the design using multiple instances of an existing 2-input, 1-bit multiplexer component. How many do you need?

A: 1
B: 8
C: 24
D: 40
E: 56

Q4. The following diagram



illustrates a 4-bit ripple-carry adder circuit, constructed using 4 full-adder instances: it computes the sum $r = x + y + ci$, given two operands x and y and a carry-in ci, and an associated carry-out co. Given the propagation delay of NOT, AND, OR and XOR gates is 10ns, 20ns, 20ns and 60ns respectively, which of the following

A: 120ns
B: 180ns
C: 240ns
D: 280ns
E: 480ns

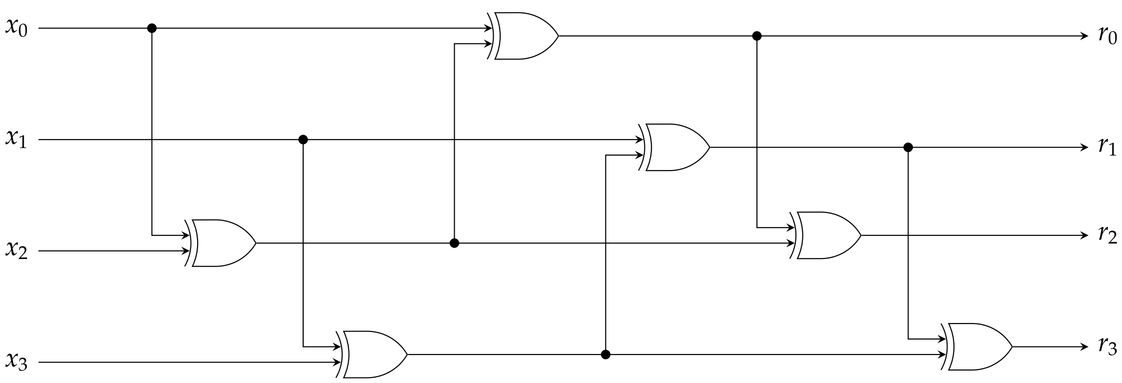
most accurately reflects the critical path of the entire circuit?

Q5. Imagine you use the ripple-carry adder in the previous question to compute an unsigned addition within some larger circuit. Having seen your design, your friend suggests they can optimise it: they claim that replacing each full-adder instance with a half-adder instance will halve the total number of logic gates required. However, they admit the optimisation does have a disadvantage. Specifically, although any value of x can be accommodated the optimised circuit can only produce the correct output for *some* values of y. Which of the following values of y

A: -1
B: 0
C: 1
D: any $2 \leq y < 8$
E: any $8 \leq y < 16$

will produce the correct output?

Q6. Consider the following combinatorial circuit



with a 4-bit input x and a 4-bit output r. Which of the following best describes the purpose of this circuit?

A: it computes the Hamming weight of x
B: it computes the parity of x
C: it swaps the most-significant 2-bit half of x with the least-significant 2-bit half of x
D: it adds the most-significant 2-bit half of x to the least-significant 2-bit half of x (treating it as an unsigned, 4-bit integer)
E: it negates x (treating it as a signed, 4-bit integer represented using two's-complement)