CSE 20, Fall 2020 - Homework 2

Due: Monday 10/19 at 11 am PDT

Instructions

Upload a single file to Gradescope for each group. All group members' names and PIDs should be on each page of the submission. Your assignments in this class will be evaluated not only on the correctness of your answers, but on your ability to present your ideas clearly and logically. You should always explain how you arrived at your conclusions, using mathematically sound reasoning. Whether you use formal proof techniques or write a more informal argument for why something is true, your answers should always be well-supported. Your goal should be to convince the reader that your results and methods are sound.

Reading Rosen Section 4.2 (p. 256) Section 1.2 (p. 20-21) Section 1.1 (p. 4-6) Section 1.3 (definition 1 and 2)

Key Concepts One's, Two's complement, logic gates and circuits, truth table, compound proposition

Problem 1 (20 points)

(a) Convert the following numbers to another representation in binary. Note that they are fixed width. Numbers are written in the following form:

$$[n]_{width,\ representation}$$

Example: $[0010]_{4,\ sign-magnitude}$ is a decimal 2 in sign-magnitude of width 4.

(i) 27 = [ ]$_{8,\text{ sign-magnitude}}$
(ii) -133 = [ ]$_{12,\text{ two's complement}}$
(iii) 74 = [ ]$_{8,\text{ two's complement}}$
(iv) -35 = [ ]$_{8,\text{ two's complement}}$

(b) What are the pros and cons of using signed magnitude and two's complement?
(i) pros of signed magnitude
(ii) cons of signed magnitude
(iii) pros of two's complement
(iv) cons of two's complement

Solution:

a) The
   i) 27 = [**0001 1011**]$_{8,\ \text{sign-magnitude}}$
   ii) -133 = [**1111 0111 1011**]$_{12,\ \text{two's complement}}$
   iii) 74 = [**0100 1010**]$_{8,\ \text{two's complement}}$
   iv) -35 = [**1101 1101**]$_{8,\ \text{two's complement}}$

b) pros of signed magnitude: ____**easy calculation**____
   cons of signed magnitude: ____**represents less number**________
   pros of two's complement: __**represents maximum numbers possible**_
   cons of two's complement: ______**harder calculations**_____

   As long as it's reasonable, points will be awarded.

Problem 2 (20 points)

RGBA, red green blue alpha, is a commonly used color space in computer science. Such representation can be written as a 4-tuple, (*r, g, b, a*), where *r* represents the red component, *g* the green component, *b* the blue component, and *a* the transparency component alpha. Each of the four values must be an element from the set $A = \{x \in \mathbb{N} \mid 0 \le x \le (255)_{10}\}$.

Definition: A RGBA color is a nonnegative integer, *n*, that has a base 16 fixed-width 8, and can be defined as

$$n = (r_1 r_2 g_1 g_2 b_1 b_2 a_1 a_2)_{16,8}$$

And A is defined above

(a) Why is each color component represented in 2 digits of hexadecimal number? Please give 1-2 sentences description.
(b) What is the green and alpha component of RGBA color $(2404399377)_{10}$? Hint: Convert the number to hexadecimal and use the definition above. For example, $(FF)_{16,2}$ is the red component in RGBA color $(FFFFFFFF)_{16,8}$
(c) If representing a RGBA color defined above (with hexadecimals) takes 8 bytes in the system, then how many bytes would representing a RGBA color take if the digits are in decimal? Assuming we want each component to be represented separately and each decimal digit takes 1 byte, for example $(0, 50, 70, 255)_{10}$ will be an example of RGBA color in decimal.
(d) What is $(FF0000FF)_{16,8}$ (opaque red) subtract $(EE82EE80)_{16,8}$ (50.2% transparent violet) in decimal? Write down your explanation and how you convert to decimal.

Solution:

a) Since $(FF)_{16}$ is 255, take 2 digits to represent the maximum number.
b) 0x8F503911, so $(50)_{16,2}$ is the green component and $(11)_{16,2}$ is the alpha component.
c) 3*4=12, each (r, g, b, a) component is < 255
d) 276632191. Trust our TA's and tutors, just reasonable process of converting to decimal is good

Problem 3 (20 points)

(a) Choose the appropriate answer and fill in the blanks. Each option might be used more than once or unused at all.

DNF (disjunctive normal form) are __**(B)**__ of __**(A)**__ (s)
CNF (conjunctive normal form) are __**(A)**__ of __**(B)**__ (s)

(A) AND (B) OR (C) NOT (D) unknowns

(b) Consider the following truth table

p	q	r	?
T	T	T	T
T	T	F	F
T	F	T	F
T	F	F	T
F	T	T	F
F	T	F	F
F	F	T	T
F	F	F	T

(i) Write down DNF describing the above table

$$(p \wedge q \wedge r) \vee (p \wedge \neg q \wedge \neg r) \vee (\neg p \wedge \neg q \wedge r) \vee (\neg p \wedge \neg q \wedge \neg r)$$

(ii) Write down CNF describing the above table

$$(\neg p \vee \neg q \vee r) \wedge (\neg p \vee q \vee \neg r) \wedge (p \vee \neg q \vee \neg r) \wedge (p \vee \neg q \vee r)$$

Problem 4 (20 points)

a) In class, we used the XOR gate to construct several logic circuits such as the full-adder circuit. Let us say you are asked to design an XOR gate, using only NOT and OR gates. Write a logically equivalent compound proposition, using only $\neg$ and $\vee$ to represent the XOR operation:

$$z = p \oplus q$$

Draw a truth table to verify your answer.

Solution:

a) The XOR operation for p and q can be written in DNF as:

$$p \oplus q \equiv (\neg p \wedge q) \vee (p \wedge \neg q)$$

The question asks to remove the AND operator $\wedge$. Recall DeMorgan's law:

$$\neg(p \wedge q) \equiv \neg p \vee \neg q$$

Applying double negation, this gives us: $\neg(\neg(p \wedge q)) \equiv p \wedge q \equiv \neg(\neg p \vee \neg q)$

Thus, we can write:

$$(\neg p \wedge q) \equiv \neg(\neg(\neg p) \vee \neg q) \equiv \neg(p \vee \neg q)$$
$$(p \wedge \neg q) \equiv \neg(\neg p \vee \neg(\neg q)) \equiv \neg(\neg p \vee q)$$

Substituting back, we get:

$$p \oplus q \equiv (\neg p \wedge q) \vee (p \wedge \neg q) \equiv (\neg(p \vee \neg q)) \vee (\neg(\neg p \vee q))$$

b) The 3-variable XOR function is defined as:

$$z = p \oplus q \oplus r$$

Simplify the proposition logic for *z* in either the CNF or DNF form by drawing a truth table.

Solution: To think of the truth table for the 3-variable XOR, write

$$z = p \oplus q \oplus r \equiv (p \oplus q) \oplus r$$

p	q	r	z
F	F	F	F
F	F	T	T
F	T	F	T
F	T	T	F
T	F	F	T
T	F	T	F
T	T	F	F
T	T	T	T

DNF form:

We represent all the rows for which $z = T$:

i) $(FFT) \equiv (\neg p \wedge \neg q \wedge r)$

ii) $(FTF) \equiv (\neg p \wedge q \wedge \neg r)$

iii) $(TFF) \equiv (p \wedge \neg q \wedge \neg r)$

iv) $(TTT) \equiv (p \wedge q \wedge r)$

Therefore, the final CNF form is:

$$z = p \oplus q \oplus r \equiv (\neg p \wedge \neg q \wedge r) \vee (\neg p \wedge q \wedge \neg r) \vee (p \wedge \neg q \wedge \neg r) \vee (p \wedge q \wedge r)$$

In CNF form, we look at the outputs where we avoid F:

i) $(FFF) \equiv \neg(\neg p \wedge \neg q \wedge \neg r) \equiv p \vee q \vee r$

ii) $(FTT) \equiv \neg(\neg p \wedge q \wedge r) \equiv p \vee \neg q \vee \neg r$

iii) $(TFT) \equiv \neg(p \wedge \neg q \wedge r) \equiv \neg p \vee q \vee \neg r$

iv) $(TTF) \equiv \neg(p \wedge q \wedge \neg r) \equiv \neg p \vee \neg q \vee r$

Therefore, the final CNF form is:

$$z = p \oplus q \oplus r \equiv (p \vee q \vee r) \wedge (p \vee \neg q \vee \neg r) \wedge (\neg p \vee q \vee \neg r) \wedge (\neg p \vee \neg q \vee r)$$

Note: Here, since we had 4 Fs and 4 Ts, both forms were equally easy (or hard!) to write.

Problem 5 (20 points)

For this question, we will consider the logic circuit for multiplying two two-bit binary numbers (we will not be considering the numbers to be signed for this problem). For this purpose, we define the result as:

$$(z_3z_2z_1z_0)_{2,4} = (x_1x_0)_{2,2} \times (y_1y_0)_{2,2}$$

with $(x_1x_0)_{2,2}$ and $(y_1y_0)_{2,2}$ the input numbers in binary.

a) Write the input-output definition table (truth table) for this operation: your table should have 16 rows of inputs, and 4 outputs (z_3, z_2, z_1, z_0) for each operation.

b) Write a compound proposition that is logically equivalent to the expression for z_3, z_2, z_1 and z_0. *Hint:* CNF and DNF forms may be useful here.

Solution:

$$(z_3z_2z_1z_0)_{2,4} = (x_1x_0)_{2,2} \times (y_1y_0)_{2,2}$$

a) This is a multiplication operation using 2-bit binary numbers:

x_1	x_0	x_1x_0	y_1	y_0	y_1y_0	z_3	z_2	z_1	z_0	$z_3z_2z_1z_0$
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	0	0	0	0	0
0	0	0	1	0	2	0	0	0	0	0
0	0	0	1	1	3	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0
0	1	1	0	1	1	0	0	0	1	1
0	1	1	1	0	2	0	0	1	0	2
0	1	1	1	1	3	0	0	1	1	3
1	0	2	0	0	0	0	0	0	0	0
1	0	2	0	1	1	0	0	1	0	2
1	0	2	1	0	2	0	1	0	0	4
1	0	2	1	1	3	0	1	1	0	6
1	1	3	0	0	0	0	0	0	0	0
1	1	3	0	1	1	0	0	1	1	3
1	1	3	1	0	2	0	1	1	0	6
1	1	3	1	1	3	1	0	0	1	9

b) Let's consider each output:

z_0 : We observe that only 4 rows produce 1, so DNF is easier here:

$z_0 \equiv (\neg x_1 \wedge x_0 \wedge \neg y_1 \wedge y_0) \vee (\neg x_1 \wedge x_0 \wedge y_1 \wedge y_0) \vee (x_1 \wedge x_0 \wedge \neg y_1 \wedge y_0) \vee (x_1 \wedge x_0 \wedge y_1 \wedge y_0)$

z_1 : We observe that only 6 rows produce 1, so DNF is again easier here:

$z_1 \equiv (\neg x_1 \wedge x_0 \wedge y_1 \wedge \neg y_0) \vee (\neg x_1 \wedge x_0 \wedge y_1 \wedge y_0) \vee (x_1 \wedge \neg x_0 \wedge \neg y_1 \wedge y_0) \vee (x_1 \wedge \neg x_0 \wedge y_1 \wedge y_0)$
$\vee (x_1 \wedge x_0 \wedge \neg y_1 \wedge y_0) \vee (x_1 \wedge x_0 \wedge y_1 \wedge \neg y_0)$

z_2 : We observe that only 3 rows produce 1, so again DNF is easier here:

$z_2 \equiv (x_1 \wedge \neg x_0 \wedge y_1 \wedge \neg y_0) \vee (x_1 \wedge \neg x_0 \wedge y_1 \wedge y_0) \vee (x_1 \wedge x_0 \wedge y_1 \wedge \neg y_0)$

z_3 : We observe that only 1 row produces 1, so again DNF is easier here:

$z_3 \equiv (x_1 \wedge x_0 \wedge y_1 \wedge y_0)$

Problem 6 (Bonus: 10 points)

Simplification of logical propositions is often a challenging aspect of digital design. You are given the following truth table:

p	*q*	*r*	Output
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Express the output as a compound proposition, whose circuits use **exactly 2** gates. You are allowed to use AND $(\wedge)$, OR $(\vee)$, NOT $(\neg)$ and XOR $(\oplus)$.

Solution: Using DNF form, we can write this as:

$$(\neg p \wedge q \wedge r) \vee (p \wedge q \wedge \neg r)$$

Here. we see that the output uses 7 gates (7 operators are used in the operation).

$$(\neg p \land q \land r) \lor (p \land q \land \neg r) \equiv (q \land \neg p \land r) \lor (q \land p \land \neg r)$$

Using the distributive property, we can take q common and rewrite as:

$$(q \land \neg p \land r) \lor (q \land p \land \neg r) \equiv q \land ((\neg p \land r) \lor (p \land \neg r))$$

Recall that:

$$(\neg p \land r) \lor (p \land \neg r) \equiv p \oplus r$$

Substituting, we get:

$$q \land ((\neg p \land r) \lor (p \land \neg r)) \equiv q \land (p \oplus r)$$

Observe that this expression uses only 2 operators (consequently only 2 gates).