CSE 20 Homework 2
Due: Sunday Jan 22, 2017 at noon

Topics Logic and Circuits

Reading Rosen Section 4.1: pages 237-240, (up to but not including Modular Arithmetic), Section 4.2: pages 245-253 (up to but not including Modular Exponentiation), Sections 1.1, 1.2, 1.3 (up to page 31).

Key Concepts Base expansion, decimal expansion, binary expansion, hexadecimal expansion, octal expansion, bit shift, DIV, MOD. Proposition, propositional variable, truth value, compound propositions, truth table, negation, conjunction, disjunction, exclusive or, conditional statements, converse, contrapositive, inverse, biconditional, bitwise operations, consistent specifications, logical equivalence, De Morgan's laws.

---

1. (**10 points**) Hexadecimal numbers are frequently used to more concisely represent data stored in binary.

a. Use the definition of base expansion from page 246 (Theorem 1), to justify why

$$(b_7b_6b_5b_4b_3b_2b_1b_0)_2 = (x_1x_0)_{16}$$

where each x_i is defined as $x_i = \begin{cases} n & \text{if } n = (b_{4i+3}b_{4i+2}b_{4i+1}b_{4i})_2 \text{ and } n < 10 \\ A & \text{if } 10 = (b_{4i+3}b_{4i+2}b_{4i+1}b_{4i})_2 \\ B & \text{if } 11 = (b_{4i+3}b_{4i+2}b_{4i+1}b_{4i})_2 \\ C & \text{if } 12 = (b_{4i+3}b_{4i+2}b_{4i+1}b_{4i})_2 \\ D & \text{if } 13 = (b_{4i+3}b_{4i+2}b_{4i+1}b_{4i})_2 \\ E & \text{if } 14 = (b_{4i+3}b_{4i+2}b_{4i+1}b_{4i})_2 \\ F & \text{if } 15 = (b_{4i+3}b_{4i+2}b_{4i+1}b_{4i})_2 \end{cases}$

Note: a calculation plus a concluding sentence or two will be ample for the justification.

b. Bitwise operations on hexadecimal numbers are defined in terms of their corresponding binary representation: to compute

$$(x_1x_0)_{16} \wedge (y_1y_0)_{16}$$

first convert each of the hexadecimal numbers to their 8 bit binary representations and then perform the bitwise operation (and similarly for $\vee, \oplus$).

Compute

$$(13)_{16} \oplus (FB)_{16}$$

as a bitwise operation. *Note: Include any relevant calculations and explanations.*

c. The **least significant bit** of a number in hexadecimal expansion is the coefficient in the ones place when we convert to binary. For example, the least significant bit of $(A2)_{16}$ is 0 because $(A2)_{16} = (10100010)_2$ has a 0 in the ones place. For which $(y_1y_0)_{16}$ and which bit operation ☺ is it true that

$$(x_1x_0)_{16} \text{ ☺ } (y_1y_0)_{16}$$

equals the least significant bit of $(x_1x_0)_{16}$.

Note: for full credit, choose the value of y_0, y_1, choose whether ☺ is $\vee, \wedge, \oplus$, and then explain why your choices work.

2. (**12 points**) In this question, you will construct a circuit that takes a pair of two-bit integers $(x_1x_0)_2$ and $(y_1y_0)_2$ and computes the four output bits for their integer product.
For example, the product of $(10)_2$ and $(11)_2$ is $(110)_2$ and could be computed by long multiplication as

$$\begin{array}{rl}
1\ 0 & \text{This is } x_1x_0 \\
\times\ 1\ 1 & \text{This is } y_1y_0 \\
\cline{1-1}
1\ 0 & \\
+1\ 0 & \\
\cline{1-1}
1\ 1\ 0 & \text{This is the product, whose four output bits will be 0110.}
\end{array}$$

Note: In this context, we allow the leading bit to be 0 *since we're using fixed-width representations.*

a. Express the two-bit output of multiplying y_0 with $(x_1x_0)_2$ as two compound proposition: one for the least significant bit, call it a_0, and one of the most significant bit, call it a_1.

   Note: the compound proposition may include the operators $\neg, \vee, \wedge, \oplus$.

b. Use part a. and our work in class on adding integers in binary to express each of the four bits of output as compound propositions with inputs x_0, x_1, y_0, y_1.

   Note: the compound proposition may include the operators $\neg, \vee, \wedge, \oplus$.

c. Draw a logic circuit with four inputs and four outputs implementing your design in part b.

   Note: the logic circuit may include the gates for AND, OR, and XOR, and each gate may have as many inputs as you need.

3. (**9 points**) A given positive integer n has binary expansion coefficients $a_{k-1}, \cdots, a_2, a_1, a_0$. Express the binary expansions of the following numbers in terms of these coefficients. For example, the binary expansion of $2n$ is $(a_{k-1}\cdots a_2a_1a_0 0)_2$.

a. $8n$

b. n *DIV* 4

c. n *MOD* 4

Note: For full credit, you must explain your reasoning and include any assumptions you make.

4. (**9 points**) Teachers in the Middle Ages supposedly tested the real-time propositional logic ability of a student via a technique known as an obligato game. In an obligato game, a number of rounds is set and in each round the teacher gives the student successive assertions that the student must either accept or reject as they are given. When the student accepts an assertion, it is added as a commitment; when the student rejects an assertion its negation is added as a commitment. The student passes the test if the consistency of all commitments is maintained throughout the test.
For example, in a two-round obligato game, where the teacher gives the proposition $p \wedge q$ and then the proposition $p \vee q$:

- If the student's sequence of answers is ACCEPT, ACCEPT then the commitments are that $p \wedge q$ (is true) and $p \vee q$ (is true). These two statements are consistent (there is some truth assignment to the variables p and q that makes both statements true) so the student passes.
- If the student's sequence of answers is ACCEPT, REJECT then the commitments are that $p \wedge q$ is true and that $\neg(p \vee q)$ is true. However, these statements are inconsistent (the first requires both p and q to be T whereas the second requires both of them to be F). In this case, the student fails the test.

Suppose that in a three-round obligato game, the teacher first gives the student the proposition $p \rightarrow q$, then the proposition $\neg(p \vee r) \vee q$, and finally the proposition q. For which of the eight possible sequences of three answers will the student pass the test? Briefly justify your answer for each sequence.

5. (**10 points**) Two compound propositions are logically equivalent if they have the same truth table. For example, the following two compound propositions are logically equivalent: $p \rightarrow q$ and $q \vee \neg p$. Using exactly k propositional variables, how many compound propositions are there that are **not** logically equivalent to each other? For example, for $k = 1$ there are only four non-logically equivalent compound propositions: p, $\neg p$, $p \vee \neg p$, and $p \wedge \neg p$; so the answer for $k = 1$ is 4. Find a formula for the answer in terms of k and briefly explain how you got that formula.

Note: If you simply try to count non-logically equivalent compound propositions, then this is a very hard problem. Instead, use the definition of what it means for compound propositions to be logically equivalent to rephrase what you need to count for this question.