LSU EE 2720-2 Homework 2 Solution Due: 28 September 2011

Solution updated :: 30 September 2011, 10:45:27 CDT (Solutions to 4b and 4c were swapped.)

Problem 1: Perform the multiplications indicated below.

Multiply the following two 8-bit unsigned binary integers into a 16-bit product: `01110010 + 10010011`.

```
Solution:
          01110010 = 0x72 = 114
          10010011 = 0x93 = 147
          --------
          01110010 <- Pad with 0 zeros.
         011100100 <- Pad with 1 zero.
      011100100000 <- Pad with 4 zeros.
   011100100000000 <- Pad with 7 zeros.
   ---------------
  0100000101110110 <- Binary to the right, below show in hex and decimal.
  0100 0001 0111 0110 = 0x4176 = 16758
```

Multiply the following two 8-bit signed 2's complement integers into a 16-bit product: `01110010 + 10010011`.

```
Solution:  Binary numbers broken into groups of 4 digits for readability.

            0111 0010 = 0x72 = 114     Multiplicand
            1001 0011 = 01101100 + 1 = 01101101 => -109
            ---------
            0111 0010 <- Pad with 0 zeros.
          0 1110 0100 <- Pad with 1 zero.
       0111 0010 0000 <- Pad with 4 zeros.
  1100 0111 0000 0000 <- Pad with 7 zeros. Used negated and sign ext m'cand
  -------------------
  1100 1111 0111 0110 <- Product
    11 0000 1000 1010 <- Two's comp of product: 0x308a = 12426, so prod is -12426

How last partial product obtained:
            0111 0010 <- Multiplicand
            1000 1110 <- Negated Multiplicand (2's complement)
   100 0111 0000 0000 <- Pad on left with 7 zeros.
  1100 0111 0000 0000 <- Make 16-bit by sign extension (repeating sign bit).
```

The theorem numbers in the problems below are from Dr. Skavantos' Handout 5, available via `http://www.ece.lsu.edu/alex/EE2720/EE2720_H05.pdf`.

Problem 2: Prove the following

(*a*) Prove theorem T5 (called *complement* in the notes, but a better name is the damned-if-you-do,-damned-if-you-don't theorem) by perfect (finite) induction.

Proof by finite induction means verifying equality for all possible values of the variables. Since there's just one variable that's $2^1 = 2$ verifications. Theorem T5 is $x + x' = 1$. For $x \to 1$ we have $1 + 1' = 1 \Rightarrow 1 + 0 = 1$ which is true by an axiom. For $x \to 0$ we have $0 + 0' = 1 \Rightarrow 0 + 1 = 1$ which is also true by an axiom. *Q.E.D.*

(*b*) Prove that

$$(x + y) \cdot (z + w) = x \cdot z + x \cdot w + y \cdot z + y \cdot w$$

using the axioms and theorems T1 to T11 (and their duals). **Do not** prove it using perfect induction or by otherwise substituting values for the variables. (For example, a proof like the following is not allowed: If $x = 0$ the statement becomes $y \cdot (z + w) = y \cdot z + y \cdot w$, true by T8, if $x = 1 \ldots$.)

One can easily see that the equality holds by applying the familiar distributive rules that apply both to ordinary algebra and Boolean algebra. The point of this problem is to identify the exact theorems.

Start by applying T8 (looking at the right-hand side first), $a \cdot b + a \cdot c = a \cdot (b + c)$ with $a \to x + y$, $b \to z$, and $c \to w$; that yields $(x + y) \cdot z + (x + y) \cdot w$. Apply T8 again to the first term, this time with $a \to z$, $b \to x$, and $c \to y$. This yields $x \cdot z + x \cdot y + (x + y) \cdot w$. Applying T8 once again to the last term yields $x \cdot z + x \cdot y + x \cdot w + y \cdot w$. A few applications of the commutativity theorem yields the original expression. *Q.E.D.*

(*c*) Draw a logic diagram (a connection of logic gates) for the left- and right-hand side of the equality in the previous problem.

(*d*) Prove that

$$x + y_1 \cdot y_2 \cdot \; \cdots \; \cdot y_n = (x + y_1) \cdot (x + y_2) \cdot \; \cdots \; \cdot (x + y_n)$$

using induction.

We start the inductive proof by proving the equality for $n = 2$, meaning we need to prove $x + y_1 y_2 = (x + y_1) \cdot (x + y_2)$. That's true by T8'. Next assume that the equality holds for $n \in [2, n-1]$ (for values of n from 2 to $n - 1$). We can re-write the original equality as

$$x + y_1 \cdot y_2 \cdot \; \cdots \; \cdot y_{n-2} \cdot (y_{n-1} \cdot y_n) = (x + y_1) \cdot (x + y_2) \cdot \; \cdots \; \cdot (x + y_{n-2}) \cdot (x + y_{n-1} y_n)$$

and using our assumption that the original equality holds for $n - 1$ the equality above is true (because we are treating $y_{n-1} y_n$ as a single variable). We already proved that $x + y \cdot z = (x + y) \cdot (x + z)$, applying that to the last product term of the right hand side yields:

$$x + y_1 \cdot y_2 \cdot \; \cdots \; \cdot y_{n-2} \cdot (y_{n-1} \cdot y_n) = (x + y_1) \cdot (x + y_2) \cdot \; \cdots \; \cdot (x + y_{n-2}) \cdot (x + y_{n-1}) \cdot (x + y_n)$$

Which is the original equality. *Q.E.D.*

Problem 3: Simplify the expressions below:

(*a*) Simplify:

$$i \cdot n + a \cdot n \cdot t \cdot i \cdot c + t \cdot a \cdot x \cdot i' + t \cdot a \cdot n \cdot x$$

A good place to start is looking at the shortest term, $i \cdot n$ to see if those same variables appear in any other. They do!, in $antic$ (using juxtaposition for logical and, since I'm tired of typing `cdot`). Therefore, by the cover theorem $antic$ disappears. That leaves $in + taxi' + tanx$. Notice that i appears un-negated in one term and negated in another, meaning there is a possibility of applying the consensus theorem. Looking further we see that yes we can: the consensus theorem is $xa + x'b + ab = xa + x'b$, letting $x \to i$, $a \to n$, and $b \to tax$. That yields the simplified expression: $in + taxi'$.

(*b*) Simplify:

$$i \cdot n + (i \cdot n)' \cdot k + c \cdot a \cdot s \cdot k$$

Notice that in appears negated and un-negated. One should see if that can be taken advantage of before demorganizing $(in)'$. A useful theorem given in class and handout 5, is $x + x'y = x + y$. Applying this here yields $in + k + cask$. By the cover theorem k kills $cask$ and so the expression reduces to $in + k$.

(c) Simplify, noting that the difference in the second term in this subproblem and the previous one is significant.

$$i \cdot n + i' \cdot n' \cdot k + n' \cdot i' \cdot b$$

In this can the best we can do is factor the $i'n'$ which yields $in + i'n'(k + b)$.

Problem 4: Perform the conversions indicated below:

(a) Convert the following to sum-of-products form, paying attention to opportunities for simplification.

$$(a + b) \cdot (a + c) \cdot (a' + d) + b \cdot a \cdot d$$

One can always get a SoP by first demorganizing (doesn't apply here) then multiplying out. But to save time lets do some simplifications first. By T8' we get $(a+bc)(a'+d)+bad$. In class we noted that $(x+y)(x'+z) = xz+x'y$. Applying that to the first term yields $ad + a'bc + bad$ which is in sum of products form. But wait!, we can use cover to get rid of the "bad" term: $ad + a'bc$.

(b) Convert the following to product-of-sums form, paying attention to opportunities for simplification.

$$[(a + b) \cdot (a + c) \cdot (a' + d) + b \cdot a \cdot d]'$$

To avoid tedium simplify before demorganizing:

$$[(a + bc) \cdot (a' + d) + bad]'$$

$$[(ad + a'bc) + bad]'$$

$$[ad + a'bc + bad]'$$

now demorganize (being careful with parentheses)

$$(ad)' \cdot (a'bc)' \cdot (bad)'$$

$$(a' + d') \cdot (a + b' + c') \cdot (b' + a' + d')$$

Lucky us!, it's now in product of sum form.

(c) Convert the following to sum-of-products form, paying attention to opportunities for simplification.

$$[(a + b) \cdot (a + c) \cdot (a' + d) + b \cdot a \cdot d]'$$

We need to do some demorganization first:

$$[(a+b)\cdot(a+c)\cdot(a'+d)]'\cdot(b\cdot a\cdot d)'$$
$$[(a+b)'+(a+c)'+(a'+d)']\cdot(b'+a'+d')$$
$$[(a'\cdot b')+(a'\cdot c')+(a\cdot d')]\cdot(b'+a'+d')$$

eliminating unneeded parenthesis:

$$(a'\cdot b'+a'\cdot c'+a\cdot d')\cdot(b'+a'+d')$$

multiplying out yields and instantly eliminating zero and duplicate terms:

$$a'b'+a'b'd'+a'c'b'+a'c'+ad'b'+ad'$$

apply cover:

$$a'b'+a'c'+ad'$$

(d) Verify your solution to the conversions above by constructing a truth table. One output column should be for the expression $(a+b)\cdot(a+c)\cdot(a'+d)+b\cdot a\cdot d$, and there should be three more columns, one for each of the conversions.

The truth table below is for problem 4.

a b c d	p4a	sol	p4b	sop	pos
0 0 0 0	0	0	1	1	1
0 0 0 1	0	0	1	1	1
0 0 1 0	0	0	1	1	1
0 0 1 1	0	0	1	1	1
0 1 0 0	0	0	1	1	1
0 1 0 1	0	0	1	1	1
0 1 1 0	1	1	0	0	0
0 1 1 1	1	1	0	0	0
1 0 0 0	0	0	1	1	1
1 0 0 1	1	1	0	0	0
1 0 1 0	0	0	1	1	1
1 0 1 1	1	1	0	0	0
1 1 0 0	0	0	1	1	1
1 1 0 1	1	1	0	0	0
1 1 1 0	0	0	1	1	1
1 1 1 1	1	1	0	0	0