CS2100: Computer Organisation

Tutorial #6: Boolean Algebra, Logic Gates and Simplification

(Week 8: 9 – 13 March 2020)

LumiNUS Discussion Questions:

D1. (a) One common mistake is the following: $A \cdot B + A' \cdot B' = 1$ … (equation 1)

This seems to be erroneously "derived" from the following rule: $X + X' = 1$

Explain why the rule is wrongly applied here.

(b) Is the following equation correct? Why?

$A \cdot B + (A \cdot B)' = 1$ … (equation 2)

D2. Given the following two 3-variable Boolean functions:

$\boldsymbol{F(A,B,C) = \Sigma\, m(0, 2, 4, 6, 7)}$
$\boldsymbol{G(A,B,C) = \Sigma\, m(1, 2, 3, 6)}$

(a) Write the product-of-maxterms expressions in ΠM notation for F and G.

(b) If $X = F + G$, write the sum-of-minterms expressions in Σm notation for X.

(c) If $Y = F \cdot G$, write the sum-of-minterms expressions in Σm notation for Y.

(d) If $Z = F \oplus G$, write the sum-of-minterms expressions in Σm notation for Z.

Do you know how to generalise the above for any arbitrary Boolean functions F and G?

[To make it easy for you to type in the LumiNUS forum, you may use Sum-m to mean Σm and Prod-M to mean ΠM. Example: Sum-m(0, 2, 4, 6, 7), Prod-M(2, 3, 5).]

D3. How many prime implicants (PIs) and essential prime implicants (EPIs) are there in each of the K-maps? [d(…) and D(…) denote don't-cares.]

(a) $F1(A,B,C,D) = \Sigma\, m(5, 8, 10, 12, 13, 14)$

(b) $F2(W,X,Y,Z) = \Pi\, M(0, 1, 2, 8, 9, 10)$

(c) $F3(K,L,M,N) = \Sigma\, m(1, 7, 10, 13, 14) + d(0, 5, 8, 15)$

(d) $F4(A,B,C,D) = \Pi\, M(4, 8, 9, 11, 12) \cdot D(2, 3, 6, 7, 10, 14)$

D4. For each of the functions in D3 above, find the simplified **POS expression**. List out all alternative answers, if any.

You are encouraged to do the above discussion questions and discuss them on LumiNUS forum. These are fundamental concepts that you must know, before you attempt the tutorial questions below.

Tutorial Questions:

1. Without referring to the book or lecture slides, name the essential theorems A, B, C and D used in the following derivation:

$$
\begin{aligned}
F(j, k, m, p) &= k' \cdot (\, j' \cdot p \cdot (j' + m')\,)' + (\, p + k' + j\,)' \\
&= k' \cdot (\, j + p' + j{\cdot}m) + p'{\cdot}k{\cdot}j' && \text{... [A; involution]} \\
&= j{\cdot}k' + k'{\cdot}p' + j{\cdot}k'{\cdot}m + p'{\cdot}k{\cdot}j' && \text{... [distributive; commutative]} \\
&= j{\cdot}k' + p'{\cdot}(\, k' + k{\cdot}j'\,) + j{\cdot}k'{\cdot}m && \text{... [associative; commutative; distributive]} \\
&= j{\cdot}k' + k'{\cdot}p' + j'{\cdot}p' + j{\cdot}k'{\cdot}m && \text{... [B; distributive; commutative]} \\
&= j{\cdot}k' + k'{\cdot}p' + j'{\cdot}p' && \text{... [associative; C]} \\
&= j{\cdot}k' + j'{\cdot}p' && \text{... [D]}
\end{aligned}
$$

Note: In writing out terms, you should write the literals in the order of significance, especially in your final answer. For instance, for the above Boolean function $F(j, k, m, p)$, you should write the final answer as $j{\cdot}k' + j'{\cdot}p'$ and not $k'{\cdot}j + j'{\cdot}p'$ or $j{\cdot}k' + p'{\cdot}j'$.

2. Using Boolean algebra, simplify each of the following expressions into simplified **sum-of-products (SOP) expressions**. Indicate the law/theorem used for each step.

   (a) $F(x,y,z) = (x + y{\cdot}z'){\cdot}(y' + y) + x'{\cdot}(y{\cdot}z' + y)$

   (b) $G(p,q,r,s) = \Pi\, M(5, 9, 13)$

   Tip: For (b), it is simpler to start with the given expression, rather than to expand it into sum-of-products/sum-of-minterms expression first.

3. Design a **divide-by-3 circuit** as follows: the input is a 4-bit unsigned binary number $ABCD$, and the output is a 3-bit unsigned binary number XYZ which is the quotient of $ABCD$ / 3. For example, if $ABCD$ = 1100 (or 12 in decimal), then XYZ = 100 (or 4 in decimal); if $ABCD$ = 0111 (or 7 in decimal), then XYZ = 010 (or 2 in decimal).

   (a) Draw the truth table and try to obtain the **simplified SOP expressions** of X, Y, and Z just from observation. (It is quite easy to obtain the simplified SOP expressions for X and Y just from observing the truth table, but harder for Z. The K-map technique is probably useful here, but we'll do that in the next question.)

   (b) Verify your answer for X by first writing out its **sum-of-minterms expressions** and then simplifying the expression from there using Boolean algebra. Write out the law/theorem you use at each step.

   (c) From the simplified SOP expressions, implement X, Y, and Z using (i) **2-level AND-OR circuits**, and (ii) 2-**level NAND-only circuits**. Assume that primed literals are not available. (Always assume that prime literals are not available unless otherwise stated.)

4. A 3-bit code *PQR* is designed to represent the 6 digits (0 – 5) in a base-6 system as follows:

Digit	3-bit code *PQR*
0	001
1	011
2	010
3	101
4	100
5	110

Design a logic circuit (you need not draw it) that takes in this 3-bit code *PQR* and generates two outputs:

- Output *Y* is 1 if the code *PQR* represents a prime number; otherwise, it is 0.
- Output *V* is 1 if the code *PQR* is valid; otherwise, it is 0.

Complete the truth table below. Using K-maps, obtain the simplified SOP expressions for *Y* and *V*. Give all alternative answers, if any. (Questions to settle before you can proceed: What are prime numbers? How do you handle invalid codes?)

P	***Q***	***R***	***Y***	***V***