LAB 1: LOGIC MINIMIZATION

This lab will give you a chance to review the basics of Boolean algebra and implement the Karnaugh Map (K-map) and Quine-McCluskey (tabular) methods to minimize Boolean functions, and to realize said Boolean functions in a logic circuit.

When prompted to implement a circuit, feel free to use Logisim-evolution or CircuitVerse.

Part 1: Boolean Algebra Review

1.A) Equivalent Operations

Consider two functions f and g, of two inputs x and y, with x as the MSB, and y as the LSB:

$$f(x,y) = \sum m(1,2) \quad \textbf{(1)}$$

$$g(x,y) = \prod m(0,4) \quad \textbf{(2)}$$

a) Draw up the truth tables of f and g. Are f and g equivalent functions? If so, is there a name for this Boolean function?
b) One of these functions, as expressed, is more naturally implemented as an OR-AND circuit, and the other as an AND-OR circuit. Which is which?
c) Implement f and g, based on your response in b).

1.B) De Morgan's Theorem

Recall the De Morgan's Theorem:

$$\overline{(a_1 + a_2 + \cdots + a_n)} = (\overline{a_1} \cdot \overline{a_2} \cdot \ldots \cdot \overline{a_n}) \quad \textbf{(3)}$$

$$\overline{(a_1 \cdot a_2 \cdot \ldots \cdot a_n)} = (\overline{a_1} + \overline{a_2} + \cdots + \overline{a_n}) \quad \textbf{(4)}$$

d) Suppose you only have access to NAND and NOR gates. Implement the OR-AND and AND-OR Circuits using only NAND and NOR gates. [!]

Part 2: The K-Map Method

Consider two functions H and L of three inputs x, y, and z, with x as the MSB and z as the LSB.

$H(x, y, z) = \sum m(1,2,4,5,6,7)$ **(5)**

$L(x, y, z) = \prod m(3,5,6,7)$ **(6)**

a) Draw up a K-map for each of these functions, and minimize them to determine an SOP or POS for each of them.
b) Implement each of these as OR-AND (for SOPs) or AND-OR circuits (for POSs).

Part 3: The Tabular Method

We will once again use the function H in Part 2.

a) Before constructing an Implicant Table, first construct an Implicant Chart for H, and determine an SOP for the two functions.
b) Now, construct an Implicant Table, and use the Quine-McCluskey method to find the prime implicants of H. When combining two minterms, keep a catalog of which implicants you are combining (e.g., if you combine and get 110-, keep track that it covers 1100 and 1101.).
c) Consider one such pairing obtained from b). And consider another function $\widetilde{H} = \sum m(p_i, p_j)$, where p_i and p_j are the two implicants combined in b). Implement it as an AND-OR circuit. Now, implement another function $H' = \sum m\ (P)$, where P is the resulting implicant from combining p_i and p_j . (e.g., if p_i = 0001 and p_j = 0000, then P = 000-.) **Are these functions equal?**
d) Now, construct an Implicant Chart for H, at each iteration of the process. What happens to the number of rows? To the number of prime implicants? To the amount of gates one would require to implement H?
e) Compare the two implementations for H from this Part, and that of Part 2. Are they the same, or different? Why or why not would that be the case? If they are different, which is the most efficient?
f) Repeat a) to e) for function L.