Appendix: Basic Boolean algebra

In this exercise we are going to develop the skill of implementing Boolean logic expressions through transistor pull-up and pull-down networks. Since we want to use as few MOSFETs as possible when designing CMOS logic gates, it is important to know how to minimize Boolean expressions.
Basic knowledge includes writing Boolean truth tables and from them extracting minimized Boolean logic equations. In a Boolean truth table, any row represents a so called *minterm* and can be represented by a square in a Karnaugh map. Any minterm can be expressed as a product of all the variables involved.

For a 2-input NAND gate the Boolean truth table and the Karnaugh map are shown in Fig. 1. Each row in the truth table represents a minterm, one for each of the squares in the Karnaugh map. The NAND function is true for three of the four minterms, and can be written as the following sum of products: $Z = \overline{A}\overline{B} + A\overline{B} + \overline{A}B$. The NAND function is false for the remaining minterm, $\overline{Z} = AB$. Boolean functions can be minimized by grouping minterms into prime implicants, i.e. into sums of products of a reduced number of inputs. For the Boolean NAND function he prime implicants are $\overline{A}$ and $\overline{B}$. Hence, $Z = \overline{A} + \overline{B}$.

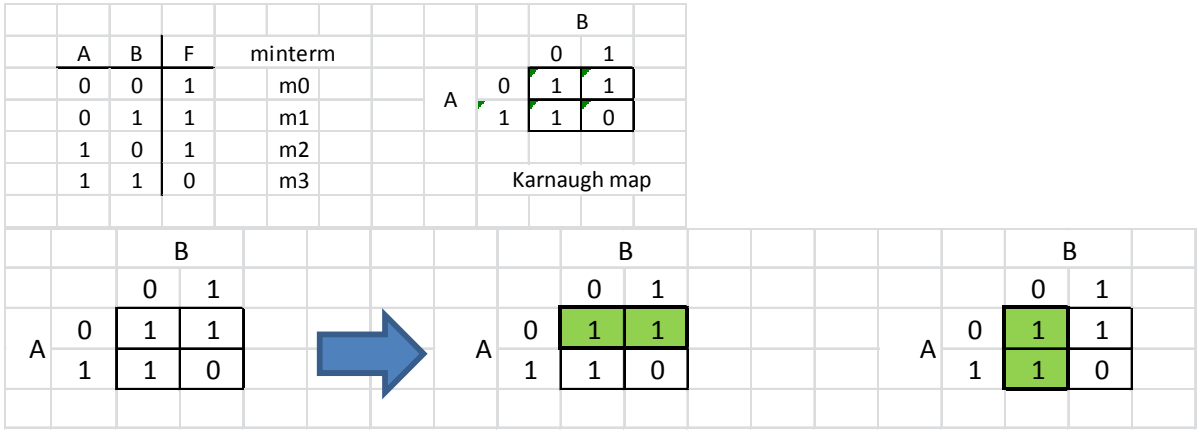

Fig. 1. Boolean truth table, Karnaugh diagram, and prime implicants.

Since the NAND function also can be written $Z = \overline{AB}$, this is a good time to remind ourselves of de Morgans theorem,

$$\overline{AB} = \overline{A} + \overline{B}, \text{ or alternatively } \overline{A + B} = \overline{A}\,\overline{B}.$$

De Morgan's theorem is illustrated graphically in figure 2.

Fig. 2. Illustration of de Morgan's theorem for NAND and NOR-gates.

A more complex Boolean function of four variables is given by $Z = AB + BC + CD$, which is a sum of the three products, or prime implicants, *AB*, *BC*, and *CD*. This example is illustrated in Fig. 3.

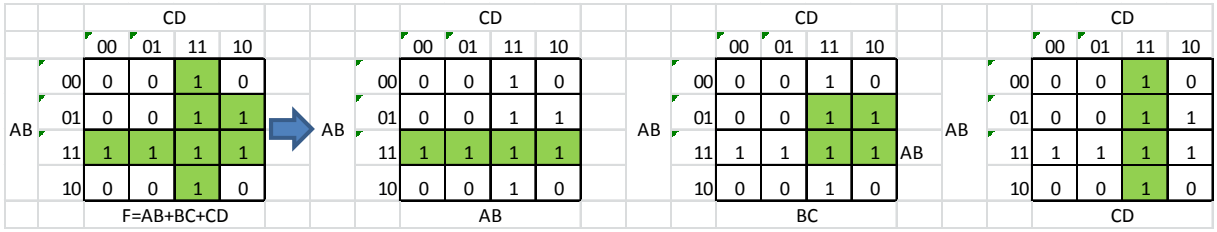

Fig. 3. Karnaugh diagram, and prime implicants for the Boolean function $Z = AB + BC + CD$.

The corresponding expression for when the Boolean function is false is given by

$$\overline{Z} = \overline{A}\overline{C} + \overline{B}\overline{C} + \overline{B}\overline{D}.$$

Applying de Morgan´s theorem on this expression we obtain

$$Z = \overline{\overline{A}\overline{C}} + \overline{\overline{B}\overline{C}} + \overline{\overline{B}\overline{D}} = (A + C)(B + C)(B + D).$$

The conclusion of this exercise is that Boolean functions can not only be written as sums of products, but also as products of sums (POS). If prime implicants are reduced minterms, these sums are reduced maxterms.

It is not very often that you derive the maxterms and the product of sums for any Boolean function from the Karnaugh map, but the way to do it is illustrated in Fig. 4. Each of the reduced maxterms are found by excluding the reduced minterm when the Boolean function is false. By excluding $\overline{A}\overline{C}$ we find the maxterm (A+C), by excluding $\overline{B}\overline{C}$ we find the maxterm (B+C), and by excluding $\overline{B}\overline{D}$ we find the maxterm (B+D).

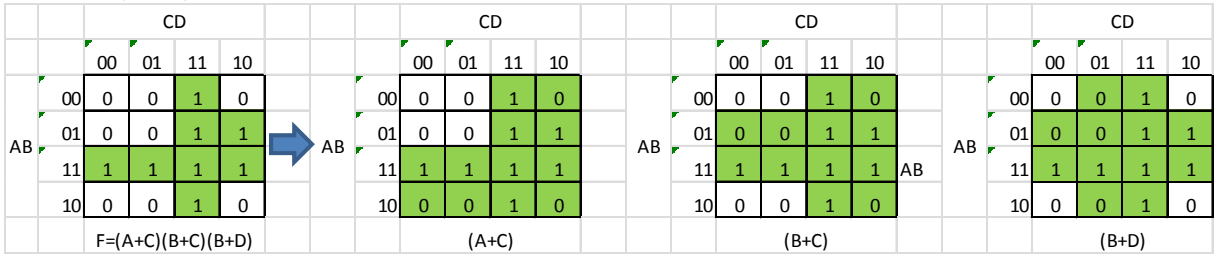

Fig. 4. Finding the reduced maxterms.