COMPUTER ARCHITECTURE
2020 - 2021

TD n°3 - Correction

Exercise 1: use of truth tables.

In a brasserie you order a ham sandwich **or** a pâté sandwich **and** a glass of beer. The waiter listens to you distractedly because he is busy.

At first, he is sure of the place of the **or** and the **and** but he hesitates about the place of the brackets.

1. Write the corresponding propositional formula for each of the possible commands.

   We define three propositional variables J, P and B whose truthfulness is interpreted as "I order a ham sandwich", "I order a pâté sandwich" and "I order a beer" respectively.

   The hesitation between the place of the brackets results in:

   $$\underbrace{(J \vee P) \wedge B}_{X} \qquad \underbrace{J \vee (P \wedge B)}_{Y}.$$

2. To be sure to satisfy the customer, it must satisfy both possible orders: write the corresponding propositional formula.

   Show with the help of a truth table that the latter is logically equivalent to bringing (a ham sandwich **or** a pâté sandwich) **and** a glass of beer.

We construct the truth table of the formula X ∧ Y to show that we have the logical equivalence X ≡ (X ∧ Y).

J	P	B	$J \vee P$	$P \wedge B$	X	Y	$X \wedge Y$
F	F	F	F	F	F	F	F
F	F	V	F	F	F	F	F
F	V	F	V	F	F	F	F
F	V	V	V	V	V	V	V
V	F	F	V	F	F	V	F
V	F	V	V	F	V	V	V
V	V	F	V	F	F	V	F
V	V	V	V	V	V	V	V

In a second step the waiter also hesitates about the place of **or** and **and**.

3. Repeat questions (1) and (2) with logical equivalences.

We repeat, this time swapping the connectors ∧ and ∨:

$$\underbrace{(J \wedge P) \vee B}_{U} \qquad \underbrace{J \wedge (P \vee B)}_{V}.$$

4. Show that he can then just bring a ham sandwich **and** (a pâté sandwich **or** a glass of beer).

We construct the truth table of the formula U ∧ V to show that we have the logical equivalence V ≡ (U ∧ V). It also recalls the formula X ∧ Y for the following question:

J	P	B	$J \wedge P$	$P \vee B$	U	V	$U \wedge V$	$X \wedge Y$
F	F	F	F	F	F	F	F	F
F	F	V	F	V	V	F	F	F
F	V	F	F	V	F	F	F	F
F	V	V	F	V	V	F	F	V
V	F	F	F	F	F	F	F	F
$\mathbf{V}$	F	$\mathbf{V}$	F	V	V	V	$\mathbf{V}$	$\mathbf{V}$
V	V	F	V	V	V	V	V	F
$\mathbf{V}$	V	$\mathbf{V}$	V	V	V	V	$\mathbf{V}$	$\mathbf{V}$

What should it provide as a minimum to satisfy the customer and answer all his hesitations?

As can be seen in the last two columns of this table, both U ∧ V and X ∧ Y can be satisfied for the interpretation J ≡ V and B ≡ V, in other words, if the waiter brings a ham sandwich and a beer, he is sure to satisfy your request even if he has misunderstood it.

<u>Exercise 2</u>: equation simplification.

We give the equation $\mathbf{t} = \mathbf{x\overline{y}} + \mathbf{z}\,(\mathbf{\overline{x}} + \mathbf{y})$.

Start by rewriting this equation without parentheses, with three terms: build the truth table, then Karnaugh's rectangular table with **xy** on the one hand and **z** on the other hand.

Deduce the simplified form of **t**.

The truth table of $\mathbf{t} = \mathbf{x\overline{y}} + \mathbf{z\overline{x}} + \mathbf{zy}$ results in the equation:

$$t = \overline{x}\,\overline{y}\,z + \overline{x}\,y\,z + x\,\overline{y}\,\overline{z} + x\,\overline{y}\,z + x\,y\,z$$

Let's simplify it with a Karnaugh table.

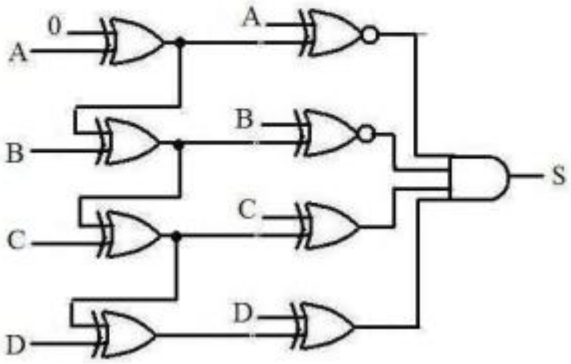
z \ xy	00	01	11	10
0	0	0	0	1
1	1	1	1	1

$$t = x\,\overline{y} + z$$

Exercise 3: about XOR gates

We give this logic circuit with four input bits A, B, C and D, and an output S.

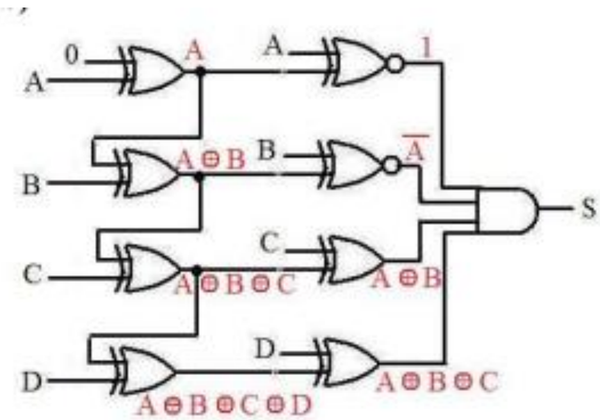
Show that there are exactly two cases for the inputs leading to S = 1 at the output and give these two cases. To do this, add to the drawing below the results obtained at the output of each of the XOR gates in the diagram.



Preamble: $0 \oplus x = x$, $x \oplus x = 0$ as seen by making $x = 0$ and then $x = 1$. Note that we also have $1 \oplus x = \overline{x}$.

To have S = 1, all inputs to the AND gate must be 1, which means: $A = 0$, $A \oplus B = 1$, hence $B = 1$, then $A \oplus B \oplus C = 1$, hence $C = 0$.

Since D is arbitrary, there are indeed two solutions for (A, B, C, D): (0, 1, 0, 0) or (0, 1, 0, 1).



Exercise 4: construct a logical function.

Let an integer from 0 to 7 be represented by 3 bits $b_2b_1b_0$. Let F be the logic function whose inputs are these 3 bits, and which takes the value 1 if the input value is 0, 3, 4 or 7 (coded in binary), and 0 otherwise.

1. Give the truth table of F, write it in its canonical form and simplify the resulting expression.

b_2	b_1	b_0	F
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

We then obtain the canonical form of F and its simplification:

$$
\begin{aligned}
F &= \bar{b}_2\bar{b}_1\bar{b}_0 + \bar{b}_2 b_1 b_0 + b_2\bar{b}_1\bar{b}_0 + b_2 b_1 b_0 \\
&= (\bar{b}_2 + b_2)(\bar{b}_1\bar{b}_0) + (\bar{b}_2 + b_2)(b_1 b_0) \\
&= \bar{b}_1\bar{b}_0 + b_1 b_0 \\
&= \overline{b_1 \oplus b_0}
\end{aligned}
$$

2. Find the previous result by writing the Karnaugh table of F.

b_2 \ b_1b_0	00	01	11	10
0	1	0	1	0
1	1	0	1	0

Table de Karnaugh de F.

By carrying out the two groupings, we find the result:

$$F = \bar{b}_1 \bar{b}_0 + b_1 b_0.$$

3. Draw the logic circuit calculating F, using only NAND gates.

Morgan's law allows us to transform OR into AND; we obtain the corresponding formula and circuit:

$$F = \overline{\overline{\bar{b}_1 \bar{b}_0} . \overline{b_1 b_0}}$$