## **Exercices**

Exercice 231 Écrire en Python une fonction xor(x,y) qui réalise l'opérateur du même nom sur les booléens.

Exercice 232 Montrer de deux manières différentes l'égalité suivante.

$$(x \wedge y) \vee (\neg y \wedge z) = (x \vee \neg y) \wedge (y \vee z)$$

**Exercice 233** Définir une fonction booléenne f sur deux variables x et y qui vaut 1 si et seulement si deux variables ont la même valeur (qu'elle soit 0 ou 1), en utilisant uniquement les opérations NON, ET, OU ou OU exclusif. Donner sa table de vérité.

Exercice 234 On considère la fonction booléenne à deux variables suivante :

$$f(x,y) = (x \land \neg y \land \neg z) \lor (\neg x \land y \land \neg z) \lor (\neg x \land \neg y \land z)$$

Donner sa table de vérité. Que fait cette fonction? Donner une expression booléenne plus simple pour cette fonction.

**Exercice 235** Expliquer comment combiner deux additionneurs 1 bit pour obtenir un additionneur 2 bits. Donner la table de vérité de cet additionneur 2 bits. Elle doit avoir seize lignes.

**Exercice 236** Écrire un programme Python qui affiche la table d'une opération logique pour tous les entiers de n bits. Par exemple, pour l'opération & (ET logique) et n=3, le programme devra afficher ceci :

Pour afficher un entier x en base 2, on peut se servir de format(x, 'b'). Si on veut que cela soit joli, avec exactement n chiffres, il faut se fatiguer un peu.

**Exercice 237** Un multiplexeur k bits permet de sélectionner une entrée parmi  $2^k$  disponibles. Un tel circuit a  $k+2^k$  bits en entrée. Les k premiers  $(a_0,\ldots,a_{k-1})$  sont les bits d'adresse, ils servent à coder le numéro de l'entrée à sélectionner parmi les  $2^k$  entrées suivantes  $(b_0,\ldots,b_{2^{k-1}})$ . La sortie s du multiplexeur est égale à la sortie sélectionnée.

Donner tout d'abord la table de vérité d'un multiplexeur 1 bit ainsi que l'expression booléenne qui définit sa sortie. À l'aide de portes logiques, définir ensuite le schéma de ce multiplexeur.