

Circuits et Architecture (CA7)

TD n° 3 : Circuits simples

Manipulation d'expressions booléennes

On rappelle les identités remarquables suivantes :

$$\begin{array}{ll}
 a + 0 = a & a \cdot 0 = 0 \\
 a + 1 = 1 & a \cdot 1 = a \\
 a + \bar{a} = 1 & a \cdot \bar{a} = 0 \\
 a + a = a & a \cdot a = a
 \end{array}$$

$$\begin{array}{lll}
 \overline{(a + b)} = \bar{a} \cdot \bar{b} & \overline{a \cdot b} = \bar{a} + \bar{b} & \text{(lois de De Morgan)} \\
 (ab)c = a(bc) & (a + b) + c = a + (b + c) & \text{(associativité)} \\
 ab = ba & a + b = b + a & \text{(commutativité)} \\
 a(b + c) = ab + ac & a + (bc) = (a + b)(a + c) & \text{(distributivité)}
 \end{array}$$

$$a \oplus b = a\bar{b} + \bar{a}b \quad a \Rightarrow b = \bar{a} + b$$

Exercice 1 – Expressions simplifiées

Utilisez les identités ci-dessus pour simplifier les expressions suivantes :

1. $\overline{(\bar{a} + b)(a + \bar{b})}$
2. $a\bar{b} + \bar{a}b + ab$
3. $(a + b + c)(\bar{a} + bc)$
4. $(a + b) \Rightarrow (a + \bar{b} + c)$
5. $(a + b + c)\overline{(ab + ac + bc)}$

Exercice 2 – Circuits combinatoires booléens

Écrire les circuits codant des fonctions à n variables ci-dessous. Pour simplifier on considérera que $n = 2^p$.

1. ET
2. UN SEUL (vrai)

Manipulation de nombres binaires

Nous considérons ici des nombres entiers **positifs** qui sont codés en base 2 en utilisant n bits. Pour simplifier la construction récursive des circuits, n sera une puissance de 2. Le codage d'un nombre entier positif b est un vecteur $b_{n-1} \dots b_0$ avec $b_i \in \{0, 1\}$ tel que :

$$b = \sum_{i=0}^{n-1} b_i 2^i$$

Le but de cette partie est de travailler sur des circuits booléens réalisant des opérations sur les entiers.

Exercice 3 – Comparaison de nombres

1. On travaille sur deux nombres à 1 bit a et b . Écrivez les circuits testant $a < b$, $a \leq b$, $a = b$, $a > b$ et $a \geq b$.
2. On suppose qu'on dispose d'un circuit prenant a et b comme entrées et ayant deux sorties : $a < b$ et $a \leq b$ (et de portes classiques AND, OR, NAND, NOR, NOT, XOR). Comment construire les trois autres fonctions $a = b$, $a > b$ et $a \geq b$?
3. On s'intéresse maintenant à des nombres à 2 bits. Écrivez le tableau de Karnaugh des fonctions $a < b$ et $a \leq b$.
4. Déduisez-en les circuits correspondants. Comment cette construction se généralise pour des nombres à n bits ?
5. On va maintenant rechercher un circuit construit **récursivement** pour résoudre le problème, avec une technique "diviser pour régner". On considère que $n = 2^d$. a se décompose en $a_h = a_{n-1} \dots a_{\frac{n}{2}}$ et $a_l = a_{\frac{n}{2}-1} \dots a_0$ et b se décompose en $b_h = b_{n-1} \dots b_{\frac{n}{2}}$ et $b_l = b_{\frac{n}{2}-1} \dots b_0$.

Montrez comment les relations d'ordre $<$ et \leq entre a_h et b_h , et entre a_l et b_l , permettent de calculer les relations d'ordre entre a et b .

Écrivez le circuit correspondant (on suppose que l'on a, en boîte noire, un circuit prenant deux nombres à $n/2$ bits et ayant deux sorties : $<$ et \leq).

6. Déduisez-en le comparateur de nombres à n bits.