



**TD numéro 3 : Circuits combinatoires (II)**

Université de Provence  
 Année 2008-2009

# 1 Comparateurs $n$ -bits

On souhaite réaliser un comparateur pour des nombres de  $n$  bits avec les opérations  $<$  (strictement plus petit) et  $\leq$  (plus petit). Nous allons procéder de deux façons différentes et étudier chacune des deux solutions en terme de nombre de portes logiques utilisées et de temps de calcul.

## Exercice 1 (Comparateur associatif)

**Question 1.1** Dessiner deux circuits  $Comp_{<}$  et  $Comp_{\leq}$  ayant tous les deux 2 entrées  $a$  et  $b$  et une sortie  $s$  valant 1 si respectivement  $a < b$  et  $a \leq b$ .

Reprenons le fonctionnement des comparateurs associatifs dans le TD précédent. On utilise un comparateur par paire de bits, qui dispose de plus de deux entrées valant les résultats de comparaison pour les bits précédents dans les deux nombres. On retrouve à la figure 1 à gauche un des ces comparateurs associatifs et sur la droite, leur assemblage pour obtenir un comparateur 3 bits.

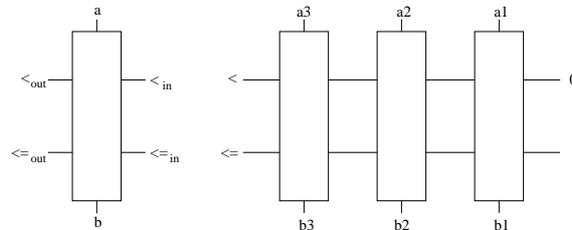


FIG. 1 –

Chaque comparateur associatif utilise le principe suivant :

- $(a_n a_{n-1} \dots a_2 a_1)_2 < (b_n b_{n-1} \dots b_2 b_1)_2$  ssi  
 $a_n < b_n$  ou  $a_n = b_n$  et  $(a_{n-1} \dots a_2 a_1)_2 < (b_{n-1} \dots b_2 b_1)_2$
- $(a_n a_{n-1} \dots a_2 a_1)_2 \leq (b_n b_{n-1} \dots b_2 b_1)_2$  si et seulement si  
 $a_n < b_n$  ou  $a_n = b_n$  et  $(a_{n-1} \dots a_2 a_1)_2 \leq (b_{n-1} \dots b_2 b_1)_2$

**Question 1.2** Selon ce principe, définir un circuit COMB permettant de construire un comparateur associatif en combinant des entrées  $<_{in}$  et  $\leq_{in}$  et les résultats des circuits  $Comp_{<}$  et  $Comp_{\leq}$ .

**Question 1.3** Pour le circuit obtenu, donner, en fonction de  $n$ , le nombre de portes logiques nécessaires à la réalisation d'un comparateur de nombres de  $n$  bits comme celui décrit figure 1.

**Question 1.4** Pour le circuit obtenu, calculer, en fonction de  $n$  et de  $t$  le temps de passage d'une porte logique pour les signaux, au bout de combien de temps les résultats de la comparaison sont disponibles sur la sortie.

**Exercice 2 (Comparateur récursif)** Etudions maintenant une autre manière de réaliser un comparateur  $n$  bits ; nous supposons ici que  $n$  est une puissance de 2, ie  $\exists d \in \mathbb{N}, n = 2^d$ .

**Question 2.1** Montrer comment les deux relations d'ordre  $<$  et  $\leq$  entre  $(a_n a_{n-1} \dots a_{\frac{n}{2}+1})_2$  et  $(b_n b_{n-1} \dots b_{\frac{n}{2}+1})_2$  et entre  $(a_{\frac{n}{2}} a_{\frac{n}{2}-1} \dots a_1)_2$  et  $(b_{\frac{n}{2}} b_{\frac{n}{2}-1} \dots b_1)_2$  permettent de calculer les relations d'ordre entre  $(a_n a_{n-1} \dots a_2 a_1)_2$  et  $(b_n b_{n-1} \dots b_2 b_1)_2$ .

**Question 2.2** Réaliser un comparateur pour  $n$  bits à l'aide de 2 comparateurs pour  $\frac{n}{2}$  bits (dont vous supposez disposer) et le circuit COMB obtenu précédemment.

On peut de manière récursive appliquer le même principe et construire chacun des comparateurs pour  $\frac{n}{2}$  bits avec 2 comparateurs pour  $\frac{n}{4}$  bits qui eux-même peuvent se construire avec .....

**Question 2.3** Décrire comment selon ce principe, on peut obtenir un comparateur pour  $n$  bits en utilisant uniquement des circuits  $Comp_{<}$ ,  $Comp_{\leq}$  et  $COMB$ .

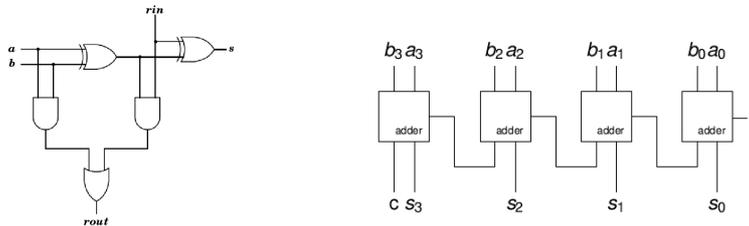
**Question 2.4** Calculer le nombre de circuits  $Comp_{<}$ ,  $Comp_{\leq}$  et  $COMB$  et donc, le nombres de portes logiques utilisé(e)s dans votre comparateur récursif pour  $n$  bits.

**NB :** le terme général de la suite  $u_n = \begin{cases} 2u_{\frac{n}{2}} + c & \text{si } n > 1 \\ k & \text{si } n = 1 \end{cases}$  est  $u_n = n * k + (n - 1) * c$ .

**Question 2.5** Calculer, en fonction de  $n$  et de  $t$  le temps de passage d'une porte logique, au bout de combien de temps les résultats de la comparaison pour  $n$  bits sont disponibles sur la sortie.

## 2 Additionneurs $n$ -bits

On rappelle qu'un additionneur 1 bit est donné par le circuit ci-dessous à gauche.



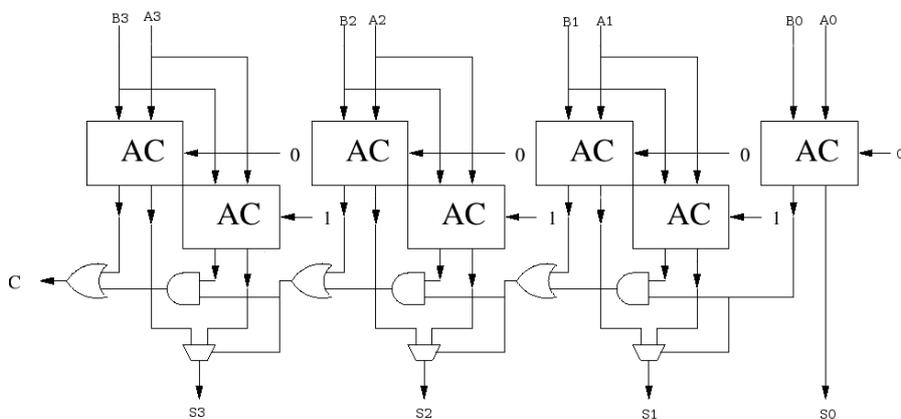
**Exercice 3** Réaliser un additionneur 1 bit à l'aide de deux demi-additionneurs 1 bit.

Au dessus à droite on trouve le circuit additionneur 4 bits vu en cours; il est appelé additionneur à propagation de retenue, puisque la retenue se propage d'un additionneur à l'autre.

**Exercice 4** Dans le cas de l'additionneur 4 bits, étendre le dispositif pour tester la validité du résultat dans le cas d'entiers codés à complément à 2, ie tester s'il y a dépassement de capacité.

**Exercice 5** En supposant que le temps de franchissement d'une porte est de  $t$ , au bout de combien de temps le résultat de l'opération est disponible pour un additionneur  $n$  bits ?

**Exercice 6 (Additionneur à sélection de retenue)** L'idée de l'additionneur à sélection de retenue est de coupler deux séries d'additionneurs, l'une avec pour retenue la valeur 1 et l'autre la valeur 0. Une fois les retenues calculées, elles sont utilisées pour sélectionner le bon résultat des additions. Voici le schéma d'un additionneur à sélection de retenue



**Question 6.1** En supposant que le temps de franchissement d'une porte est de  $t$ , au bout de combien de temps le résultat de l'opération est disponible pour un additionneur  $n$  bits ?