

TD7 – Circuits logiques structurés

1. Compérateurs

Ce type de circuit (figure 1) compare des entiers naturels codés suivant le code usuel (base 2), sur n bits.

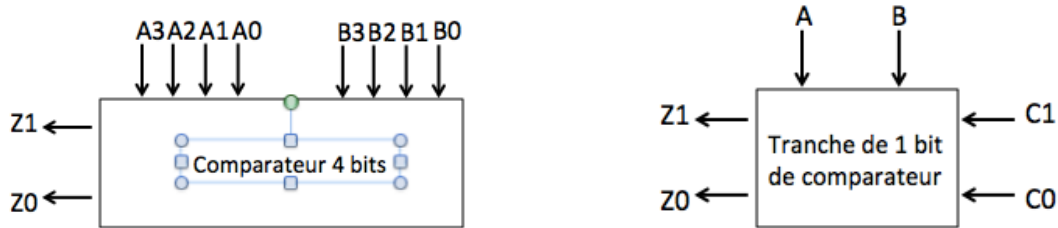


Figure 1 : Compérateurs

- Décomposer la fonction logique $a_3a_2a_1a_0 < b_3b_2b_1b_0$ en partant des bits de poids fort.
- Le comparateur 4 bits est basé sur la propagation des résultats des poids forts vers les poids faibles. Donner les tables de vérité et une expression simplifiée en NAND et inverseurs de la tranche 1 bit de comparateur en supposant le codage suivant : $Z1Z0=00$ si égalité, 01 si $A < B$ et 10 si $A > B$.
- Donner le temps de retard exprimé en nombre de temps de propagation de portes NAND, noté t_p , pour le comparateur 4 bits réalisé avec 4 tranches de 1 bit. Même question pour un comparateur 8 bits et pour un comparateur 16 bits.

2. Réalisation d'un incrémenteur 4 bits

Le circuit de la figure 2 est appelé *demi-additionneur* ; il est constitué d'une porte XOR et d'une porte ET. Le temps de propagation d'une porte ET est t_{pET} et celui d'une porte XOR est t_{pXOR} .

On veut réaliser un incrémenteur 4 bits (figure 3). Les nombres X ($e_3e_2e_1e_0$) et Y ($s_3s_2s_1s_0$) sont des entiers naturels sur 4 bits.

Lorsque $inc = 0$, on a $Y=X$.

Lorsque $inc = 1$, on a $Y=X+1$.

$C = 1$ lorsque l'incréméntation fournit un résultat incorrect ($inc=1$ et $X=15$).

- Donner un schéma de réalisation de l'incrémenteur 4 bits avec des demi-additionneurs.
- Donner le temps de retard entre les entrées et la sortie S_3 d'une part, entre les entrées et deb d'autre part, lorsque $inc=1$ en fonction de t_{pET} et de t_{pXOR} .

Temps de retard retenue : $4t_{pXOR}$

Temps de retard S_3 : $3t_{pXOR} + 1t_{pET}$

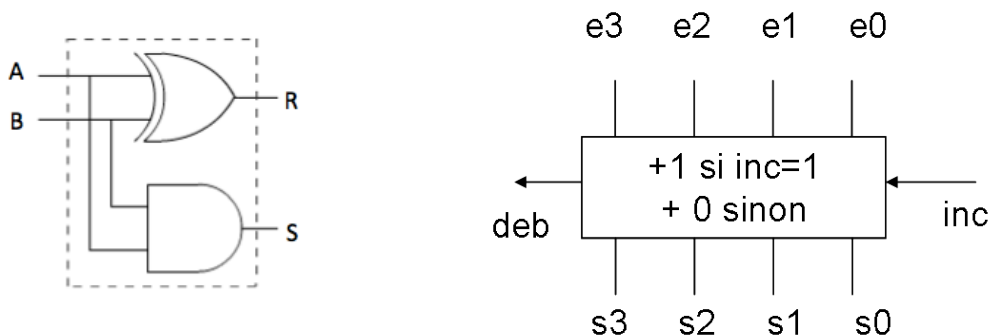


Figure 2 : demi-additionneur

Figure 3 : incrémenteur

3. Additionneur BCD (optionnel)

En notation BCD (décimal codé binaire, chaque chiffre décimal est représenté sur un quartet ; par exemple 17 est représenté par 00010111)

On veut réaliser un additionneur modulo 10 pour additionner deux chiffres BCD : pour deux chiffres BCD $A=A_3A_2A_1A_0$ et $B=B_3B_2B_1B_0$, la sortie de ce circuit est $S=S_3S_2S_1S_0$ et sa retenue de sortie est r_b , avec $S = A+B$ modulo 10 et $r_s=1$ si $A+B \geq 10$.

L'addition modulo 10 de deux nombres $A=A_3A_2A_1A_0$ et $B=B_3B_2B_1B_0$ peut être réalisée en deux étapes :

- Ajouter A et B (addition binaire).
- Si le résultat est supérieur ou égal à 10_{10} ajouter 6 (il y a alors retenue pour l'étage suivant), sinon ajouter 0. Ne pas tenir compte du débordement du 2ème étage d'additionneur.

On dispose d'additionneurs binaires 4 bits dont les entrées sont $X=X_3X_2X_1X_0$ et $Y=Y_3Y_2Y_1Y_0$ et la retenue d'entrée r_e . Les sorties sont $Z_3Z_2Z_1Z_0$ et la retenue de sortie r_s

1. On injecte A et B dans l'additionneur 4 bits. Quelles sont les valeurs possibles de $A+B$ (valeur décimale); quelles sont les valeurs de Z et r_s correspondantes ; quelles doivent être les valeurs de S et r_b ?
2. Donner le schéma logique de l'additionneur BCD réalisé à partir d'additionneurs binaires 4 bits et de portes NAND.

Lorsqu'on ajoute deux nombres décimaux, avec retenue d'entrée, la somme obtenue est comprise entre 0 et 19 ($9+9+1$). A la sortie d'un additionneur 4 bits (modulo 16), le résultat est incorrect si la somme est entre 10 et 19, soit lorsqu'il y a une retenue de sortie (16 à 19), soit lorsqu'il n'y a pas de retenue de sortie et que la sortie est entre 10 et 15.

La correction (ajouter +6) a donc lieu lorsque $r_s = 1$ ou ($r_s=0$ et $Z_3(Z_2+Z_1)$) Correction = $r_s + Z_3(Z_2+Z_1)$

La correction à apporter consiste à ajouter 0000, où $C = r_s + Z_3(Z_2+Z_1)$, soit 0110 ou 0000 selon la valeur de C.

