Architecture des ordinateurs

TD 4 : Circuits combinatoires

Arnaud Giersch, Benoît Meister et Frédéric Vivien

1. Exprimer la fonction **xor** comme un produit de sommes et réaliser le circuit logique correspondant. Même question en exprimant **xor** comme une somme de produits.
2. La fonction **nand** formant un groupe logique complet, réaliser, uniquement avec des portes **nand**, les circuits logiques **not, and, or** et **xor** (les formules sont rappelées ci-dessous).
   - $\text{not}(A) = \text{nand}(A, A)$
   - $\text{and}(A, B) = \text{nand}(\text{nand}(A, B), \text{nand}(A, B))$
   - $\text{or}(A, B) = \text{nand}(\text{nand}(A, A), \text{nand}(B, B))$
   - $\text{xor}(A, B) = \text{nand}(\text{nand}(\text{nand}(A, A), B), \text{nand}(A, \text{nand}(B, B)))$
3. Réaliser un circuit logique qui implémente la fonction F.

$$F = (A + B + C) \cdot (A + \overline{B} + \overline{C}) \cdot (\overline{A} + \overline{B} + \overline{C})$$

4. Un *générateur de parité impaire* est une fonction qui retourne 1 si le nombre de bits à 1 est impair et 0 sinon. Définir cette fonction pour un mot de 4 bits. Donner un circuit logique implémentant cette fonction.
5. Rappeler les principes d'un demi-additionneur puis d'un additionneur complet. Déduire de ces principes un circuit logique qui implémente le complément à 2 sur n bits.
6. Réaliser un circuit pour un décrémenteur à n bits.
7. **Le soustracteur**
   (a) Réaliser un demi-soustracteur (table de vérité et circuit).
   (b) Réaliser un soustracteur binaire complet (ou *étage de soustracteur*) selon deux modes :
      i. avec deux demi-soustracteurs ;
      ii. avec un demi-additionneur et un demi-soustracteur.
   (c) Réaliser un soustracteur parallèle pour mots de 8 bits.
8. **Le (dé)multiplexeur**

   Un *multiplexeur* est un circuit logique qui dispose de 2^n entrées, d'une unique sortie et de n lignes de sélection. Son principe de fonctionnement consiste à connecter, selon la configuration binaire présente sur les n lignes de sélection, l'une des entrées à la sortie. Les n lignes de sélection différencient 2^n configurations binaires, chacune de ces configurations correspondant à l'entrée du multiplexeur qui doit être connectée à la sortie.

   Un *démultiplexeur*, pour sa part, est un circuit logique qui dispose d'une unique entrée, de 2^n sorties et de n lignes de sélection. Son principe de fonctionnement, à l'inverse de celui du multiplexeur, consiste à connecter, selon la configuration binaire présente sur les lignes de sélection, l'entrée à l'une des sorties.

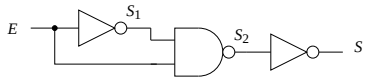
   (a) Réaliser un multiplexeur à quatre voies (c'est-à-dire un multiplexeur à quatre entrées).
   (b) Réaliser un démultiplexeur à quatre voies (c'est-à-dire un démultiplexeur à quatre sorties).
9. **Les hasards logiques**

   Le *temps de passage d'une porte logique* est la durée entre l'instant où les signaux sont appliqués à l'entrée et celui où leur effet se répercute en sortie. Jusqu'à présent, ce temps de passage a été ignoré dans un souci de simplification. Toutefois, le temps de passage d'une porte logique n'est jamais nul (de l'ordre de 5 à 25 ns).

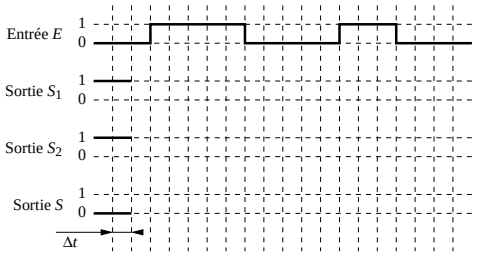
Si un étage logique est construit à l'aide de portes logiques (c'est-à-dire si la sortie d'une porte logique attaque l'une des entrées de la porte logique suivante) alors le temps de passage de l'étage est au moins égal à la somme des temps de passage des portes logiques qui le composent : dans ce cas, les temps de passage s'ajoutent. Il en résulte qu'un changement des données en entrée d'un montage, non seulement mettra un certain temps à se répercuter en sortie, mais pourra en plus provoquer des changements d'état (impulsions) non souhaités à la sortie. De telles impulsions parasites sont appelées *hasards logiques*.

(a) Mise en évidence d'un hasard logique.

i. Exprimer la valeur de la sortie S du circuit ci-dessous en fonction de son entrée E.

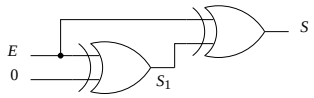


ii. Compléter le chronogramme suivant de ce circuit (on considère que toutes les portes logiques mises en jeu ont un même temps de passage Δt).

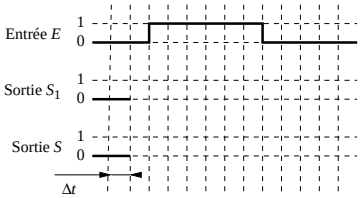


(b) Exemples de mise à profit des hasards logiques : *détecteur de transitions*.

i. Exprimer la valeur de la sortie S du circuit ci-dessous en fonction de son entrée E.



ii. Compléter le chronogramme suivant de ce circuit (on considère que toutes les portes logiques mises en jeu ont un même temps de passage Δt).



iii. Réaliser un détecteur de transitions pour lequel la durée des impulsions en S est de $3\Delta t$.

(c) Réaliser un *détecteur de front montant*, c'est-à-dire un détecteur de transitions qui ne répond que lorsque le signal d'entrée passe d'un niveau bas à un niveau haut.

(d) Réaliser un *détecteur de front descendant*, c'est-à-dire un détecteur de transitions qui ne répond que lorsque le signal d'entrée passe d'un niveau haut à un niveau bas.