

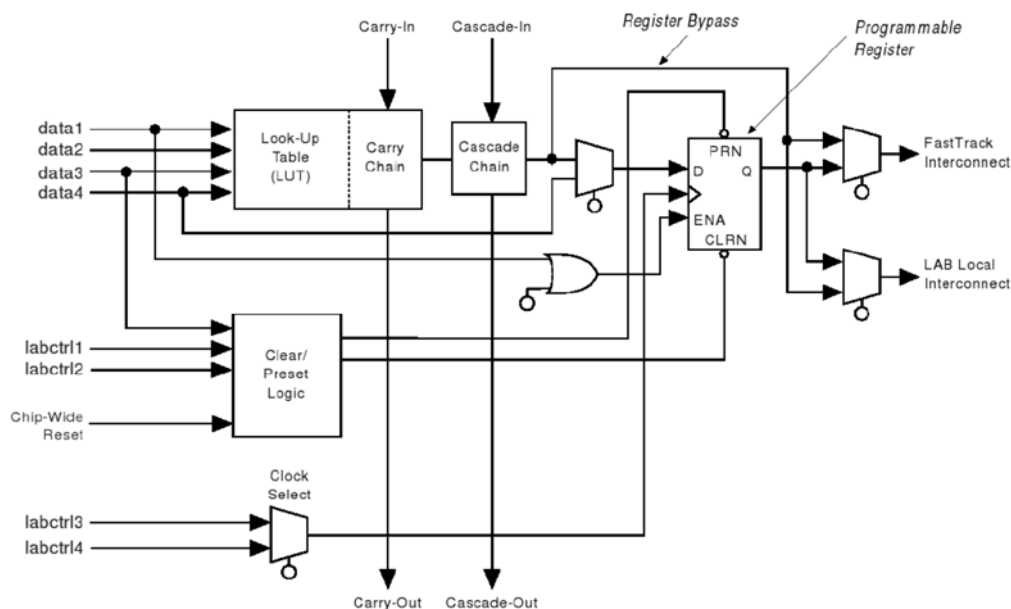
## Module EN 201 (les circuits FPGA)

### TD n°4 – synthèse d'additionneur dans un circuit FPGA

#### Introduction

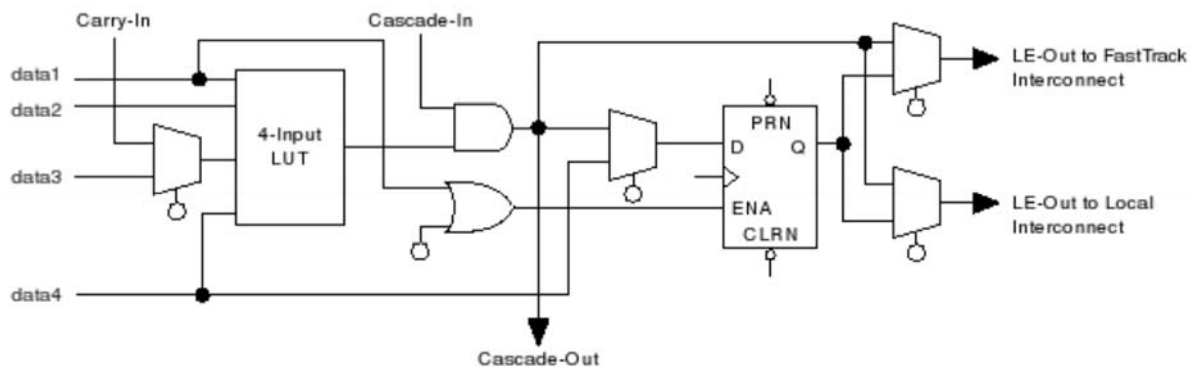
Les éléments logiques LE correspondent à la plus fine granularité pour le circuit FPGA. Une structure classique est présentée sur la figure suivante. Ils sont composés d'une table LUT à quatre entrées permettant de réaliser des fonctions combinatoires à quatre variables. Le second élément important est un registre programmable, autorisant avec la table LUT la gestion de fonctions de logique synchrone. Enfin, des chemins spécifiques sont intégrés. Ils sont dédiés à la propagation des retenues des opérations arithmétiques de base et à l'utilisation en cascade de plusieurs LE pour réaliser des fonctions comportant plus de quatre entrées. Il faut remarquer que ces cellules ont quatre modes distincts de fonctionnement :

- mode normal ;
- mode arithmétique ;
- mode compteur / décompteur ;
- mode compteur avec remise à zéro synchrone.



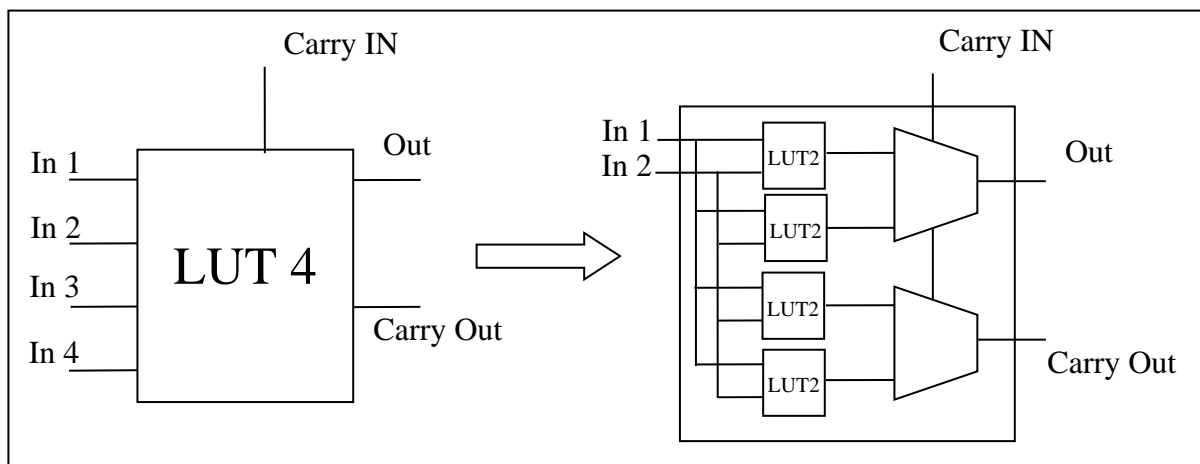
Architecture d'une cellule logique

**Le mode de fonctionnement normal** propose une table LUT à quatre entrées permettant de créer des fonctions combinatoires classiques. Le schéma structurel d'un élément logique fonctionnant dans ce mode est présenté sur la figure suivante. La présence d'un chemin d'interconnexion dédié et rapide entre les LE d'un même niveau permet de réaliser des fonctions combinatoires plus complexes utilisant plus de quatre entrées. Ce chemin est repéré par les entrées *Cascade In* et *Cascade Out* sur le schéma de la figure suivante. Le mode «normal» ne propose pas de chemin optimisé pour les propagations de retenues. Il est spécifiquement utilisé dans le cas de fonctions combinatoires simples et isolées.



*Élément logique en mode de fonctionnement «normal»*

**Le mode arithmétique** propose quatre LUTs à deux entrées au lieu de la seule LUT à quatre entrées du mode normal. Cette architecture est particulièrement indiquée pour la réalisation d'additionneurs ou de fonctions équivalentes. Ce mode comble l'une des lacunes du mode «normal» en proposant un chemin adapté à la propagation rapide de retenues. En contrepartie, les fonctions logiques que le LE est capable de traiter sont limitées à deux entrées de données.



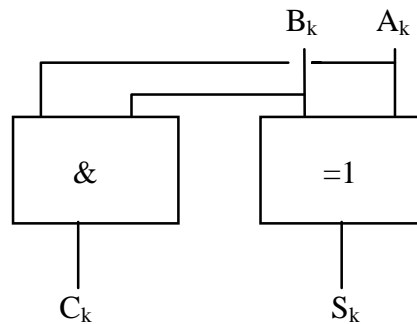
*Élément logique en mode de fonctionnement arithmétique*

**Question 1 : synthèse d'un additionneur binaire complet à base de LUT4**

Le demi-additionneur binaire est un circuit qui prend en entrée 2 nombres de 1 bit ( $A_k$  et  $B_k$ ) et génère leur somme ( $S_k$ ) et une retenue ( $C_k$ ).

$A_k$	$B_k$	$S_k$	$C_k$
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Le schéma ci-après représente la structure d'un "demi-additionneur binaire" :



A partir des informations ci-dessus, on souhaite réaliser un additionneur binaire complet qui prenne en compte une retenue ( $C_{k-1}$ ) générée à partir d'un étage précédent.

1-1 Etablir la table de vérité donnant la somme  $S_k$  et la retenue sortante  $C_k$  en fonction de  $A_k$ ,  $B_k$  et  $C_{k-1}$ .

1-2 Dans un premier temps nous considérons que l'unique mode de fonctionnement des éléments logiques du FPGA est le mode de fonctionnement normal. Chacun des éléments logiques dans ce mode comporte une LUT à 4 entrées et 1 sortie. Dans ce cas, dessiner un schéma précis à base de LUT à 4 entrées réalisant l'additionneur binaire complet. Sur ce schéma entourer les éléments logiques.

1-3 Quel est l'équivalent en capacité mémoire distribuée (en octets) utilisé dans ce cas ?

**Question 2 : synthèse d'un additionneur binaire complet à base de LUT2**

2-1 Tracer les tableaux de Karnaugh correspondants à la table de vérité de la question précédente et exprimer les fonctions sous forme réduite.

2-2 On souhaite réaliser un additionneur complet à partir de deux demi-additionneurs et d'un bloc combinatoire  $g$ . Dessinez le schéma de l'additionneur complet.

2-3 Nous considérons maintenant que les éléments logiques disposent d'un mode de fonctionnement arithmétique. Chacun des éléments logiques dans ce mode comporte quatre LUTs à 2 entrées et 1 sortie (les entrées des quatre LUTs sont les mêmes) et deux multiplexeurs 2 vers 1 connectés aux sorties des LUTs. Dans ce cas, dessiner un schéma précis utilisant ces éléments logiques pour réaliser l'additionneur binaire.

2-4 Donner la table de vérité de chacune des LUT utilisées.

2-5 Quel est l'équivalent en capacité mémoire distribuée (en octets) utilisé dans ce cas ?

**Question 3 : synthèse d'un additionneur/soustracteur  $n$  bits**

3-1 En utilisant le module de l'additionneur complet, donnez le synoptique d'un additionneur de  $n$  bits capable d'effectuer l'opération  $S = A + B$ , (+ représente ici l'opérateur d'addition arithmétique).

3-2 On souhaite transformer le montage précédent en un additionneur / soustracteur. On rappelle que dans la représentation en complément à 2,  $A - B = A + (-B) = A + \overline{B} + 1$ . Proposer le schéma d'un additionneur / soustracteur capable de manipuler des nombres de 4 bits codés dans le système de représentation du complément à 2. Cet additionneur / soustracteur possèdera une entrée de commande *SOUSTRAC* qui sera utilisée comme suit :

- *SOUSTRAC* = 0, fonctionnement en additionneur
- *SOUSTRAC* = 1, fonctionnement en soustracteur

3-3 En justifiant la réponse, indiquer le nombre minimum de LUT4 et l'équivalent en capacité mémoire distribuée (en octets) nécessaires pour réaliser un additionneur / soustracteur binaire prenant en entrée deux mots de 48 bits.

3-4 En justifiant la réponse, indiquer le nombre minimum de LUT2 et l'équivalent en capacité mémoire distribuée (en octets) nécessaires pour réaliser un additionneur / soustracteur binaire prenant en entrée deux mots de 48 bits.

3-5 De quelle alternative disposons-nous avec les circuits FPGA ?

***Question 4 : problème de débordement***

Le montage précédent ne traite pas les problèmes de débordement de capacité. A partir de la table de vérité proposée en question 1.1, donner l'équation de fonctionnement d'un bit de sortie  $OV$  ( $OV = \text{overflow}$ ) en fonction des retenues  $C_{n-1}$  et  $C_{n-2}$ . Ce bit  $OV$  sera positionné à "1" lorsqu'il y aura débordement de capacité en arithmétique signée.