Synthèse et implémentation de circuits arithmétiques sur FPGA

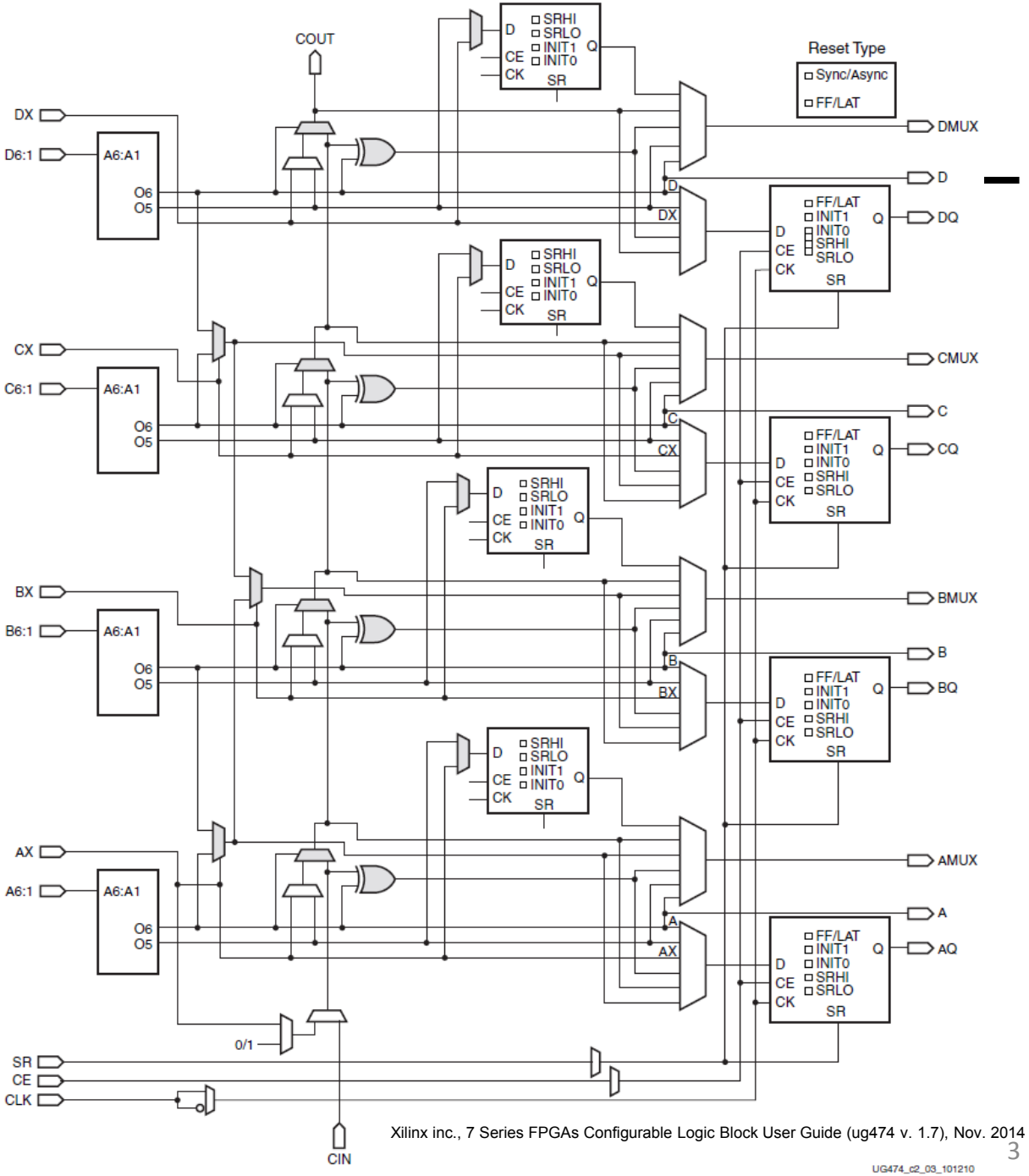
Synthèse et implémentation de circuits arithmétiques sur FPGA
Sujets de ce thème

- Ressources de calcul d'une tranche de type L
- Implémentation de l'addition et de la soustraction dans une tranche
- Valeur absolue
- Multiplication par une constante
- Blocs DSP48E pour l'implémentation de la multiplication et de la multiplication-accumulation
- Décalage et modulo

FPGA de Xilinx série 7 : tranche de type L (SLICEL)

- Une tranche de type L comprend:
  - Quatre tables de correspondance (Look-up Table – LUT) à 6 entrées (A6:A1) et deux sorties O6 et O5.
  - Trois multiplexeurs (en gris, verticaux) pour réaliser des fonctions logiques de 7 ou 8 entrées.
  - Des portes logiques pour l'addition rapide.
  - Huit éléments à mémoire.
  - Des multiplexeurs programmables (en blanc) pour router les signaux.

Comment faire pour implémenter des fonctions arithmétiques avec ce circuit?

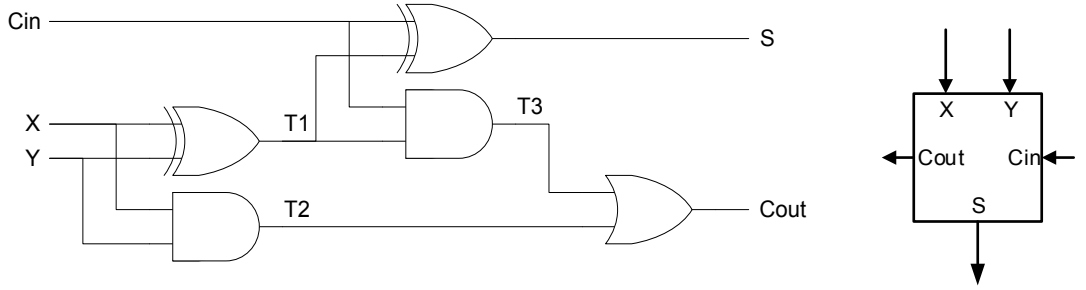


Xilinx inc., 7 Series FPGAs Configurable Logic Block User Guide (ug474 v. 1.7), Nov. 2014

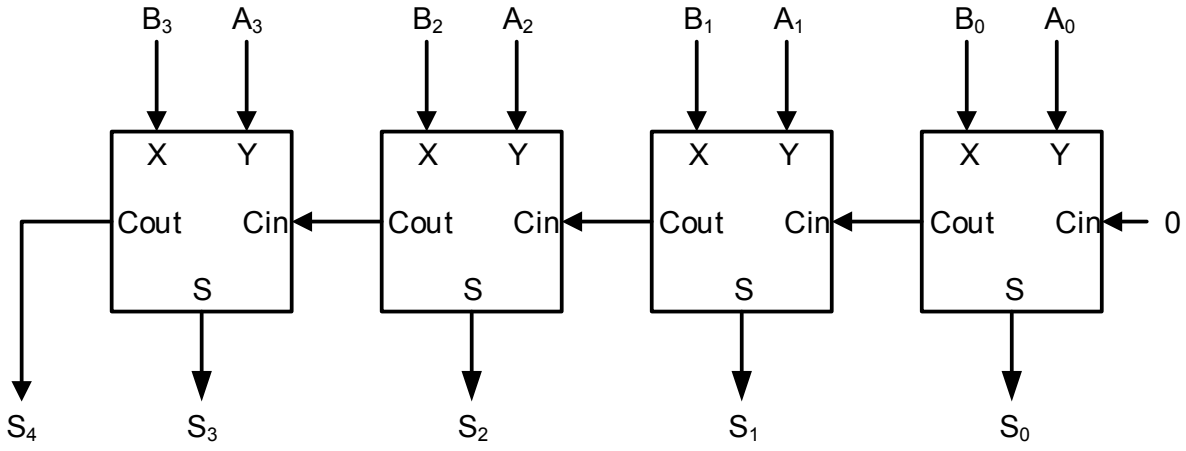
UG474_c2_03_101210

Architecture d'un additionneur

- La configuration d'additionneur la plus simple est l'additionneur à retenue propagée (*ripple carry adder*).
- Le module de base est un compresseur 3:2 qui prend trois bits en entrée et encode leur somme sur 2 bits.
- Une cascade de compresseurs 3:2 réalise une addition à plusieurs bits.



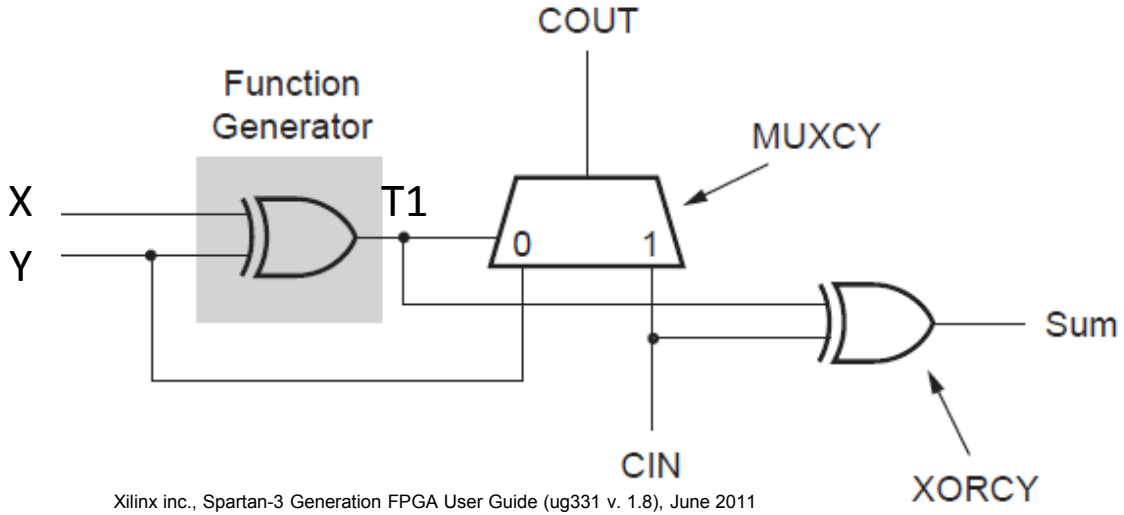
Cin	X	Y	Cout	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



	0110
9	1001
+ 3	0011
12	01100

Implémentation de l'addition dans une tranche d'un FPGA

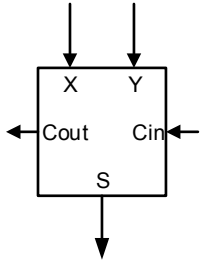
- Les FPGA de Xilinx implémentent le module du compresseur avec un circuit spécial dédié en reformulant les équations du compresseur 3:2.
- La première porte OUX est réalisée dans une LUT.
- Le multiplexeur et la deuxième porte OUX font partie du circuit dédié de chaque tranche.
- Les signaux de retenue sont propagés à l'intérieur d'une tranche et de tranche en tranche.



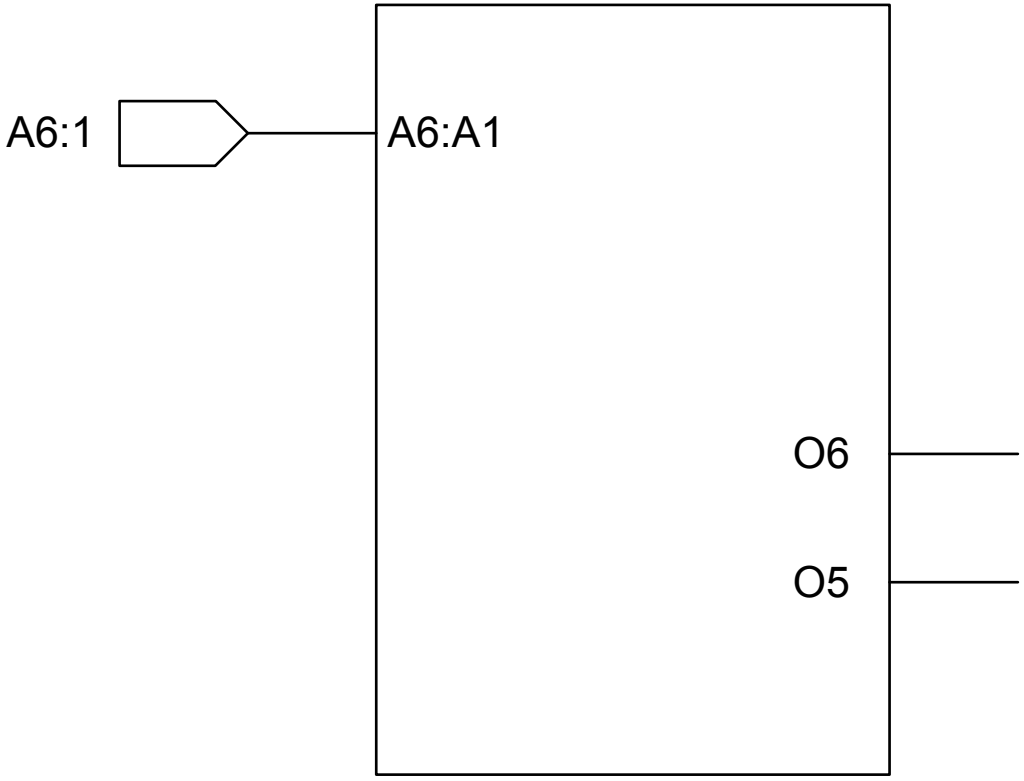
Xilinx inc., Spartan-3 Generation FPGA User Guide (ug331 v. 1.8), June 2011

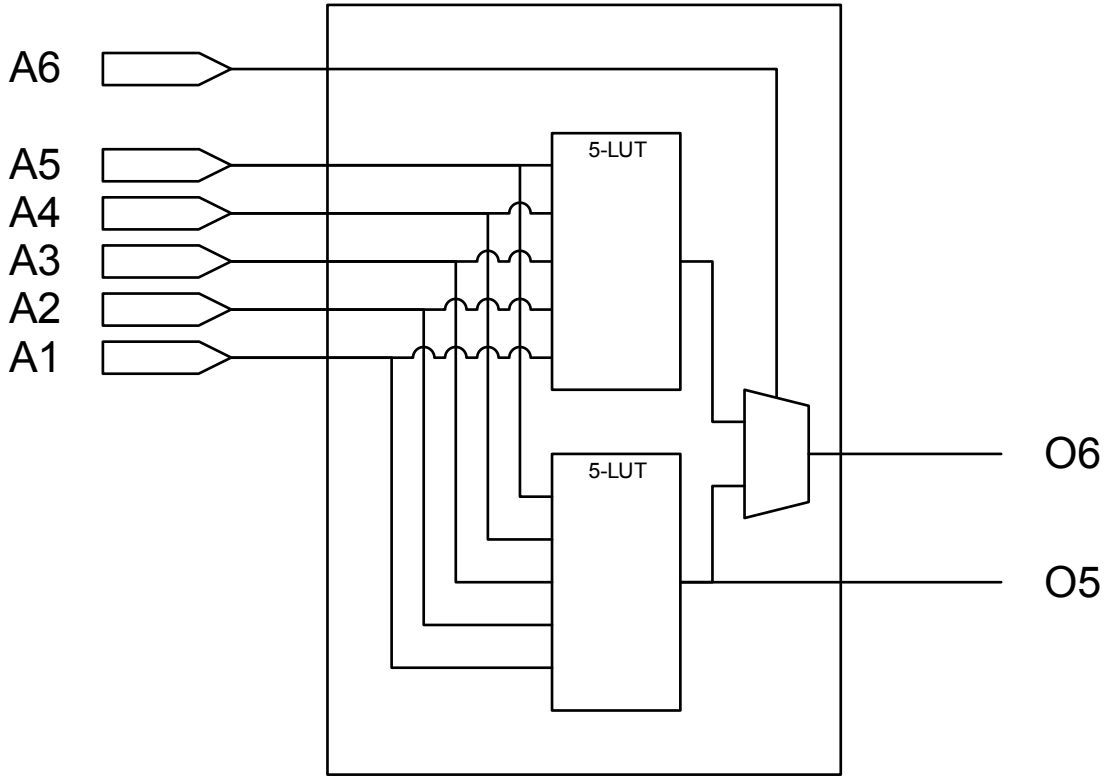
Cin	X	Y	T1 = X xor Y	Cout	S
0	0	0	0	0	0
0	0	1	1	0	1
0	1	0	1	0	1
0	1	1	0	1	0
1	0	0	0	0	1
1	0	1	1	1	0
1	1	0	1	1	0
1	1	1	0	1	1

$$T1 = X \text{ xor } Y$$
$$S = T1 \text{ xor } Cin$$
$$Cout = T1 \cdot Cin + T1' \cdot Y$$
$$= T1 \cdot Cin + T1' \cdot X$$



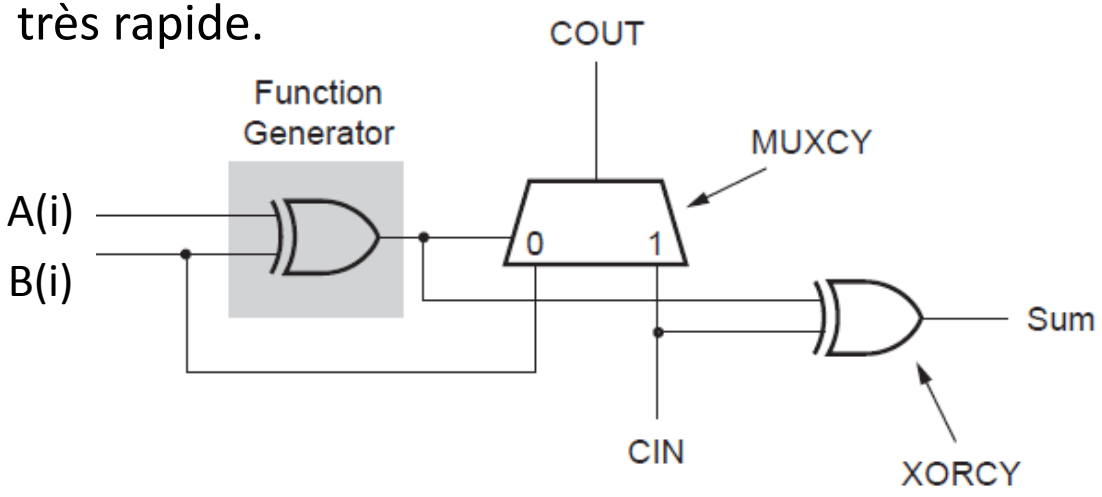
Fonctionnement interne des LUT à 6 entrées A6:A1 et deux sorties O6 et O5





Implémentation de l'addition dans une tranche d'un FPGA

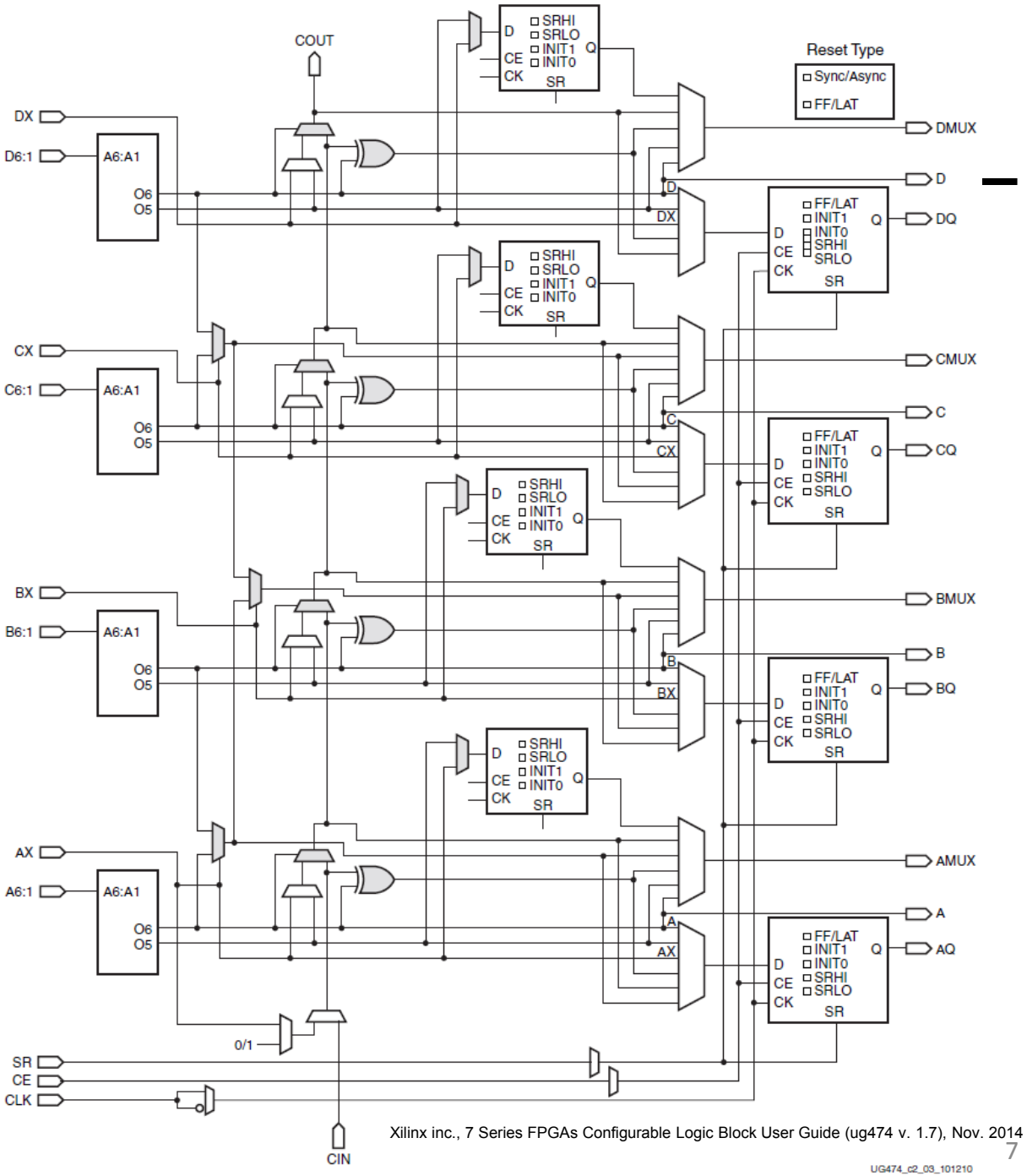
- Chaque tranche peut réaliser l'addition ou la soustraction de deux nombres de 4 bits.
- La configuration pré-connectée et les interconnexions entre tranches rendent le circuit très rapide.



Xilinx inc., Spartan-3 Generation FPGA User Guide (ug331 v. 1.8), June 2011

En VHDL:

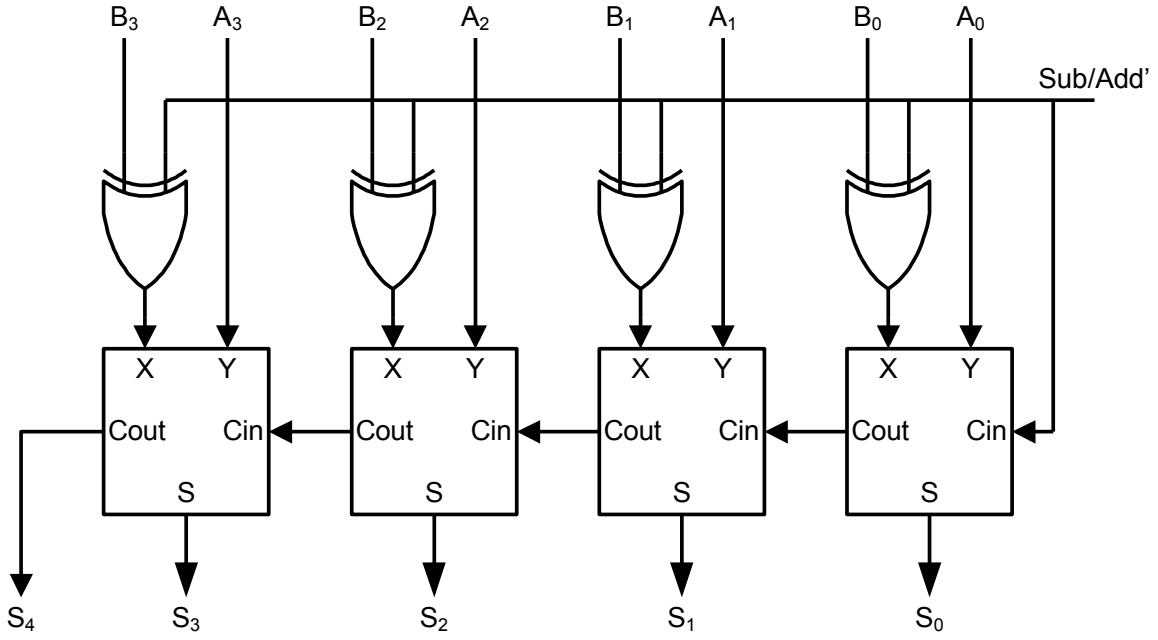
```
signal A, B, sum : signed (W - 1 downto 0);
…
sum <= A + B;
```



Xilinx inc., 7 Series FPGAs Configurable Logic Block User Guide (ug474 v. 1.7), Nov. 2014

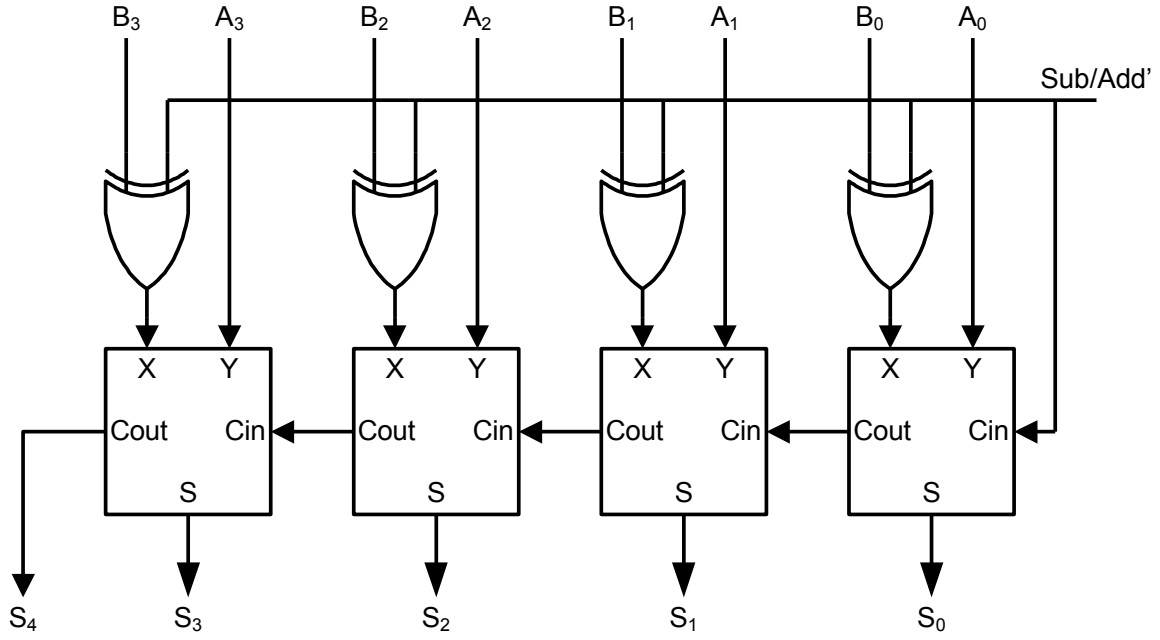
Architecture d'un additionneur-soustracteur

- Nous supposons une représentation à complément à deux.
- La soustraction A – B peut être faite par A + (-B).
- Changer le signe d'un nombre consiste à inverser ses bits et ajouter 1.
- Quand Sub/Add' = 0, une addition est faite.
- Quand Sub/Add' = 1, les bits de B sont inversés et on ajoute une retenue à l'additionneur de poids faible: la soustraction est réalisée.
- L'implémentation sur une tranche de type L est laissée en exercice.



Valeur absolue

- La valeur absolue s'implémente avec un additionneur-soustracteur. Si le bit de signe a une valeur de 1, on fait une inversion des bits et une addition de 1.



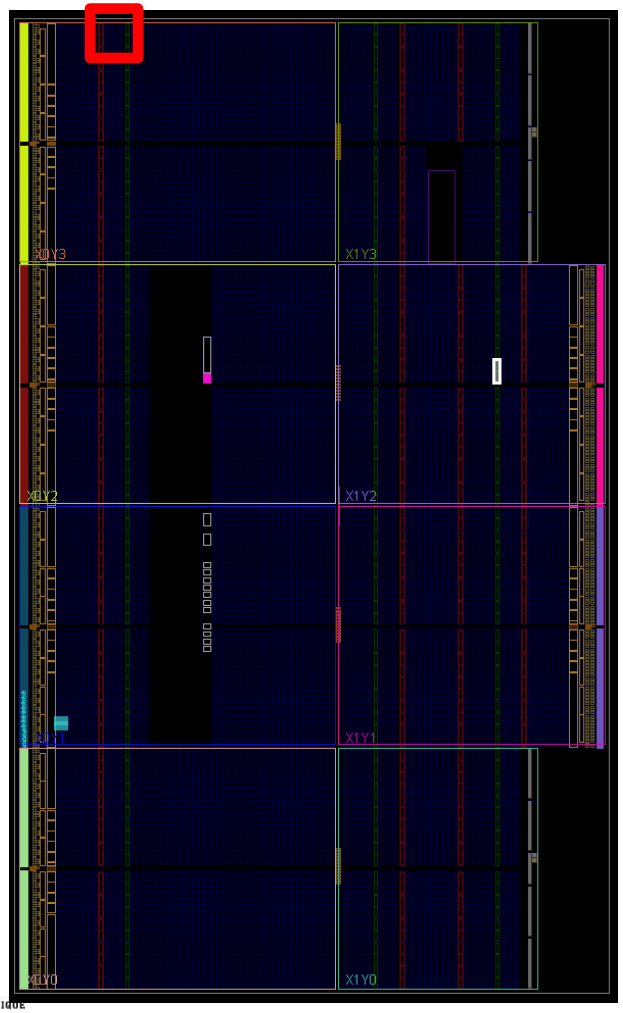
Multiplication par une constante

- Quand un des termes de la multiplication est constant, la technique de la décomposition en produits partiels peut grandement simplifier les calculs.
- Par exemple
  $A \times 10 = A \times (8 + 2) = A \times 8 + A \times 2 = A << 3 + A << 1$
- Le coût de l'opération est donc limité à un seul additionneur.
- Un synthétiseur peut implémenter la multiplication par une constante avec une combinaison d'additions et de décalages.

- On peut exploiter les produits partiels négatifs pour avoir le plus faible coût possible.
- Par exemple:
  $A \times 15 = A \times (8 + 4 + 2 + 1) =$
  $A << 3 + A << 2 + A << 1 + A$
  ce qui prend trois additions.
- Une approche moins couteuse serait:
  $A \times 15 = A \times (16 – 1) = A << 4 – A$, ce qui ne prend qu'une seule addition (ou soustraction).
- Il existe des méthodes pour trouver la représentation d'un nombre utilisant un minimum de chiffres signés.

Architecture du FPGA XC7A100T et tranches de calcul DSP48E1

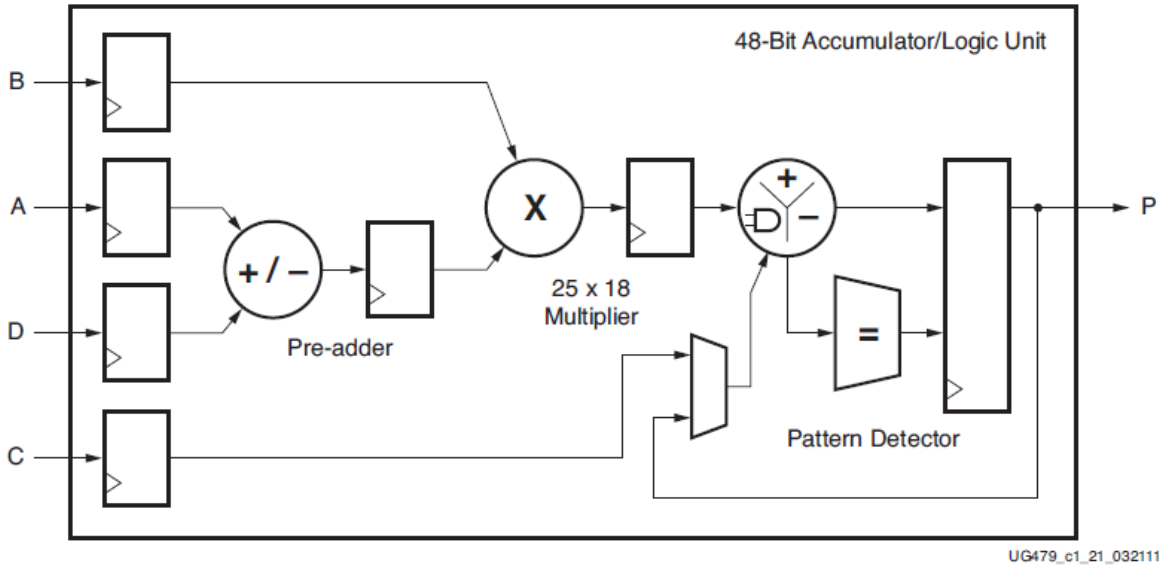
Vue d'ensemble du FPGA XC7A100T



Gros plan sur la région du rectangle rouge

Tranches de calcul DSP48E1 et implémentation de la multiplication

- Les FPGA de série 7 de Xilinx incluent des tranches spécialisées pour les calculs de traitement de signal, les blocs DSP48E1.
- Les FPGA de type XC7A100T en contiennent 240.
- Ces blocs incluent entres autres un multiplicateur de 25 × 18 bits en complément à deux.
- Les multiplications de très petits nombres (e.g. 4 × 4) se font plutôt dans les tranches avec LUT.
- Les multiplications de très grands nombres se font en utilisant plusieurs multiplicateurs de 25 × 18 et en combinant les produits.



Xilinx inc., 7 Series DSP48E1 Slice User Guide (ug479 v. 1.8), November 2014

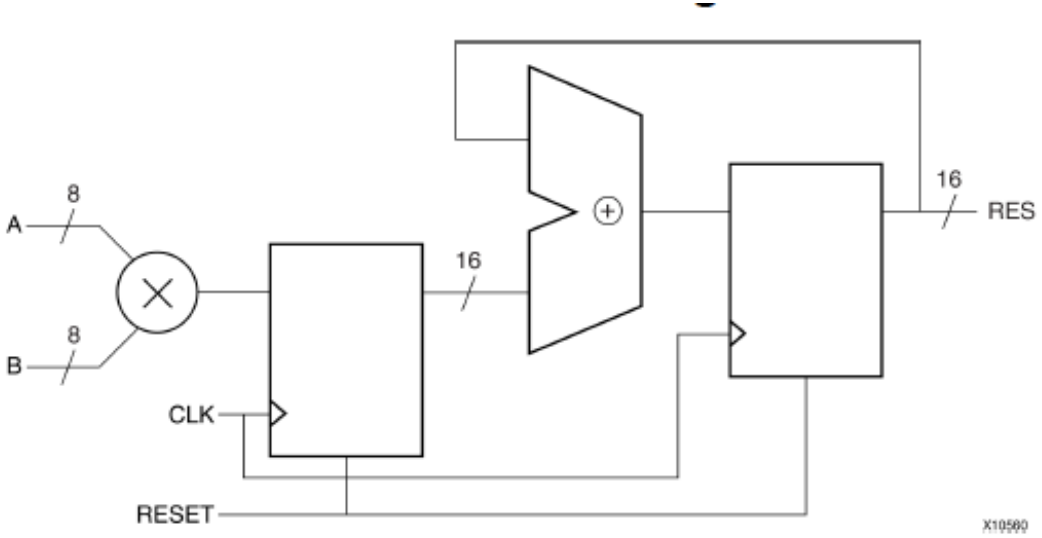
En VHDL:

```
Signal Produit : signed(2 * W – 1 downto 0);
Signal A, B : signed(W – 1 downto 0);
…
Produit <= A * B;
```

Multiplication-accumulation

- Les tranches de calcul DSP48E peuvent être configurés pour des fonctions plus complexes comme la multiplication-accumulation sur 48 bits.
- En anglais on appelle cette opération *Multiply and Accumulate* (MAC).
- La multiplication-accumulation réalise l'opération S = S + A * B
- Elle est très fréquente en traitement de signal pour calculer la convolution ou le produit scalaire de deux vecteurs.

$$y(n) = h(n) * x(n)$$
$$= \sum_{k=0}^{M} h(k)x(n-k)$$
$$= h(0)x(n) + h(1)x(n-1) + h(2)x(n-2) + \ldots + h(M)x(n-M)$$



Xilinx inc., XST user guide (ug627 v. 12.4), December 2010

```
signal S: signed(2 * W - 1 downto 0);
signal A, B : signed(W - 1 downto 0);
…
process (clk)
begin
if rising_edge(clk) then
        S <= S + mult;
        mult <= A * B
end if;
end process;
```

Décalage, modulo et redimensionnement

1. Un décalage par un nombre constant de bits ($A \times 2^N$, $A / 2^N$, shift_left(A, N), shift_right(A, N) ne nécessite aucune logique, seulement une sélection de fils.
2. Le décalage vers la gauche par un nombre arbitraire de bits $N \leq 17$ s'implémente dans une tranche DSP48E en fixant une des entrées à 2^N.
3. Le décalage vers la droite par un nombre arbitraire de bits s'implémente aussi avec une tranche DSP48E. Voir les exercices en classe.
4. Le modulo et le redimensionnement (resize(arg, newsize)) ne nécessitent aucune logique, seulement une sélection de fils.

Vous devriez maintenant être capable de …

- Décrire et utiliser les ressources de calcul spécialisées disponibles sur un FPGA: propagation des retenues et blocs DSPs. (B2, B3)
- Effectuer le processus de synthèse et d'implémentation d'opérations arithmétiques simples sur les structures de calcul disponibles sur un FPGA. (B3)

Code	Niveau (http://fr.wikipedia.org/wiki/Taxonomie_de_Bloom)
B1	Connaissance – mémoriser de l'information.
B2	Compréhension – interpréter l'information.
B3	Application – confronter les connaissances à des cas pratiques simples.
B4	Analyse – décomposer un problème, cas pratiques plus complexes.
B5	Synthèse – expression personnelle, cas pratiques plus complexes.