Master 2 - SEM

Concepts avancés d'Architecture

Cours 1.1 : Rappels circuits logiques

Année 2022-2023

Pr R. BOUDOUR

Citation

La culture, c'est ce qui reste quand on a tout oublié !

Glossaire

ASIC : Application Spécific Integrated Circuit
circuit intégré pour application spécifique

SOC : System On Chip
Système sur puce

IP : Intellectual Property (bloc IP ou « IP core ») ou Composant virtuel
propriété intellectuelle

FPGA : Field Programmable Gate Array
réseau de portes programmables

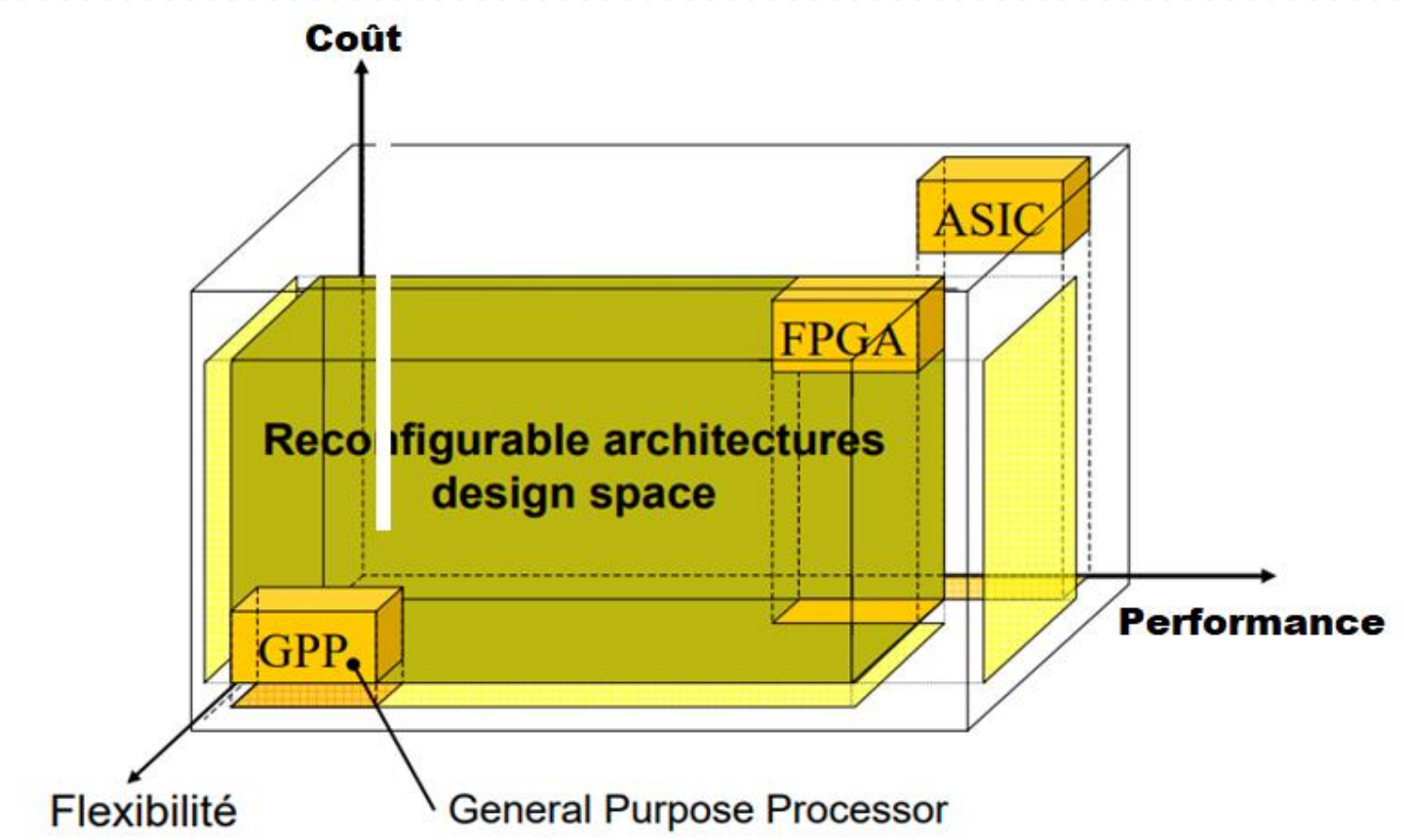
PLD : Programmable Logic Devices

CPLD : Complex Programmable Logic Devices

Rappels : Types de circuit de traitement

- **Microprocesseurs, microcontrôleurs, DSP, SoC**
  - Avantages : universalité ; exécution par une unité centrale (CPU) de programmes (langages C, …) qui permettent de traiter toutes sortes de problèmes
- **ASICs (Application Specific Integrated Circuit)**
  - Avantages : vitesse, robustesse, sécurité, consommation faible, intégration de fonctions analogiques
  - Inconvénients : durée de fabrication, coût de développement élevé (rentable pour de grandes quantités)
- **PLDs (Programmable Logic Device)**
  - Avantages : vitesse (c'est une logique « câblée »), densité élevée, développement rapide (catalogue de composants IP), flexibilité (reconfigurable très rapidement)
  - Deux types de circuits :
    - CPLD (Complex Programmable Logic Device)
    - FPGA (Field Programmable Gate Array)

Rappels : Types de circuit de traitement



Propriétés des circuis CPLD et FPGA

- Un circuit logique programmable est un circuit intégré comportant un très grand nombre de ressources logiques (des fonctions élémentaires combinatoires et séquentielles) dont la plupart des interconnexions sont laissées ouvertes.
- Le concepteur conçoit la fonction globale du circuit à l'aide d'un langage de haut niveau (HDL : Hardware Description Language).
- Un outil de synthèse convertit le code HDL en description normalisée de blocs logiques et d'interconnexions ; un outil de configuration réalise les connexions matérielles.
- Le circuit est reconfigurable.

Applications types de PLD

- Historiquement…
  - Prototypage avant lancement en fabrication d'un ASIC
  - Logique d'appoint (en plus) aux microprocesseurs (gestion d'entrées-sorties, prétraitement, …)
- Et aussi…
  - Systèmes numériques complets embarqués
  - Systèmes de mesure et de traitement rapides pour le test de performance d'autres appareils électroniques

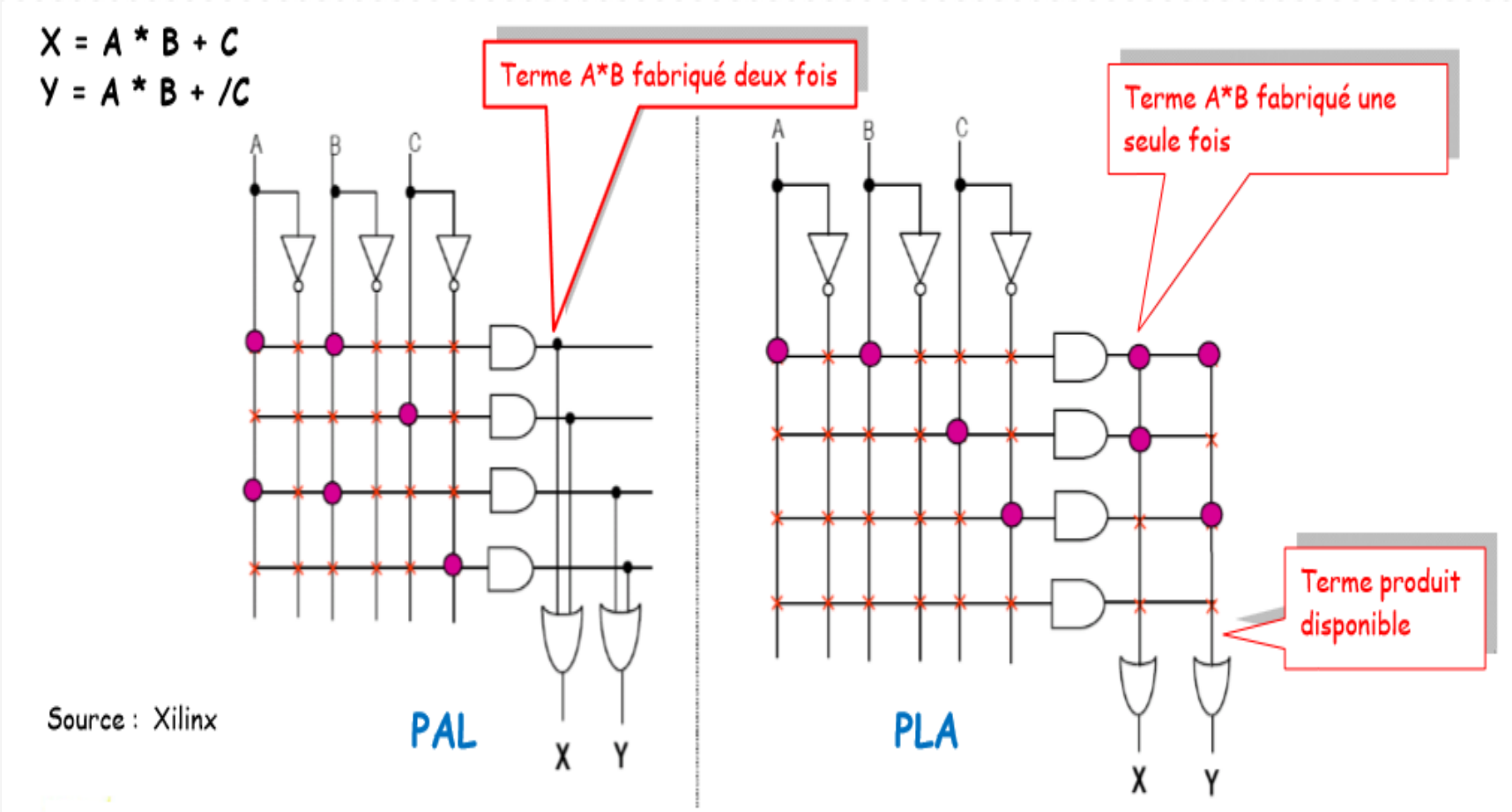
Principaux fabricants de PLD

- Depuis 1982
  - XILINX
  - ALTERA
  - LATTICE
  - ACTEL

CPLD

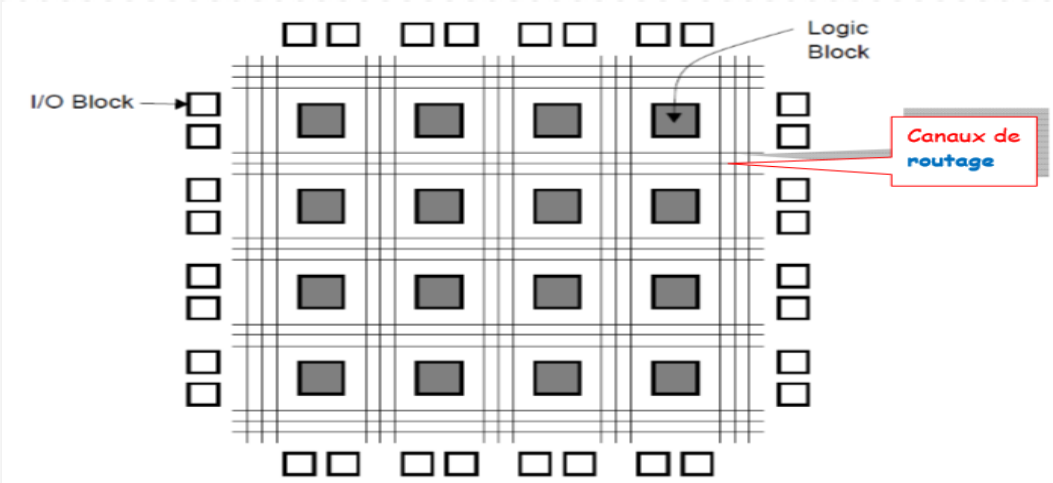
- CPLD : Complex Programmable Logic Device
- Un CPLD est un ensemble de macro-cellules : gros réseaux AND et OR (de type PAL ou PLA) associés à des flip-flops. Les portes logiques et les macro-cellules sont reliées entre elles à l'aide de points de connexions configurables.

Exemple


X = A * B + C
Y = A * B + /C
Terme A*B fabriqué deux fois
Terme A*B fabriqué une seule fois
Terme produit disponible
A
B
C
PAL
PLA
X
Y
Source : Xilinx

FPGA

- FPGA : Field Programmable Gate Array
- Un FPGA est un ensemble constitué de petits blocs logiques : cellules combinatoires et flip-flops et d'un réseau d'interconnexions très souple.

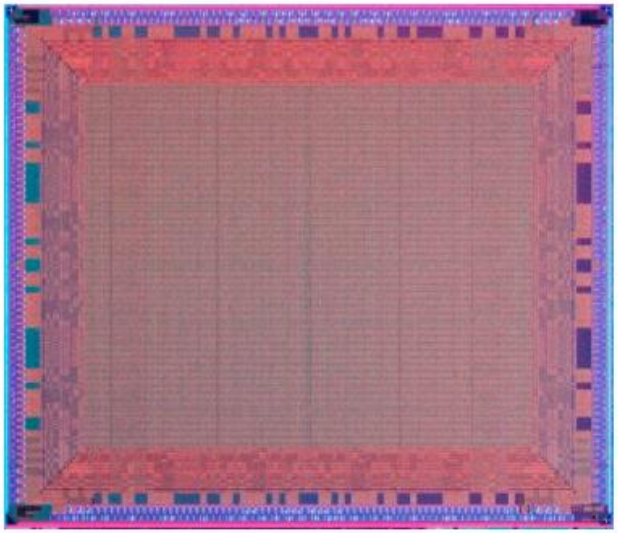


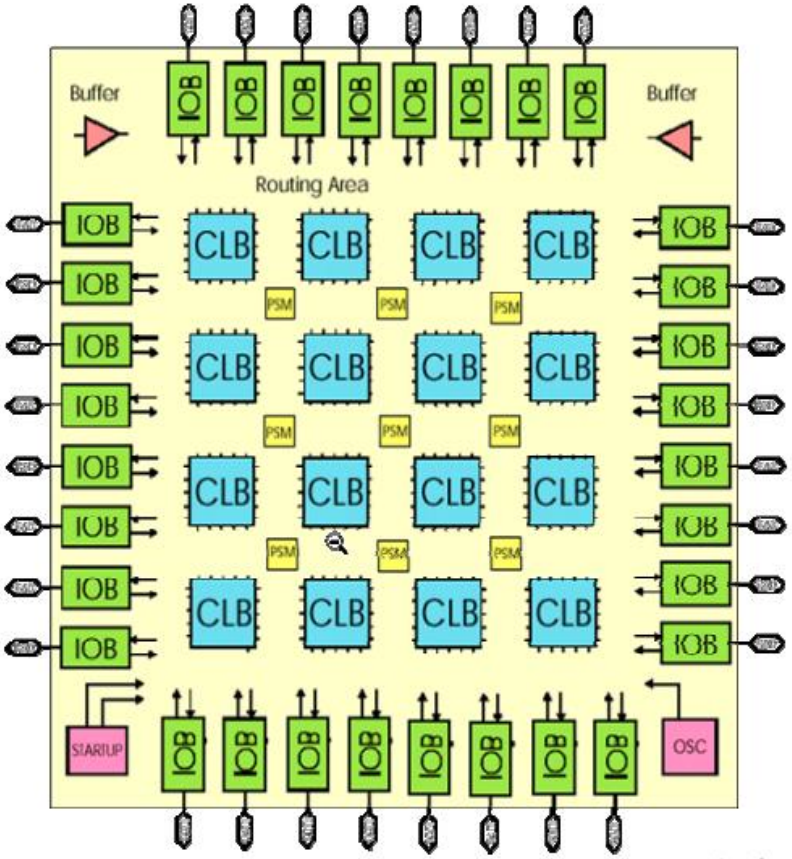
FPGA

Matrice de blocs reconfigurables

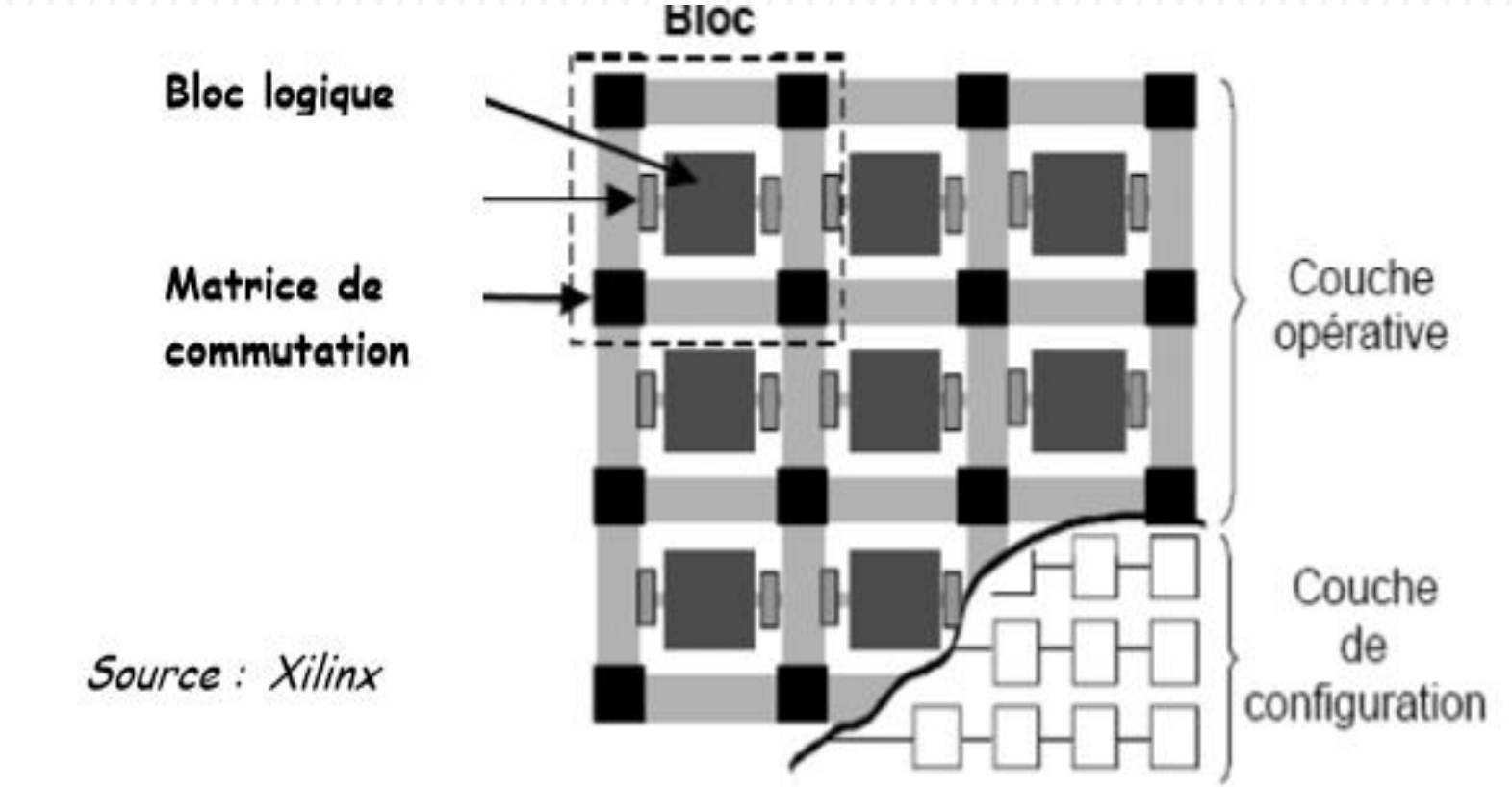
Interconnexion reconfigurable

CLB = LUT + Sequentiel éléments





Plans d'un circuit FPGA



Plans d'un circuit FPGA

- **La couche opérative** comporte une structure régulière de blocs logiques identiques entourés d'un réseau d'interconnexions
- **La couche de configuration** est un ensemble de cellules mémoires (SRAM) qui, d'une part, configurent les blocs logiques et d'autre part, établissent leurs interconnexions (chemins de routage)

Configuration

- Configurer : Programmer (un élément d'un système) pour assurer son fonctionnement selon un certain mode (تهيئة)
- La configuration est le processus qui charge le design dans la SRAM afin de programmer les fonctions des différents blocs et de réaliser leurs interconnexions.
- Une mémoire de configuration contient toutes les informations concernant la programmation des CLB et des IOB ainsi que l'état des connexions.
- Cette configuration est réalisée, à la mise sous tension, par le chargement d'un fichier binaire contenu généralement dans une mémoire flash série qui se trouve à côté du FPGA sur la carte

FPGA

- Les FPGA sont fabriqués en grands volumes, ce qui permet de partager les coûts de conception par tous les clients. Un FPGA coûte quelques euros, pour les plus simples, à quelques centaines (voire milliers) pour les plus complexes.
- Pour des petites séries, les FPGA sont bien plus accessibles que des ASIC. Bien entendu, pour des grands volumes, une solution ASIC est plus rentable :
  - Les mécanismes de configuration des FPGA réduisent les performances par rapport à des solutions ASIC : La vitesse est moindre et la consommation d'énergie supérieure.
  - Mais le caractère reconfigurable permet d'optimiser le système dans le temps.
    - On peut imaginer de mettre rapidement sur le marché une première solution, puis de l'améliorer dans le temps (**Time-To-Market** réduit).

FPGA

- Les FPGA fournissent les très hautes performances du matériel mais avec une utilisation simple et rapide par configuration logicielle.
  - Ainsi, on les retrouve dans de nombreux domaines d'applications : calcul à hautes performances, systèmes embarqués, télécommunications, routeurs de réseaux, réseaux sans fil, traitement du signal et des images, imagerie médicale, vision par ordinateur, cryptologie, dispositifs de sécurité, capteurs biomédicaux, bioinformatique, prototypage de circuits, etc.
- Les FPGA sont devenus le support matériel de base de systèmes sur puce complets ou SoC (**System on Chip**). Toutes les fonctionnalités sont alors intégrées dans le FPGA.

FPGA vs CPLD

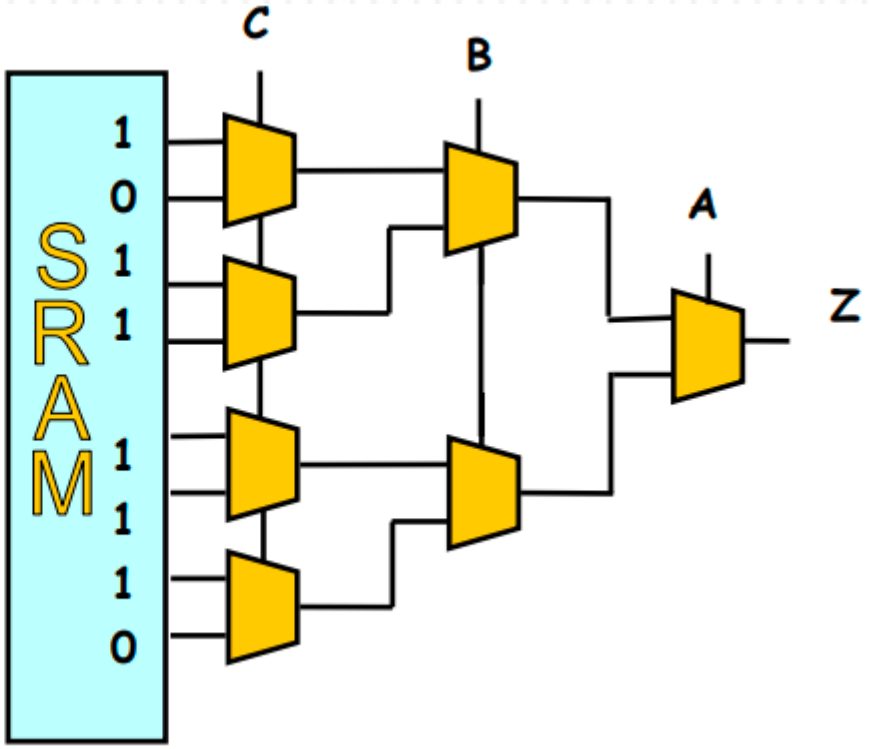
- Les FPGA sont plus complexes (Millions de portes) que les CPLD.
- Les FPGA ont plus de fonctions intégrées de haut niveau que les CPLD.
- Les FPGA utilisent des tables de consultation (LUT) tandis que les CPLD utilisent une somme de produits.
- Les CPLD ont une mémoire non volatile, contrairement aux FPGA.

Réalisation d'une LUT

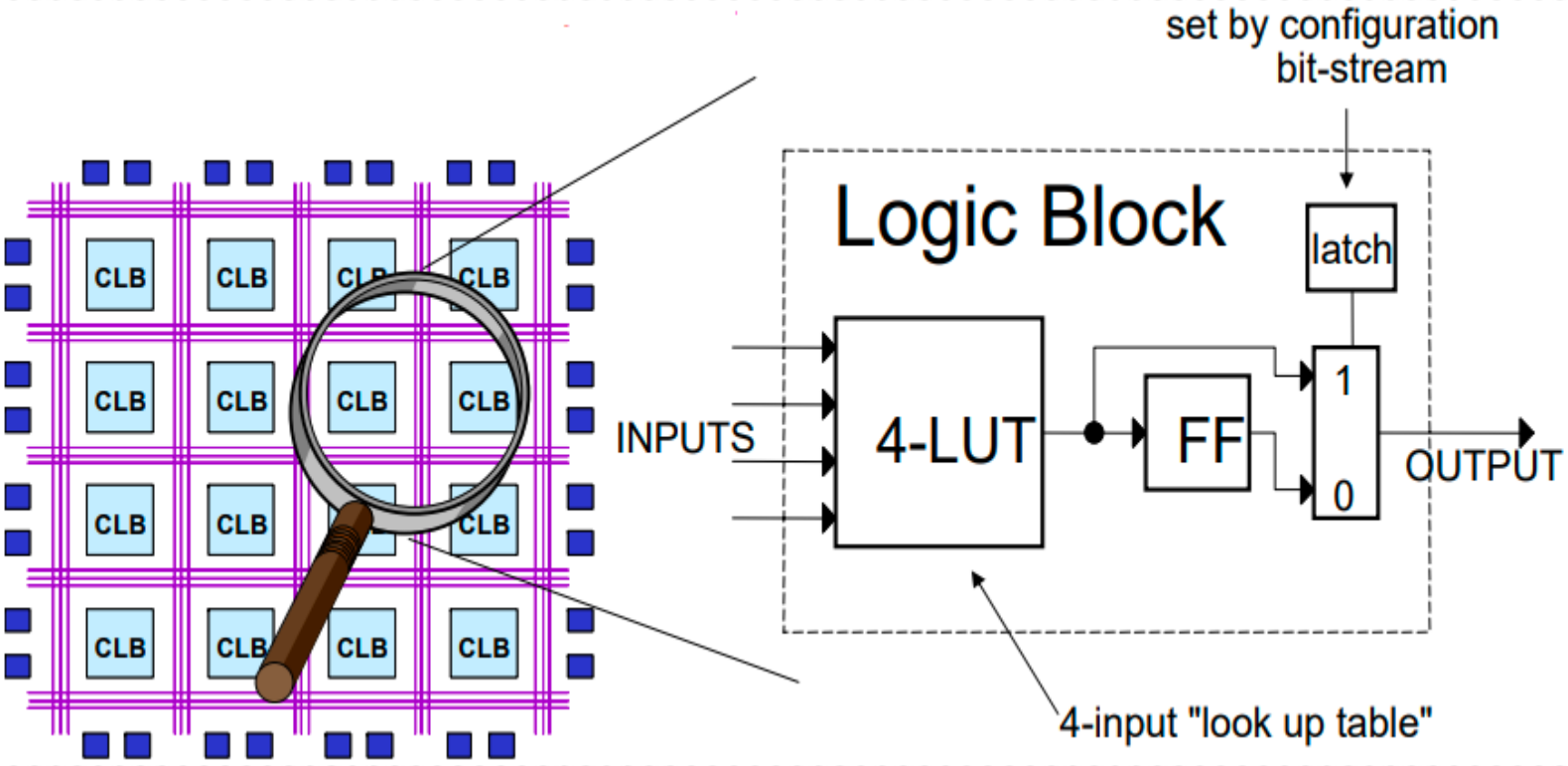
A	B	C	Z
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

« Look Up Table »

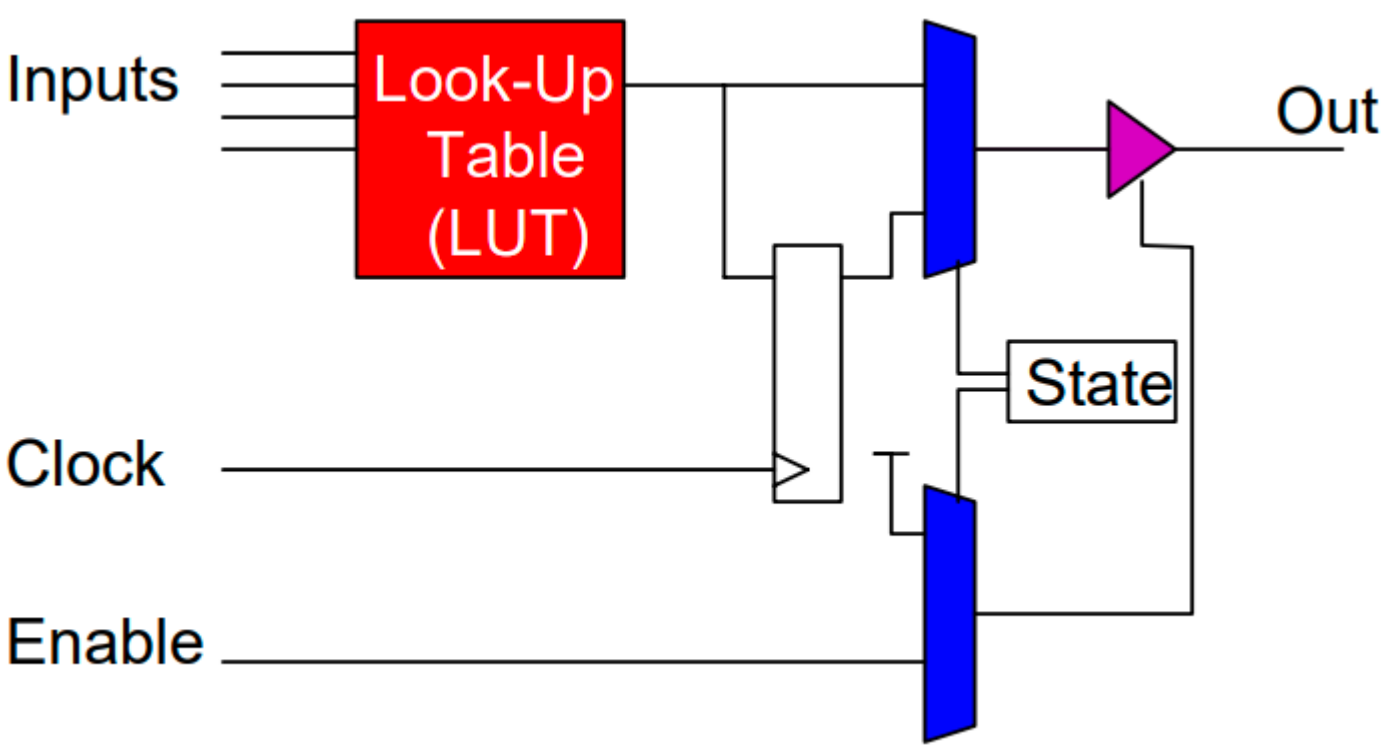
LUT 3



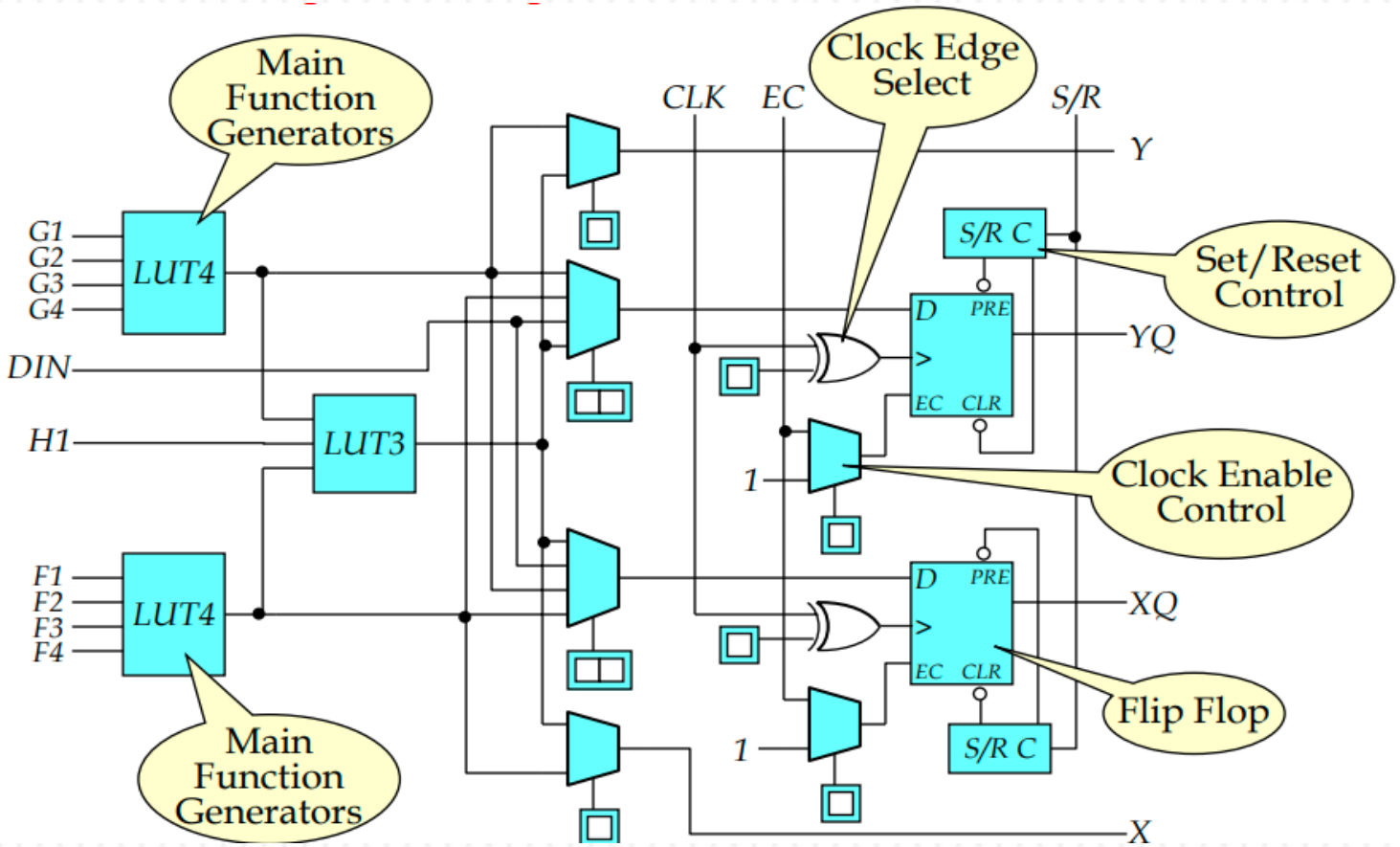
FPGA Reconfiguration



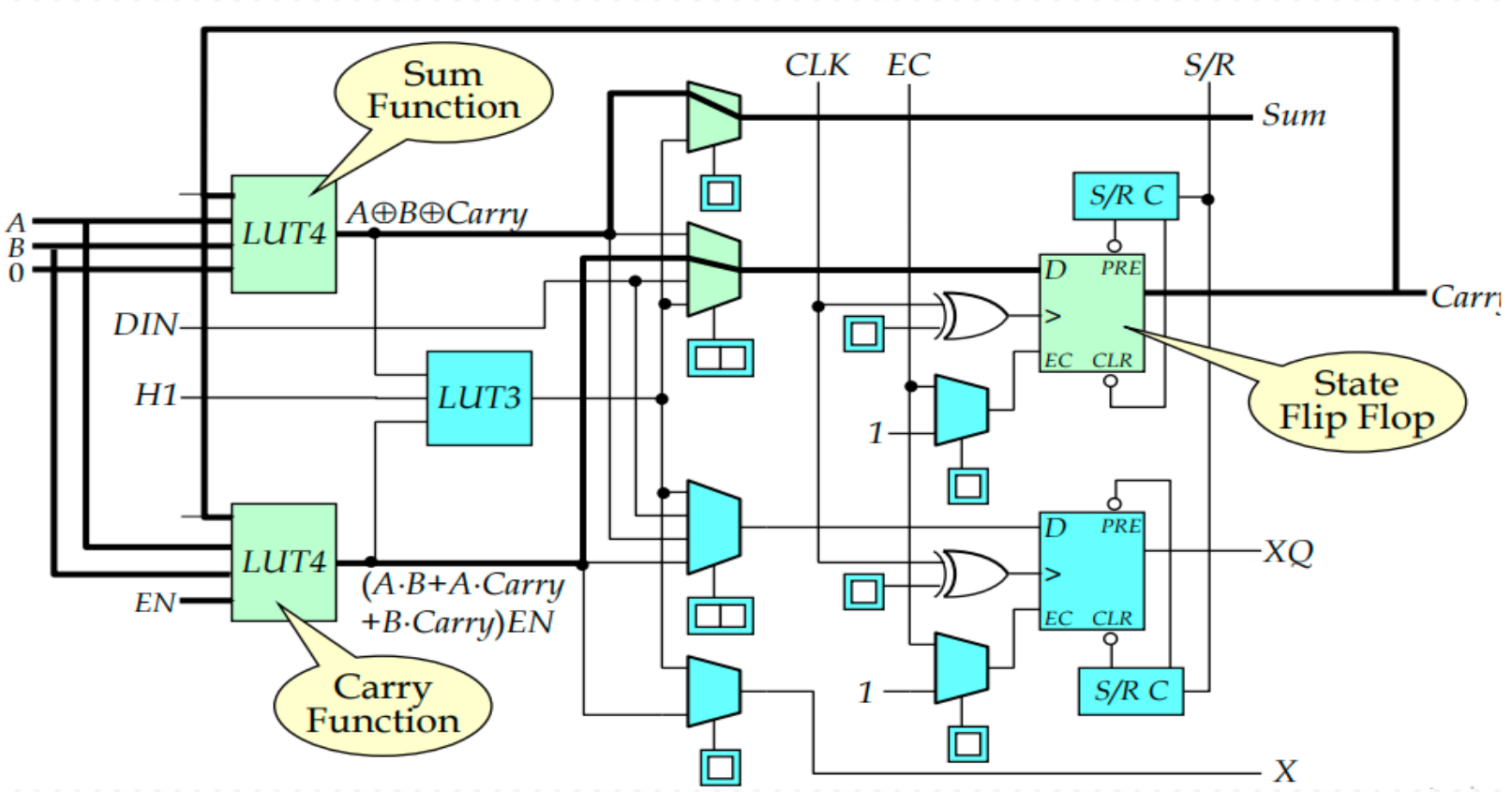
Generic reconfigurable block



Configurabble block logic



Exemple d'un additionneur



Le bitstream

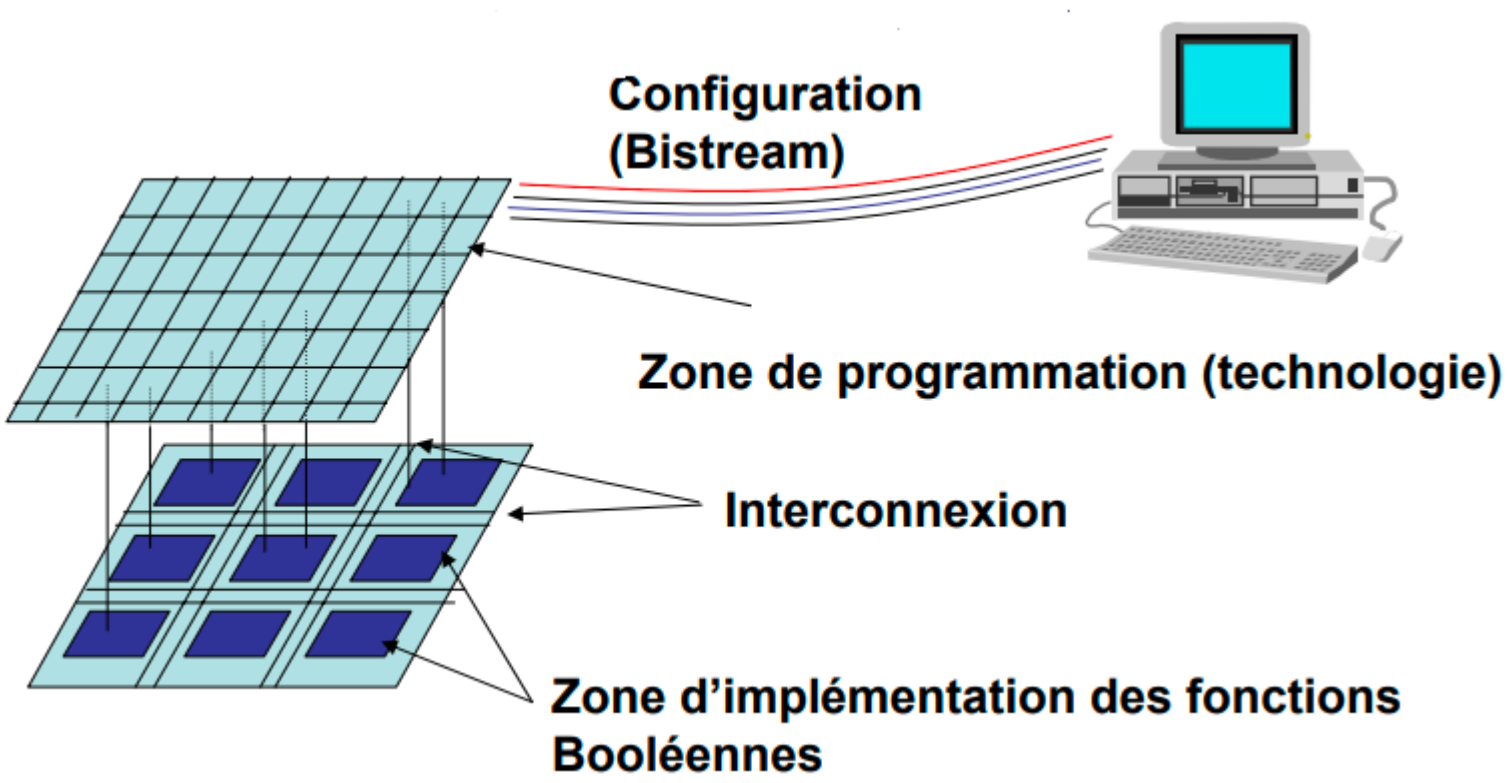
- Le bitstream représente le contenu de la mémoire de configuration destiné à déterminer :
  - les fonctions des blocs logiques,
  - les interconnexions entre les blocs.

Il ne s'agit donc nullement

du code d'un programme à exécuter par le circuit,

mais bien d'une description de matériel.

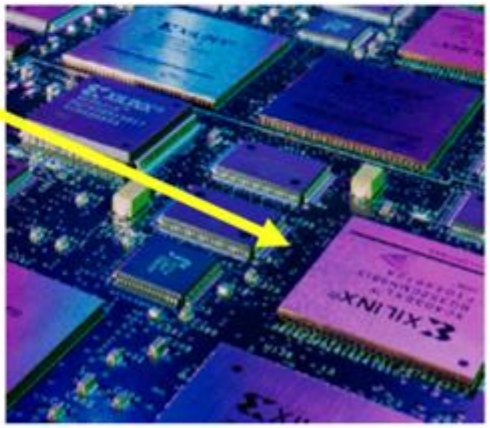
Principe simplifié



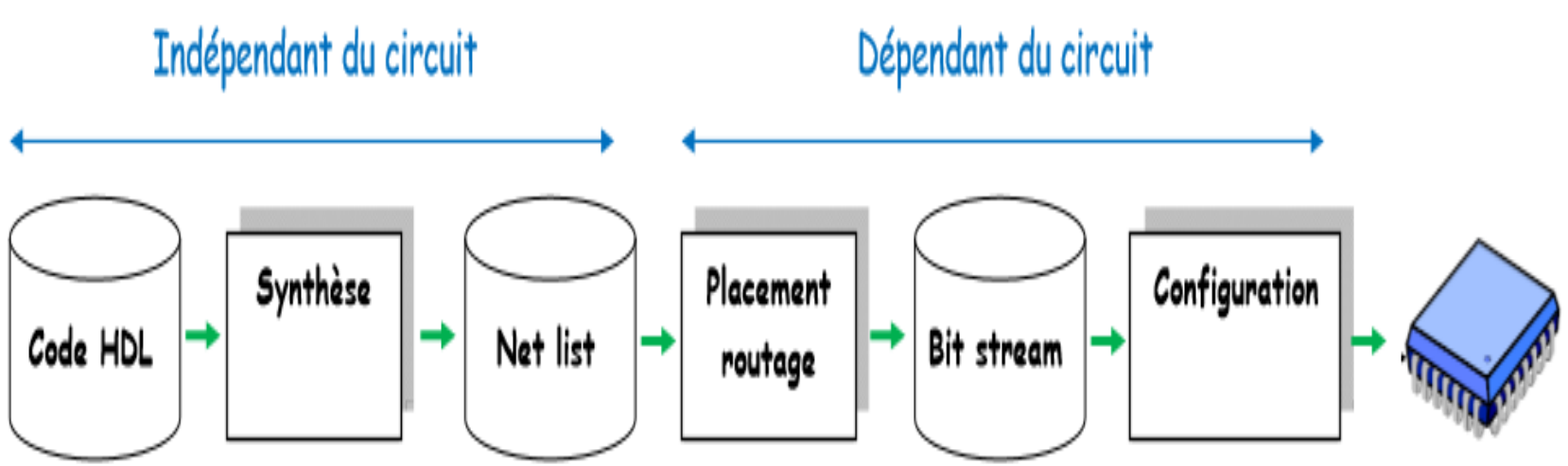
Le bitstream

- Le bitstream décrit la configuration de tous les éléments configurables du circuit
- Un transfert de bitstream est nécessaire lors de la mise sous tension et à chaque reconfiguration

10010010011110010110



Processus de production du bitsream



- **Netlist** décrit les composants et les connexions entre-eux (liste des interconnexions)
- **Placement :** sélection des blocs logiques du circuit
- **Routage** : interconnexion des blocs logiques

Notion de couche logique

- Une couche logique est un couple (bloc logique, interconnexion vers un autre bloc).
- Une couche logique induit pour tout signal électrique un délai :
  - temps de réponse du bloc logique (ns)
  - + temps de réponse du commutateur de connexion (ns).
- Les algorithmes de placement et de routage cherchent donc à optimiser le nombre de couches logiques à traverser.

Configuration

- La mémoire SRAM (volatile) du circuit FPGA contient la configuration « active ».
- A la mise hors tension, le contenu de la SRAM est perdu !
  - La configuration doit donc être mémorisée dans une mémoire de secours Flash PROM.
  - A la mise sous tension, ou sur demande particulière, le contenu de la Flash PROM est copié dans la SRAM.

Configuration

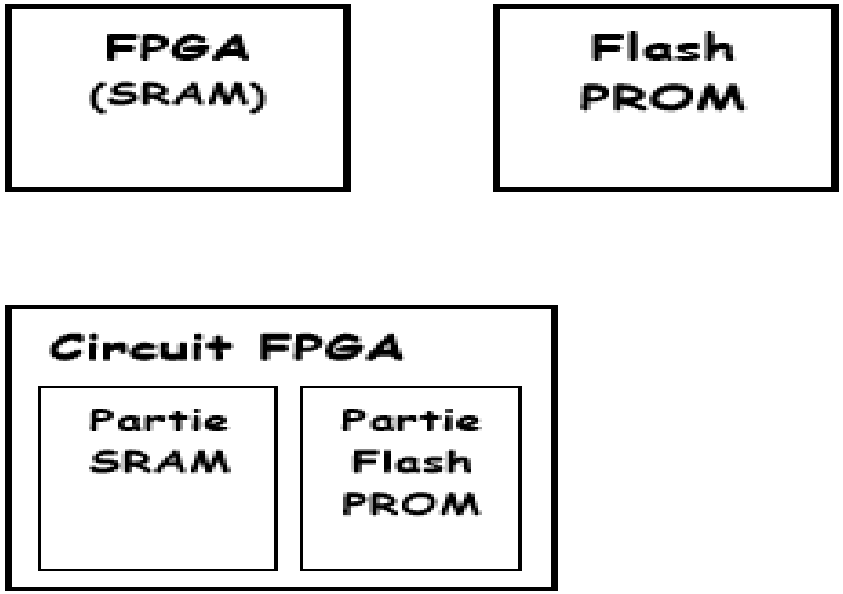
- Sur le marché, les fondateurs (Xilinx, Altera, Atmel) ont développé plusieurs familles de FPGA, concurrentes ou complémentaires, parfois moins performantes, parfois mieux adaptées. L'architecture, retenue par Xilinx se présente sous forme de deux couches :
  - une couche appelée circuit configurable,
  - une couche de mémoire SRAM (SelectRAM).
- La couche dite circuit configurable est constituée d'une matrice de blocs logiques configurables CLB (Configurable Logic Block). Les CLBs permettent de réaliser des fonctions combinatoires et des fonctions séquentielles. Tout autour de ces CLBs, on trouve des blocs entrées/sorties IOB dont le rôle est de gérer les entrées-sorties réalisant l'interface avec les modules extérieurs

Configuration

- La couche SRAM est un ensemble de blocs de mémoire RAM configurables de différentes manières. Leur capacité est de 18KB.
- Les blocs de la SRAM peuvent être mis en cascade pour réaliser de larges zones de stockage enfouies dans le FPGA.
- La mise au point de la configuration d'un FPGA consiste dans un premier temps à décrire ses données de configuration sous forme d'une succession de bits qu'on appelle bitstream/fichier de configuration.
- Ce bitstream est stocké dans une ROM externe au FPGA. Ainsi à la mise sous tension du FPGA, sa mémoire SRAM est chargée à partir de cette ROM et la configuration est prise en compte. Ensuite une vérification est effectuée pour tester si le fonctionnement réel correspond bien à l'attente du concepteur

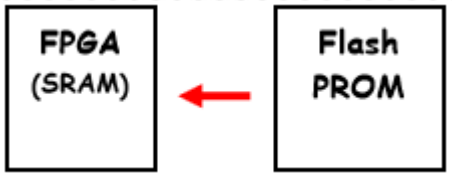
Configuration

- **Deux possibilités** :
  - **Deux CI flash séparés**
  - **Un seul CI flash intégré**

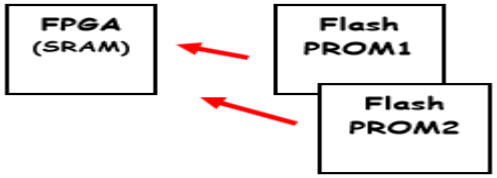


Types de configuration

- **Configuration statique**
  - Si une seule architecture suffit par application



- **Configuration semi-dynamique**
  - Plusieurs configurations possibles, pour des applications différentes



- **Configuration dynamique (auto-reconfiguration)**
  - Si l'architecture doit évoluer en cours de traitement : chargement continu dans certaines parties de la SRAM, sans arrêter le fonctionnement des sections inchangées

Types de configuration

- La logique reconfigurable est une branche émergente de l'architecture informatique qui cherche à construire des systèmes informatiques flexibles et dynamiques.
  - C'est-à-dire des systèmes ayant la propriété de pouvoir être modifiés après leur conception et pendant leur exécution, chose qui permet d'atteindre des niveaux de performance très élevés

Configuration dynamique

- La **Re-configuration dynamique des FPGAs** consiste à changer la programmation de ces circuits logiques programmables alors qu'ils sont en activité.
- La reconfiguration dynamique permet d'augmenter **l'efficacité** d'un FPGA en allouant ses ressources logiques à plusieurs tâches. Ainsi, en utilisant la notion de mémoire virtuelle qui permet entre autres d'exécuter un programme de taille plus grande que la mémoire physique disponible,
  - un FPGA reconfigurable dynamiquement peut offrir une quantité importante de ressources logiques virtuelles et les mapper au moment de l'exécution sur une quantité réduite de ressources physiquement disponibles.

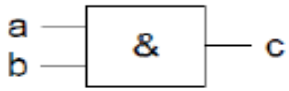
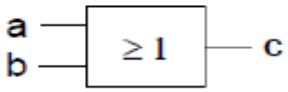
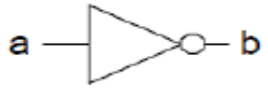
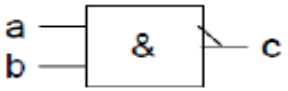
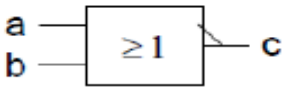
Configuration dynamique

- Un autre avantage qui suscite un intérêt croissant chez les chercheurs, est la **flexibilité** introduite par la reconfiguration dynamique des FPGA qui se manifeste à deux niveaux :
  - Le premier niveau permet au concepteur, de facilement adapter son application pour faire face à de nouvelles contraintes ou pour remplacer un algorithme par un autre plus efficace. Cette flexibilité assure une certaine évolution du système.
  - Le second niveau se situe au choix des tâches à exécuter, qui peut se faire dynamiquement, par exemple en fonction des données à traiter. On peut donc tout à fait imaginer que les données d'entrée puissent influer en temps réel sur l'enchaînement des algorithmes. Le matériel devient extrêmement malléable et contrôlable par logiciel, il peut être utilisé à des instants différents pour exécuter des opérations différentes.

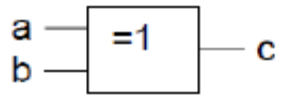
Configuration dynamique

- Théoriquement, la reconfiguration dynamique n'est pas réservée à une catégorie particulière de FPGA. Cependant, certains sont plus adaptés que d'autres pour satisfaire les contraintes temps réel.
- Le FPGA idéal pour construire un système reconfigurable dynamiquement doit posséder une vitesse d'exécution élevée et une durée de reconfiguration très faible. Malheureusement, les FPGAs commercialisés à l'heure actuelle ne réunissent que partiellement ces deux caractéristiques.
  - Cependant, un FPGA offrant une seule des deux caractéristiques sans que la deuxième ne soit excessivement pénalisante, peut être utilisé efficacement pour construire un système dynamique.
  - La durée de re-configuration a aussi un impact sur les performances d'un système dynamique, particulièrement si la modification de la fonctionnalité intervient fréquemment. La meilleure utilisation d'un FPGA dans un système dynamique est alors de servir de coprocesseur à une machine hôte pour accélérer des traitements répétitifs sur un volume de données important.

Logique

Fonction	Modèle Français	Modèle Américain
ET		
OU		
NON		
NON ET		
NON OU		

a	b	c
0	0	0
0	1	1
1	0	1
1	1	0



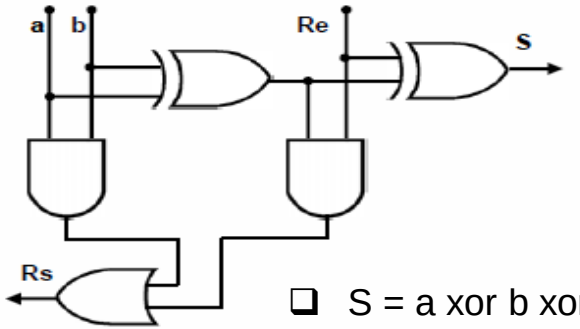
Demi additionneur et additionneur

- Demi additionneur 1 bit
  - S = a xor b
  - Rs = a.b

a	b	S	Rs
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

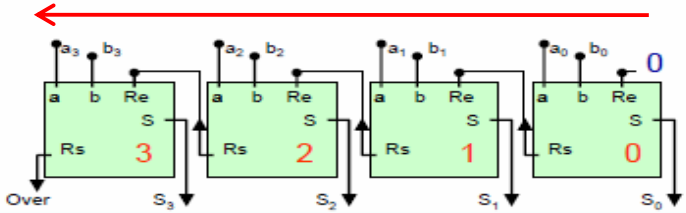
- **Additionneur**
  - **Cellule addition 1 bit**

a	b	Re	S	Rs
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



- S = a xor b xor Re
- R. = a.b + Re(a xor b)

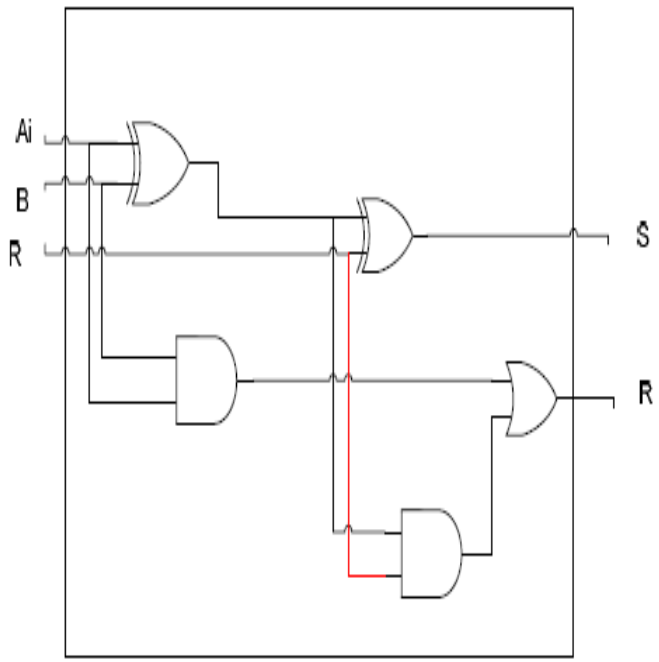
  - **Additionneur 4 bits**
    - **Slice**



Demi-additionneur et additionneur

$$R_i = A_i.B_i + R_{i-1}.(B_i \oplus A_i)$$

$$S_i = A_i \oplus B_i \oplus R_{i-1}$$



$$R_i = A_i.B_i + R_{i-1}.(B_i \oplus A_i)$$

$$S_i = A_i \oplus B_i \oplus R_{i-1}$$

Si on pose $X = A_i \oplus B_i$ et $Y = A_i B_i$

On obtient :

$$R_i = Y + R_{i-1}.X$$

$$S_i = X \oplus R_{i-1}$$

et si on pose $Z = X \oplus R_{i-1}$ et $T = R_{i-1}.X$

On obtient :

$$R_i = Y + T$$

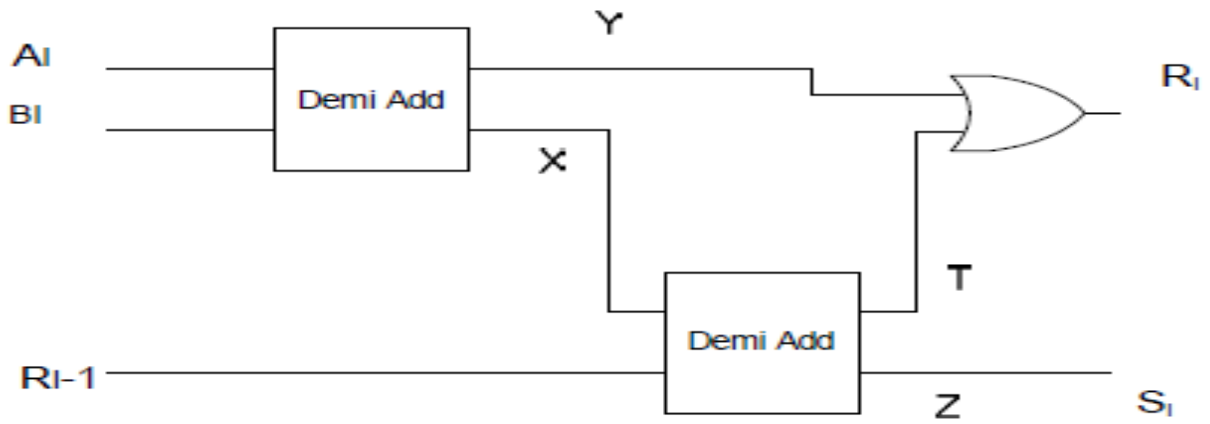
$$S_i = Z$$

Demi-additionneur et additionneur

$$X = A_i \oplus B_i$$
$$Y = A_i B_i$$
$$Z = X \oplus R_{i-1}$$
$$T = R_{i-1}.X$$
$$R_i = Y + T$$
$$S_i = Z$$

On remarque que X et Y sont les sorties d'un demi additionneur ayant comme entrées A et B

On remarque que Z et T sont les sorties d'un demi additionneur ayant comme entrées X et R_{i-1}



Spécificités des SE

- Les SE sont des systèmes complexes :
  - parfois conçus à partir de composants existants (IP)
  - incluant des parties matérielles et des parties logicielles
  - et généralement soumis à de nombreuses contraintes (coût, performances, consommation, fiabilité, …)

Les méthodes de conception doivent pouvoir prendre en compte tous ces aspects.

Spécificités des SE

- Il faut également :
  - prendre en compte le "Time To Market" (un retard d'un mois à la mise sur le marché peut induire une perte de l'ordre de 30% dans la rentabilité !),
  - considérer que les aspects liés à la vérification peuvent représenter de 60 à 70% du cycle de conception,
  - mais aussi que faciliter maintenance et mises à jour est crucial, …

D'où l'intérêt de modéliser et simuler avant de synthétiser et de fabriquer le dispositif. Un flot de conception à partir du niveau ESL (Electronic System Level) est souvent adopté.


Thank's