

Partie III : Cours

CIRCUITS LOGIQUES PROGRAMMABLES : PLD, CPLD et FPGA

III.1 INTRODUCTION

Au début des années 80, les développeurs disposaient de circuits de type PLD, facilement programmables mais ne pouvant contenir des designs trop complexes. Les ASICS quant à eux supportaient des systèmes de grande complexité mais n'avaient pas les propriétés de configuration des PLD. Il manquait donc un type de circuits permettant la réalisation des systèmes complexes, tout en offrant une configuration rapide et peu onéreuse. C'est pourquoi en 1984, Ross Freeman, Bernie Vonderschmit, et Jim Barnett fondent la compagnie Xilinx. En 1985, ils introduisaient sur le marché le premier FPGA, le XC2064, et offrent ainsi une alternative aux précédentes approches.

Depuis la commercialisation du premier pré-diffusé programmable (FPGA) en 1985, l'utilisation de ces circuits ne cesse de s'étendre à des domaines et applications variés. Les circuits FPGA, permettent d'implémenter des systèmes numériques aussi complexes que ceux réalisés jusqu'alors grâce aux ASICS, tout en ayant le grand avantage de pouvoir être programmés électriquement.

L'intérêt suscité par les FPGA est dû essentiellement à leurs prix abordables, facilité de mise en œuvre et flexibilité. En outre, les coûts fixes et délais de fabrications, en comparaison avec les circuits spécifiques (ASIC), sont totalement éliminés. Cependant, ils présentent une faible densité d'intégration de portes logiques et atteignent des fréquences de travail relativement faibles devant les ASIC.

Les circuits FPGA sont principalement composés d'un tableau d'éléments plus ou moins complexes pouvant être configurés, ainsi que d'un réseau complexe de connexions également configurables.

Dans ce chapitre nous allons décrire d'une manière générale l'architecture des circuits FPGA ainsi que les étapes nécessaires au développement d'un projet sur le circuit FPGA **Virtex-II V2MB1000**, disponible dans notre laboratoire, en partant de la programmation en utilisant le langage de description matériel VHDL, jusqu'au chargement sur la carte ce qui permet d'utiliser par la suite les ressources de cette carte pour implémenter des algorithmes MPPT discutés dans le chapitre précédent.

III.2 LES DIFFERENTES TECHNOLOGIES UTILISEES POUR LES FPGA

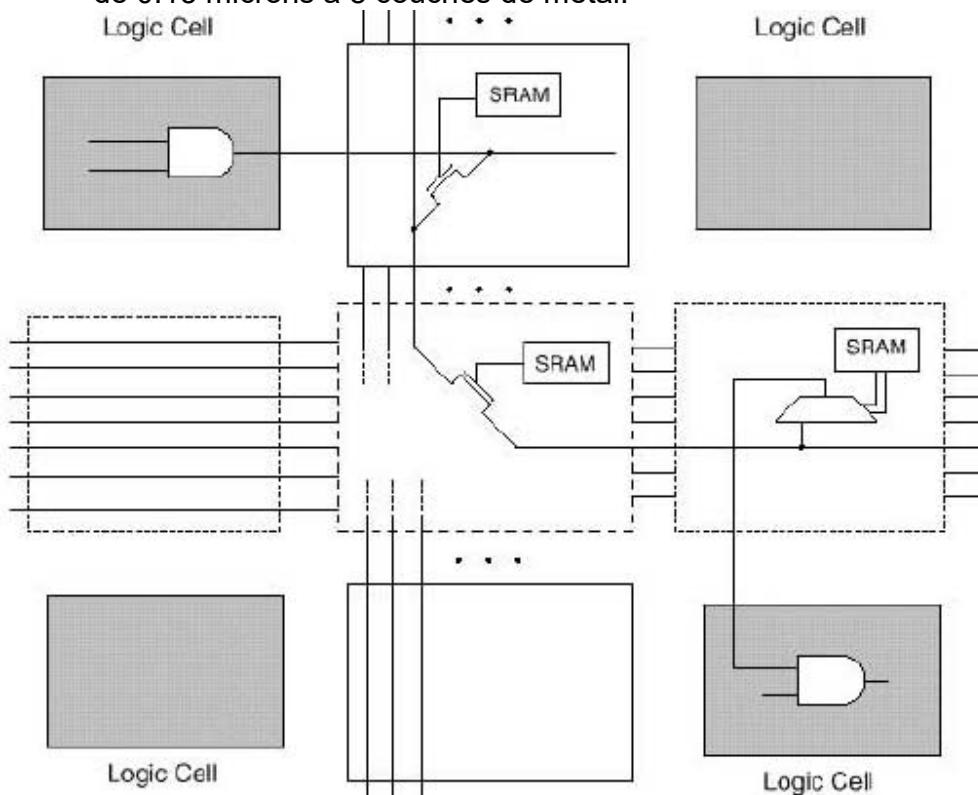
Elles sont au nombre de quatre:

- Static RAM: Les connexions sont réalisées en rendant les transistors passant. L'avantage de cette technologie est qu'elle permet une reconfiguration rapide au sein même du circuit. Le principal désavantage est la surface nécessaire pour la SRAM. Ils nécessitent l'utilisation d'une mémoire standard chargée à l'initialisation. Ils font appel à la technologie CMOS.

Voici quelques exemples de datasheet de FPGA utilisant la technologie SRAM:

- [Xilinx XC3000](#)
- [Xilinx XC5200](#)
- [Xilinx XC4000](#)
- [Xilinx Spartan](#)
- [Xilinx Spartan-II](#)
- [Xilinx Virtex-II](#)
- [Actel Série 40MX et 42MX](#)

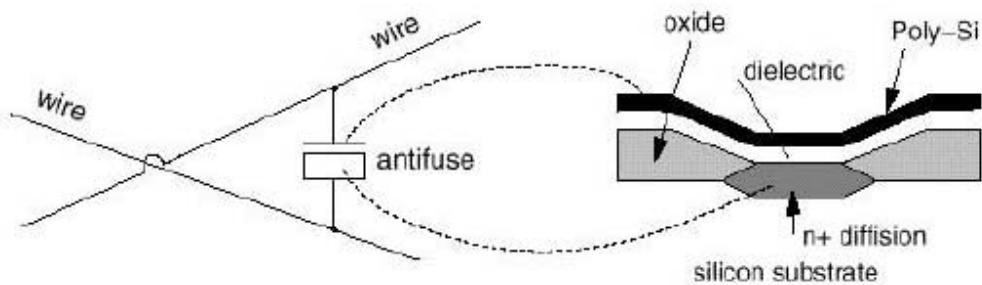
Les FPGA comportant le plus grand nombre de porte se trouve dans les séries Virtex-II. Leurs densités s'étendent de 40 mille à 8 million de portes systèmes. Ce qui est énorme sachant que l'on estime qu'un FPGA de 100 mille portes système suffit amplement à émuler complètement un microcontrôleur de type 8051. Ils possèdent une fréquence d'horloge de maximum 420 MHz et utilisent un process de 0.15 microns à 8 couches de métal.



- La technologie anti-Fuse: Un état anti-fuse réside en un état d'haute impédance. Il peut être programmé dans un état de faible impédance ou état "fused". Il s'agit d'une technologie moins chère que la SRAM, elle permet d'atteindre des vitesses plus élevées et occupe moins de place sur le circuit. Par contre, un tel FPGA ne peut être programmé qu'une seule fois. Performances électriques supérieures à la technologie SRAM (minimisation des effets RC due à la faible surface). Résistance de l'ordre d'une centaine d'ohm et effet capacitif de quelques femtofarads.

Voici quelques exemples de datasheet de FPGA utilisant la technologie anti-fuse:

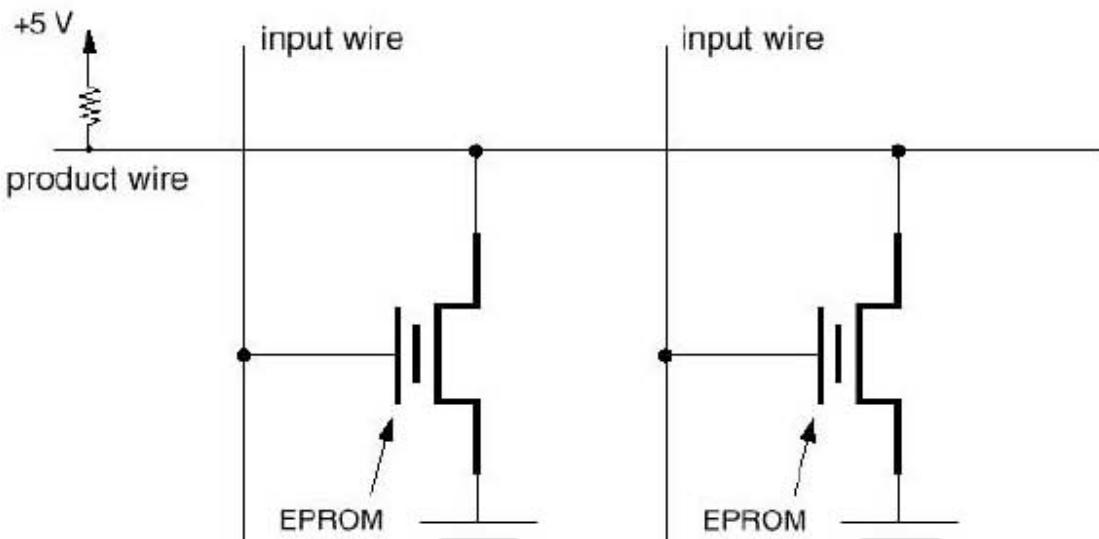
- [Actel série EX](#)
- [Actel série SX-A](#)



- **EPROM/EEPROM:** Cette méthode est la même que celle utilisée dans les mémoires EPROM. L'EEPROM est reprogrammable. La puce fonctionne seule. La surface moyenne et les caractéristiques électriques sont semblables à la SRAM.

Voici quelques exemples de datasheet de FPGA utilisant la technologie EEPROM:

- Actel PROASIC



Résumé des différentes technologies

Nom:	Reprogrammable:	Volatile:	Technologie:
Fuse	Non	Non	Bipolaire
EPROM	Oui, en dehors du circuit	Non	UVCMOS
EEPROM	Oui, dans le circuit	Non	EECMOS
SRAM	Oui, dans le circuit	Oui	CMOS
Antifuse	Non	Non	CMOS

III.3 FPGA

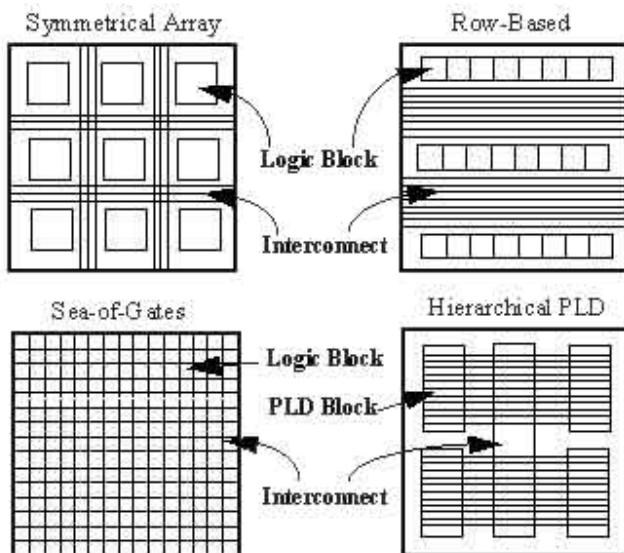
Présentation rapide: Les FPGA (Field Programmable Gate Arrays ou "réseaux logiques programmables") sont des composants VLSI entièrement reconfigurables ce qui permet de les reprogrammer à volonté afin d'accélérer notamment certaines phases de calculs. L'avantage de ce genre de circuit est sa grande souplesse qui permet de les réutiliser à volonté dans des algorithmes différents en un temps très court. Le progrès de ces technologies permet de faire des composants toujours plus rapides et à plus haute intégration, ce qui permet de programmer des applications importantes.

Introduction: Les circuits FPGA sont constitués d'une matrice de blocs logiques programmables entourés de blocs d'entrée sortie programmable. L'ensemble est relié par un réseau d'interconnexions programmable.

Les FPGA sont bien distincts des autres familles de circuits programmables tout en offrant le plus haut niveau d'intégration logique.

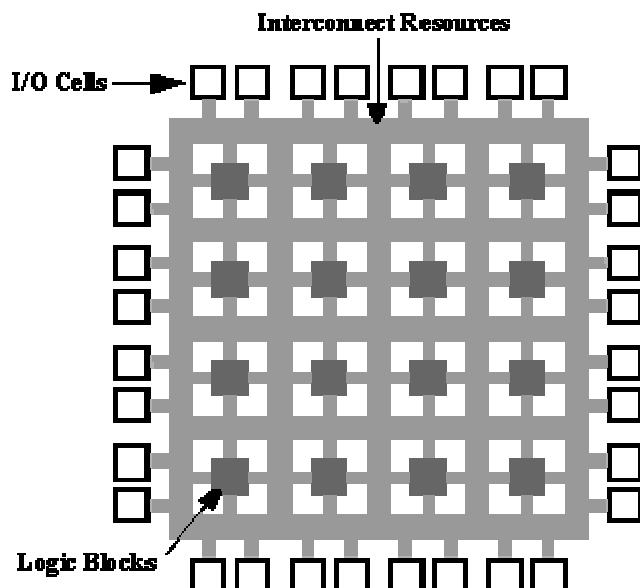
Il y a 4 principales catégories disponibles commercialement:

- Tableau symétrique.
- En colonne.
- Mers de portes.
- Les PLD hiérarchique.



Les différentes classes de FPGA.

Voici la structure interne d'un FPGA de type matrice symétrique. Il s'agit de l'architecture que l'on retrouve dans les FPGA de la série XC4000 de chez Xilinx.



Structure interne d'un FPGA

L'utilisateur peut programmer la fonction réalisée par chaque cellule (appelée CLB par Xilinx: Configurable Logic Block):

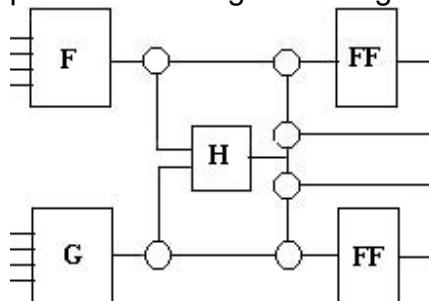
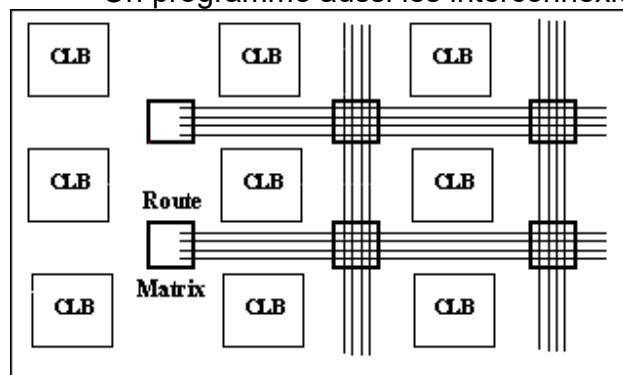


Schéma bloc d'une cellule

On programme aussi les interconnexions entre les cellules:



Les interconnexions entre les cellules d'un FPGA

De même que les entrées et sorties du circuit. L'avantage des FPGA est de pouvoir être configuré sur place, sans envoi du circuit chez le fabricant, ce qui permet de les

utiliser quelques minutes après leur conceptions. Les FPGA les plus récents sont configurables en une centaine de millisecondes. Les FPGA sont utilisés pour un développement rapide et bon marché des ASIC.

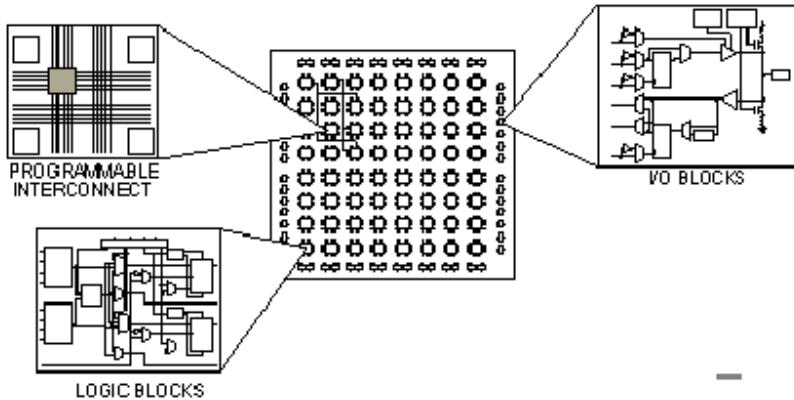
Inventés par la société Xilinx, le FPGA, dans la famille des ASICs, se situe entre les réseaux logiques programmables et les prédiffusés. C'est donc un composant standard combinant la densité et les performances d'un prédiffusé avec la souplesse due à la reprogrammation des PLD. Cette configuration évite le passage chez le fondeur et tous les inconvénients qui en découlent.

Architecture détaillée de la série XC4000 de Xilinx

Introduction: L'architecture, retenue par Xilinx, se présente sous forme de deux couches :

- une couche appelée circuit configurable,
- une couche réseau mémoire SRAM.

La couche dite 'circuit configurable' est constituée d'une matrice de blocs logiques configurables CLB permettant de réaliser des fonctions combinatoires et des fonctions séquentielles. Tout autour de ces blocs logiques configurables, nous trouvons des blocs entrées/sorties IOB dont le rôle est de gérer les entrées-sorties réalisant l'interface avec les modules extérieurs (cf. figure). La programmation du circuit FPGA appelé aussi LCA (logic cells arrays) consistera par le biais de l'application d'un potentiel adéquat sur la grille de certains transistors à effet de champ à interconnecter les éléments des CLB et des IOB afin de réaliser les fonctions souhaitées et d'assurer la propagation des signaux. Ces potentiels sont tout simplement mémorisés dans le réseau mémoire SRAM.



Architecture interne d'un FPGA.

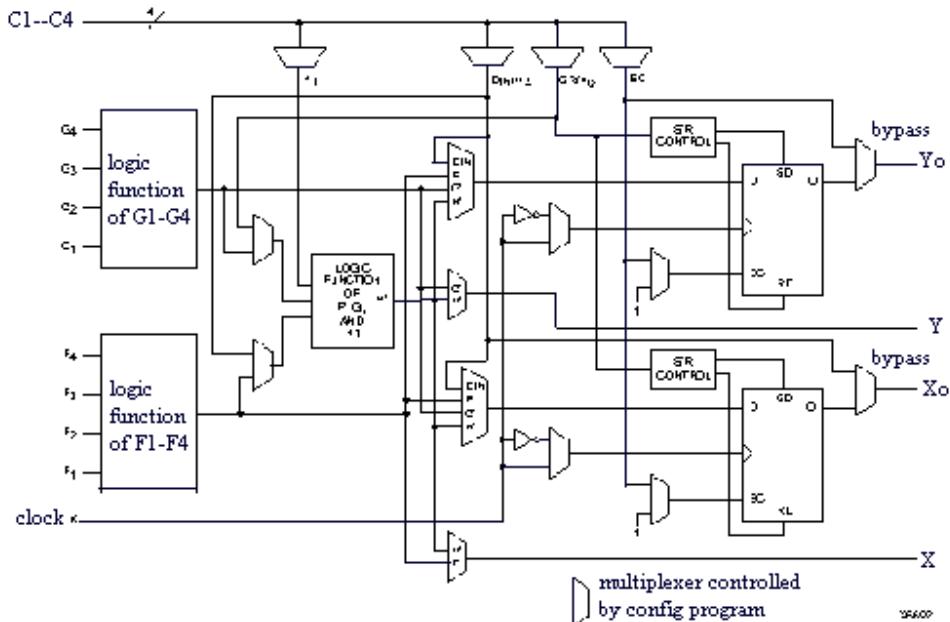
Les circuits FPGA du fabricant Xilinx utilisent deux types de cellules de base :

- les cellules d'entrées/sorties appelés IOB (input output bloc),
- les cellules logiques appelées CLB (configurable logic bloc). Ces différentes cellules sont reliées entre elles par un réseau d'interconnexions configurable.

a) Les CLB (configurable logic bloc)

Les blocs logiques configurables sont les éléments déterminants des performances du FPGA. Chaque bloc est composé d'un bloc de logique combinatoire composé de deux générateurs de fonctions à quatre entrées et d'un bloc de mémorisation synchronisation composé de deux bascules

D. Quatre autres entrées permettent d'effectuer les connexions internes entre les différents éléments du CLB. La figure ci-dessous, nous montre le schéma d'un CLB.



b) Cellule logique (CLB)

Voyons d'abord le bloc logique combinatoire qui possède deux générateurs de fonctions F' et G' à quatre entrées indépendantes ($F_1 \dots F_4, G_1 \dots G_4$), lesquelles offrent aux concepteurs une flexibilité de développement importante car la majorité des fonctions aléatoires à concevoir n'excède pas quatre variables. Les deux fonctions sont générées à partir d'une table de vérité câblée inscrite dans une zone mémoire, rendant ainsi les délais de propagation pour chaque générateur de fonction indépendants de celle à réaliser. Une troisième fonction H' est réalisée à partir des sorties F' et G' et d'une troisième variable d'entrée H_1 sortant d'un bloc composé de quatre signaux de contrôle $H_1, D_{in}, S/R, E_{c}$. Les signaux des générateurs de fonction peuvent sortir du CLB, soit par la sortie X , pour les fonctions F' et G' , soit Y pour les fonctions G' et H' . Ainsi un CLB peut être utilisé pour réaliser:

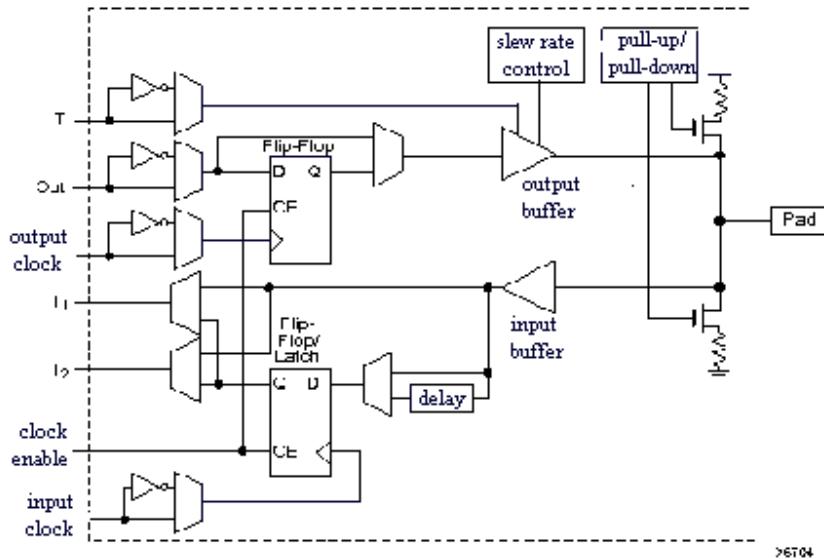
- o deux fonctions indépendantes à quatre entrées indépendantes, plus une troisième fonction de trois variables indépendantes
- o ou toute fonction à cinq variables
- o ou toute fonction à quatre variables et une autre avec quelques fonctions à six variables
- o ou certaines fonctions jusqu'à neufs variables.

L'intégration de fonctions à nombreuses variables diminue le nombre de CLB nécessaires, les délais de propagation des signaux et par conséquent augmente la densité et la vitesse du circuit. Les sorties de ces blocs logiques peuvent être

appliquées à des bascules au nombre de deux ou directement à la sortie du CLB (sorties X et Y). Chaque bascule présente deux modes de fonctionnement : un mode '**flip-flop**' avec comme donnée à mémoriser, soit l'une des fonctions F', G', H' soit l'entrée directe DIN et un mode **latch**. La donnée peut être mémorisée sur un **front montant ou descendant** de l'horloge (CLK). Les sorties de ces deux bascules correspondent aux sorties du CLB XQ et YQ. Un mode dit de "verrouillage" exploite une entrée S/R qui peut être programmée soit en mode SET, mise à 1 de la bascule, soit en Reset, mise à zéro de la bascule. Ces deux entrées coexistent avec une autre entrée laquelle n'est pas représentée sur la figure, appelée le global Set/Reset. Cette entrée initialise le circuit FPGA à chaque mise sous tension, à chaque configuration, en commandant toutes les bascules au même instant soit à '1', soit à '0'. Elle agit également lors d'un niveau actif sur le fil RESET lequel peut être connecté à n'importe quelle entrée du circuit FPGA. Un mode optionnel des CLB est la configuration en mémoire RAM de 16x2bits ou 32x1bit ou 16x1bit (en mode DUAL port). Les entrées F1 à F4 et G1 à G4 deviennent des lignes d'adresses sélectionnant une cellule mémoire particulière. La fonctionnalité des signaux de contrôle est modifiée dans cette configuration, les lignes H1, DIN et S/R deviennent respectivement les deux données D0, D1 (RAM 16x2bits) d'entrée et le signal de validation d'écriture WE. Le contenu de la cellule mémoire (D0 et D1) est accessible aux sorties des générateurs de fonctions F' et G'. Ces données peuvent sortir du CLB à travers ses sorties X et Y ou alors en passant par les deux bascules.

c) Les IOB (input output bloc)

La figure présente la structure de ce bloc. Ces blocs entrée/sortie permettent l'interface entre les broches du composant FPGA et la logique interne développée à l'intérieur du composant. Ils sont présents sur toute la périphérie du circuit FPGA. Chaque bloc IOB contrôle une broche du composant et il peut être défini en entrée, en sortie, en signaux bidirectionnels ou être inutilisé (haute impédance).



Input Output Block (IOB).

- Configuration en entrée

Premièrement, le signal d'entrée traverse un buffer qui selon sa programmation peut détecter soit des seuils TTL ou soit des seuils CMOS. Il peut être routé directement sur une entrée directe de la logique du circuit FPGA ou sur une entrée synchronisée. Cette synchronisation est réalisée à l'aide d'une bascule de type D, le changement d'état peut se faire sur un front montant ou descendant. De plus, cette entrée peut être retardée de quelques nanosecondes pour compenser le retard pris par le signal d'horloge lors de son passage par l'amplificateur. Le choix de la configuration de l'entrée s'effectue grâce à un multiplexeur (program controlled multiplexer). Un bit positionné dans une case mémoire commande ce dernier.

- Configuration en sortie

Nous distinguons les possibilités suivantes :

- a) inversion ou non du signal avant son application à l'IOB,
 - b) synchronisation du signal sur des fronts montants ou descendants d'horloge,
 - c) mise en place d'un " pull-up " ou " pull-down " dans le but de limiter la consommation des entrées sorties inutilisées,
 - d) signaux en logique trois états ou deux états. Le contrôle de mise en haute impédance et la réalisation des lignes bidirectionnelles sont commandés par le signal de commande Out Enable lequel peut être inversé ou non. Chaque sortie peut délivrer un courant de 12mA. Ainsi toutes ces possibilités permettent au concepteur de connecter au mieux une architecture avec les périphériques extérieurs.

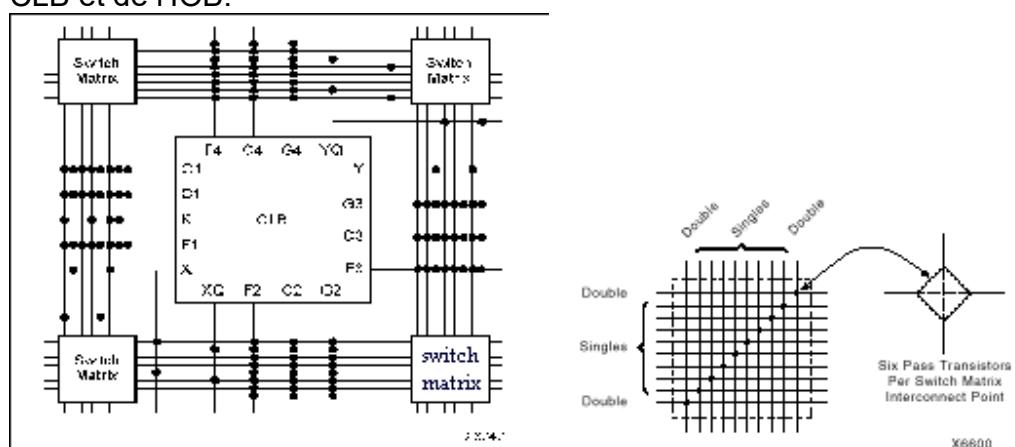
- **Les différents types d'interconnexions**

Les connexions internes dans les circuits FPGA sont composées de segments métallisés. Parallèlement à ces lignes, nous trouvons des matrices programmables réparties sur la totalité du circuit, horizontalement et verticalement entre les divers CLB. Elles permettent les connexions entre les diverses lignes, celles-ci sont assurées par des transistors MOS dont l'état est contrôlé par des cellules de mémoire vive ou RAM. Le rôle de ces interconnexions est de relier avec un maximum d'efficacité les blocs logiques et les entrées/sorties afin que le taux d'utilisation dans un circuit donné soit le plus élevé possible. Pour parvenir à cet objectif, Xilinx propose trois sortes d'interconnexions selon la longueur et la destination des liaisons. Nous disposons :

- d'interconnexions à usage général,
- d'interconnexions directes,
- de longues lignes.

- **Les interconnexions à usage général**

Ce système fonctionne en une grille de cinq segments métalliques verticaux et quatre segments horizontaux positionnés entre les rangées et les colonnes de CLB et de l'IOB.

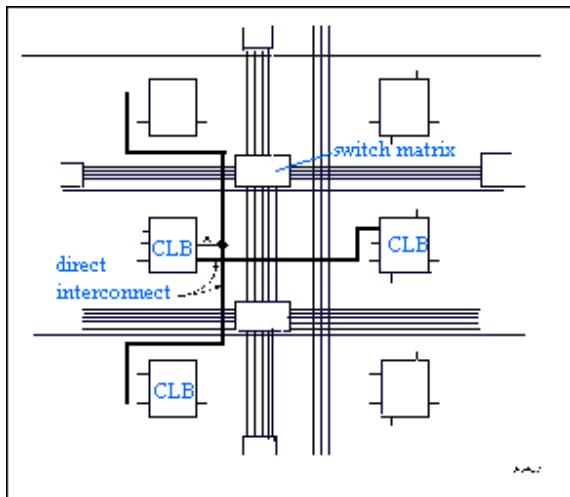


Connexions à usage général et détail d'une matrice de commutation.

Des aiguilleurs appelés aussi matrices de commutation sont situés à chaque intersection. Leur rôle est de raccorder les segments entre eux selon diverses configurations, ils assurent ainsi la communication des signaux d'une voie sur l'autre. Ces interconnexions sont utilisées pour relier un CLB à n'importe quel autre. Pour éviter que les signaux traversant les grandes lignes ne soient affaiblis, nous trouvons généralement des buffers implantés en haut et à droite de chaque matrice de commutation.

- **Les interconnexions directes**

Ces interconnexions permettent l'établissement de liaisons entre les CLB et les IOB avec un maximum d'efficacité en terme de vitesse et d'occupation du circuit. De plus, il est possible de connecter directement certaines entrées d'un CLB aux sorties d'un autre.

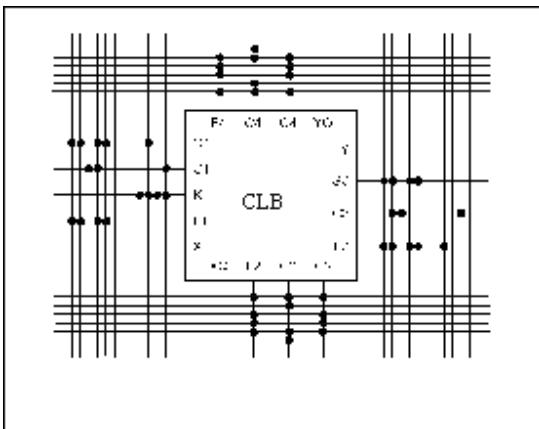


Les interconnexions directes.

Pour chaque bloc logique configurable, la sortie X peut être connectée directement aux entrées C ou D du CLB situé au-dessus et les entrées A ou B du CLB situé au-dessous. Quant à la sortie Y, elle peut être connectée à l'entrée B du CLB placé immédiatement à sa droite. Pour chaque bloc logique adjacent à un bloc entrée/sortie, les connexions sont possibles avec les entrées I ou les sorties O suivant leur position sur le circuit.

- **Les longues lignes**

Les longues lignes sont de longs segments métallisés parcourant toute la longueur et la largeur du composant, elles permettent éventuellement de transmettre avec un minimum de retard les signaux entre les différents éléments dans le but d'assurer un synchronisme aussi parfait que possible. De plus, ces longues lignes permettent d'éviter la multiplicité des points d'interconnexion.



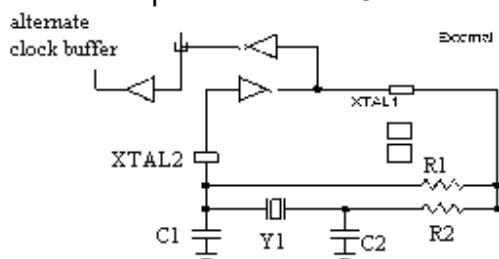
Les longues lignes.

- **Performances des interconnexions**

Les performances des interconnexions dépendent du type de connexions utilisées. Pour les interconnexions à usage général, les délais générés dépendent du nombre de segments et de la quantité d'aiguilleurs employés. Le délai de propagation de signaux utilisant les connexions directes est minimum pour une connectique de bloc à bloc. Quant aux segments utilisés pour les longues lignes, ils possèdent une faible résistance mais une capacité importante. De plus, si on utilise un aiguilleur, sa résistance s'ajoute à celle existante.

- **L'oscillateur à quartz**

Placé dans un angle de la puce, il peut être activé lors de la phase de programmation pour réaliser un oscillateur. Il utilise deux IOB voisins, pour réaliser l'oscillateur dont le schéma est présenté ci-dessous. Cette oscillateur ne peut être réalisé que dans un angle de la puce où se trouve l'amplificateur prévu à cet effet. Il est évident que si l'oscillateur n'est pas utilisé, les deux IOB sont utilisables au même titre que les autres IOB.



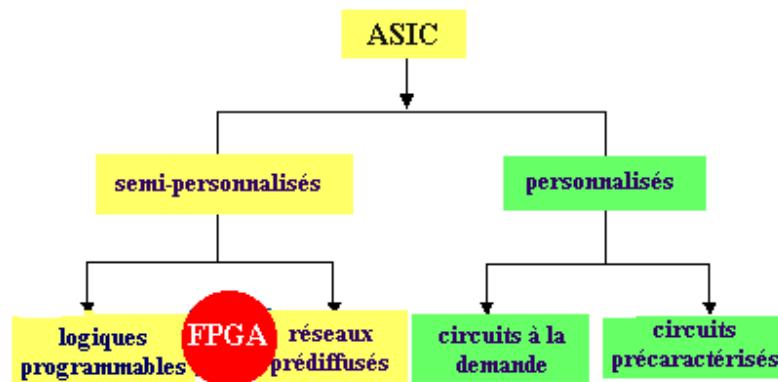
L'oscillateur à quartz.

III.4 ASICs

Les circuits ASIC constituent la troisième génération de circuits intégrés qui a vu le jour au début des années 80. En comparaison des circuits intégrés standards et figés proposés par les fabricants, l'ASIC présente une personnalisation de son fonctionnement, selon l'utilisation, accompagnée d'une réduction du temps de développement, d'une augmentation de la densité d'intégration et de la vitesse de fonctionnement. En outre sa

personnalisation lui confère un autre avantage industriel, c'est évidemment la confidentialité. Ce concept d'abord développé autour du silicium s'est ensuite étendu à d'autres matériaux pour les applications microondes ou très rapides (GaAs par exemple).

Par définition, les circuits ASIC regroupent tous les circuits dont la fonction peut être personnalisée d'une manière ou d'une autre en vue d'une application spécifique, par opposition aux circuits standards dont la fonction est définie et parfaitement décrite dans le catalogue de composants. Les ASIC peuvent être classés en plusieurs catégories selon leur niveau d'intégration [LUCA94], en fait un ASIC est défini par sa structure de base (réseau programmable, cellule de base, matrice, etc.). Sous le terme ASIC deux familles sont regroupées, les semi-personnalisés et les personnalisés.

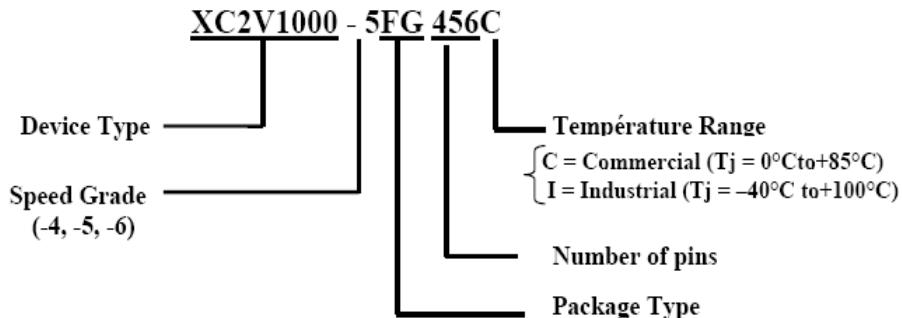


Avantages et inconvénients de l'utilisation d'ASIC: D'une manière générale l'utilisation d'un ASIC conduit à de nombreux avantages provenant essentiellement de la réduction de la taille des systèmes. Il en ressort :

- Réduction du nombre de composants sur le circuit imprimé. La consommation et l'encombrement s'en trouvent considérablement réduits.
- Le concept ASIC par définition assure une optimisation maximale du circuit à réaliser. Nous disposons alors d'un circuit intégré correspondant réellement à nos propres besoins.
- La personnalisation du circuit donne une confidentialité au concepteur et une protection industrielle.
- Enfin, ce type de composant augmente la complexité du circuit, sa vitesse de fonctionnement et sa fiabilité.
- Dans l'approche des circuits prédiffusés et personnalisés, l'inconvénient majeur réside dans le fait du passage obligatoire chez le fondeur ce qui implique des frais de développement élevés du circuit. En général le fondeur ne souhaite pas intervenir dans la phase de conception ; sa tâche est de réaliser le composant à partir des masques. Dans le but de réduire les surcoûts dus aux modifications, il s'avère nécessaire d'être rigoureux lors de la phase de développement de telle sorte que le circuit prototype fonctionne dès les premiers essais : c'est réussi dans environ 60% des cas. De plus dans de nombreuses applications, l'utilisateur doit concevoir les programmes de testabilité.

Nomenclature des circuits FPGA

Les circuits FPGA suivent la nomenclature suivante, selon un exemple donné:



• **Device type:** il montre le type de la famille qui est dans notre exemple VIRTEX-II.

• **Speed grade:** il montre la vitesse du composant selon la technologie.

Terminons ce point par un bref aperçu des avantages et désavantages de l'approche FPGA en commençant par les désavantages:

- Une fréquence d'horloge moins élevée, pour une même application, en comparaison avec les ASICs.
- Une plus grande place nécessaire sur le silicium, beaucoup de logique additionnelle étant nécessaire au bon fonctionnement de la programmation du FPGA.
- Dans la même lignée, l'ensemble des éléments programmables ne sont jamais entièrement utilisés. Leur nombre dépend de la taille du design, mais ne peut pas atteindre 100%, le routage devenant quasiment impossible lorsque le circuit est presque plein.

Et enfin les avantages des FPGAs:

- Un faible coût de développement, le prototype ne nécessitant pas de réalisation matériel, mais seulement des tests successifs sur un FPGA.
- Peu de risques, dans le sens où une erreur de design est très vite corrigée et n'implique pas la création d'un nouveau circuit.
- Une grande rapidité lors de la réalisation d'un prototype.

Références

1. Van den Bout, D. (1998). *The practical Xilinx designer lab book*. Prentice Hall.
2. Crockett, L. H., Elliot, R. A., Enderwitz, M. A., & Stewart, R. W. (2014). *The Zynq Book: Embedded Processing with the Arm Cortex-A9 on the Xilinx Zynq-7000 All Programmable Soc*. Strathclyde Academic Media
3. Rahman, A. (2008). *FPGA based design and applications*. Springer Publishing Company, Incorporated.
4. https://www.academia.edu/30401224/Le_langage_VHDL