

Les FPGA

Technologie, architecture et utilisation¹

1. Des circuits matériels configurables

Les FPGA (*Field Programmable Gate Array*) sont des circuits numériques matériels configurables dédiés à l'électronique numériques [BOS 10], [MAX 04]. A l'état initial ils ne peuvent rien faire mais disposent d'une importante quantité (dépendant de la technologie utilisée) de ressources matérielles opérationnelles dont on peut configurer la fonction. Ces ressources sont, principalement, des blocs élémentaires logiques (pour réaliser des fonctions booléennes), des mémoires RAM, des opérateurs arithmétiques (qui travaillent en virgule fixe), des ressources de routage interne et des entrées/sorties. Ces ressources configurables sont reliées par un réseau dense de lignes de routage et de lignes de transport des horloges. Ces lignes de routage sont aussi configurables.

En plus de ces ressources, un FPGA est composé d'une mémoire interne de configuration. Chaque point de cette mémoire correspond à la configuration d'un élément d'une des ressources opérationnelles. Cette mémoire est, dans la plupart des cas, réalisée avec une des trois technologies suivantes : ANTIFUSIBLE (la plus ancienne, configurable une seule fois), FLASH (non-volatile) ou SRAM (volatile, la plus utilisée, représente plus de 80 % du marché).

Comme le montre la figure 1, pour réaliser une application avec un FPGA il faut décrire le circuit électronique à réaliser avec un langage de description matérielle comme le VHDL² (*Very High Speed Integrated Circuit Hardware Description Language*). Puis il faut synthétiser cette description en circuit électronique. Cette étape et les suivantes peuvent se faire avec des logiciels gratuits fournis par le fabricant de circuit. Enfin après une étape de placement et routage qui prend en

¹ Rédigé par Lilian BOSSUET.

² D'autres langages comme Verilog et SystemC existent, mais VHDL reste le plus utilisé en Europe actuellement.

compte l'architecture du FPGA, un fichier de configuration appelé *bitstream* est généré. Celui-ci permet de spécifier au FPGA lors de la configuration la position des points de la mémoire de configuration.

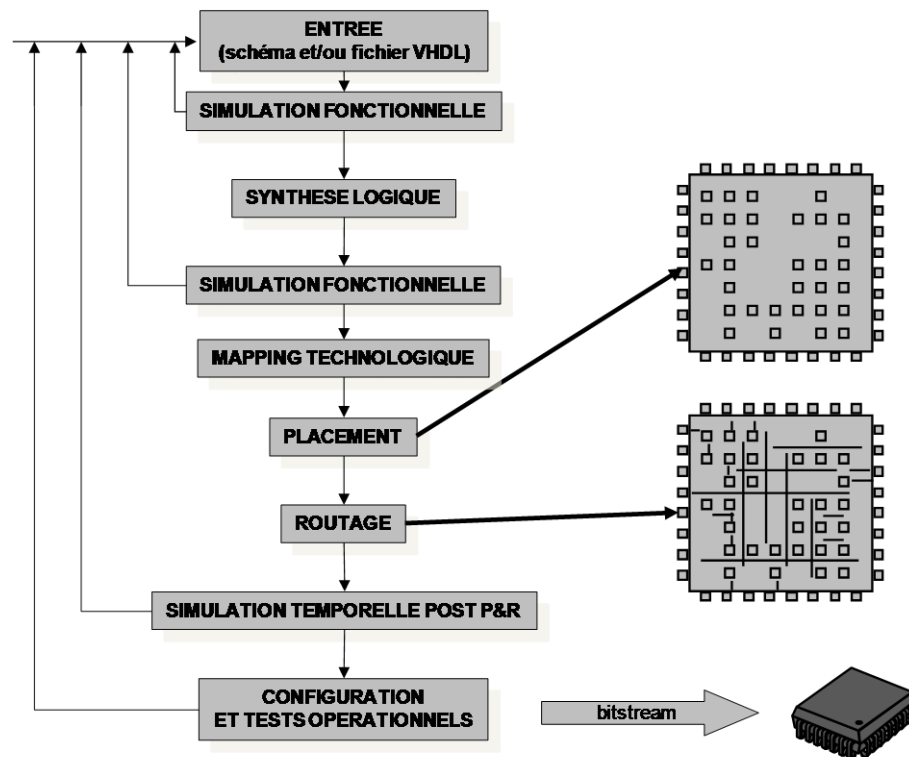


Figure X.1. Flot classique de conception FPGA.

Parmi les principaux fabricants de FPGA dans le monde on peut citer : Xilinx (n°1 du marché des FPGA, 53% de part de marché en 2009) [XIL 10], Altera (n°2 du marché des FPGA, 36 % de part de marché en 2009) [ALT 10], Actel (n°1 du marché des FPGA Antifusibles et FLASH) [ACT 10], Atmel, QuickLogic, Lattice, M2000 (cœurs de FPGA).

Les FPGA sont les plus récents des circuits numériques, à ce titre ils sont toujours en pleine évolution. Leur architecture a évolué ces dernières années, tout comme la granularité et le type de ressources logiques configurables. Une des évolutions particulièrement intéressante concerne leur utilisation. La reconfiguration partielle et la reconfiguration dynamique ont ouvert un nouvel espace d'application pour ces circuits qui prennent de plus en plus de part dans le marché des circuits

numériques matériels (hors microprocesseur) dominée par les ASIC³ (*Application Specific Integrated Circuit*). Ce chapitre fait un point sur ces évolutions en débutant par un rappel historique de l'évolution de ces circuits.

2. Historique

2.1. Considérations technologiques

En 1984 la société américaine Xilinx fut précurseur du domaine en lançant le premier circuit FPGA commercial, le XC2000. Ce composant avait une capacité maximum de 1500 portes logiques. La technologie utilisée était alors une technologie aluminium à 2µm avec 2 niveaux de métallisation. Xilinx sera suivi un peu plus tard, et jamais lâché, par son plus sérieux concurrent Altera qui lança en 1992 la famille de FPGA FLEX 8000 dont la capacité maximum atteignait 15000 portes logiques.

Ces composants apportent de nouveaux concepts grâce à leur universalité applicative et à leurs possibilités de reconfiguration. Longtemps ils ne pouvaient être intégrés directement aux produits commerciaux, à part pour réaliser quelques fonctions logiques. Ils étaient pourtant très utilisés, et ils le sont toujours, dans les bureaux d'études pour le prototypage d'ASIC. Leur faculté de reconfiguration offre aux concepteurs un processus de développement avec des possibilités de conception itérative. Cependant, à leur début ils n'avaient pas des capacités, en nombre de portes logiques équivalentes, suffisantes pour permettre la réalisation d'applications complètes, de plus, leurs performances étaient très en dessous de celles des ASIC. Les outils dédiés aux FPGA offraient des performances limitées car ils étaient issus des outils de conception d'ASIC qui ne prenaient pas en compte toutes les spécificités des composants reconfigurables.

Depuis les années 2000 des évolutions majeures ont été apportées. Les technologies utilisées pour les FPGA sont les mêmes que celles utilisées pour les ASIC. Par exemple, la technologie cuivre est mise en œuvre pour la réalisation des métallisations au sein des FPGA. Elle permet une réduction d'environ 70% des temps de propagation des signaux le long des métallisations par rapport à la technologie aluminium.

En 2000, les technologies utilisées étaient des technologies CMOS 0,15µm avec 8 niveaux de métallisation, aujourd'hui la technologie utilisée est la technologie CMOS 40 nm avec plus de 10 ou 12 niveaux de métallisation en cuivre.

³ Un ASIC est un circuit électronique réalisé *sur mesure*. Ce sont les circuits les plus performants mais aussi les plus chers et longs à réaliser.

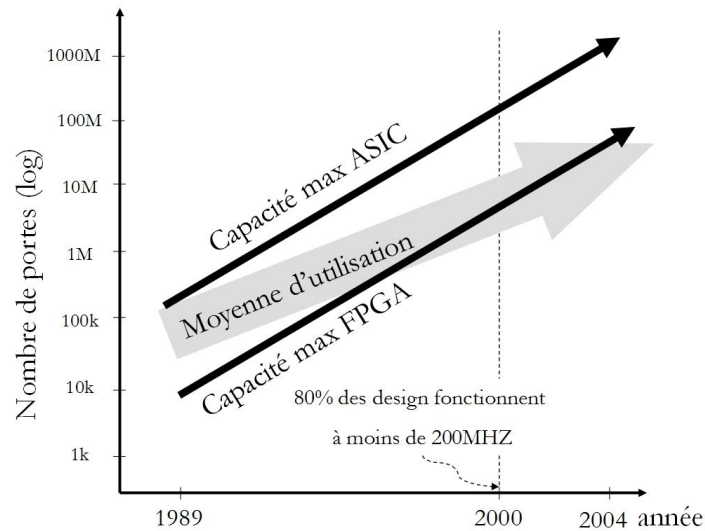


Figure 2. Densité d'intégration des circuits ASIC et FPGA par rapport aux besoins moyens des applications.

En 2000 et 2001 les deux concurrents Xilinx et Altera ont franchi une nouvelle étape au niveau de la densité d'intégration en sortant respectivement leurs circuits Virtex et Apex-II dont les capacités maximums avoisinaient les 4 millions de portes logiques équivalentes avec en plus l'introduction de larges bancs de mémoire embarquée. Aujourd'hui, les fréquences de fonctionnement de ces circuits sont de l'ordre de quelques centaines de mégaHertz, (ces dernières sont en réalité très dépendantes de l'application). Bien que ces valeurs soient relativement réduites par rapport aux ASIC, depuis les années 2000, elles sont suffisantes pour une très large majorité d'applications actuelles comme l'illustre la figure 2. En conclusion, les capacités des FPGA permettent d'offrir aux concepteurs une solution supplémentaire de réalisation pour une majorité d'applications avec des vitesses de fonctionnement convenables.

2.2. Considérations économiques

Dans un contexte économique mondial incertain les FPGA apparaissent comme une solution flexible bien adaptée aux contraintes économiques telles que le temps de mise sur le marché et le potentiel d'évolution ou de flexibilité des produits.

De plus, le modèle économique lié aux FPGA, qui est un modèle linéaire devient de plus en plus avantageux par rapport aux modèles ASIC dont le coût du premier

échantillon fabriqué rend l'amortissement d'une telle solution long et viable économiquement uniquement pour de très grandes productions. La figure 3 montre que le nombre de circuits fabriqués à partir duquel la solution ASIC est économiquement plus rentable (point de *cross-over*) tend à augmenter avec l'évolution des technologies. Par exemple, avec la technologie utilisée en 2003 (90 nm) la solution ASIC est intéressante à partir d'environ un million de circuits à fabriquer (et donc à vendre). Les solutions FPGA (SRAM) sont donc de plus en plus intéressantes tant d'un point de vue technique qu'économique.

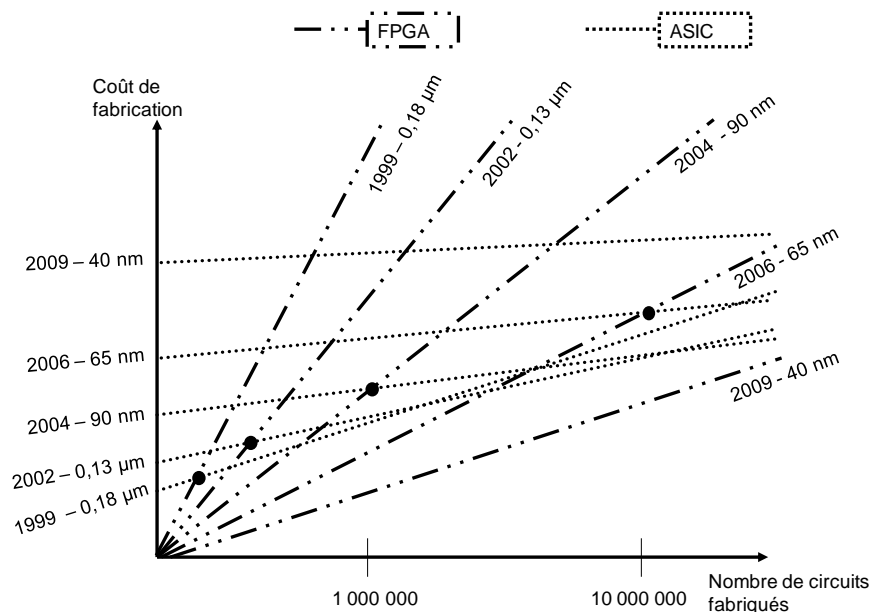


Figure 3. Modèle de coûts pour les composants ASIC et FPGA (SRAM) en fonction de la technologie utilisée.

2.3. Evolutions

Les FPGA n'ont pas fini d'évoluer, en particulier il y a un rapprochement des systèmes reconfigurables avec les systèmes programmables. Effectivement, de nombreux fabricants de FPGA proposent aujourd'hui des circuits qui embarquent un ou plusieurs microprocesseurs dans leur matrice. Une des évolutions architecturales possibles est de remplacer les éléments logiques de bases par des microprocesseurs élémentaires. Ainsi des architectures systoliques reconfigurables seraient développées. Cependant il ne s'agit pas d'une vision futuriste, les architectures reconfigurables de gros grains existent déjà et sont promues à un bel avenir avec la réduction des technologies. Pour imaginer la complexité potentielle de ces circuits, quelques chiffres : avec la technologie SRAM 40nm un FPGA haut de gamme peut

contenir plusieurs milliards de transistors et peut réaliser plus de 2000 multiplications en virgule fixe avec des mots de 18 bits en moins de 20 ns. Ces chiffres laissent songeur sur les possibilités offertes aujourd'hui aux développeurs.

La suite de ce chapitre présente les architectures classiques de FPGA, ainsi que les éléments configurables que l'on trouve dans ces circuits.

3. Architectures des FPGA

L'architecture d'un FPGA est principalement décrite par la topologie des ressources de routages et des éléments logiques configurables de base. Il existe deux architectures classiques, l'architecture îlot de calcul (initialement utilisée dans les composants Xilinx) et l'architecture hiérarchique (initialement utilisée dans les composants Altera). Cependant, une tendance apparaît avec les dernières générations de circuits, les architectures sont principalement de style îlots de calculs avec une légère hiérarchique (un ou deux niveaux de cluster hiérarchique). Les sections suivantes donnent quelques détails sur ces architectures. Dans le passé, d'autres architectures ont été utilisées. On peut citer l'architecture de routage logarithmique utilisée par Xilinx pour son circuit XC6000. Malheureusement, les outils de placement routage n'étaient pas adaptés à cette architecture pour permettre au concepteur d'en tirer pleinement partie.

3.1. Architecture îlot de calcul

L'architecture la plus communément utilisée pour réaliser ces circuits est de type *îlot de calcul*. Dans ce cas les ressources configurables sont disposées sous formes de matrice, comme on peut le voir sur la figure 4. Des lignes de routage sont disposées horizontalement et verticalement autour des ressources configurables. Des blocs de connexion relient les ressources configurables aux lignes de connexion. Des matrices de connexion relient les lignes de routage horizontales et verticales.

L'utilisation de matrices de connexions configurables est indispensable pour assurer la connectivité des modules, mais les matrices de connexions configurables dégradent les caractéristiques des signaux, diminuent les performances (fréquence de fonctionnement et consommation de puissance) et nécessitent des outils de placement-routage efficaces [WIL 97]. Sur la figure 4, on peut voir en gras des liaisons point à point entre deux éléments configurables. Ces liaisons utilisent : les ports d'entrées/sorties des éléments configurables, les connexions configurables qui permettent la connexion des éléments configurables au réseau de routage, les lignes de routages et les matrices de connexions configurables. Autant d'éléments parcourus qui dégradent les performances du circuit mais qui permettent une flexibilité importante.

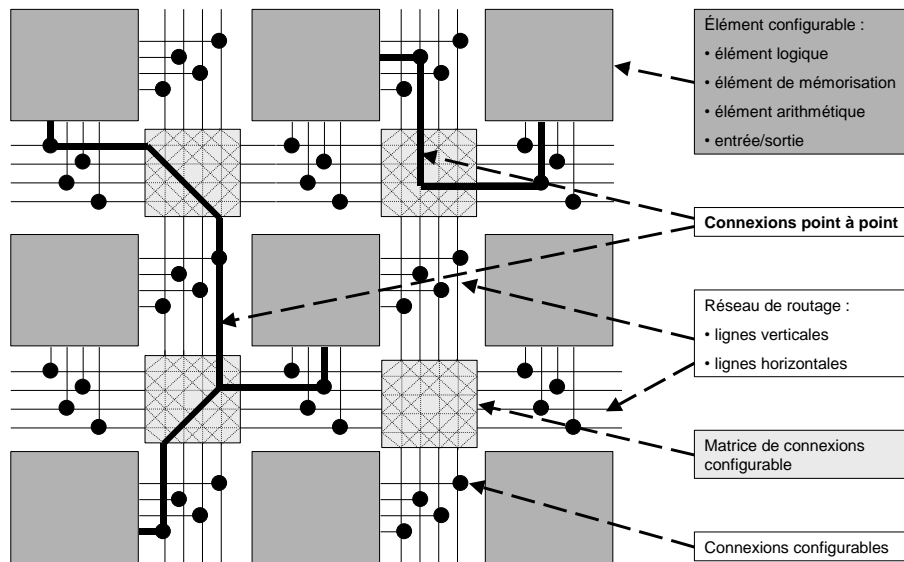


Figure 4. Architecture îlot de calcul, typique des FPGA actuels [BOS 04].

3.2. Architecture hiérarchique

L'architecture hiérarchique couramment utilisée pour les circuits FPGA est constituée de quatre ou trois niveaux. A chacun de ces niveaux des ressources de routage sont disponibles pour communiquer entre les éléments propres du circuit. La figure 5 schématise une architecture hiérarchique à quatre niveaux en utilisant un exemple d'architecture couramment rencontrée dans les FPGA ACTEL. Au niveau le plus haut de la hiérarchie, le circuit est constitué de tuiles agencées matriciellement. Les tuiles sont constituées de clusters logiques et de bancs de mémoires. Enfin, les clusters logiques regroupent les éléments logiques (et/ou arithmétiques) configurables. Ce style d'architecture peut être très efficace énergétiquement car elle permet de localiser les communications intenses et limite l'utilisation de longues lignes de routage. Cependant, il est nécessaire pour les algorithmes de placement et routage de prendre en compte les caractéristiques spécifiques de ces architectures [BOS 04].

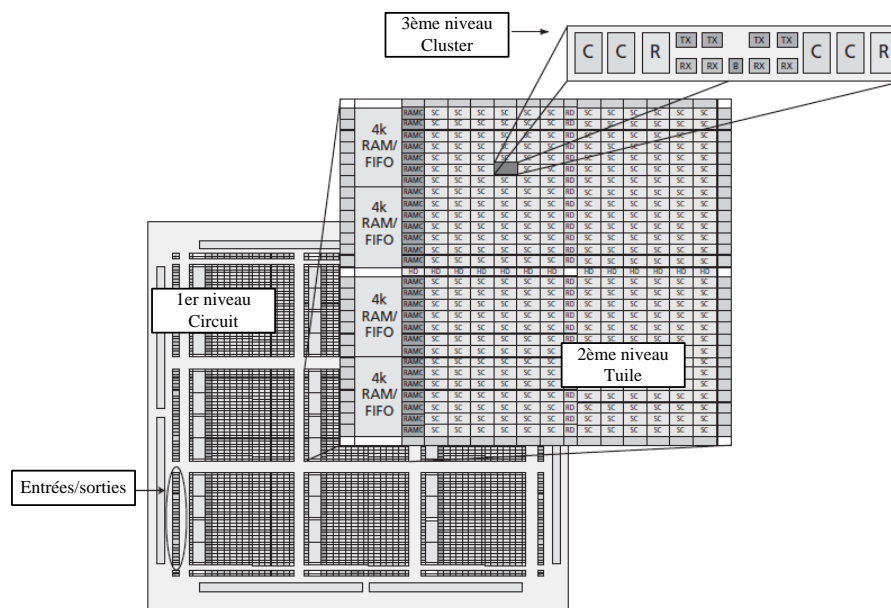


Figure 5. Exemple d'Architecture hiérarchique à quatre niveaux (circuit, tuiles, clusters, éléments configurables) que l'on rencontre fréquemment dans les circuits ACTEL [ACT 10a].

4. Ressources fonctionnelles configurables

4.1. Éléments logiques et mémoires embarquées

Dans la plupart des cas l'élément logique configurable de base des FPGA se compose d'une LUT (*Look Up Table*⁵) avec un nombre d'entrées allant de 4 à 8 pour les dernières générations, d'une chaîne de propagation rapide de la retenue et d'un registre de sortie afin d'assurer la synchronisation des signaux (très utile pour l'implémentation de calculs pipelinés). Ces éléments configurables peuvent être rassemblés en clusters hiérarchiques afin de favoriser une connectivité locale et rapide (cas des composants Altera). La figure 6 illustre l'augmentation de la taille des éléments logiques configurables (appelé *SLICE* chez Xilinx) entre deux générations successives de FPGA Xilinx : le Virtex4 [XIL 10b] et le Virtex6 [XIL 10c]. Cette différence entraîne donc la nécessité de spécifier correctement les résultats de synthèse étant donné qu'un SLICE Virtex4 ne correspond pas à un SLICE Virtex6.

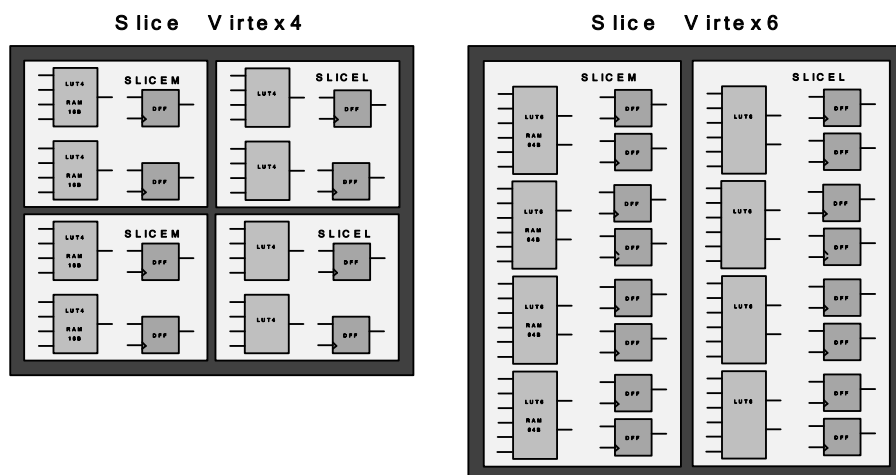


Figure 6. *Éléments logiques configurables (simplifiés) des circuits Xilinx Virtex4 et Virtex6.*

Quelques rares circuits ne comportent pas de LUT et sont constitués de cellules configurables basées sur des multiplexeurs. Ce fut le cas du circuit Xilinx XC6000 qui n'est plus disponible. C'est le cas pour certains composants ACTEL comme les

⁵ Une LUT peut être considérée comme une petite mémoire RAM dans laquelle on mémorise la table de vérité d'une fonction logique. Une LUT à 4 entrées et 1 sortie, classiquement utilisée dans les FPGA, est donc équivalente à une RAM 16 bits.

familles FPGA de technologie FLASH ProASIC-3 et Fusion ainsi que la famille de FPGA de technologie anti-fusible Axcelerator [ACTEL 10b].

Rapidement, afin de réaliser complètement des applications modernes, les FPGA ont dû se doter d'éléments configurables de mémorisation (apparition en 1999 dans les composants Virtex et Apex de Xilinx et Altera). Sans ceux-ci la mémoire synthétisée doit être distribuée sur les LUT, ce qui laisse peu de place pour les traitements.

4.2. Ressources arithmétiques de gros grain

Nombreuses sont les applications qui nécessitent la synthèse d'opérateurs du type multiplieur, additionneur et multiplieur/accumulateur. S'il est possible, grâce aux chaînes de propagation rapide de la retenue de réaliser sur un petit nombre de LUT des additionneurs efficaces, ce n'est pas le cas pour des multiplieurs très coûteux en ressources. Les industriels ont donc choisi d'implanter de façon matérielle des multiplieurs reconfigurables (la reconfiguration intervient en particulier sur la taille des données à traiter) au sein même de la matrice de grain fin. En positionnant ces multiplieurs près des colonnes d'éléments mémoires et d'éléments reconfigurables de grain fin il est possible de synthétiser des opérateurs MAC (Multiplieur Accumulateur). Cette solution fut retenue par Xilinx pour les composants Virtex-II. Altera a choisi d'implanter dans les circuits de la famille Stratix des opérateurs câblés plus complexes pouvant directement être configurés en opérateurs MAC que l'on trouve également sur les nouvelles générations de FPGA Xilinx.

La figure 7 donne le schéma (simplifié) d'un élément arithmétique configurable de dernière génération (DSP48E Slice) que l'on trouve dans les composants Xilinx Virtex-6. Ceux-ci comportent, entre autres, un multiplieur prenant en entrée un mot de 25 bits et un mot de 18 bits, une unité arithmétique (opération addition ou soustraction) et logique ainsi que des chaînes de retour pour les calculs itératifs (filtre IIR par exemple). En fonction du circuit choisi le nombre de DSP48E Slice est plus ou moins grand, il atteint 2016 éléments pour le plus gros des circuits Virtex-6 SXT qui est une version spécialisée pour le traitement numérique du signal.

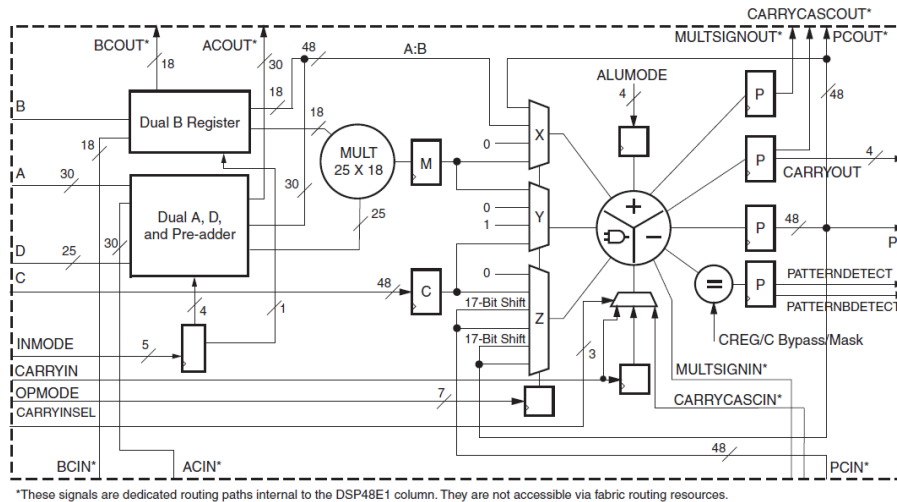


Figure 7. Elément arithmétique configurable du circuit Xilinx Virtex6 [XIL 09a].

4.3. Ressources programmables embarquées

Depuis les années 2000, la densité d'intégration des circuits FPGA permet de regrouper, sur une même puce, une matrice d'éléments matériels (logiques, mémoires, opérateurs arithmétiques, entrées-sorties) configurables et un ou plusieurs systèmes à microprocesseurs. Ce type de circuit permet de profiter du parallélisme de calcul offert par l'architecture matérielle et du contrôle séquentiel efficace offert par le système programmable (microprocesseur). Aussi en tirant parti des propriétés respectives des systèmes programmables et des systèmes reconfigurables il est possible d'améliorer l'adéquation du système global avec l'application développée. Dans ce cas l'utilisation de méthodes de conception conjointe logicielle/matérielle est indispensable et demande un effort important en développement d'outils [CTI 98].

Plusieurs architectures existent aujourd'hui, la figure 8 illustre ces différentes possibilités. Dans certains circuits, la partie matérielle configurable et la partie programmable sont séparées par un bus spécifique. La partie programmable comprend le système à microprocesseur dans son ensemble : cœur de processeur, mémoires caches, périphériques, interface etc ... Ce fut le cas du premier circuit commercial embarquant un cœur de processeur, le circuit Altera Excalibur qui regroupait une matrice FPGA APEX 20KE et un cœur de processeur ARM9 (32 bits) fonctionnement à 100MHz accompagné de deux fois 8kilo-octets de mémoire cache (instructions + données). Malheureusement, au moment de la sortie de ce composant, les outils n'étaient pas matures pour permettre une utilisation simple et efficace dans un contexte industriel.

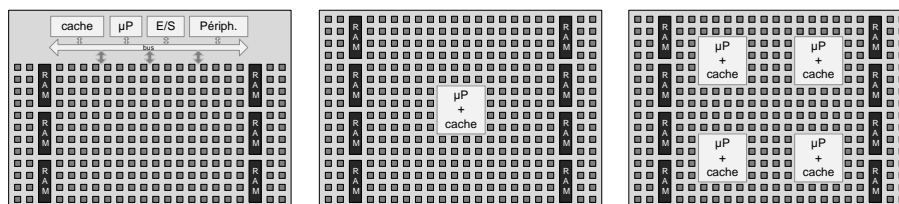


Figure 8. *Trois architectures possibles de circuits mixtes FPGA-microprocesseur(s) [BOS 10].*

Certains circuits embarquent *profondément* au cœur de la matrice configurable le ou les processeurs. Ceux-ci ne sont pas nécessairement accompagnés de leur système, mais des outils logiciels permettent de configurer une partie de la logique pour constituer le système complet. Cela permet ainsi une plus grande flexibilité dans le choix du système programmable. C'est cette solution qui fut choisie par Xilinx pour son premier composant mixte Virtex-II Pro. Celui-ci était composé d'une matrice Virtex-II Pro et de un à quatre cœurs IBM PowerPC 405 (32 bits) cadencés à 400MHz disposant de deux fois 16 kilo-octets de mémoire cache (instructions et données). Cette même architecture est toujours utilisée pour les composants Virtex de dernière génération.

Il faut noter qu'il est possible de réaliser un circuit mixte sans disposer physiquement d'un cœur de processeur embarqué dans le circuit FPGA. Dans ce cas l'utilisation d'un cœur synthétisable (dit *cœur soft*), fourni gratuitement par le fabricant de circuit, est une solution très efficace. Les cœurs synthétisables 32 bits Altera NIOS [ALT 09] et Xilinx MicroBlaze [SUN 09] sont par exemple très utilisés. Bien entendu, les performances de ces cœurs sont inférieures à celles des cœurs embarqués, mais ils permettent une plus grande flexibilité de configuration.

5. Configuration et reconfiguration des FPGA

La configuration des FPGA peut prendre différentes formes, que ce soit pour configurer les ressources opératoires ou que ce soit pour configurer les réseaux de routage. Cependant, les FPGA de technologie anti-fusible ne sont configurables qu'une seule fois. En anglais, on parle de circuits OTP (*One Time Programmable*). Notons l'utilisation en anglais du mot « *programmable* » alors que les circuits FPGA ne sont pas programmables (ils n'exécutent pas un programme) mais ils sont configurables (la configuration mémorisée de tous les éléments fonctionnels et de routage du FPGA lui permet de réaliser une application). Les seuls FPGA reconfigurables (configurables plusieurs fois) sont les circuits de technologie FLASH (non volatile) et les circuits de technologie SRAM (volatile et qui nécessite la sauvegarde externe dans une mémoire non volatile de la configuration)

Dans ces deux derniers cas, la reconfiguration de l'architecture peut être mise en œuvre de plusieurs façons au cours de l'exécution de l'application. Elle peut intervenir une seule fois sans remise en cause durant l'exécution de l'application. Nous parlerons dans ce cas de reconfiguration statique. Les processus de reconfiguration et d'exécution de l'application sont, dans ce cas, distincts et clairement séparés dans le temps. Une nouvelle reconfiguration sera effectuée dans deux cas. Tout d'abord elle peut être la conséquence d'une perte de la configuration pouvant être due pour certains dispositifs à un arrêt de la source d'alimentation (cas des FPGA de technologie SRAM sans sauvegarde externe de la configuration). Mais elle peut aussi faire suite à une modification de l'application par le concepteur si celle-ci est défectueuse ou peut être améliorée (cas du prototypage).

Cependant une étude approfondie de l'exécution d'une application peut éventuellement mettre en évidence que certaines parties configurées de l'architecture ne sont nécessaires qu'un temps court vis à vis de la durée de l'exécution, et peuvent par contre utiliser une place conséquente dans l'architecture. Le ratio entre le temps d'utilisation de la partie configurée et la taille qu'elle occupe sur l'architecture peut être faible. D'où l'idée d'introduire un dynamisme temporel à la reconfiguration. Lorsqu'une partie de l'application est exécutée, et dans le cas où elle ne l'est plus dans un temps proche, on peut reconfigurer les éléments qui lui étaient dédiés afin de les utiliser pour une autre partie de l'application. Cette reconfiguration se fait concurrentiellement à l'exécution de l'application.

La reconfiguration dynamique permet donc d'optimiser la surface configurée dans le temps. Etant donné qu'il s'agit de modifier dans le temps uniquement une partie des éléments configurables, il faut nécessairement utiliser une architecture à reconfiguration partielle. Ici l'inconvénient vient qu'il est nécessaire de bien déterminer le partitionnement dans le temps de l'application afin de profiter de toute la surface du circuit. Il peut exister des problèmes de fragmentation comme sur les disques durs des ordinateurs [COM 99]. De plus, il faut correctement établir les communications entre les partitions [DEL 07].

La figure 9 présente schématiquement le déroulement de la reconfiguration durant l'exécution de l'application dans deux cas : la reconfiguration complète et la reconfiguration partielle dynamique. Durant le temps d'exécution de l'application la reconfiguration complète est statique contrairement à la reconfiguration partielle qui sous certaines conditions permet de reconfigurer une partie du circuit en cours de fonctionnement sans affecter les parties configurées qui ne doivent pas être modifiées, on parle alors de configuration dynamique.

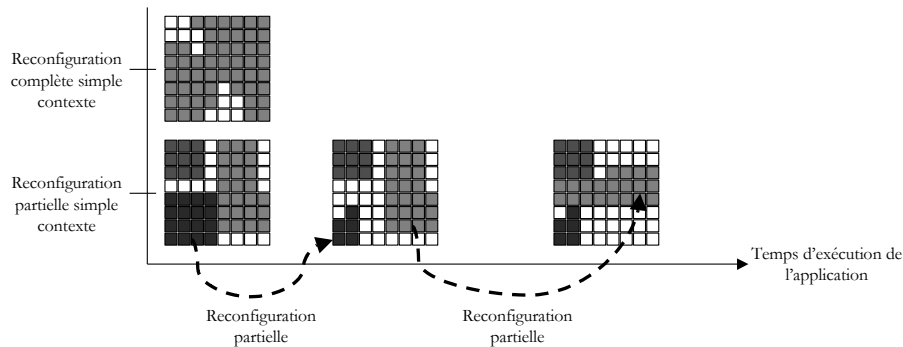


Figure 9. *La reconfiguration simple contexte et partielle durant le temps d'exécution de l'application.*

Pour chacun des cas de la figure 9, un carré gris représente un élément configuré, un carré blanc représente un élément non utilisé. Si deux carrés sont de niveaux de gris différents alors c'est qu'ils sont dédiés à la réalisation de deux parties (ou tâches) distinctes de l'application (notamment dans le cas de la reconfiguration partielle).

Dans tous les cas la gestion de la configuration, en particulier avec la notion de reconfiguration dynamique, est complexe. Le plus souvent il faut ajouter un circuit auxiliaire pour la gérer. Ce circuit peut être un processeur externe ou interne à l'architecture. Il est possible d'utiliser les mémoires internes à l'architecture comme mémoires caches de configurations. C'est la solution proposée par Xilinx dans son système d'auto-reconfiguration, basé sur l'utilisation d'un processeur embarqué de type MicroBlaze [BLO 03], [ULM 04]. Un chapitre de ce traité reviendra plus en détails sur la reconfiguration dynamique des FPGA.

6. Tendances

Quelques tendances apparaissent ces dernières années dont l'objectif est d'augmenter l'efficacité des FPGA. Sans vouloir être exhaustif nous pouvons en décrire quelques unes. Au niveau de la mémoire de configuration, les densités de configuration et la maîtrise des technologies ont permis l'émergence de composants de technologie mixte FLASH-SRAM, tel que les composants Lattice XP2 [LAT 10] et Xilinx Spartan AN [XIL 09b]. L'offre commerciale a évolué en proposant de plus en plus des séries spécialisées à l'intérieur d'une même famille de composants. Par exemple il existe aujourd'hui des FPGA spécialisés pour le traitement numérique du signal, pour l'utilisation de processeurs embarqués, pour les communications très haut débit (GigaBits Ethernet) etc...

De nombreux FPGA embarquent des parties analogiques, dans la plupart des cas il s'agit de convertisseurs analogiques numériques aux performances limitées qui permettent au composant d'évaluer son environnement (température, pression etc...). Cependant, le circuit FPGA, de technologie FLASH, ACTEL FUSION propose une utilisation sur plusieurs voies analogiques de plusieurs convertisseurs analogique numérique ce qui permet de connecter directement le FPGA à un nombre plus important de sources (jusqu'à une trentaine d'entrées analogiques) [ACT 10c].

De nombreuses évolutions ont eu lieu dans le domaine des applications, parmi celle-ci il est à noter l'importance croissante des FPGA dans le domaine de la sécurité et de la cryptographie appliquée. Cette évolution a entraîné d'importantes questions en ce qui concerne la sécurité des FPGA et de leur configuration [BOS 04b], de nombreux travaux sont en cours dans ce domaine [BOS 08].

7. Conclusion

La part de marché des FPGA dans le marché globale des circuits matériels pour l'électronique numérique ne cesse d'augmenter. Les évolutions technologiques et architecturales qui ont eu successivement lieu depuis le début des années 2000 ont fait de ces circuits de réels et rentables alternatives aux classiques ASIC. Avec ces évolutions c'est tout un nouveau domaine de l'électronique numérique qui s'est ouvert. Aujourd'hui les FPGA sont utilisés dans tous les domaines, des systèmes embarqués aux systèmes de communications, ils sont au cœur d'un important champ de recherche académique et industrielle.

8. Bibliographie

- [ALT 10a] ALTERA Coporation <http://www.altera.com>
- [ALT 09] ALTERA Corporation, NIOS II Processor Reference Handbook, ver9.1, November 2009.
- [ACT 10a] ACTEL Coporation. www.actel.com
- [ACT 10b] ACTEL Corporation, Devices Overview, 2010
<http://www.actel.com/products/devices.aspx>
- [ACT 10c] ACTEL Corporation, Mixed Signal FPGAs, 2010
<http://www.actel.com/products/mixed-signal-fpga/default.aspx>
- [BLO 03] B. Blodget, P. James-Roxby, E. Keller, S. McMillan and P. Sundararajan. « A Self-reconfiguration Platform », in *proceeding of 13th International Conference on Field-Programmable Logic and Applications, FPL'03*, Lisbon, Portugal, September 2003.
- [BOS 08] L. BOSSUET, G. GOGNIAT, *La sécurité matérielle des systèmes embarqués*, Chapitre 5 du Traité IC2, série réseaux et télécoms, Les systèmes embarqués

© Lilian Bossuet, 2010

communicants : mobilité, sécurité, autonomie, aux éditions Hermes Science, septembre 2008.

- [BOS 04a] L. Bossuet, *Exploration de l'espace de conception des architectures reconfigurables*, Thèse de doctorat, Université de Bretagne Sud, Lorient, septembre 2004.
- [BOS 04b] L. Bossuet, G. Gogniat, and W. Burleson, « Dynamically Configurable Security for SRAM FPGA Bitstreams », in *Proceedings of the 11th Reconfigurable Architectures Workshop (RAW 2004)*, Santa Fé, New Mexico, USA, avril 2004.
- [BOS 10] L. Bossuet, *Architecture Conception et Utilisation des FPGA*, Cours de l'ENSEIRB 2010, disponible librement à l'adresse
http://www.enseirb-matmeca.fr/~bossuet/Bossuet_cours_FPGA_ENSEIRB.pdf
- [COM 99] Katherine Compton. *Programming Architectures For Run-Time Reconfigurable Systems*. Master's Thesis, Dept of ECE, Northwestern University, Evanston, IL USA. December 1999
- [CTI 98] C.T.I. COMETE (CENT, LIRMM, TIMA, IRESTE, IRISA, LAMI). *CODESIGN, conception conjointe logiciel-matériel*. Collection technique et scientifique des télécommunications (CTST), édition Eyrolles, Juin 1998.
- [DEL 07] J.P. Delahaye, *Plate-forme hétérogène reconfigurable : application à la radio logicielle*, Université de Rennes 1, Rennes, avril 2007.
- [LAT 10] Lattice, FlexiFLASH Architecture web page,
<http://www.latticesemi.com/products/fpga/xp2/flexiflasharchitecture.cfm>
- [MAX 04] C. MAXFIELD, *The Deign Warrior's Guide to FPGAs*, ELSEVIER, ISBN 0-7506-7604-3, 2004.
- [SUN 09] N. Sundaramoorthy, *Simplifying Embedded Hardware and Software Development with Targeted Reference Designs*, Xilinx White Paper, December, 2009.
- [ULM 04] M. Ulmann, M. Hübner, B. Grimm, J. Becker. « An FPGA Run-Time System for Dynamical On-Demand Reconfiguration », in *proceeding of the 11th IEEE Reconfigurable Architectures Workshop, RAW 2004*, Santa Fé, New Mexico, USA, 2004.
- [WIL 97] S. J.E. Wilton. *Architectures and Algorithms for Field Programmable Gate Arrays with Embedded Memory*. Ph.D Thesis, University of Toronto, Canada, 1997.
- [XIL 10a] XILINX Coporation, <http://www.xilinx.com>
- [XIL 10b] Xilinx, Virtex-4 FPGA Data Sheets,
<http://www.xilinx.com/support/documentation/virtex-4.htm>
- [XIL 10c] Xilinx, Virtex-6 FPGA Data Sheets,
<http://www.xilinx.com/support/documentation/virtex-6.htm>
- [XIL 10d] Xilinx, *Embedded Processor Block in Virtex-5 FPGAs*, Xilinx User Guide, UG200, February 2010.

Les FPGA, technologie, architecture et utilisation

- [XIL 09a] Xilinx, Virtex-6 FPGA DSP48E1 Slice, Xilinx User Guide, UG369 v1.2, September 16, 2009.
- [XIL 09b] Xilinx, Spartan-3AN FPGA In-System Flash User Guide, Xilinx User Guide, UG333 v2.1, January 15, 2009.