Université de Bretagne Sud

Habilitation à Diriger des Recherches
Sciences pour L'ingénieur, Mention Electronique

Contribution au domaine de la conception des Systèmes Embarqués Reconfigurables

Partie 3

Par

Guy Gogniat

Laboratoire LESTER
Université de Bretagne Sud – CNRS FRE 2734
Centre de Recherche
56321 Lorient Cedex
France

Table des matières

Partie 3 : Travaux de recherche détaillés et perspectives

Cette troisième partie présente de façon approfondie les différents travaux que j'ai menés depuis l'obtention de mon doctorat. Elle propose tout d'abord une introduction afin de positionner les différentes contributions suivant les 3 axes de recherche autour desquels s'articulent mes travaux. Ensuite chaque axe est détaillé et une sélection de certains travaux est proposée afin d'illustrer l'activité menée. Enfin une conclusion et des perspectives sont proposées afin de préciser les actions envisagées dans l'avenir.

1. Introduction

La dimension numérique de nos sociétés a profondément muté en l'espace d'une dizaine d'années. Il apparaît clairement aujourd'hui que son assise est durable et que le mouvement ne pourra qu'accroître et se décupler dans l'avenir. Les utilisateurs se sont familiarisés avec cette nouvelle dimension technologique et en attendent aujourd'hui davantage afin de les accompagner dans leur vie quotidienne professionnelle et personnelle. Il est clair que l'avenir sera connecté et multimédia, l'utilisateur aura en permanence un accès au monde numérique, ce qui n'est pas sans poser de problèmes entre la richesse et le confort que cela apportera à l'utilisateur mais aussi le risque de servilité face à cette quantité incroyable d'information et d'instantanéité.

Parallèlement à ces profonds changements sociétaux, une autre mutation est en marche. La face cachée de cette révolution technologique se situe dans les laboratoires où les chercheurs, les concepteurs, les ingénieurs développent et imaginent les nouvelles technologies de demain. De nombreux défis sont à relever afin de matérialiser ces rêves technologiques.

En effet, que de changements depuis 1994, date de ma première expérience dans le monde de la recherche, que d'évolutions architecturales, quelle accélération des ruptures technologiques. En un peu plus de dix ans le paysage des systèmes numériques a profondément changé.

Dans les années 1990 les architectures hétérogènes embarquées faisaient leur apparition. Certes le modèle architectural était modeste avec un processeur et un ASIC, mais les concepts étaient posés. Aujourd'hui les architectures se sont énormément complexifiées avec l'émergence des systèmes multiprocesseurs sur silicium [Jerraya 2004]. Plusieurs processeurs pouvant être connectés à travers des architectures de communication très diversifiées. Dans les années 1990 le bus était l'élément fondamental d'une communication. Depuis, les bus hiérarchiques, les matrices d'interconnexion et maintenant les réseaux sur silicium ont fait leur apparition [Evain 2006a][Evain 2006b]. Les communications deviennent un des points les plus critiques pour les concepteurs. Comment faire communiquer des dizaines de composants interconnectés ?

Les sauts technologiques se sont également enchaînés à un rythme soutenu comme l'avait anticipé l'incontournable loi de Moore [Moore 1965]. Au milieu des années 1990 la technologie CMOS était en 0.35 µm, puis le cap du submicronique a été franchi au milieu des années 2000. Aujourd'hui le 45 nm voit le jour et bientôt les progrès nous promettent du 32 nm. Parallèlement de nouvelles technologies s'installent dans notre paysage et sont promises à un bel avenir. Il s'agit par exemple des nano tubes de carbone mais aussi des technologies moléculaires [ITRS 2007].

Comment être exhaustif tant les changements ont été nombreux et spectaculaires, à l'image de l'évolution des technologies reconfigurables. Dans les années 1985 les premiers FPGA font leur apparition et s'installent progressivement comme éléments incontournables des systèmes numériques. Leurs architectures ont énormément évolué en suivant toujours les besoins des applications. Progressivement des éléments de calcul gros grain, des mémoires, des connexions rapides et des processeurs embarqués se sont introduits dans ces composants [Wilton 1999][Chow 1999a][Chow 1999b][Betz 1999][Clifford 2000][Ahmed 2000][Xilinx 2007]. Les FPGA sont devenus aujourd'hui des systèmes sur silicium intégrant plusieurs centaines de millions de transistors et utilisant les dernières technologies afin de faire bénéficier les utilisateurs des meilleures performances actuelles.

Les FPGA sont les précurseurs des systèmes de demain où des centaines de "cellules" seront interconnectées à travers des réseaux de communication configurables conduisant à

la définition de systèmes flexibles. Mais continuons ce rapide survol des changements majeurs de ces dix dernières années.

Le domaine des applications a également profondément évolué avec l'apparition à un rythme soutenu de nouveaux standards que cela soit pour les télécommunications, le multimédia ou la sécurité. Par exemple, le domaine des télécommunications a vu se succéder en l'espace de quelques années la 2G avec le GSM et le CDMA, puis la 2.5G dans les années 2000 avec les standards GPRS et EDGE, ensuite la 3G avec l'UMTS et le CDMA2000 et la 3.5G avec le HSPDA et enfin la 4G qui va apparaître prochainement et qui devrait être déployée en Europe dans les années 2015 [Kim 2006].

Les téléphones équipés de cette technologie pourront télécharger des films de 2h en l'espace de seulement 10 secondes, soit environ 1300 fois plus rapidement que la 3.5G lancée au japon en 2006 [Rouffet 2005].

Parallèlement à ces standards le domaine des réseaux a également vu un grand nombre de solutions apparaître avec notamment les technologies Wi-Fi et Bluetooth. Cette multiplication des standards et le rythme soutenu de leurs apparitions n'est pas sans poser de problèmes pour les concepteurs qui doivent offrir aux utilisateurs un nombre élevé d'interfaces de communication tout en garantissant la possibilité de mise à niveau de leur produit afin d'anticiper les évolutions prévisibles.

Ces évolutions majeures au niveau des architectures et des applications ont inévitablement eu un impact significatif sur les méthodologies de conception. Le rythme des innovations est soutenu dans les laboratoires de recherches et au sein des organismes internationaux tels que l'IEEE [IEEE 2007] ou l'OMG [OMG 2007]. Certes leurs transferts vers le monde industriel est parfois moins spectaculaire tant il est difficile de rendre robuste ces outils de grande complexité et de faire adopter de nouvelles méthodologies de conception aux concepteurs; les changements culturels prennent du temps mais sont néanmoins incontournables.

Dans les années 1990 le codesign faisait ses premiers pas avec des articles de référence tel que celui de Gupta et De Micheli intitulé "Hardware-Software Cosynthesis for Digital Systems" [Gupta 1993]. Bien que le modèle architectural visé était relativement simple, puisque composé d'un processeur connecté à un ASIC via un bus, il n'en demeure pas moins que les idées étaient présentes.

Malheureusement, plus de 10 ans après, les outils de codesign n'ont pas encore réussi à s'imposer dans l'industrie tant le problème est complexe et l'espace des solutions est vaste. Certaines tentatives, notamment par Cadence avec l'outil Virtual Component Co-Design (VCC) au début des années 2000 [Santarini 2000], ont été menées mais ces dernières n'ont pas réussi à convaincre les concepteurs malgré quelques bonnes réussites.

Toutefois, une évolution forte résultant de ces premiers pas vers le codesign est l'émergence depuis quelques années du domaine de la conception au niveau ESL (*Electronic System-Level Design*) [ESL 2005][Burton 2006][Martin 2007]. L'idée est de développer un flot de conception unifié où le concepteur puisse progressivement raffiner sa spécification afin d'aller d'une description comportementale vers une description RTL. Pour cela le concepteur s'appui sur un langage unique comme par exemple SystemC qui a fait son apparition au début des années 2000 et qui aujourd'hui s'impose de plus en plus chez les concepteurs [SystemC 2005].

En 2004, 40,4% des compagnies internationales sondées par l'agence *Open SystemC Initiative* (OSCI) avaient déclaré avoir utilisé SystemC pour un ou plusieurs de leurs projets. En 2007, les tendances évoluent positivement avec 15% des compagnies ayant développées plus de dix projets en SystemC et plus de 50% déclarant avoir utilisé SystemC pour un ou plusieurs de leurs projets [OSCI 2007]. Cette évolution repose sur la capacité du langage SystemC à décrire différents niveaux d'abstraction, depuis le niveau TLM (*Transaction Level Modeling*) jusqu'au niveau RTL en introduisant plus ou moins finement des modèles temporels [Ghenassia 2005]. Des travaux sont encore nécessaires afin de mieux préciser les niveaux de raffinements car aujourd'hui les frontières sont encore floues

et parfois contradictoires. Les bases des approches unifiées sont posées et il semble indéniable que l'avenir verra leur positionnement se renforcer.

Une autre évolution forte, connexe au problème exposé à l'instant, concerne le rapprochement toujours plus marqué entre les domaines du matériel et du logiciel. Cette évolution résulte de l'intégration massive de logiciel dans les systèmes embarqués; la taille de code au sein de ces systèmes doublant tous les 10 mois. Cette tendance exponentielle s'explique en autre par le besoin de réutilisation nécessaire afin de faire face aux exigences du marché qui requièrent un renouveau constant des produits.

Dans les années 1980 écrire quelques dizaines de milliers de lignes de code assembleur pour des systèmes embarqués était une opération héroïque, aujourd'hui un téléphone portable contient près de cinq millions de lignes de code C ou C++ [Walls 2005]. Cet accroissement de complexité s'est également accompagné par l'intégration courante de systèmes d'exploitation dans les systèmes embarqués. Ces évolutions ont un impact significatif sur les outils et pour les concepteurs qui voient leurs métiers évoluer.

Dans les années 1990 un concepteur se spécialisait soit vers le domaine de l'analogique soit vers le domaine du numérique, aujourd'hui ce spectre s'ouvre largement avec l'apparition du logiciel pour les systèmes embarqués. Aussi, le concepteur de systèmes numériques doit également appréhender les technologies du logiciel. Les conséquences au niveau des outils ne sont pas négligeables puisqu'elles imposent de pouvoir dimensionner au plus tôt les plateformes d'exécution et d'en évaluer leurs performances. Pour cela une élévation du niveau d'abstraction se fait sentir et depuis quelques années les approches basées sur le langage UML et dirigées par des transformations de modèles commencent à apparaître [UML 1997][MDA 2003]. Ce type d'approche vise à se détacher des plateformes d'exécution afin de pouvoir mettre en œuvre des vérifications fonctionnelles et non fonctionnelles à plusieurs niveaux de conception.

D'autres évolutions ont vu le jour ces dernières années avec le développement d'outils permettant la mise en évidence des potentiels d'accélération au sein d'une application [Synfora 2007]. Ces outils souvent associés à des processeurs configurables permettent d'étendre l'architecture afin de mettre en œuvre des accélérateurs matériels [Tensilica 2007]. D'autres travaux se sont également intéressés à la gestion de la reconfiguration des systèmes [Compton 1999][Blodget 2003][Ulmann 2004][Delahaye 2004]. Ce point, encore délicat, nécessitera davantage d'études avant d'aboutir à des solutions matures. Concernant les outils il est incontournable de citer les changements au niveau des métriques d'exploration de l'espace de conception. En effet, la consommation s'installe depuis quelques années comme un des éléments majeurs à optimiser. La fiabilité devient également importante du fait des ruptures technologiques et des densités d'intégration.

Cette analyse bien évidemment non exhaustive des évolutions passées et de certaines tendances à venir est intéressante afin de mettre en regard les travaux que j'ai menés depuis l'obtention de mon doctorat en 1997 et souligner leur cohérence par rapport aux évolutions auxquelles nous assistons. L'ensemble des travaux menés depuis 1997 couvre un large spectre, comme nous le verrons dans la suite de ce document, afin d'adresser différents points très importants des systèmes embarqués. Il s'agit des flots de conception, des architectures reconfigurables et des nouvelles contraintes à savoir la sécurité.

Pour conclure cette introduction et comme indiqué au début de ce chapitre, la révolution technologique se situe également dans les laboratoires où les chercheurs, les concepteurs et les ingénieurs doivent imaginer et développer les technologies de demain. A travers l'ensemble de mes travaux j'ai modestement participé à cette révolution en essayant d'anticiper les évolutions et les besoins à venir aussi bien en ce qui concerne les flots de conception à travers le langage UML et les approches de pré partitionnement, que les architectures à travers le caractère dynamique des futurs systèmes autour des architectures reconfigurables. Le domaine des applications a également été profondément abordé à travers notamment la problématique émergente de la sécurité des systèmes.

La suite de ce document reprend et détaille les différentes activités de recherche que j'ai menées ces dernières années afin d'illustrer les problématiques qui ont été abordées. Le chapitre suivant propose tout d'abord un positionnement de mes travaux selon trois axes de recherche : les systèmes embarqués (HW/SW codesign), les architectures reconfigurables et la sécurité des systèmes embarqués. Les Chapitres 3, 4 et 5 détaillent ensuite chacun des trois axes et proposent pour chaque axe deux exemples de travaux afin d'illustrer les contributions qui ont été apportées. Une fiche de synthèse est également proposée pour chaque axe afin d'établir un bilan succinct des travaux. Enfin, le Chapitre 6 conclut ce document et propose plusieurs thèmes importants à adresser dans le futur afin de faire face aux évolutions à venir et aux défis à relever.

2. Positionnement des travaux

Comme introduit dans le chapitre précédent les systèmes embarqués présentent une complexité croissante. Afin d'illustrer ce propos et de positionner plus précisément les travaux de recherche menés ces dernières années, il est intéressant de s'attarder sur trois exemples représentatifs de systèmes intégrés qualifiés de SoC (*System on Chip*).

La Figure 1 décrit l'architecture d'un système multistandard (802.11a et MCCDMA) pour des applications du type WLAN [Cambonie 2003]. Ce circuit, qui est développé par la société ST Microelectronics, contient un grand nombre d'IP et une architecture de communication complexe basée sur des bus hiérarchiques afin de garantir les débits en parallèle entre les différentes ressources du système. Par ailleurs, il possède des unités reconfigurables du type grain fin (i.e. FPGA) et gros grain afin de pouvoir gérer à la fois et de façon efficace des opérations au niveau bit mais également au niveau mot. L'introduction d'unités reconfigurables au sein du système est intéressante et permet de garantir une certaine pérennité au niveau du composant afin de faire face à de nouvelles évolutions des standards mais aussi d'effectuer des mises à jour matérielles du système.

Le deuxième exemple d'architecture correspond au circuit MP211 développé par la société NEC pour des applications du type téléphone cellulaire (Figure 2) [Torii 2005]. Là encore un grand nombre de ressources est intégré au sein du SoC avec trois CPU, un DSP et des processeurs spécialisés (e.g. processeur graphique 2D/3D). Les applications visées sont les traitements multimédias comme la vidéophonie ou encore le standard DVB-H qui est une adaptation du DVB-T, le système pour la télévision terrestre numérique, aux exigences des récepteurs de portables. Il est également intéressant d'observer qu'une unité matérielle dédiée à la sécurité a fait son apparition. La sécurité pour ce type d'application devient donc un paramètre important de la conception et il apparaît clairement que sa mise en œuvre ne sera plus exclusivement gérée de façon logicielle afin de pouvoir respecter les contraintes de performances (débit et consommation) mais aussi afin de garantir un

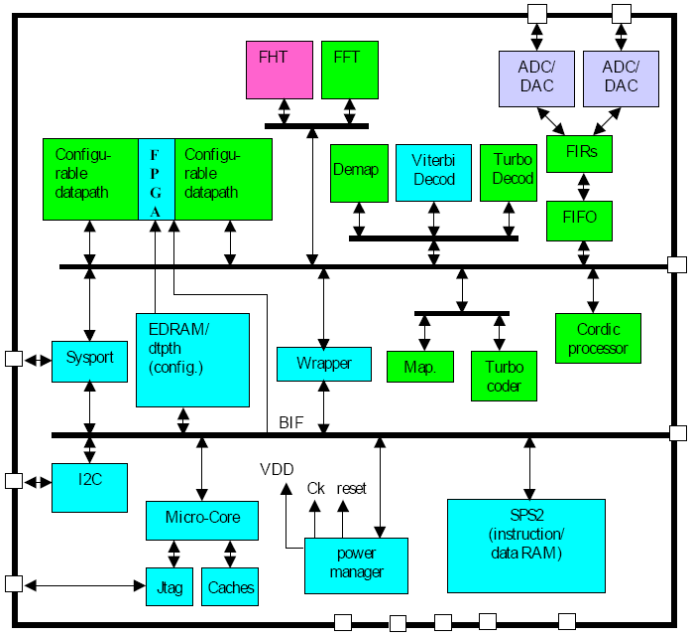


Figure 1 • Architecture du composant multistandard WLAN (802.11a et MCCDMA) de ST Microelectronics.

meilleur niveau de sécurité notamment contre les attaques par canaux cachés [Guilley 2004].

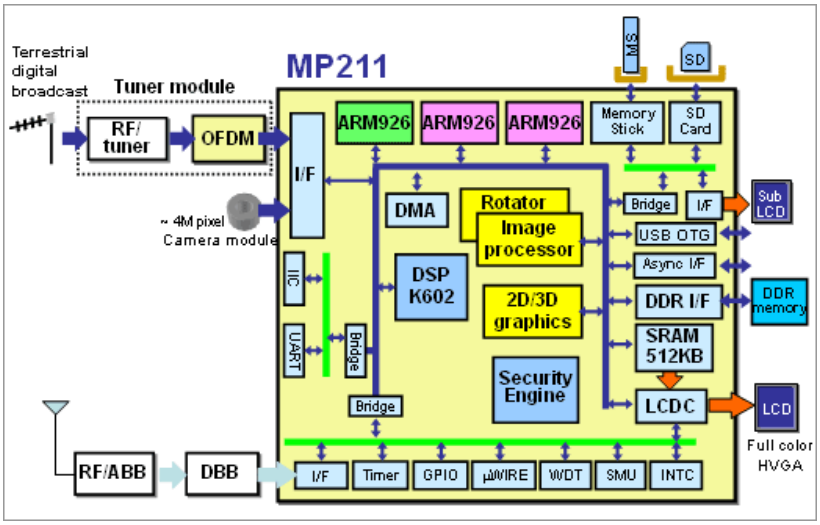


Figure 2 • Architecture du composant MP211 pour les applications téléphone cellulaire de NEC.

Le troisième exemple d'architecture correspond au composant Tera-scale en cours de développement dans les laboratoires d'Intel (Figure 3) [Tera 2007][Shekhar 2007]. Cet exemple est intéressant car il illustre les évolutions à venir en terme d'intégration de composants ou des dizaines de processeurs seront connectés au sein d'un SoC. La définition de l'architecture de communication sera un point critique afin de tirer profit du parallélisme possible et de réduire les conflits au niveau des échanges de données. Par ailleurs comme dans l'exemple précédent des processeurs dédiés seront présents afin de prendre en charge les calculs spécifiques à des classes d'applications (e.g. cryptographie, graphisme).

Cette présentation d'architectures de SoC pourrait encore se décliner selon de nombreux exemples mais ces trois systèmes sont intéressants car ils illustrent les tendances qui se dessinent et auxquelles nous seront confrontés. Parmi celles-ci on peut citer le parallélisme

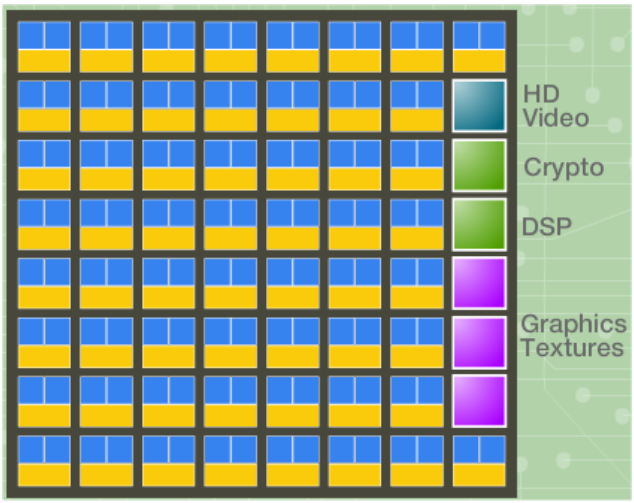


Figure 3 • Architecture du composant Tera-scale d'Intel.

massif, la reconfiguration et la sécurité des systèmes. La gestion de la consommation est également un point critique. En tant que concepteur il est donc important de se projeter dans les années à venir afin d'anticiper les besoins en méthodologies de conception et en architectures des systèmes.

Les travaux que j'ai menés ces dernières années s'inscrivent dans ces problématiques et adressent certains points particuliers. Durant ma thèse, le problème des interconnexions a été étudié [Gogniat 1997]. En effet, face à la difficulté croissante d'intégration des composants au sein d'une architecture hétérogène mes travaux ont conduit à la définition d'une architecture de communication homogène basée sur des bus et des FIFO afin de minimiser le coût des communications et faciliter l'intégration des composants [Gogniat 2000]. Chaque composant est connecté à l'architecture de communication à travers une interface générique. Je me suis également intéressé aux étapes terminales du cycle de conception : synthèse des communications et intégration des composants dans l'architecture cible. Depuis les architectures de communication du type NoC (*Network on Chip*) ont fait leur apparition et il est raisonnable de penser que l'avenir verra leur utilisation se généraliser. En effet, les concepteurs atteignent aujourd'hui les limites des architectures à base de bus pour des systèmes complexes intégrant des dizaines de ressources (IP et processeurs) [Donghyun 2007][Cidon 2007]. Toutefois, les concepts associés aux NoC devront encore s'étendre afin d'introduire davantage d'adaptabilité au sein des communications. La gestion dynamique de la qualité de service sera un paramètre important afin de garantir le bon fonctionnement du système [Ciordas 2007].

Mes travaux se sont poursuivis dans le domaine de la conception des systèmes complexes. Mon activité de recherche s'est articulée autour de trois axes de recherche : les systèmes embarqués (HW/SW codesign), les architectures reconfigurables et la sécurité des systèmes embarqués. Ces trois axes visent à faire le lien entre les spécifications de haut niveau et les architectures sous-jacentes pour les systèmes embarqués.

L'axe de recherche HW/SW codesign s'intéresse aux méthodologies de conception et plus particulièrement aux techniques permettant d'explorer l'espace de conception à partir d'une spécification de haut niveau. L'objectif de cet axe de recherche est de lever les verrous de conception relatifs à la mise en œuvre de systèmes complexes et d'anticiper les besoins futurs en terme de flot de conception. Pour cela des travaux autour du langage UML et des transformations de modèles ont été et sont actuellement menés [Rouxel 2006a]. Il s'agit de proposer des approches de conception dites MDA basées sur des modèles pour des applications à comportement statique et dynamique. D'autres travaux, visant à définir un pré partitionnement fonctionnel avant la définition précise de l'architecture, ont été développés [Maalej 2006]. Cette étape bien qu'encore peu étudiée aujourd'hui dans les flots de conception verra sont intérêt se renforcer à l'avenir dans la mesure où l'étape de conception fonctionnelle ne pourra plus être gérée manuellement. Plusieurs études concernant les OS ont également eu lieu dans la mesure où leur intégration dans les systèmes embarqués devient de plus en plus systématique.

L'axe de recherche concernant les architectures reconfigurables s'est intéressé à l'exploration des architectures grain fin et gros grain. L'objectif était initialement de pouvoir évaluer très tôt dans le flot de conception les performances d'une application implémentée sur une architecture FPGA [Bilavarn 2006]. Ensuite ces travaux ont été étendus afin d'adresser les limites des architectures FPGA à savoir la granularité des ressources [Bossuet 2007]. Pour cela nous avons développé un flot d'exploration permettant de définir l'architecture et les ressources à mettre en œuvre afin d'optimiser l'efficacité de l'architecture, notamment du point de vu énergétique [Bossuet 2004]. Très tôt nous avons pressentie l'apport de la reconfiguration partielle des FPGA, aussi dés 2003 nous avons développé un démonstrateur de la reconfiguration dynamique partielle pour une chaîne de modulation multistandard QPSK et 8PSK [Delahaye 2004]. Par la suite nous avons conservé une activité autour de ce domaine afin de mettre en œuvre l'auto reconfiguration

dynamique partielle et ainsi anticiper les besoins des futurs systèmes embarqués qui devront dynamiquement s'adapter à leur environnement d'exécution.

L'axe de recherche concernant la sécurité des systèmes embarqués s'intéresse au problème de plus en plus sensible de la protection des systèmes autonomes et mouvants. Ce domaine de recherche est récent du point de vu des systèmes embarqués et il est clair que l'activité de recherche s'y rattachant va se renforcer dans les années à venir. Les premières études que nous avons menées dans ce domaine s'intéressaient à la protection des bitstreams pour les composants FPGA afin de garantir la confidentialité des données de configuration [Bossuet 2006a]. Ce problème bien que partiellement résolu aujourd'hui par les industriels reste cependant ouvert à de nouvelles évolutions dans la mesure où de nombreuses IP sont intégrées au sein d'un système reconfigurable. Ces IP devant être protégées et confidentielles les unes par rapport aux autres, des perspectives de recherche intéressantes sont à étudier. Un autre travail que nous avons mené concerne la sécurité des systèmes à base de FPGA qui permettent la mise en œuvre d'une sécurité dynamique [Gogniat 2006]. Ce type d'approche est intéressant car il permet d'adapter le niveau de sécurité en fonction de la menace. Ce point est essentiel afin de faire face aux contraintes très fortes des systèmes embarqués, notamment en ce qui concerne la consommation. Plusieurs études concernant la protection des échanges de données entre des zones sécurisées et non sécurisées sont également en cours [Vaslin 2007][Wanderley 2007]. Enfin, l'extension des OS afin de mettre en œuvre des mécanismes de protection matérielle est aussi étudiée. Ce point est fondamental afin de déployer des solutions de sécurités complètes.

Mes activités de recherche (codesign, reconfigurable, sécurité) interagissent donc à travers ces trois axes de recherche. Les travaux menés suivant ces axes se renforcent les uns les autres et une contribution dans un domaine suscite de nouvelles idées dans un autre. Ce point me semble important et positif car il participe à l'éveil et à la maturité scientifique d'un chercheur. La Figure 4 illustre ces trois domaines de compétences et leurs dénominateurs communs. L'épine dorsale de mes travaux correspond à la notion de systèmes embarqués (*System on Chip – SoC*). Ces derniers peuvent être reconfigurables dynamiquement et/ou adaptatifs. L'objectif systématique étant d'aboutir à la définition d'une architecture adaptée à une ou plusieurs applications. Les axes 1 et 2 s'intéressent également aux outils d'aide à la conception. Dans la suite je détaille mon activité de recherche selon ces trois axes.

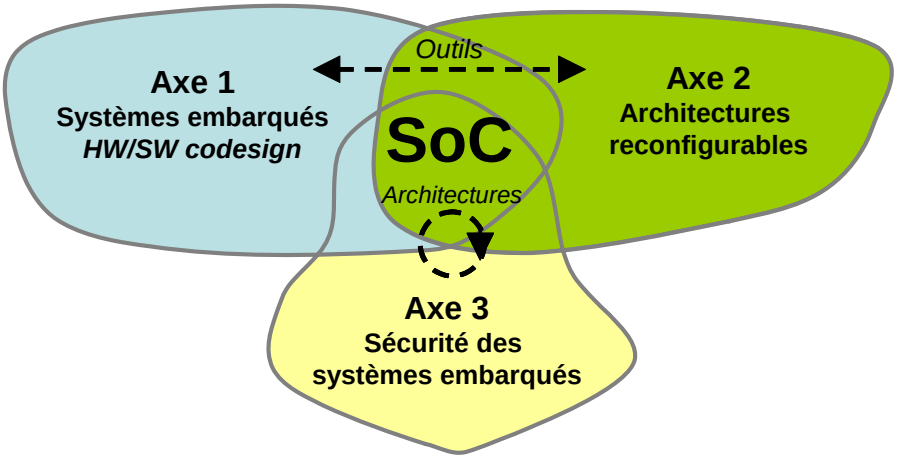


Figure 4 • Positionnement et interactions des 3 axes de recherche.

3. Axe 1 : Systèmes embarqués (HW/SW codesign)

3.1 Introduction

Le concepteur de systèmes électroniques dispose d'un éventail de solutions matérielles pour réaliser un système conforme à son cahier des charges, en termes de consommation d'énergie, de performances temporelles, d'occupation d'espace, d'opportunité de reconfiguration ou d'évolutivité. Il peut utiliser des cartes déjà existantes si elles correspondent à ses besoins, ou créer une architecture matérielle dédiée à son application avec tous les choix architecturaux que cela implique. Il doit cependant trouver l'architecture matérielle la mieux adaptée à ses besoins, avec la diversité de composants matériels existants. Les ASIC sont des composants dédiés très performants mais limités à n'exécuter que ce pour quoi ils ont été conçus.

Les processeurs généralistes ou spécifiques à certains traitements (DSP), sont programmables et disposent donc d'un peu de flexibilité (on peut modifier le programme mais pas l'architecture du composant). Les FPGA sont très flexibles de part la reconfiguration matérielle qu'ils offrent, et possèdent désormais des ressources de calcul importantes avec les processeurs qu'ils intègrent (ou synthétisables). Dans le cas des SoC, comme nous l'avons vu précédemment, toutes ces ressources utiles aux concepteurs sont dans le même composant, et l'espace des solutions architecturales se "limite" aux

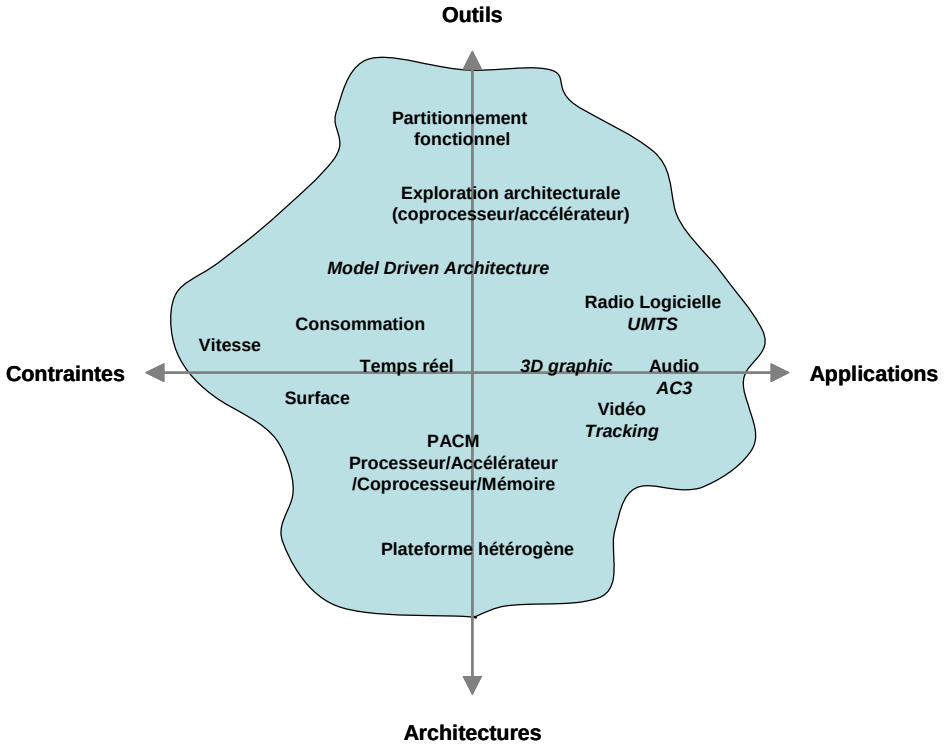


Figure 5 • Couverture de l'espace d'exploration (outils, contraintes, applications et architectures) de l'axe 1.

possibilités de combinaison des ressources potentiellement contenues. Autant dire que le nombre de solutions est très large. Les critères pris en compte dans ces décisions sont fonction des multiples contraintes de natures différentes provenant à la fois de l'application à réaliser et de l'architecture retenue.

Avec les contraintes de temps de mise sur le marché, il est donc primordial de proposer aux concepteurs un outil leur permettant d'estimer l'adéquation entre des architectures et des applications, au plus tôt dans le flot de conception (niveau système), afin de valider des solutions d'implantation conformes au cahier des charges.

L'objectif de cet axe de recherche est donc de développer de nouvelles méthodes et de nouveaux outils de conception afin de lever les verrous de conception entre les spécifications fonctionnelles (i.e. au niveau système) et les architectures hétérogènes sous-jacentes. Les architectures considérées sont du type système sur silicium (SoC) et les contraintes de conception sont essentiellement la vitesse et la surface. Plusieurs contributions ont été apportées au sein de cet axe :

- Métriques au niveau système et partitionnement fonctionnel pour la conception des SoC (2002/2007)
- Exploration autour du modèle d'architecture hétérogène PACM, i.e. processeur/accélérateur/coprocesseur/mémoire (2003/en cours)
- Approche MDA (Model Driven Architecture) pour la radio logicielle à comportement statique (2003/2006)
- Approche MDA pour les systèmes à comportement dynamique (2007/en cours)

La Figure 5 illustre les travaux menés dans un espace à 4 dimensions : outils, contraintes, applications et architectures. En ce qui concerne les outils, les travaux menés couvrent plusieurs niveaux depuis le partitionnement fonctionnel jusqu'au prototypage rapide en passant par des approches de conception dirigées par les modèles. En ce qui concerne les contraintes, la vitesse, la surface et la consommation sont considérées. La contrainte de temps réel est également prise en compte afin de respecter les besoins des applications de télécommunication et du multimédia. Les architectures sont hétérogènes et caractérisées par la cohabitation entre des ressources logicielles et matérielles (coprocesseur, accélérateur). L'espace couvert par cet axe de recherche est donc relativement vaste à l'image du problème à traiter.

Bien évidemment la mise en place de ces différentes activités de recherche s'est effectuée progressivement comme l'illustrent les Figures 6 et 7. Les premiers travaux se sont intéressés aux systèmes hétérogènes et au problème de la synthèse des interfaces entre les unités logicielle et matérielle. Ensuite les premières études autour de la conception MDA ont été réalisées. Il est intéressant de noter qu'en 2004 ces activités de recherche débutaient également au niveau international à travers notamment le SDR Forum [SDR 2007] et l'OMG [OMG 2007]. Parallèlement le problème du partitionnement fonctionnel a été abordé afin de faire face à la complexité croissante des applications. L'identification des accélérateurs matériels sur la base d'une architecture dite PACM (Processeur Accumulateur Coprocesseur Mémoire) a ensuite été étudiée. Ce type d'approche se relève très intéressante d'un point de vue efficacité énergétique et commence a être déployée dans de nombreux systèmes [Synfora 2007][Tensilica 2007]. Plus récemment l'étude des OS pour les systèmes embarqués a débutée. Il s'agit d'analyser le surcoût en temps et en consommation lié à l'OS afin de pouvoir au plus tôt dimensionner correctement le système. Enfin, les derniers travaux portent sur l'extension de l'approche MDA aux applications à comportement dynamique. Ce point est fondamental afin de pouvoir appréhender et concevoir efficacement les futurs systèmes adaptatifs.

3.2 Présentation des travaux

Afin d'illustrer l'activité menée au sein de cet axe de recherche la suite de cette section présente deux études adressant le problème de la conception de systèmes complexes. La première étude correspond aux travaux de Issam Maalej actuellement en fin de thèse et porte sur l'exploration au niveau système à travers un partitionnement fonctionnel [Maalej 2006]. La deuxième étude correspond aux travaux de thèse de Samuel Rouxel actuellement ingénieur R&D au sein de CRESITT Industrie à Orléans et porte sur une approche de conception dirigée par les modèles [Rouxel 2006a].

Métriques au niveau système et partitionnement fonctionnel pour la conception des SoC

L'étape d'exploration architecturale est une étape critique du flot de conception des systèmes embarqués dans la mesure où les décisions prises à ce niveau impactent très fortement sur les performances finales du système. En effet, pour une application donnée, une multitude d'architectures peuvent être considérées. Cet espace d'exploration contient des architectures non réalisables, des architectures réalisables mais ne satisfaisant pas les contraintes, et enfin des architectures réalisables et respectant les contraintes. Parmi cette dernière catégorie les architectures proposant les performances optimales doivent être

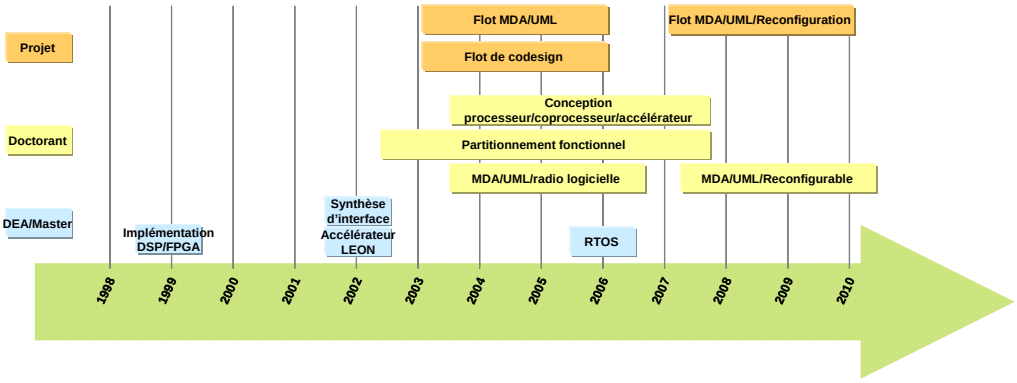


Figure 6 • Déroulement des travaux concernant l'axe de recherche Systèmes Embarqués.

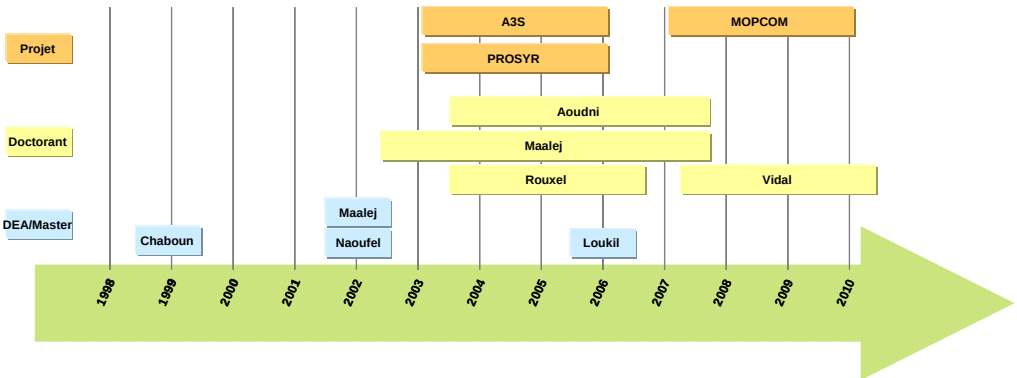


Figure 7 • Etudiants, doctorants et projets impliqués dans l'axe de recherche Systèmes Embarqués.

identifiées. Il incombe donc à l'étape d'exploration de trouver pour une application donnée une où plusieurs architectures adéquates qui satisfassent les contraintes et qui optimisent les performances.

Ces travaux de recherche adressent le problème de l'exploration des architectures multiprocesseurs pour les applications actuelles et futures qui comportent des centaines de tâches et des volumes de données échangés extrêmement importants [Bautista 2007]. Au delà des difficultés liées à l'étape d'exploration des architectures traditionnelles, le concepteur doit donc faire face à plusieurs nouveaux défis induits par l'accroissement des niveaux de complexité [Yoon 2007]. En effet, le parallélisme a été fortement accentué ces dernières années demandant plus de ressources de communications, de mémorisation et d'exécution [Martin 2006]. Les ressources logicielles sont de plus en plus nombreuses au sein d'une architecture et également de plus en plus variées [Flake 2006]. Par ailleurs, l'intégration croissante au sein des architectures introduit une nouvelle dimension dans l'espace à explorer : la dimension spatiale. Cette notion consiste à analyser l'influence de la proximité des ressources les unes par rapport aux autres sur les performances de l'architecture [Hu 2006][Sassatelli 2007].

Les approches d'exploration traditionnelles consistent à explorer l'espace des architectures en prenant en compte plusieurs informations relatives aux paramètres technologiques de l'architecture cible. Toutefois l'estimation des performances dans le cadre de telles approches a souvent un coût élevé du fait du nombre important de ressources technologiques potentielles (logicielles, matérielles et communications). Aussi, ce type d'exploration est complexe et atteint aujourd'hui ses limites du fait de la croissance continue du nombre d'architectures possibles.

Afin d'appréhender ces niveaux de complexité une approche consiste à élever le niveau d'abstraction des architectures afin de gagner en rapidité d'exploration [Ha 2007][Atat 2007]. Certes, plus le niveau d'abstraction est élevé, plus l'espace des architectures est grand, mais parallèlement les concepteurs parcourent plus rapidement l'espace de conception à haut niveau qu'à bas niveau. En effet certaines caractéristiques de bas niveau sont inutiles afin d'effectuer les premiers dimensionnements d'un système, d'autant que l'obtention de ces caractéristiques de bas niveau engendrent souvent un temps d'estimation important.

Afin de faire face à cet accroissement de l'espace de conception, nous proposons donc une approche de conception qui débute l'exploration à un niveau supérieur (fonctionnel) afin de réduire l'espace des architectures pour l'exploration traditionnelle et réduire par conséquent les coûts de l'estimation des performances et de l'exploration. La réduction de l'espace des architectures n'est pas faite au détriment de la qualité de l'exploration. En effet, l'analyse tient compte d'un nombre important de paramètres pour permettre d'éliminer les architectures les moins performantes et guider l'exploration vers un espace réduit certes mais contenant l'essentiel des architectures les plus performantes. Une approche en deux étapes est donc proposée comme présenté sur la Figure 8 [Maalej 2006].

- La première étape, nommée pré-exploration, consiste à analyser la spécification de l'application décrite par un graphe de tâches en considérant l'analyse de l'application et des tâches qui la composent. Pour cela, nous projetons l'espace d'exploration à trois dimensions temps, surface consommation sur un espace à six dimensions basé sur 6 métriques (EDE, DEIC, CIC, MEM, IA, TC) qui ont été développées et formalisées [Maalej 2004]. Ces métriques ont pour but d'optimiser la distribution des échanges des données et le partage des données entre les ressources du système. Elles permettent également d'optimiser la distribution des contraintes de débit et le regroupement des tâches dans l'architecture afin d'optimiser les performances. Ce nouvel espace a pour intérêt de réduire le coût de l'étape d'exploration dans la mesure où les métriques sont moins dépendantes de la technologie et représentent donc un ensemble d'informations considérablement moins coûteux à élaborer. L'exploration de ce nouvel espace est

effectuée par un algorithme génétique multi-objectif (six objectifs) développé au cours de ce travail.

- La deuxième étape plus traditionnelle, nommée exploration, explore l'espace réduit par l'étape précédente afin de converger vers une architecture adéquate à l'application. Cette deuxième étape se base sur l'analyse faîte par la première étape pour étudier l'espace des architectures en considérant le modèle d'architecture dit multi PACM.

L'approche proposée a été validée sur plusieurs applications. La première étude s'est basée sur un émetteur UMTS [Rouxel 2006b]. Cette application a permis de valider les métriques et leur analyse par l'algorithme génétique. Le résultat de l'exploration automatique a ainsi permis de mettre en évidence plusieurs architectures optimisées et notamment la même architecture que celle résultant d'une optimisation manuelle. Le gain de temps au niveau du cycle de conception peut donc être significatif et fortement soulager le concepteur. La seconde étude a porté sur une application de codage audio AC3 [ATSC 1995]. Cette application a permis de valider le flot proposé en deux étapes. L'outil d'exploration logiciel/matériel CODEF [Auguin 2001] a été utilisé pour la deuxième étape du flot d'exploration. La combinaison des deux outils a permis de réduire l'espace de conception et de mettre en évidence certaines architectures optimisées non identifiées par un processus de conception en une seule étape. Enfin la troisième étude a testé l'efficacité de l'approche proposée pour des applications contenant un nombre de tâches élevé.

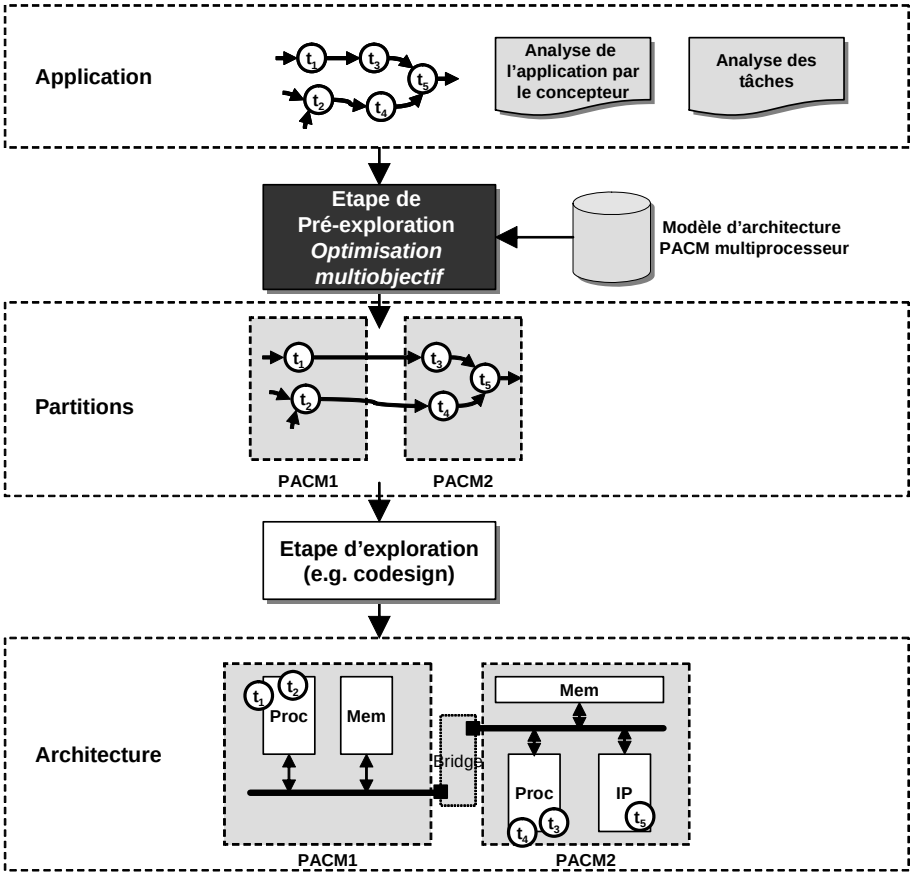


Figure 8 • Flot de conception pour l'exploration au niveau système d'une application. Deux étapes d'exploration sont mises en œuvre afin de progressivement réduire l'espace de conception.

Plusieurs scénarios ont été considérés depuis une spécification d'une dizaine de tâches jusqu'à une cinquantaine de tâches et pour une architecture contenant une dizaine de processeurs. Cette troisième application (suivi d'objets dans une image, ICAM) [Auguin 2003] a démontré la capacité de l'approche proposée à appréhender un espace de conception étendu aussi bien du point de vu de l'application que de l'architecture.

Approche MDA (Model Driven Architecture) pour la conception de systèmes hétérogènes

La méthodologie développée dans ces travaux est basée sur le langage UML, et suit une approche MDA (*Model Driven Architecture*). Ce type d'approche, dirigé par les modèles, se développe rapidement et plusieurs experts internationaux y voient le moyen d'appréhender la complexité des futurs systèmes hétérogènes. Plusieurs évènements majeurs (i.e. DAC et DATE) s'intéressent aux opportunités offertes par l'approche MDA [UML4SoC 2004]. Il est clair qu'un mouvement en direction de ces approches a été initié ces dernières années et que l'avenir verra son influence se renforcer [Martin 2005].

Dans la méthode proposée, le concepteur de SoC utilise des IP matérielles et logicielles déjà disponibles dans sa bibliothèque de ressources afin de réaliser ses systèmes [Rouxel 2006a][Rouxel 2006b]. Dès lors, la fonctionnalité même du composant IP n'est plus à démontrer, car elle est assurée par le vendeur ou le fournisseur de l'IP. De même, le concepteur dispose de toutes les caractéristiques de performances, de surface et de consommation propre à chacune des IP qu'il détient. Ces travaux montrent que ces informations peuvent être utilisées pour la spécification d'une partie du système à haut niveau. Le flot de conception A3S [A3S 2005], représenté sur la Figure 9 est basé sur ces considérations et propose une modélisation et une spécification au niveau système en 4 étapes :

- Modélisation de l'architecture : le concepteur peut indifféremment débuter par la modélisation de son architecture matérielle ou de son application. Une bibliothèque de composants dits "matériels" fournit un ensemble de composants matériels qu'il peut utiliser pour créer son architecture. Ces composants sont paramétrables en fonction des configurations d'utilisation requises par le système à mettre en œuvre. Le concepteur peut donc modéliser son SoC par l'assemblage de différents composants spécifiés.
- Modélisation de l'application : sachant que le concepteur peut réutiliser des codes logiciels de traitements déjà développés dans d'autres applications, sachant également que les applications peuvent être décomposées en tâches spécifiques, une seconde bibliothèque, de composants dits "logiciels", fournit un ensemble de composants logiciels représentant des fonctionnalités. Le concepteur peut donc modéliser son application de manière indépendante de toute cible architecturale par l'assemblage de composants logiciels "génériques" sous la forme d'un graphe de tâches.

La modélisation réalisée, qualifiée de PIM (*Platform Independent Model*), est indépendante de la plate-forme matérielle. L'avantage réside dans la simplicité des modifications à apporter à l'application le cas échéant, où tout remplacement, insertion, suppression d'une fonction, ou réutilisation d'une partie de l'application peut se faire sans impact majeur sur l'architecture matérielle.

- Déploiement : Une fois l'application logicielle et l'architecture matérielle modélisées, le concepteur peut alors décider des choix d'implantation à mettre en oeuvre et à tester pour son système. Il décide alors manuellement de la réalisation matérielle ou logicielle de chacune des fonctions du graphe de tâches. La modélisation, qualifiée de PSM (*Platform Specific Model*) devient alors dépendante de l'architecture matérielle. Il se doit alors, de spécifier les contraintes imposées par ses choix architecturaux pour chaque composant logiciel instancié.

- Vérifications et Analyse : A chacune des étapes précédentes, des vérifications sont effectuées pour valider les modèles réalisés. Elles avertissent le concepteur des éventuelles erreurs de modélisation ou de spécifications commises. Elles permettent de vérifier le respect des contraintes d'utilisation des composants matériels et d'approuver la cohérence des choix d'implantation retenus. Ces vérifications accomplies, les analyses de faisabilité et d'estimation de performances peuvent avoir lieu. Elles déterminent la faisabilité de l'implantation et renvoient les résultats d'ordonnançabilité et de performance.

La méthodologie de conception mise en œuvre à travers le flot A3S peut être assimilée à du prototypage virtuel. En effet, le résultat de l'analyse de l'implantation d'une application sur une architecture est issu des paramètres spécifiés par le concepteur. Ces paramètres correspondent aux caractéristiques réelles des composants matériels ainsi qu'aux caractéristiques des IP liées aux composants logiciels, issues d'implantations réelles et fournies avec l'IP.

Ce prototypage repose sur des modélisations UML effectuées dans l'atelier de développement Objecteering [Objecteering 2007], à partir de métamodèles UML définis (e.g. profil A3S). L'expérimentation choisie pour tester et valider la méthodologie développée au travers du profil UML A3S, est une application UMTS [UMTS 1999]. C'est une application complexe avec des contraintes temps réel fortes imposées par le standard. Il s'agit d'une application type pour confronter la méthode et le profil développé aux besoins d'un concepteur dans le cadre du processus de conception à haut niveau d'abstraction d'un SoC, appliqué aux systèmes Radio Logicielle [SDR 2007].

Plusieurs configurations donnant lieu à des systèmes UMTS différents, en terme de réalisation et de performances attendues, ont été effectuées afin d'obtenir des estimations de faisabilité et de performances pour les 4 possibilités de l'espace de conception considéré et détaillé dans la suite (Table 1).

Des vérifications automatiques, développées dans le profil A3S, ont été lancées à chacune des étapes du flot de conception couvert par l'outil. La cohérence de chacune des

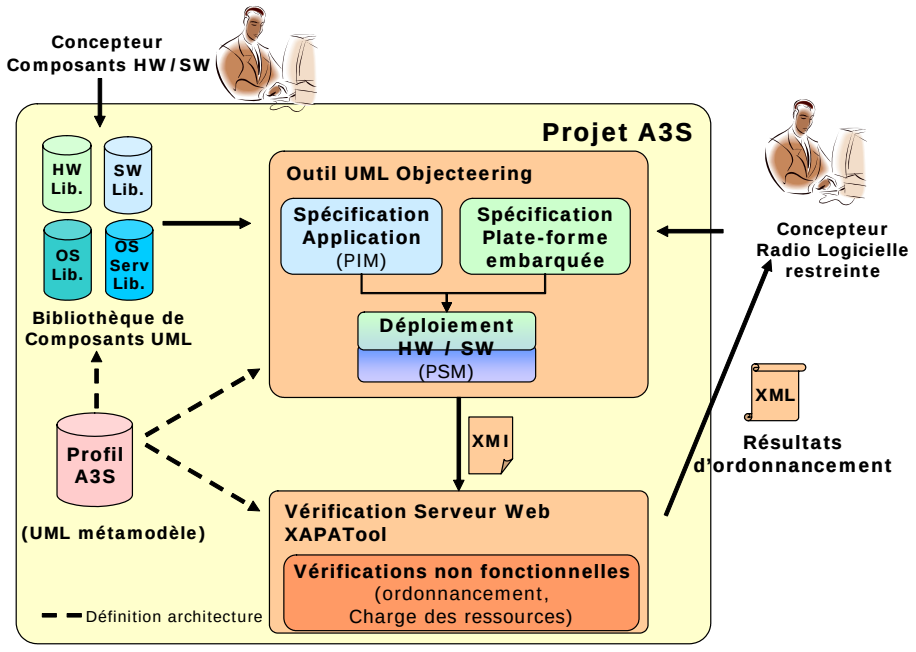


Figure 9 • Flot de conception du projet A3S. Approche MDA et vérification non fonctionnelles.

modélisations a ainsi pu être établie, stipulant au concepteur qu'aucune erreur structurelle de modélisation et de spécification n'a été commise.

Par ailleurs, la traduction des contraintes d'implantation issues de la mise en œuvre des composants logiciels sur les instances des composants matériels, est considérée au cours de la phase d'analyse. En effet, les divers choix d'implantation effectués influent sur le calcul de l'ordonnancement des fonctions et sur les performances temporelles obtenues. Dans le cas où toutes les fonctions sont implantées sur le même composant matériel, les temps de communication inter-fonction peuvent être considérés comme nuls. En revanche, dans le cas où deux fonctions qui interagissent sont implantées sur deux composants matériels différents, les temps de communications ne sont plus considérés comme négligeables et un temps supplémentaire doit être ajouté aux temps d'exécution des fonctions. Ceci n'est possible qu'après analyse des choix d'implantation en étroite relation avec les graphes de tâches issus des diagrammes d'activité.

Les résultats des analyses sont fournit par l'outil XAPAT qui a été développé dans le cadre du projet A3S et qui intègre l'outil RTDT précédemment développé au sein de notre équipe de recherche [Rouxel 2006c][Tmar 2006]. Lorsque les modélisations de l'application UMTS sont effectuées et qu'aucune erreur de conception n'est détectée un ficher XMI correspondant au système est généré et envoyé dans l'outil XAPAT qui calcule automatiquement l'ordonnançabilité du système (Table 1). Ces résultats sont obtenus par le calcul de l'ordonnancement des blocs fonctionnels vu comme des tâches indépendantes (algorithme du *Rate Monotonic*). Lorsqu'un ordonnancement est possible les résultats sont également renvoyés sous la forme d'un gantt.

Table 1 • Taux d'utilisation des composants pour l'application UMTS en fonction des contraintes applicatives.

Application Emetteur/Récepteur UMTS	débit bit 117 kbits/s (configuration 1)				débit bit 950 kbits/s (configuration 2)			
	DSP_A	DSP_C	FPGA_A	Temps (ms)	DSP_A	DSP_C	FPGA_A	Temps (ms)
Emetteur UMTS								
Implantation logicielle	96.6%	3.4%	-	9.99	96.6%	5.1%	-	10.33
Implantation logicielle/matérielle	11.4%	3.4%	66%	7.96	11.4%	5.1%	66%	8.29
Récepteur UMTS								
Implantation logicielle	185%	4.6%	-	19.27	185%	5%	-	19.33
Implantation logicielle/matérielle	17.1%	4.6%	71.2%	9.44	17.2%	5%	71.2%	9.49

L'analyse des résultats renvoyés, traduit la faisabilité du système dans la configuration 1 (débit 117 kbits/s) en ce qui concerne l'émetteur UMTS (solution purement logicielle composée de deux DSP). En effet, les taux d'utilisation des DSP, bien qu'élevés, restent inférieurs à 100% et la contrainte temporelle d'une trame est respectée (9ms<10ms). En revanche, la même configuration du récepteur UMTS n'est pas envisageable puisque la charge de travail requise au niveau du DSP_A est supérieure à la charge admissible. Il est impossible de faire fonctionner un DSP au delà de 100%. Il faut donc dans ce cas alléger la charge de travail du DSP_A. De plus la contrainte temporelle n'est pas respectée (19.27ms). Cette solution d'implantation n'est donc pas envisageable dans le cadre de l'application visée avec ces contraintes.

Dans le cas d'une contrainte applicative plus forte (configuration 2, 950 kbits/s), il se trouve que l'implantation purement logicielle, suffisante pour l'application de l'émetteur fonctionnant avec un débit de 117 kbits/s, ne l'est plus. Le DSP_C voit sa charge de travail

augmenter légèrement (+1,7%) et la contrainte applicative n'est plus respectée (10.33ms). Pour ce qui est de l'application du récepteur, elle n'était pas réalisable avec des contraintes applicatives plus faibles. Les résultats obtenus confirment qu'elle ne l'est pas davantage avec des contraintes plus fortes.

La solution repose donc sur un changement de choix d'implantation afin de pouvoir absorber les besoins en performances (solution hétérogène matérielle/logicielle). Le changement d'implantation de la fonction la plus critique en matériel (mise en œuvre sur FPGA) influe considérablement sur les performances des 2 configurations. Celles-ci deviennent conformes aux attentes du concepteur. L'implantation du récepteur pour la configuration 1 avec un débit de 117 kbits/s était impossible de manière uniquement logicielle. Elle devient réalisable avec l'implantation de la fonction MFL (fonction de filtrage) en matériel (FPGA_A). Les taux d'utilisation des DSP_A se réduisent avec l'allégement de la charge de travail, reportée sur le FPGA (71,2%), notamment celui utilisé pour l'application du récepteur UMTS qui passe de 185% (irréalisable) à 17,1%. Les temps d'exécution sont également revus à la baisse grâce au traitement effectué en matériel, ce qui conforte (dans le cas de l'émetteur) et rend possible (dans le cas du récepteur) le respect des contraintes temporelles. La dernière configuration du système consiste à appliquer des contraintes applicatives plus sévères (configuration 2, 950 kbits/s). Cette utilisation mixte de DSP et de FPGA apparaît une fois de plus comme une solution architecturale en adéquation avec l'application, en dépit des contraintes imposées, puisque les taux d'occupation sont tous corrects et la contrainte temporelle respectée pour l'émetteur et le récepteur.

3.3 Conclusion

La conception de systèmes embarqués hétérogènes est complexe et requiert de nombreuses compétences afin d'aboutir à la définition d'une solution efficace. A travers les travaux menés dans cet axe de recherche nous avons principalement adressé le problème de la conception à haut niveau aussi bien pour le pré partitionnement fonctionnel que pour le prototypage virtuel à travers une approche basée sur UML. Nous avons également développé plusieurs démonstrateurs afin d'appréhender les gains possibles au niveau des architectures lors de l'utilisation d'accélérateurs matériels et/ou de coprocesseurs. Ces analyses nous ont conduit à proposer une approche de conception permettant l'identification et l'implémentation de coprocesseurs pour des applications complexes développées en langage C. Nous avons également mené des études autour de la notion d'OS pour les applications embarquées afin d'anticiper leur utilisation systématique dans les futures systèmes mobiles. Enfin nous débutons des travaux concernant la modélisation du caractère dynamique des applications et des architectures à travers une approche basée sur UML.

Afin de mener à bien ces travaux 4 doctorants ont participé ou participent actuellement au projet [Rouxel 2006/T] [Maalej 200X/T] [Aoudni 200X/T] [Vidal 20XX/T] et 4 stagiaires de DEA ou de Master [Chaboun 1999/D] [Maalej 2002/D] [Naoufel 2002/D] [Loukil 2005/D].

L'ensemble des travaux menés au sein de cet axe de recherche a conduit à 21 publications scientifiques (1 participation à un ouvrage de synthèse, 18 conférences internationales, 2 conférences nationales) [Rouxel 2006/O] [Maalej 2006/CI] [Rouxel 2006b/CI] [Rouxel 2006a/CI] [Aoudni 2006c/CI] [Aoudni 2006b/CI] [Aoudni 2006a/CI] [Rouxel 2005a/CI] [Moy 2004/CI] [Denef 2004/CI] [Delautre 2004b/CI] [Aoudni 2004b/CI] [Aoudni 2004a/CI] [Maalej 2004b/CI] [Maalej 2004a/CI] [Delautre 2004a/CI] [Maalej 2003/CI] [Maalej 2002/CI] [Diguet 2000/CI] [MACGTT 2002/CN] [Maalej 2002/CN].

3.4 Fiche de synthèse des travaux

Co-encadrements de thèses

[Rouxel 2006/T] Samuel Rouxel 2003/2006 – Bourse Contrat RNRT

Modélisation et caractérisation de plates-formes SoC hétérogènes : Application à la Radio Logicielle

Thèse de Doctorat soutenue le 5 décembre 2006, en co-encadrement avec le Pr. Jean Luc Philippe (50%) **–** Situation : Ingénieur R&D CRESITT Industrie, Orléans, France

[Maalej 200X/T] Issam Maalej 2002/2007 – Bourse CMCU

Métriques au niveau système et partitionnement fonctionnel pour la conception des SoC
Soutenance prévue en 2007, en co-encadrement avec les Pr. Jean-Luc Philippe (25%) et Pr. Mohamed Abid (25%)

[Aoudni 200X/T] Yassine Aoudni 2003/2007 – Bourse CMCU

Mise en œuvre d'applications réactives sur SoC : proposition d'une démarche de validation

Soutenance prévue en 2007, en co-encadrement avec les Pr. Jean-Luc Philippe (25%) et Pr. Mohamed Abid (25%)

[Vidal 20XX/T] Jorgiano Marcio Bruno Vidal 2007/2010 – Bourse Contrat RNTL

Reconfiguration dynamique des systèmes : de la modélisation à la validation

Soutenance prévue en 2010, en co-encadrement avec le Pr. Jean Luc Philippe (50%)

Encadrement de stages de DEA et de Master

[Chaboun 1999/D] Said Chaboun
Etude et implémentation d'algorithmes de compression de signaux audio sur des cibles hétérogènes
DEA Rennes, année 1998/1999

[Maalej 2002/D] Issam Maalej
IP de communication générique à base de bus pour les systèmes embarqués
DEA Sfax, Tunisie, année 2001/2002

[Naoufel 2002/D] Ismail Naoufel
Modélisation et intégration d'un accélérateur matériel sur le système à base du processeur LEON
Master Sfax, Tunisie, année 2001/2002

[Loukil 2006/D] Kais Loukil
Estimation du temps d'exécution des systèmes temps réel sur puce
Master Sfax, Tunisie, année 2005/2006

Collaborations scientifiques

[PROSYR 2006] Projet CMCU PROSYR

PROtotypage de SYstèmes Réactifs : Application à la conception des systèmes sur puce
Type : Comité Mixte franco-tunisien pour la Coopération Universitaire
Durée : 2003/2006
Partenaires : LESTER, ENIS

[MOPCOM 2009] Projet MOPCOM

Modélisation et spécialisatiOn de Plates-formes et COmposants MDA pour SOC/SOPC
Type : Projet ANR/RNTL – Pôle de compétitivité Images et Réseaux
Durée : 2007/2009
Partenaires : THALES, THOMSON, ENSIETA, IETR/Supelec, IRISA, LESTER, SODIUS.

[A3S 2005] Projet A3S

Adéquation Architecture - Application Système
Type : Projet RNRT
Durée : 2003/2005
Partenaires : THALES Communications, SOFTEAM, Mitsubishi Electric ITE, LESTER

[MACGTT 2002] Projet MACGTT

Méthode d'Aide à la Conception Globale des Terminaux de Télécommunications
Type : Projet CNRS
Durée : 2000/2002
Partenaires : I3S, LASTI, LESTER

Publications scientifiques

[Rouxel 2006/O] S. Rouxel, G. Gogniat, J-P. Diguet, J-L. Philippe and C. Moy, **Chapter 7. Schedulability Analysis and MDD**, From MDD Concepts to Experiments and Illustrations Edited by: J-P. Babau, J. Champeau, S. Gérard International Scientific and Technical Encyclopedia, September 2006, pages 111 – 130

[Maalej 2006/CI] I. Maalej, G. Gogniat, J-L. Philippe, M. Abid, **Genetic algorithm for high level analysis and architecture exploration**, *IP Based Design 2006 Workshop*, December 2006, Grenoble, France

[Rouxel 2006b/CI] S. Rouxel, G. Gogniat, J-P. Diguet, J-L. Philippe, C. Moy, **System Level Design with UML: a Unified Approach**, *IEEE Symposium on Industrial Embedded System (IES'06)*, October 2006, Antibes Juan-Les-Pins, France

[Rouxel 2006a/CI] S. Rouxel, G. Gogniat, J-P. Diguet, J-L. Philippe, C. Moy, **A3S Method and Tools for Analysis of Real-Time Embedded Systems**, *International Workshop on Modeling and Analysis of Real-Time and Embedded Systems (MARTES'06)*, October 2006, Genova, Italy

[Aoudni 2006c/CI] Y. Aoudni, G. Gogniat, K. Loukil, J-L. Philippe, M. Abid, **Mapping SoC Architecture Solutions for an Application based on PACM Model**, *IEEE International Symposium on Industrial Electronics*, July 9-13, 2006, Montréal, Canada

[Aoudni 2006b/CI] Y. Aoudni, G. Gogniat, J-L. Philippe, M. Abid, **Custom Instruction Integration Method within Reconfigurable SoC and FPGA Devices**, *The International Conference on Microelectronics (ICM 2006)*, December 16-19, 2006, Dhahran, Saudi Arabia

[Aoudni 2006a/CI] Y. Aoudni, G. Gogniat, K. Loukil, J-L. Philippe, M. Abid, **Method for Embedded Application Prototyping based on SoC Platform and Architecture Model**, *IEEE 1st International Conference on Design and Test of Integrated Systems in Nanoscale Technology*, September 05-07, 2006 Tunis, Tunisia

[Rouxel 2005a/CI] S. Rouxel, G. Gogniat, J-P. Diguet, J-E. Goubard, C. Moy, N. Bulteau, **UML Framework for PIM and PSM Verification of SDR Systems**, *Software Defined Radio Technical Conference*, November 2005, Anaheim, USA

[Moy 2004/CI] C. Moy, M. Raulet, S. Rouxel, J-P. Diguet, G. Gogniat, P. Desfray, N. Bulteau, J-E. Goubard, Y. Denef, **UML Profile for Waveform SPS abstraction**, *SDR Forum Technical Conference*, November 2004, Phoenix, Arizona, USA

[Denef 2004/CI] Y. Denef, J-E. Goubard, G.Gogniat, S. Rouxel, J-P. Diguet, C. Moy and N. Bulteau, **UML Profile for SDR hardware/software adequacy verification**, *First Annual Software-Based Communications Workshop: From Mobile to Agile Communications*, September 2004, Arlington, USA

[Delautre 2004b/CI] A. Delautre, J-E. Goubard, G.Gogniat, S. Rouxel, J-P. Diguet, C. Moy and N. Bulteau, **UML profile towards waveform performances verification**, *Wireless World Research Forum (WWRF)*, June 2004, Oslo, Norway

[Aoudni 2004b/CI] Y. Aoudni, N. Ben Amor, G. Gogniat, J-L. Philippe, M. Abid, **Platform and Architecture Adequacy in SoC environment: a case study**, *The 16th IEEE International Conference on Microelectronics (ICM 2004)*, December 6-8, 2004, Tunis, Tunisia

[Aoudni 2004a/CI] Y. Aoudni, N. Ben Amor, G. Gogniat, J-L. Philippe, M. Abid, **IP Processor Core Platform Selection According to SoC Architecture: a case study**, *IP Based SOC design 2004*, December 8-9, 2004, Grenoble, France

[Maalej 2004b/CI] I. Maalej, G. Gogniat, M. Abid, J-L. Philippe, **High level analysis of multiprocessor system on chip**, *Embedded Real-Time Systems Implementation Workshop (ERTSI 2004)*, December 5-8, 2004, Lisbon, Portugal

[Maalej 2004a/CI] I. Maalej, G. Gogniat, M. Abid, J-L. Philippe, **Metrics for multiprocessor system on chip**, *The 16th IEEE International Conference on Microelectronics (ICM 2004)*, December 6-8, 2004, Tunis, Tunisia

[Delautre 2004a/CI] A. Delautre, J-E. Goubard, G. Gogniat, S. Rouxel, J-P. Diguet, C. Moy, N. Bulteau, **Verification of System coherency at early Architecture Design Stage**, *SDR Forum, Hardware Abstraction Layer Working Group*, April 21-23, 2004, Germany

[Maalej 2003/CI] I. Maalej, G. Gogniat, M. Abid, J-L. Philippe, **Interface Design Approach For System On Chip Based On Configuration**, *IEEE International Symposium on Circuits and Systems (ISCAS 2003)*, 25-28 May, 2003, Bangkok, Thailand

[Maalej 2002/CI] I. Maalej, G. Gogniat, M. Abid, J-L. Philippe, **Design of communication interface based on configuration for system on chip**, *IP Based Design'2002 Workshop*, October 2002, Grenoble, France

[Diguet 2000/CI] J-P. Diguet, G. Gogniat, P. Danielo, M. Auguin, J-L. Philippe, **The SPF model**, *Forum on Design Language (FDL)*, September 2000, Tübingen, Germany

[MACGTT 2002/CN] Projet MACGTT, **Vers une approche unifiée pour la conception globale des terminaux de télécommunications**, *JFAAA'02*, décembre, 2002, Monastir, Tunisie

[Maalej 2002/CN] I. Maalej, G. Gogniat, M. Abid, J-L. Philippe, **Conception d'interface pour processeur embarqué dans les systèmes sur puce**, *JFAAA'02*, décembre, 2002, Monastir, Tunisie

4. Axe 2 : Architectures reconfigurables

4.1 Introduction

Depuis le début des années 2000 les architectures reconfigurables ont connu des évolutions majeures, notamment aux niveaux technologique et architectural [Radumovic 1998] [Compton 2000][Hartenstein 2001][Schaumont 2001]. Les FPGA, de part la régularité de leur structure, tirent profit des dernières avancées technologiques afin d'afficher des performances toujours plus élevées. Par exemple, la technologie cuivre utilisée depuis 2001 dans les composants de chez Xilinx permet une réduction d'environ 70% des temps de propagation des signaux le long des métallisations par rapport à la technologie aluminium [Bossuet 2006b]. La technologie CMOS 65nm permet également des gains en consommation dynamique de l'ordre de 35% par rapport à la technologie CMOS 90nm. Les outils de conception associés aux FPGA ont également fortement évolué et permettent la réalisation rapide d'applications complexes intégrant un grand nombre de ressources aussi bien matérielles que logicielles [Xilinx 2007]. 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 [Tredennick 2003].

Toutefois, les architectures reconfigurables ne se limitent pas à la mise en œuvre de ressources qualifiées de grain fin mais intègrent également de façon croissante de l'hétérogénéité (e.g. mémoires, multiplieurs, processeurs) afin de répondre efficacement aux défis des nouvelles applications impliquant des traitements intensifs et des contraintes de performances sévères [Sassatelli 2002]. Par exemple, les ressources de communications, des bus aux réseaux de routages, sont souvent intégrées conjointement au sein d'une même architecture afin de faciliter les communications locales comme globales. Les systèmes sur puce mettent le plus souvent en œuvre des zones programmables (coeurs de processeurs) couplées plus ou moins étroitement avec des zones reconfigurables (de grain fin et/ou de gros grain). Les structures mises en oeuvre sont le plus souvent hiérarchiques mais peuvent avoir des topologies différentes selon les besoins de déroulement des traitements ou pour les communications [Bossuet 2004].

Ces architectures se présentent donc aujourd'hui comme une réponse intéressante au challenge des systèmes sur puce. Elles permettent l'élaboration de nouvelles applications tirant profit de leurs caractéristiques propres telles qu'un fort parallélisme matériel et des possibilités de reconfiguration statique et/ou dynamique (c'est à dire en cours d'exécution). Le domaine de la Radio Logicielle est un exemple représentatif pour lequel les architectures reconfigurables dynamiquement apportent une réponse significative afin de relever le défi de la multiplicité des standards de communication et de leur complexité intrinsèque [Wang 2007][Delahaye 2007]. Les architectures reconfigurables sont donc au centre d'une importante révolution technologique de l'électronique numérique.

Cet axe de recherche adresse plusieurs verrous liés à la mise en œuvre d'applications sur les composants reconfigurables. Il s'intéresse notamment à l'exploration de l'espace de conception des architectures reconfigurables gros grain et grain fin [Bossuet 2004][Bilavarn 2006]. Il s'agit d'évaluer les performances en termes de vitesse, surface et consommation d'une application implémentée sur une architecture reconfigurable. Des travaux sur l'auto reconfiguration partielle de composants FPGA sont également menés afin de favoriser l'adaptabilité des systèmes embarqués [Delahaye 2004]. La décision liée à la reconfiguration est également un point fondamental afin d'adapter correctement

l'architecture en fonction des contraintes sur le système. Plusieurs contributions ont donc été apportées suivant cet axe :

- Exploration architecturale et estimation de performances pour les FPGA (1999/2002)
- Exploration architecturale pour les architectures reconfigurables gros grain/grain fin (2001/2004)
- Reconfiguration dynamique des FPGA (2003/ en cours)
- Systèmes adaptatifs (2006/ en cours)

La Figure 10 positionne les travaux menés selon les 4 dimensions définies précédemment (outils, contraintes, applications et architectures). L'espace couvert par ces travaux est plus réduit et s'intéresse uniquement aux architectures reconfigurables. La contrainte considérée, en dehors de la vitesse, surface et consommation, est l'adaptabilité. Il s'agit de la reconfiguration dynamique qui permet de mettre en œuvre sur une même architecture différents systèmes et cela de façon dynamique (au cours de l'exécution). Les applications considérées sont principalement du domaine des télécommunications et du multimédia.

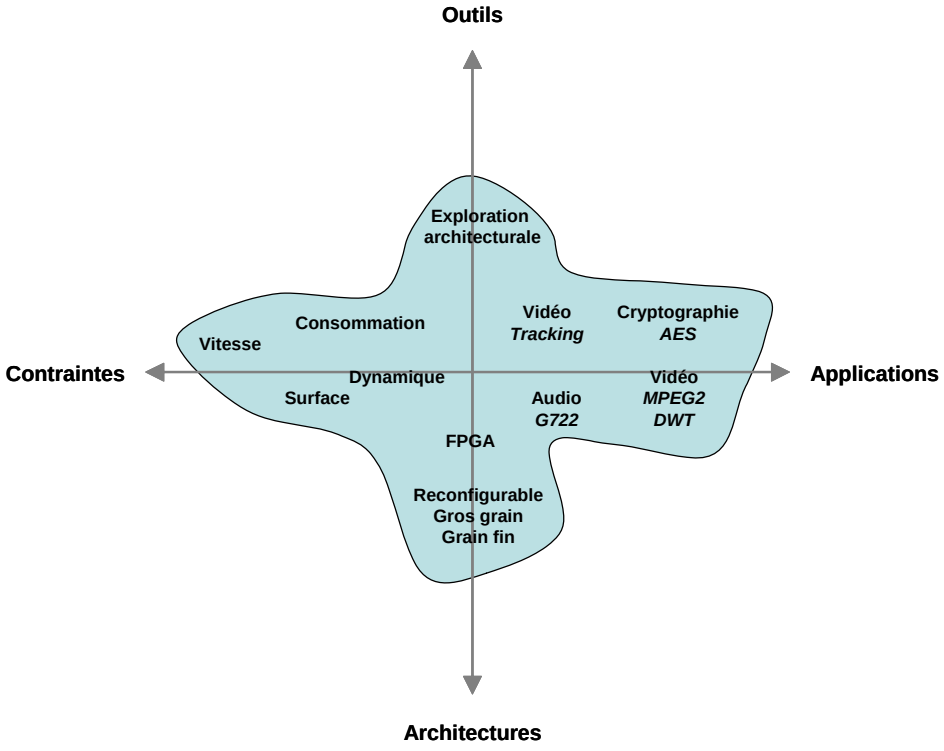


Figure 10 • Couverture de l'espace d'exploration (outils, contraintes, applications et architectures) de l'axe 2.

L'activité de recherche développée au sein de cet axe s'est d'abord intéressée au problème de l'exploration de l'espace de conception comme l'illustrent les Figures 11 et 12. Pour cela différentes études ont été menées concernant la modélisation des architectures, l'analyse de la consommation et la mise en œuvre d'applications sur FPGA [Rouxel 2002]. Ces études ont été accompagnées de projets permettant de confronter nos expériences et d'avancer conjointement avec les autres acteurs du domaine, principalement au niveau national mais également au niveau international. En 2003 nous avons débuté nos travaux sur la reconfiguration dynamique partielle et avons développé plusieurs démonstrateurs

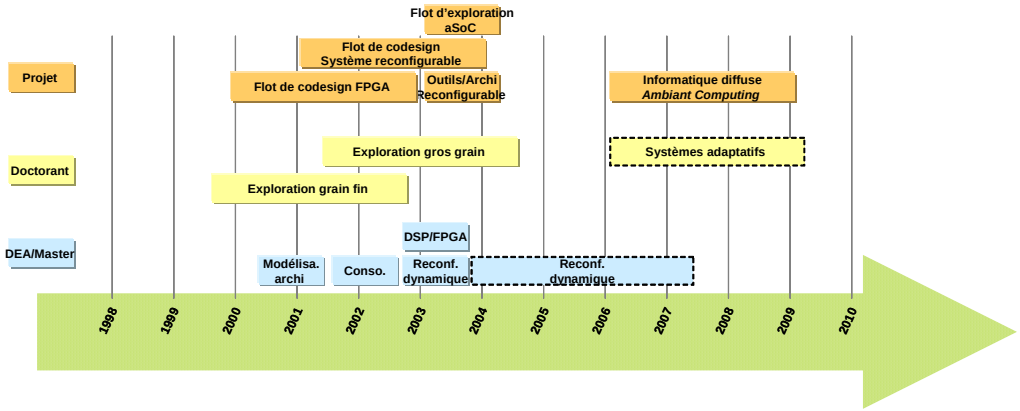


Figure 11 • Déroulement des travaux concernant l'axe de recherche Architectures Reconfigurables.

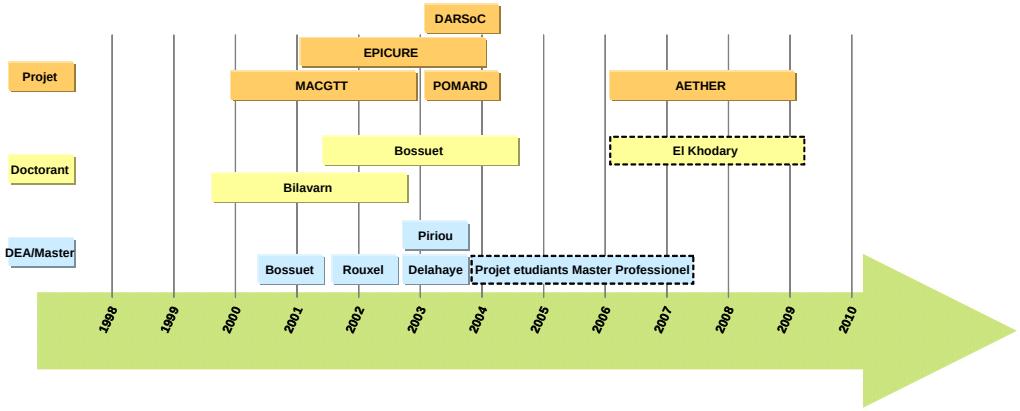


Figure 12 • Etudiants, doctorants et projets impliqués dans l'axe de recherche Architectures Reconfigurables.

significatifs afin de mettre en évidence les avantages et les limites des solutions actuelles. Nous continuons cette activité à travers des projets étudiants et des collaborations non contractuelles avec différents partenaires travaillant sur le sujet (e.g. Supelec/IETR). Il me semble très important de conserver la maîtrise de cette technologie et des flots de conception associés car il m'apparaît très clairement qu'à l'avenir les systèmes embarqués utiliseront massivement la reconfiguration dynamique afin de s'adapter en temps réel à leur environnement. Nous en sommes encore au stade embryonnaire de cette technologie et les perspectives sont particulièrement attractives. Plus récemment et dans la continuité des travaux menés nous avons naturellement débuté une activité de recherche autour de l'informatique diffuse (*ambient computing* en anglais) qui vise à anticiper ces futurs systèmes adaptatifs [AETHER 2006]. Il s'agit de définir les mécanismes permettant de déployer dynamiquement une ou plusieurs applications sur une plateforme d'exécution.

4.2 Présentation des travaux

Afin d'illustrer l'activité menée suivant cet axe de recherche la suite de cette section présente deux études adressant le problème de l'exploration de l'espace de conception des architectures reconfigurables. La première étude correspond aux travaux de thèse de

Sébastien Bilavarn actuellement Maître de Conférences à l'Ecole Polytechnique de l'Université de Nice – Sophia Antipolis et porte sur l'exploration architecturale et l'estimation de performance pour les FPGA [Bilavarn 2002]. La deuxième étude correspond aux travaux de thèse de Lilian Bossuet actuellement Maître de Conférences à l'Ecole Nationale Supérieure d'Electronique, Informatique & Radiocommunications de Bordeaux (ENSEIRB) et porte sur l'exploration architecturale pour les architectures reconfigurables gros grain/grain fin [Bossuet 2004].

Exploration architecturale et estimation de performances pour les FPGA

Ces travaux portent sur l'estimation en temps et en surface de fonctions candidates à une implémentation matérielle, plus précisément sur des composants reconfigurables du type FPGA. L'estimateur développé peut être vu comme un outil permettant de vérifier la faisabilité de l'intégration d'un système sur un FPGA, à partir d'une spécification comportementale issue d'un code de haut niveau (i.e. fonction décrite en langage C).

Le rapport précision/complexité est une caractéristique importante pour un estimateur puisqu'il conditionne l'exploration de l'espace de recherche (exploration du parallélisme, différentes fréquences d'horloge, différentes allocations) et l'évaluation de plusieurs composants cibles en un temps raisonnable. Par ailleurs, étant données les récentes avancées dans le domaine de la synthèse de haut niveau et le degré de maturité atteint par les outils de synthèse architecturale [Sentieys 1993][Bondalapati 1999][Gupta 2002][Catapult 2004], la méthode d'exploration développée ne se justifie que si elle apporte une amélioration significative du cycle de conception.

Aussi, lors de la définition des techniques d'estimation, la complexité des algorithmes a été un critère prépondérant. De plus, il est important de ne pas opposer les approches d'exploration et les approches de synthèse architecturale. Au contraire, leur complémentarité laisse entrevoir les perspectives d'une méthodologie de conception prometteuse où l'exploration et la synthèse interagissent afin d'identifier et concevoir des accélérateurs matériels permettant d'optimiser l'efficacité énergétique des solutions d'exécution pour les systèmes embarqués.

L'exploration architecturale développée dans ces travaux se déroule en deux étapes principales qui sont l'estimation au niveau structurel et l'estimation au niveau physique (Figure 13). Une première phase de pré estimation permet de vérifier le bon dimensionnement du composant candidat à l'implémentation. Elle se base d'une part sur le nombre de broches d'entrées/sorties qui est comparé au nombre de données d'entrées/sorties de la spécification, décrite en langage C et traduite sous la forme d'un graphe flot de données et de contrôle hiérarchique [Diguet 2000], et d'autre part sur le nombre de ressources de mémorisation dont dispose le FPGA. Le processus d'exploration/estimation ne débute qu'à l'issu de cette étape.

L'estimation structurelle a ensuite pour but de fournir plusieurs solutions architecturales pour la spécification en cours d'analyse. Elle se base tout d'abord sur une étape d'allocation de ressources et de calcul de la fréquence d'horloge, puis un ordonnancement partiel est réalisé. Cet ordonnancement ne concerne que les blocs de base de la spécification et est effectué pour plusieurs contraintes de temps, permettant ainsi la définition de plusieurs solutions. L'estimation globale du système est obtenue par l'application de combinaisons qui dépendent des types de contrôle (structures conditionnelles/itératives) et d'exécution (séquentielle/concurrente) présents dans l'application.

Par exemple, on obtient l'estimation d'une structure conditionnelle à partir des résultats d'estimation de la condition et des branches. L'estimation de l'exécution séquentielle de deux graphes est obtenue à partir des résultats d'estimation de ces deux graphes. L'exploration hiérarchique permet ainsi de combiner les résultats jusqu'à atteindre le plus haut niveau de la hiérarchie, qui correspond à l'application toute entière. À la fin de cette étape, on dispose de courbes de caractérisation temps/coût de toute l'application

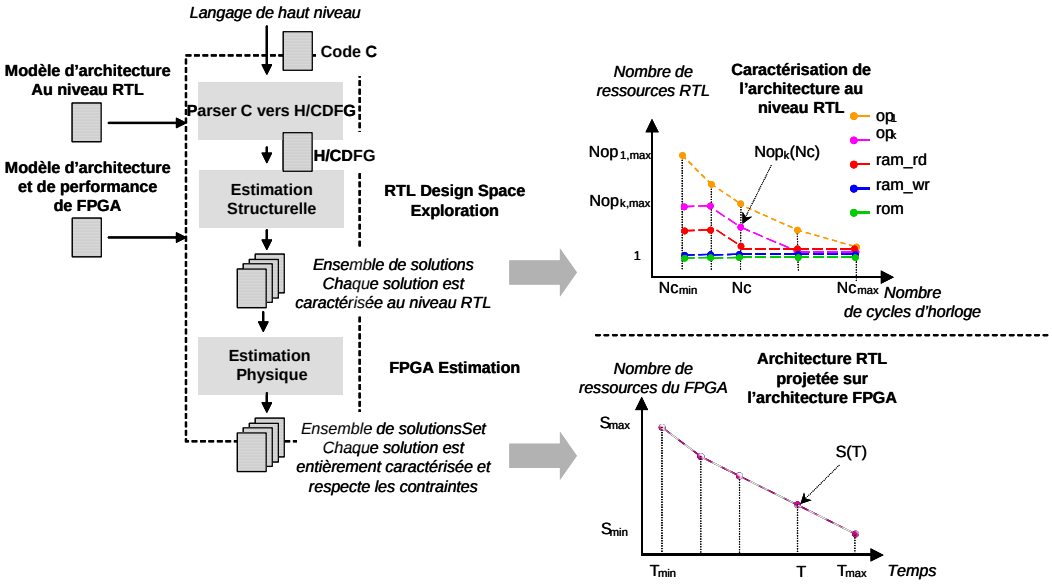


Figure 13 • Flot d'exploration pour les architectures FPGA. Depuis la description de l'application en langage C jusqu'à l'estimation des performances en nombre de ressources et en temps d'exécution.

(représentées en haut à droite de la Figure 13). Il y a une courbe de caractérisation par ressource (de traitement et de mémorisation) nécessaire à la réalisation de l'application. Chaque point d'une courbe donne le résultat d'estimation du nombre de ressources à mettre en œuvre pour un nombre de cycles d'horloge (en considérant une architecture au niveau RTL). On peut ainsi, en positionnant une contrainte de temps, connaître une estimation des besoins en termes de ressources matérielles pour la réalisation de l'application. Si la contrainte de temps est forte (c'est à dire que l'application doit être exécutée dans un temps court) alors on sélectionne des réalisations parallèles comptant de nombreuses ressources, nécessitant ainsi une réalisation à fort parallélisme pour l'application.

L'estimation physique traduit ensuite les courbes temps/coût de caractérisation structurelle (où le temps est exprimé en nombre de cycles) en courbes de caractérisation physique (exprimée en taux d'occupation des ressources du FPGA/unité de temps physique comme illustré en bas à droite de la Figure 13). Une analyse des unités de traitement, contrôle et mémorisation est effectuée pour une caractérisation complète de l'application. Cette étape nécessite la connaissance précise des caractéristiques de la technologie cible. C'est le rôle du fichier technologique (modèle d'architecture et de performance de FPGA) qui contient les paramètres physiques tels que les temps de traversée des opérateurs, les temps d'accès aux mémoires, ainsi que leur coût en surface. À l'issu de l'étape d'estimation physique, le concepteur dispose des valeurs physiques de compromis temps/surface pour différentes solutions architecturales. Ces valeurs permettent de vérifier la faisabilité (taux d'occupation inférieur à 100%) ou le respect des contraintes (vitesse d'exécution) du système intégré sur le composant candidat.

La méthode développée intègre comme indiqué ci-dessus l'estimation des unités de traitement, de mémorisation et de contrôle, ce qui constitue un point original dans la mesure où il existe très peu d'approches s'intéressant à ces trois aspects simultanément [Bilavarn 2006]. Ce point est nécessaire pour obtenir une estimation globale d'une application sur une architecture matérielle reconfigurable. La validation de la méthode s'est portée sur des systèmes représentatifs de deux classes d'applications (orientées vers du traitement, du contrôle ou de la mémorisation de données) : codage audiofréquence et traitement d'image.

La projection physique sur FPGA donne une estimation précise des performances de l'application (15 % en moyenne) ce qui est satisfaisant étant donné le niveau d'abstraction de la spécification initiale.

Bien que ces travaux visent essentiellement les composants reconfigurables, ils constituent un point de départ intéressant dans le cadre de l'élaboration d'une technique d'estimation matérielle qui puisse aussi s'appliquer aux ASICs. Un point également fondamental est la définition d'une méthode d'estimation qui soit au maximum indépendante d'une technologie donnée, afin de rendre son utilisation la plus large possible. Aussi, dans le cadre de ces travaux, nous nous sommes efforcés de rendre l'approche la plus générique possible.

Exploration architecturale pour les architectures reconfigurables gros grain/grain fin

Face à l'évolution des architectures reconfigurables qui intègrent de façon croissante des ressources de calcul hétérogènes et multi granularités, la définition d'une approche d'exploration de l'espace de conception générique se fait ressentir. Le terme générique s'inscrit principalement dans le choix architectural (architectures grain fin, gros grain et hétérogène) ainsi que dans le choix du domaine d'applications (e.g. traitement des images, cryptographie, télécommunication). L'exploration doit porter sur l'aspect architectural de la cible mais aussi sur l'implémentation de l'application, c'est à dire viser simultanément l'architecture physique et l'architecture logique (Figure 14). En s'appuyant sur l'outil Design Trotter développé au sein du laboratoire [LeMoullec 2003a][LeMoullec 2003b], ces travaux proposent une méthode d'exploration (appelée projection architecturale) qui répond à ces exigences en tirant parti d'estimations effectuées à haut niveau d'abstraction et intervenant très tôt (dés les phases de spécifications) dans le flot de développement des applications.

Ce travail est original en plusieurs points : Tout d'abord, peu d'études sont menées dans le domaine, en partie de part l'intérêt relativement récent des chercheurs pour la définition de méthodes de conception conjointe appliquées aux technologies matérielles reconfigurables. De plus, ces études ne prennent généralement pas en compte les structures de contrôle intervenant dans les applications complexes, ce qui limite leur utilisation à l'estimation de chemins de données. L'approche développée dans ces travaux, la spécification supporte différentes structures de contrôle telles que les boucles et les

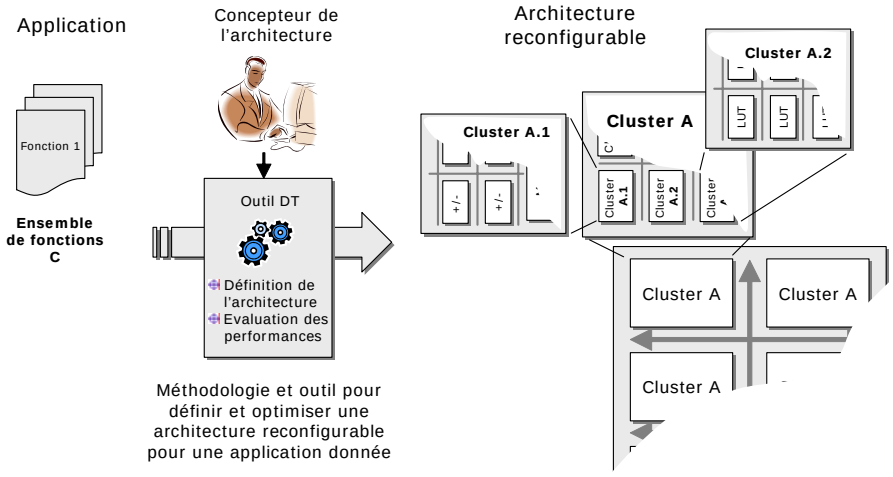


Figure 14 • Flot d'exploration pour des architectures reconfigurables multi-granularité. Depuis la spécification en langage C jusqu'à l'estimation des performances.

structures conditionnelles. Ainsi, l'estimation d'applications complexes incluant par exemple des structures hiérarchiques et des données multidimentionnelles peut être effectuée.

La projection architecturale utilise les résultats de l'estimation système [LeMoullec 2003a] ainsi qu'une modélisation hiérarchique fonctionnelle des architectures [Bossuet 2002]. Les estimations obtenues en résultats sont l'estimation de la distribution hiérarchique des communications dans l'architecture et l'estimation du taux d'utilisation des ressources de traitement de l'architecture. Ces paramètres permettent d'évaluer l'efficacité en consommation de puissance d'une architecture. Ces estimations sont ensuite utilisées par l'utilisateur lors de l'exploration architecturale pour modifier en conséquence l'architecture modélisée et converger vers la définition d'une architecture efficace.

La méthode d'exploration architecturale développée vise donc à définir une architecture efficace du point de vue de la consommation de puissance pour une application (et pour un domaine d'applications par extension). Cette méthode exige un flot complexe que nous schématisons sur la Figure 15.

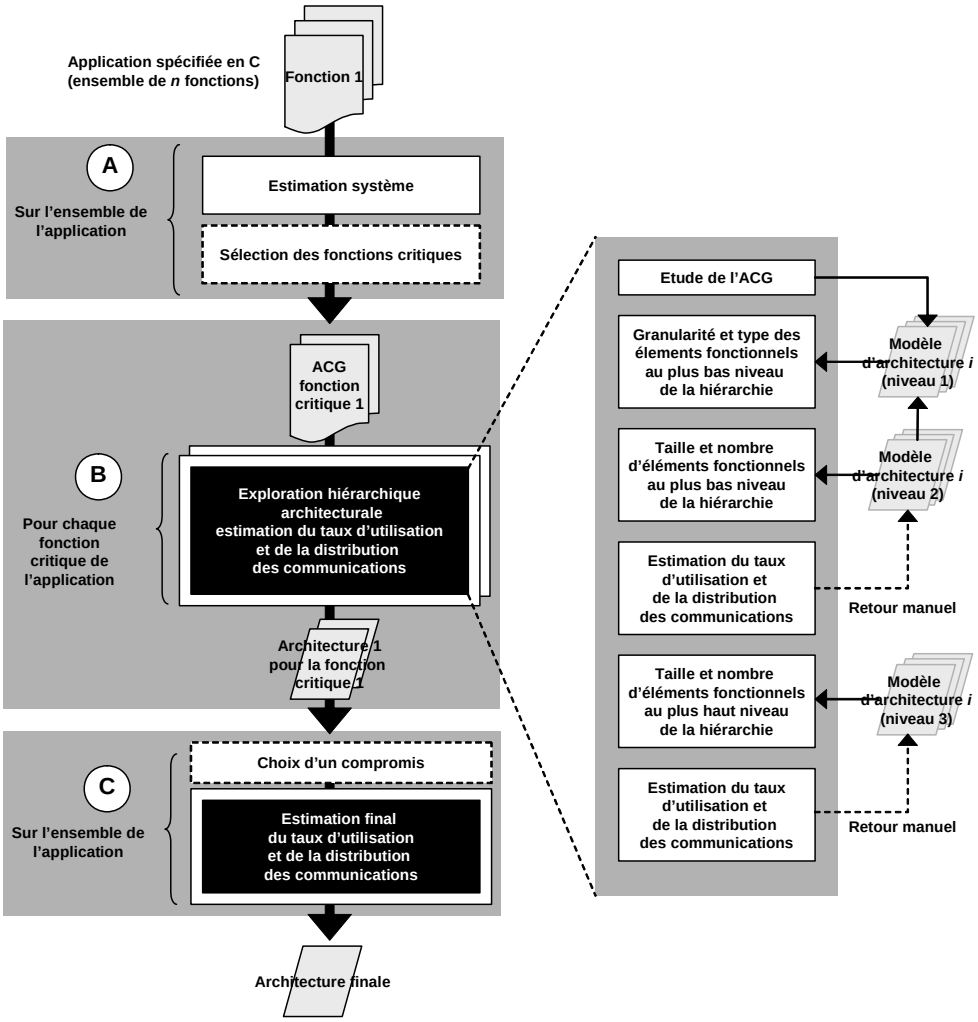


Figure 15 • Flot d'exploration hiérarchique pour les fonctions critiques et pour l'application complète.

La première étape de ce flot (étape A sur la Figure 15) correspond à la mise en œuvre de l'estimation système. L'application, spécifiée en langage C, est le plus souvent décomposée en *n* fonctions et dans ce cas une spécification sous la forme d'un graphe HCDFG (*Hierarchical Control and Data Flow Graph*) est générée pour chaque fonction. Nos expériences sur l'exploration architecturale nous ont rapidement montré que certaines fonctions de l'application, le plus souvent une ou deux, avaient un impact très fort sur les performances finales de l'application [Bossuet 2004]. Aussi, il nous est apparu plus judicieux et plus efficace de parcourir le flot d'exploration uniquement pour ces fonctions que nous appelons fonctions critiques. Effectivement, si le flot est parcouru pour les *n* fonctions de l'application il en résultera potentiellement *n* architectures, hors si l'on souhaite définir une architecture homogène structurellement il est extrêmement compliqué, voir inenvisageable, de trouver un compromis parmi *n* dès que *n* atteint plusieurs unités. Afin de déterminer clairement les fonctions critiques et à partir de nos résultats expérimentaux nous avons établi trois critères de criticité :

- Le degré de parallélisme d'exécution de la fonction va nous permettre de vérifier que la solution d'ordonnancement retenue pour chaque fonction est bien adaptée à une implémentation matérielle massivement parallèle. Si il n'est pas possible de réaliser la fonction de façon parallèle elle sera réalisée séquentiellement ce qui peut entraîner un ralentissement de l'application. La fonction peut donc être critique d'un point de vue temporel.
- La localité potentielle des communications dans l'architecture représente le nombre moyen de communications par ressource de traitement. C'est l'ACG (*Average Communication Graph*) généré d'après le HCDFG initial qui nous permet d'obtenir rapidement le nombre de communications. Un problème se pose si le nombre de ressources de traitement est faible pour un grand nombre de communications. Il faut alors prévoir des ressources de routage particulièrement adaptées ainsi que des ports d'entrées/sorties des éléments hiérarchiques assez larges pour le flux de communications. Il ne s'agit ici que d'une étude rapide nous permettant de discriminer les fonctions qui n'ont pas un nombre important de communications dans leur réalisation.
- La congestion temporelle potentielle des ressources de routage représente le nombre moyen de communications par cycle. Le problème ici vise l'occupation temporelle des ressources de routage. Effectivement il peut y avoir des problèmes si le nombre de communications est grand pour un nombre de cycles relativement faible. Le problème se pose particulièrement pour les grands nombres de communications. Il faut alors prévoir une répartition des communications telle que les congestions soient évitées.

Une fois les fonctions critiques sélectionnées le flot d'exploration (partie B de la Figure 15) est parcouru pour chacune de ces fonctions. La première étape vise à établir la granularité et le type des éléments fonctionnels (opérateurs et mémoires) contenu dans les clusters (ou éléments hiérarchiques) de niveau le plus bas de hiérarchie (niveau 1). Cette première étape consiste à étudier la répartition des communications dans le graphe ACG de la fonction critique et permet de modéliser le premier niveau de hiérarchie de l'architecture. Les deux étapes suivantes vont s'appuyer sur les résultats de la projection architecturale (estimation du taux d'utilisation des éléments fonctionnels et distribution hiérarchique des communications) pour déterminer le nombre et la taille des éléments fonctionnels.

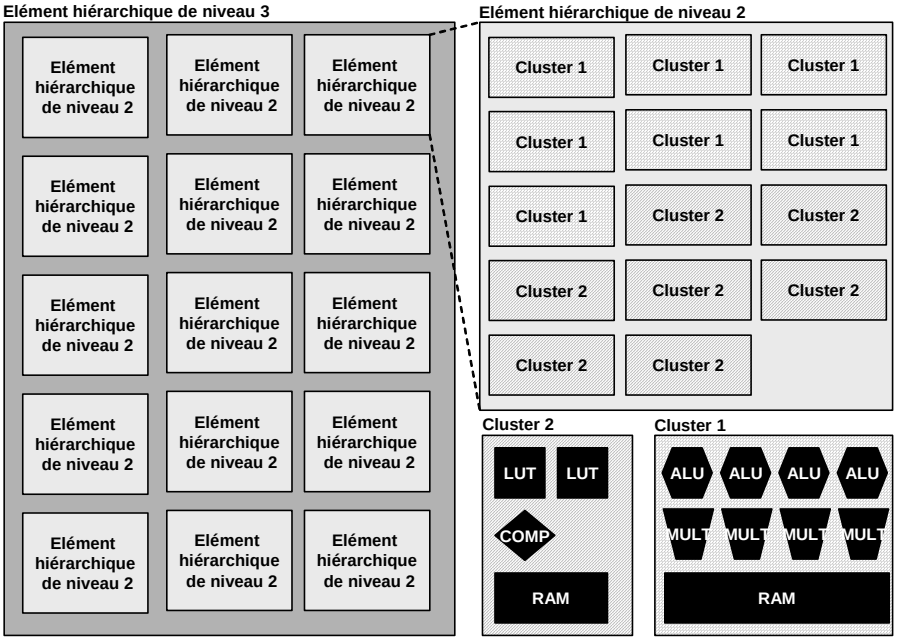
De cette façon les éléments hiérarchiques au niveau supérieur sont modélisables. Les objectifs du concepteur lors de l'exploration sont :

- Obtenir une distribution des communications dans l'architecture telle que le plus grand nombre possible de communications soit réalisé au niveau bas de la hiérarchie c'est à dire dans les éléments hiérarchiques qui contiennent les éléments fonctionnels.

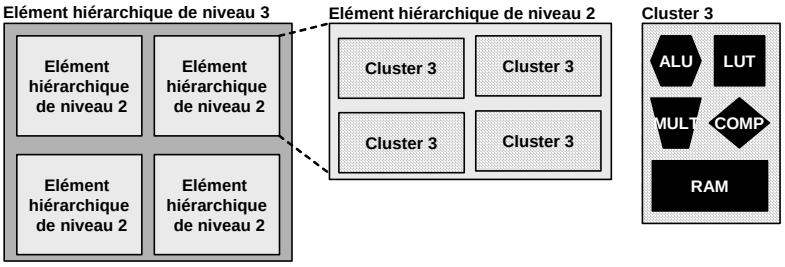
- Ne pas chercher à modéliser des éléments hiérarchiques de taille trop importante car il faut être capable d'assurer que les communications dans ces éléments sont à coûts constants.
- Enfin, pour une plus grande efficacité en consommation de puissance, obtenir un taux d'utilisation des éléments fonctionnels le plus grand possible en particulier pour les ressources de gros grain (ALU, additionneur).

Lors de cette phase d'exploration le concepteur change manuellement les paramètres de l'architecture modélisée pour améliorer les estimations, c'est donc une démarche itérative et interactive. Une dernière phase d'exploration, qui ne remet pas en question l'exploration au niveau hiérarchique inférieur, vise à déterminer le nombre d'éléments hiérarchiques dans les niveaux supérieurs de hiérarchie.

Lorsque la phase d'exploration architecturale est terminée pour l'ensemble des fonctions critiques, il est nécessaire de faire converger les architectures définies pour chaque fonction critique. Dans une première approche nous avons proposé de définir l'architecture



a) Architecture reconfigurable identifiée pour l'application MPEG2



b) Architecture reconfigurable identifiée pour l'application AES

Figure 16 • Représentations schématiques des architectures hiérarchiques définies pour le codeur MPEG2 et le bloc de chiffrement AES.

"compromis" avec les éléments hiérarchiques de taille maximum entre les différentes architectures correspondant aux fonctions critiques, mais cette approche mérite d'être raffinée. Le taux d'utilisation des éléments fonctionnels et la distribution hiérarchique des communications dans l'architecture compromis sont alors estimés pour l'ensemble des fonctions de l'application.

Afin de valider ces travaux, le flot d'exploration est illustré sur deux applications : une application de traitement des images : le codeur MPEG2 [MPEG2 2000] et une application de cryptographie : le codeur AES (sans générateur de clef) [Daemen 2002]. Ces deux applications n'ont pas la même complexité, le codeur AES est plus simple et sa spécification ne contient qu'une fonction. Le codeur MPEG2 est spécifié avec neuf fonctions en langage C, dont deux sont critiques pour les performances en consommation de puissance.

L'étude de l'ACG pour les deux applications ne donne pas le même résultat. Effectivement nous trouvons que plus de 90% des communications pour le codeur MPEG2 sont relatives à des échanges entre ressources de traitement de gros grain et entre ces mêmes ressources et les ressources de mémorisation. Par contre dans le cas du codeur AES les communications sont réparties entre les ressources de traitement de gros grain, de grain fin et les ressources de mémorisation. Les étapes du flot d'exploration nous permettent de déterminer pour les deux applications, le nombre d'opérateurs de grain fin et de gros grain ainsi que la taille (en nombre de mots) des mémoires dans les clusters de bas niveau.

La Figure 16 donne une représentation schématique des architectures hiérarchiques définies pour les deux applications. Nous voyons que l'architecture définie pour le codeur MPEG2 à deux clusters différents au plus bas niveau de hiérarchie : un cluster avec des opérateurs de gros grain (cluster 1) et un cluster avec des opérateurs de grain fin (cluster 2). Ce qui n'est pas le cas de l'architecture définie pour AES puisqu'il n'y a qu'un seul type de cluster au niveau le plus bas (cluster 3) dans lequel sont embarqués les opérateurs de grain fin et de gros grain. Finalement pour deux domaines d'application différents nous pouvons converger vers des architectures différentes.

Les estimations obtenues pour ces deux applications sont données dans la Table 2. La dernière ligne de cette table (AES*) donne les estimations obtenues en termes de distribution des communications pour le codeur AES avec l'architecture définie pour MPEG2.

Table 2 • Caractéristiques des architectures explorées pour les applications MPEG2 et AES.

Application	Taux d'utilisation				Distribution des communications		
	ADD/SUB	MUL	COMP	LUT	Niveau 3	Niveau 2	Niveau 1
MPEG2	67,0 %	70,0 %	13,0 %	2.0 %	29 %	8%	63 %
AES	63,8 %	100 %	100 %	93.8 %	21 %	10 %	69 %
AES*	-	-	-	-	36 %	14 %	50 %

Les distributions hiérarchiques des communications obtenues par estimation pour les deux applications sont satisfaisantes puisque dans les deux cas la part des communications au plus bas niveau de hiérarchie est supérieure à 60% du nombre total de communications. La dernière ligne de la Table 2 nous donne l'estimation de la distribution pour le codeur AES pour l'architecture définie pour le codeur MPEG2, nous pouvons voir que les résultats estimés sont fortement dégradés puisqu'il y a une diminution de 19% des communications au niveau bas et une augmentation de 15% des communications au niveau haut. Cette expérience souligne que le concepteur peut définir des architectures efficaces pour un

domaine d'application et définir une architecture différente pour un autre domaine d'application. En effet, cette tendance se retrouve également chez les fabricants d'architectures reconfigurables qui tendent de plus en plus à spécialiser leurs architectures par domaine d'application [Xilinx 2007].

4.3 Conclusion

Le domaine des architectures reconfigurables est extrêmement actif depuis une dizaine d'années avec des évolutions importantes au niveau des architectures et des outils d'aide à la conception. Afin d'accompagner ces évolutions nous avons mené plusieurs études suivant cet axe de recherche. Des méthodologies (et les outils associés) d'exploration de l'espace de conception pour des architectures FPGA et pour des architectures reconfigurables hétérogènes et hiérarchiques ont été proposées. Des travaux concernant l'analyse des coûts en consommation et des gains en performance des solutions reconfigurables par rapport aux solutions programmables ont également été menés. Enfin, nous avons développé plusieurs démonstrateurs afin de mettre en œuvre les mécanismes de reconfiguration dynamique et d'auto reconfiguration.

Afin de mener à bien ces travaux 2 doctorants ont participé au projet [Bossuet 2004/T] [Bilavarn 2002/T] et 4 stagiaires de DEA [Piriou 2003/D] [Delahaye 2003/D] [Rouxel 2002/D] [Bossuet 2001/D].

Les travaux menés au sein de cet axe de recherche ont conduit à 23 publications scientifiques (5 revues, 1 participation à un ouvrage de synthèse, 11 conférences internationales, 6 conférences nationales) [Bossuet 2007/R] [Diguet 2006/R] [Bilavarn 2006/R] [Bossuet 2006b/R] [Delahaye 2004/R] [Delahaye 2004/CI] [Bossuet 2003/O] [Bossuet 2005/CI] [Bossuet 2003c/CI] [Bossuet 2003b/CI] [Bilavarn 2003b/CI] [Bossuet 2003a/CI] [Bilavarn 2003a/CI] [Bossuet 2002/CI] [Bilavarn 2000/CI] [Bilavarn 2000/CI] [Bilavarn 1999/CI] [Bossuet 2005/CN] [Bossuet 2002/CN] [Bossuet 2002/CN] [Bilavarn 2001/CN] [Bilavarn 2000/CN] [Bilavarn 1999/CN].

4.4 Fiche de synthèse des travaux

Co-encadrements de thèses

Sébastien Bilavarn *Années de thèse : 1999/2002*

Exploration Architecturale au Niveau Comportementale – Application aux FPGAs
Thèse de Doctorat soutenue le 28 février 2002, en co-encadrement avec le Pr. Jean-Luc Philippe (50%) – situation : Maître de Conférences à l'Ecole Polytechnique de l'Université de Nice – Sophia Antipolis

Lilian Bossuet *Années de thèse : 2001/2004*

Méthodologie d'exploration des architectures reconfigurables
Thèse de Doctorat soutenue le 10 septembre 2004, en co-encadrement avec le Pr. Jean-Luc Philippe (50%) – situation : Maître de Conférences à l'Ecole Nationale Supérieure d'Electronique, Informatique & Radiocommunications de Bordeaux (ENSEIRB)

Encadrement de stages de DEA et de Master

[Rouxel 2002/D] Samuel Rouxel

Caractérisation de l'impact du routage sur les performances (vitesse et consommation de puissance) d'un FPGA
DEA Lorient, année 2001/2002

[Piriou 2003/D] Erwan Piriou

Comparaison de performance entre DSP et FPGA pour des applications de traitement du signal et des images
DEA Rennes, année 2002/2003

[Delahaye 2003/D] Jean Philippe Delahaye

Systèmes radio dynamiquement reconfigurables sur des architectures hétérogènes
DEA Orsay, année 2002/2003

[Bossuet 2001/D] Lilian Bossuet

Modélisation d'architectures reconfigurables embarquées
DEA Rennes, année 2000/2001

Collaborations scientifiques

[DARSoC 2003] Projet DARSoC

Dynamic Adaptative and Reconfigurable System on Chip
Type : Projet sur fond propre
Durée : 2002/2003
Partenaires : LESTER, VSPG

[AETHER 2008] Projet AETHER

Self-Adaptive Embedded Technologies for Pervasive Computing Architectures
Type : Projet Européen IST-FET (4th call ACA / FP6)
Durée : 2006/2008

[POMARD 2004] projet POMARD

Projet Outils, Méthodes et Architectures pour la Reconfiguration Dynamique
Type : Équipe Projet CNRS
Durée : 2003/2004
Partenaires : R2D2, LIEN, LIRMM, LE2I, ETIS, LIST, A&S, LESTER

[EPICURE 2003] Projet EPICURE

Environnement de Partitionnement et de Co-développement adapté aux processeurs à architectures REconfigurables
Type : Projet RNTL
Durée : 2001/2003
Partenaires : I3S, LESTER, CEA/List, THALES Communications, Esterel-Technologies

Publications scientifiques

Bossuet 2007/R] L. Bossuet, G. Gogniat, J-L. Philippe, **Communication-Oriented Design Space Exploration for Reconfigurable Architectures**, EURASIP Journal on Embedded Systems, Volume 2007 (2007), Article ID 23496, 20 pages, doi:10.1155/2007/23496

[Diguet 2006/R] J-P. Diguet, G. Gogniat, J-L. Philippe, Y. Le Moullec, S. Bilavarn, C. Gamrat, K. Ben Chehida, M. Auguin, X. Fornari, P. Kajfasz, **EPICURE: A Partitioning and CoDesign Framework For Reconfigurable Computing**, Journal of Microprocessors and Microsystems - Elsevier, Volume 30, Issue 6 , 4 September 2006, Pages 367-387, Special Issue on FPGA's, Edited by Morris Chang and Dan Lo

[Bilavarn 2006/R] S. Bilavarn, G. Gogniat, J-L. Philippe, L. Bossuet, **Low Complexity Design Space Exploration from Early Specifications**, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 25, No. 10, October 2006, pages 1950-1968

[Bossuet 2006b/R] L. Bossuet, G. Gogniat, J-L. Philippe, **Exploration de l'espace de conception des architectures reconfigurables**, Revue Technique et Science Informatiques, Architecture des ordinateurs, sous la direction de Marc Daumas et Dominique Lavenier, Volume 25, n°7, pages 921 – 946, TSI, Lavoisier 2006

[Delahaye 2004/R] J-P. Delahaye, G. Gogniat, C. Roland, P. Bomel, **Software Radio and Dynamic Reconfiguration on a DSP/FPGA platform**, Frequenz, Journal of Telecommunications, pages 152-159, N°58, 5-6/2004

[Bossuet 2003/O] L. Bossuet, G. Gogniat, J-P. Diguet, J-L. Philippe, **Chapter 4: Modeling (A Modeling Method for Reconfigurable Architectures)**, System-on Chip for Real Time Applications, The Kluwer International Series in Engineering and Computer Science, Vol. 711. Wael Badawy, Graham A. Julien (Eds), 2003, pages 170 – 180

[Bossuet 2005/CI] L. Bossuet, G. Gogniat, J.L. Philippe, **Generic Design Space Exploration for Reconfigurables Architectures**, *In 12th IEEE Reconfigurable Architectures Workshop, RAW 2005, Workshop of IEEE IPDPS 05*, April 4-5, 2005, Denver, Colorado, USA

[Delahaye 2004/CI] J-P. Delahaye, G. Gogniat, C. Roland, P. Bomel, **Software Radio and Dynamic Reconfiguration on a DSP/FPGA platform**, *The 3rd Workshop on Software Radios*, March 17-18, 2004, Karlsruhe, Germany

[Bossuet 2003c/CI] L. Bossuet, G. Gogniat, J-L. Philippe, **Communication costs driven design space exploration for reconfigurable architectures**, *13th International Conference on Field Programmable Logic and Applications*, September 1-3, 2003, Lisbon, Portugal

[Bossuet 2003b/CI] L. Bossuet, G. Gogniat, J-L. Philippe, **Fast Design Space Exploration Method for Reconfigurable Architectures**, *The International Conference on Engineering of Reconfigurable Systems and Algorithms (ERSA'03)*, June 23-26, 2003, Las Vegas, Nevada, USA

[Bilavarn 2003b/CI] S. Bilavarn, G. Gogniat, J-L. Philippe, **Fast Prototyping of Reconfigurable Architectures: An Estimation And Exploration Methodology from System-Level Specifications**, *Eleventh ACM International Symposium on Field-Programmable Gate Arrays*, February 23-25 2003, Monterey, California, USA

[Bossuet 2003a/CI] L. Bossuet, W. Burleson, G. Gogniat, V. Anand, A. Laffely, J-L. Philippe, **Targeting Tiled Architectures in Design Exploration**, *10th Reconfigurable Architectures Workshop (RAW 2003)*, April 22, 2003, Nice, France

[Bilavarn 2003a/CI] S. Bilavarn, G. Gogniat, J-L. Philippe, L. Bossuet, **Fast Prototyping of Reconfigurable Architectures From a C Program**, *IEEE International Symposium on Circuits and Systems (ISCAS 2003)*, 25-28 May, 2003, Bangkok, Thailand

[Bossuet 2002/CI] L. Bossuet, G. Gogniat, J-P. Diguet, J-L. Philippe, **A Modeling Method for Reconfigurable Architectures**, *International Workshop on System-on-Chip for Real-Time Applications*, July 6-7, 2002, Banff, Canada

[Bilavarn 2000/CI] S. Bilavarn, G. Gogniat, J-L. Philippe, **Area Time Power Estimation for FPGA Based Designs at a Behavioral Level**, *ICECS 2000*, December 2000, Beyrouth, Lebanon

[Bilavarn 2000/CI] S. Bilavarn, G. Gogniat, J.L. Philippe, **FPGA Area Time Power Estimation for DSP Applications**, *ICSPAT 2000*, October 2000, Dallas, TX, USA

[Bilavarn 1999/CI] S. Bilavarn, G. Gogniat, J. L. Philippe, **A Hardware-Software Codesign Methodology for Heterogeneous Architecture Estimation**, *ICSPAT 1999*, November 1-4, 1999, Orlando, Florida

[Bossuet 2005/CN] L. Bossuet, G. Gogniat, J-L. Philippe, **Méthode d'exploration de l'espace de conception ciblant des architectures reconfigurables**, *Journée IEEE Francophones sur l'Adéquation Algorithme Architecture (JFAAA'05*), 18-21 janvier, 2005, Dijon

[Bossuet 2002/CN] L. Bossuet, G. Gogniat, J-L. Philippe, **Flot d'exploration de l'espace de conception des architectures reconfigurable**, *JFAAA'02*, décembre, 2002, Monastir, Tunisie

[Bossuet 2002/CN] L. Bossuet, G. Gogniat, J-L. Philippe, **Méthode d'estimation relative des performances des architectures de FPGA**, *Colloque CAO*, 15-17 mai, 2002, Paris

[Bilavarn 2001/CN] S. Bilavarn, G. Gogniat, J-L. Philippe, **Estimation de performances à un niveau comportemental pour l'implantation sur composants FPGA**, *Sympa'7*, avril 2001, Paris

[Bilavarn 2000/CN] S. Bilavarn, J-P. Diguet, G. Gogniat, Y. Le Moullec, J-L. Philippe, **Méthode de Conception d'Architectures Hétérogènes pour les Applications de Traitement Numérique du Signal**, *JNRDM 2000*, mai, 2000, Montpellier

[Bilavarn 1999/CN] S. Bilavarn, G. Gogniat, J-L. Philippe, **Estimation d'Architectures Hétérogènes pour la Conception Conjointe Logicielle/Matérielle**, *Colloque CAO*, 10-12 mai, 1999, Fuveau

5. Axe 3 : Sécurité des systèmes embarqués

5.1 Introduction

On assiste actuellement à une mutation du comportement des utilisateurs vis-à-vis des technologies numériques, en effet ces dernières deviennent de plus en plus diffuses et visent à offrir l'accès à l'information n'importe où et n'importe quand. De nombreux standards de communication sans fils voient le jour et permettent aux utilisateurs de se connecter quel que soit l'environnement qui les entoure. Internet représente la source essentielle de cette information et le nombre d'utilisateurs ne cesse de croître. Parallèlement les terminaux mobiles offrent toujours davantage de services afin de faciliter l'échange de l'information aussi bien pour des raisons professionnelles que personnelles. Cependant cette vision de l'informatique diffuse peut se trouver menacée si la sécurité des terminaux mobiles n'est pas garantie [Kocher 2004]. Les premier virus visant les téléphones portables sont apparus récemment et il est vraisemblable que d'autres apparaîtront prochainement [Dagon 2004]. Aujourd'hui la réponse à ces attaques passe presque exclusivement par la mise en place de couches logicielles ayant pour but de détecter tout programme non désiré. Une telle solution n'est pas sans limite [Martin 2004], les attaques répétées sur Internet en font quotidiennement la démonstration.

Dans ce contexte et dans la perspective d'un accroissement massif des systèmes mobiles embarqués communicants (cartes à puce pour des transactions financières, suivi médical, télécommunications, set top box...), il est important d'imaginer de nouvelles solutions alternatives [Toshiba 2007]. Les protections à mettre en œuvre au niveau des systèmes électroniques ne doivent donc pas uniquement viser les composants logiciels mais également les composants matériels [Anderson 1996][Anderson 1997][Anderson2001]. De plus, la sécurité de ces systèmes doit être adressée au travers de multiples objectifs fondamentaux tels que la confidentialité, l'intégrité, la disponibilité, l'authentification, le non-reniement et la contrôlabilité.

De nouvelles techniques basées entre autre sur l'analyse matérielle du comportement du système doivent être proposées afin d'anticiper et détecter toute attaques contre le terminal (notion de surveillance active) [Arora 2005][Wolf 2006]. La définition d'une architecture sécurisée pour le domaine des systèmes embarqués doit également s'appuyer sur deux concepts essentiels, faible dégradation des performances et faible surcoût en consommation afin de ne pas pénaliser le système [Gogniat 2006]. Par ailleurs, afin de garantir au niveau matériel l'intimité et la confidentialité des systèmes et des données, les objectifs suivants doivent être adressés : i) protection des données privées qui correspondent principalement aux clés des algorithmes de cryptographie et aux données confidentielles, ii) protection de la conception contre le piratage et le *reverse-engineering* ce qui revient à garantir la confidentialité et l'intégrité des IP (*Intellectual Propertie*), et iii) protection du système de telle sorte qu'aucune personne malintentionnée ne puisse en prendre le contrôle.

Les technologies reconfigurables dynamiquement peuvent apportées des réponses intéressantes au problème de la sécurité dans la mesure où elles présentent de nombreux atouts afin de proposer des contres mesures efficaces contre les attaques matérielles tout en garantissant des performances élevées [Gogniat 2005a]. Certains aspects ne sont pas spécifiques aux technologies reconfigurables mais sont davantage le résultat des conceptions logique et physique comme par exemple les caractéristiques sans symptôme et résistant. Toutefois, une caractéristique essentielle doit être adressée lorsque le thème de la sécurité devient une priorité, il s'agit de l'adaptabilité ou flexibilité dynamique. En effet,

l'adaptabilité permet au système de réagir et d'évoluer en fonction de son état et de celui de l'environnement. Cette notion est essentielle afin de mettre en œuvre des mécanismes de protections efficaces qui s'adaptent en fonction des menaces.

Il est clair que le problème est complexe, aussi les problématiques adressées suivant cet axe de recherche sont multiples et visent toutes à renforcer la sécurité des systèmes embarqués (protection dynamique du système, protection des configurations, protection des bus…). L'objectif est de proposer des solutions matérielles permettant de minimiser le coût lié à la sécurité d'un système. Plusieurs contributions peuvent être citées :

- Sécurisation des FPGA du type SRAM/protection du bitstream par une approche dynamique (2003/2004)
- Architecture sécurisée pour les systèmes embarqués (2004/2005)
- Confidentialité et intégrité des données entre processeur et mémoire (2006/en cours)
- Réduction du surcoût lié à la sécurité par une approche de compression (2006/en cours)

La Figure 17 positionne les travaux menés suivant cet axe. Actuellement la dimension outil n'est pas considérée bien qu'un besoin important soit nécessaire dans ce domaine afin de proposer des flots de conception orientés sécurité. Quelques travaux existent aujourd'hui [Schaumont 2006][Verbauwhede 2007] mais ces derniers sont très insuffisants, aussi il indéniable qu'à l'avenir ce point deviendra critique.

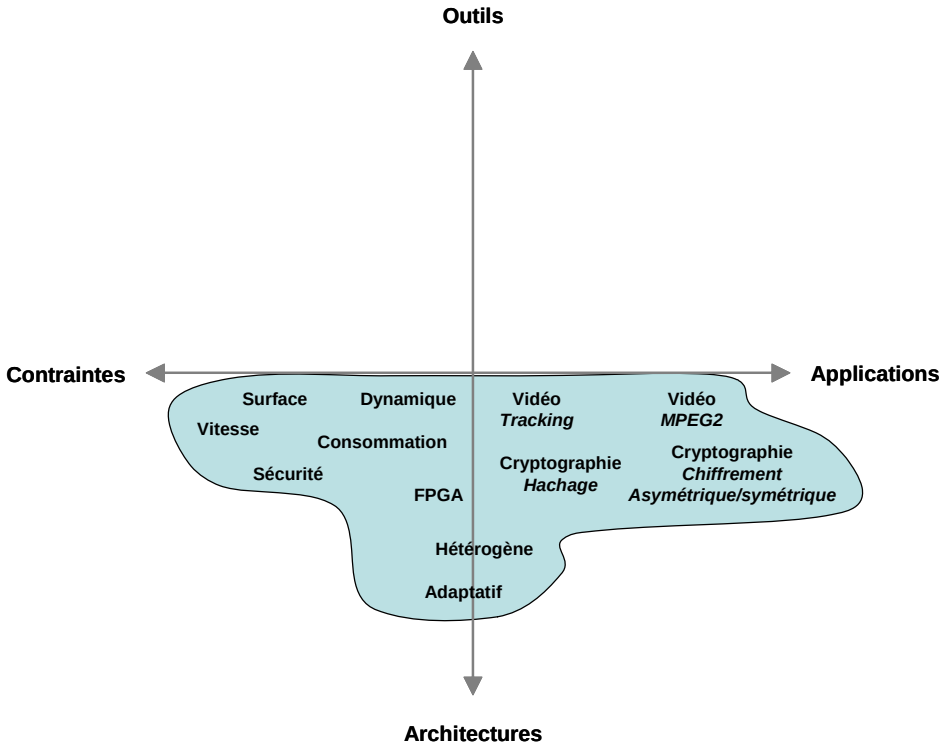


Figure 17 • Couverture de l'espace d'exploration (outils, contraintes, applications et architectures) de l'axe 4.

L'activité développée au sein de cet axe de recherche s'est d'abord intéressé à la protection de la configuration des FPGA [Bossuet 2006a] comme l'illustrent les Figures 18 et 19. Ensuite, l'utilisation des FPGA s'est étendue afin de mettre en œuvre des primitives

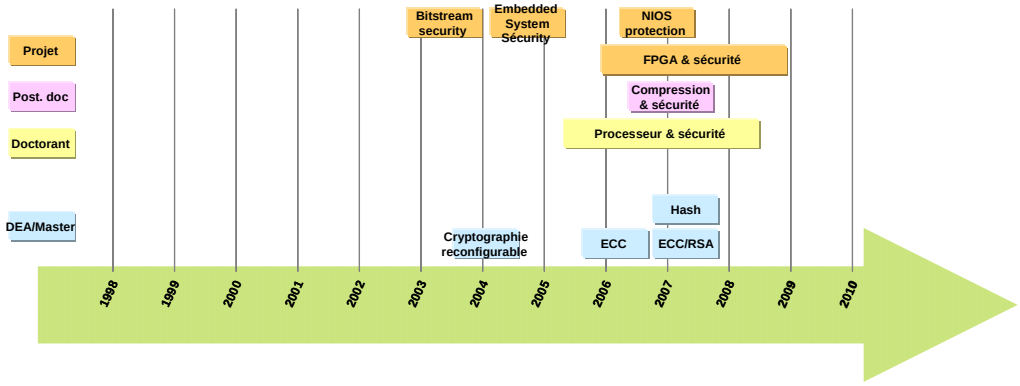


Figure 18 • Déroulement des travaux concernant l'axe de recherche Sécurité des systèmes embarqués.

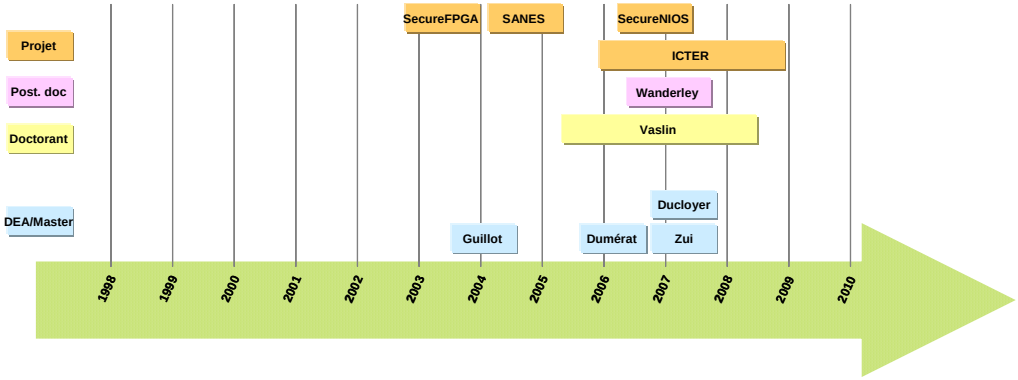


Figure 19 • Etudiants, doctorants et projets impliqués dans l'axe de recherche Sécurité des systèmes embarqués.

de sécurité (blocs de chiffrement symétriques). En novembre 2004, j'ai effectué un séjour de recherche de 10 mois à l'Université du Massachusetts à Amherst afin de travailler sur l'apport des technologies reconfigurables vis-à-vis de la sécurité [Gogniat 2005b]. J'ai proposé une architecture sécurisée basée sur un ensemble de capteurs et de moniteurs permettant de surveiller l'activité du système. L'architecture est reconfigurable dynamiquement afin d'adapter son niveau de sécurité en fonction de la menace à un instant donné. Depuis plusieurs travaux ont été menées. Une étude concerne la mise en œuvre de la confidentialité/intégrité des données entre un processeur et une mémoire [Vaslin 2007]. Une autre étude concerne la réduction du surcoût lié à la sécurité lors des accès à la mémoire par une approche de compression [Wanderley 2007]. Enfin, une dernière étude concerne la prise en compte des solutions matérielles de sécurité au niveau de l'OS.

5.2 Présentation des travaux

Afin d'illustrer l'activité menée au sein de cet axe de recherche la suite de cette section présente deux études adressant le problème de la sécurité des systèmes embarqués. La première étude correspond aux travaux que j'ai menés à l'Université du Massachusetts à Amherst et porte sur l'apport des FPGA pour la sécurité [Gogniat 2006]. La deuxième étude

correspond aux travaux de Romain Vaslin actuellement en thèse au laboratoire et porte sur la protection des échanges entre un processeur et sa mémoire [Vaslin 2007].

Architecture sécurisée pour les systèmes embarqués

L'approche considérée pour protéger les systèmes embarqués est d'imaginer un support architectural permettant la mise en place de la prévention, de la détection et de la correction des attaques. La plupart des systèmes embarqués sont implémentés sous la forme d'un système sur silicium, où tous les composants du système (processeur, mémoire, I/O) sont intégrés en un seul circuit. Nous proposons d'étendre la fonctionnalité de ces systèmes en intégrant également une matrice reconfigurable et des moniteurs de sécurité afin de renforcer la sécurité globale. L'utilisation de moniteurs permet de détecter les comportements anormaux du système [Nash 2005][Arora 2005][Wolf 2006]. Des mécanismes de défense matériels peuvent alors être mis en œuvre afin de contrer les attaques. Une telle approche est intéressant car l'analyse du système et la protection sont définies dans des unités matérielles dédiées et non directement au sein de l'application comme pour les techniques de défenses logicielles. Par ailleurs, les mécanismes de sécurité peuvent être mis à jour si besoin (de façon dynamique) ce qui assure la pérennité du système de protection.

La Figure 20 présente une vue générale de l'architecture. Plusieurs moniteurs sont considérés afin de surveiller différentes sources d'information du système. Le nombre et la complexité des moniteurs sont bien évidemment des paramètres importants car ils conditionnent directement les surcoûts liés à l'architecture de sécurité. Le rôle de ces moniteurs est de détecter les attaques contre le système. Pour cela, l'activité normale des modules sous surveillance est caractérisée et comparée en permanence avec l'activité réelle du système. Les notions d'autonomie et d'adaptabilité des moniteurs sont importantes afin

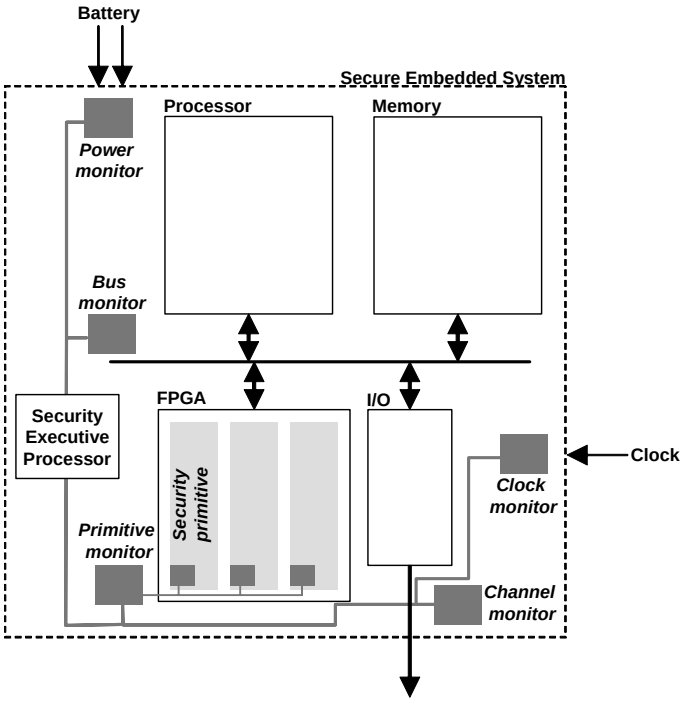


Figure 20 • Architecture SANES. La matrice reconfigurable contient les primitives de sécurité et les moniteurs afin de protéger le système.

de construire un réseau de surveillance efficace. En effet, les moniteurs sont autonomes afin de correspondre à un système tolérant aux fautes; si un moniteur est attaqué les autres doivent être en mesure de continuer à assurer la sécurité du système. Les moniteurs sont distribués à différents endroits du système afin d'analyser les points faibles de l'architecture (e.g. batteries, bus, primitives de sécurité, canaux de communication).

Différents niveaux de réaction peuvent être considérés en fonction du type d'attaque auquel le système doit faire face. Les réactions de type réflexes sont réalisées directement par un moniteur sans concertation avec les autres unités de sécurité. Dans ce cas le temps de réaction est très rapide. Les réactions de type globales sont mises en œuvre lorsqu'une attaque implique une modification importante du système. Dans ce cas, les moniteurs échangent des informations afin de définir une nouvelle configuration. Un tel scénario permet de détecter des attaques plus complexes mais implique également un temps de réaction plus long. Les moniteurs sont connectés par réseau sur silicium sécurisé. Ce réseau est également connecté à une unité de contrôle globale appelée SEP (*Security Executive Processor*) dont le rôle est d'assurer le lien sécurisé entre l'environnement extérieur et le système. Le contrôleur SEP correspond à une couche logicielle permettant d'instancier à distance de nouveaux moniteurs et de mettre à jour les politiques de sécurité des moniteurs existants. En cas de comportement anormal, le contrôleur SEP peut prendre le contrôle du système du point de vue matériel. Il peut par exemple annuler la gestion du niveau de batterie ou déconnecter des entrées/sorties. Ces extensions ont été apportées afin d'apporter une réponse globale au domaine des terminaux mobiles sécurisés.

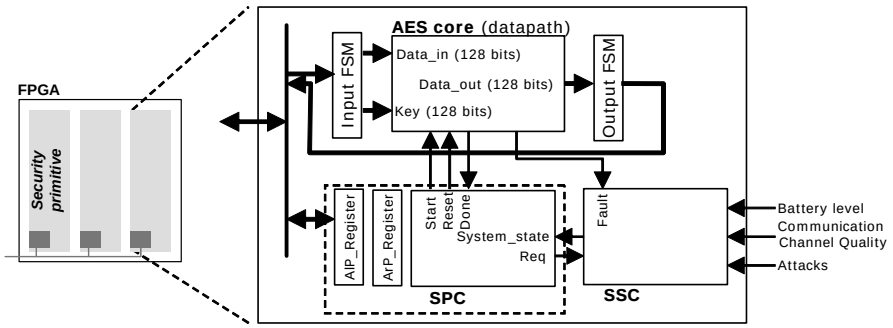


Figure 21 • Architecture de la primitive de sécurité. Le contrôleur SPC gère la politique de performance et le contrôleur SSC la politique de sécurité afin de détecter tout comportement suspect du système.

L'architecture reconfigurable au sein du système permet l'implémentation de primitives de sécurité. Une primitive de sécurité correspond à un accélérateur matériel adaptatif réalisant un algorithme lié à la sécurité (e.g. cryptage, filtrage IP, gestion de clés). Un système contient en général plusieurs primitives de sécurité qui travaillent indépendamment. Les objectifs de ces modules sont :

- L'accélération des calculs des algorithmes de sécurité comparée à une exécution logicielle;
- L'ajout de flexibilité comparé à une implémentation fixe afin de mettre à jour une primitive ou basculer d'une primitive à une autre en fonction des besoins du système et des contraintes à respecter;
- De fournir différents compromis en terme de débit, surface, latence, fiabilité, consommation et énergie afin de respecter les contraintes temps réel.

La figure 21 présente l'architecture de la primitive de sécurité pour l'algorithme de cryptage AES 128 bits. Trois unités composent l'architecture, 1) le chemin de données de la primitive de sécurité, 2) le contrôleur de la primitive de sécurité (SPC, *Security Primitive*

Controller), et 3) le contrôleur de sécurité du système (SSC, *System Security Controller*) qui est un moniteur. Le SPC est connecté au chemin de données afin de gérer la flexibilité de la primitive (gestion de la politique de performance). Les tâches de contrôle du SPC sont relatives à la gestion de la reconfiguration du chemin de données afin d'adapter ou de changer l'architecture de ce dernier. Le SPC est connecté au processeur du système afin de déterminer la configuration devant être mise en œuvre sur la primitive de sécurité (cette dernière dépend des protocoles ayant été négociés ou des standards à mettre en œuvre). Par exemple dans le cas des algorithmes de cryptage à clé secrète, cela correspond aux paramètres de l'algorithme (i.e. taille de la clé, mode, valeur de la clé). Un contrôleur SSC est également connecté à chaque primitive afin de surveiller l'activité de cette dernière et également contrôler l'état du système afin de détecter si des fautes ou des comportements suspects apparaissent. Le rôle du SSC est de détecter les attaques contre la primitive. Le contrôleur SSC est connecté aux autres moniteurs du système afin de pouvoir corréler les évènements apparaissant (e.g. batterie, bus, primitives de sécurité, canaux de communication).

Afin d'illustrer les concepts développés dans ces travaux nous avons défini une primitive de sécurité reconfigurable et deux moniteurs matériels. Le cas d'étude s'intéresse à l'algorithme AES [Daemen 2002] dans la mesure où ce standard FIPS a été sélectionné par le NIST afin de remplacer l'algorithme DES. De plus, AES joue actuellement et va jouer un rôle grandissant dans les réseaux de communication privés (IPSec) afin de garantir des communications sécurisées entre plusieurs utilisateurs.

Toutes les expérimentations ont été menées en considérant un composant FPGA Xilinx Virtex-II Pro. Les deux registres du contrôleur SPC contiennent respectivement les paramètres algorithmiques et architecturaux. Dans cette étude, les paramètres algorithmiques correspondent au type d'algorithme considéré (i.e. AES), au mode d'exécution (i.e. itératif, non itératif) et à la taille des clés secrètes (i.e. 128 bits). Les paramètres architecturaux correspondent à la fiabilité (i.e. sans, détection de fautes, tolérance aux fautes), au débit, à la surface (taux d'utilisation du composant) et à la consommation énergétique.

Quatre implémentations différentes ont été considérées afin de montrer la flexibilité pouvant être obtenue au sein d'une primitive de sécurité et afin de quantifier le coût de la sécurité. Les quatre implémentations sont les suivantes : mode itératif (FB), mode non itératif (N_FB), mode itératif avec détection de fautes (FB_FD) et mode itératif avec tolérance aux fautes (FB_FT). Pour cette étude nous avons considéré une clé de 128 bits. Les solutions du type itératif offrent des débits de l'ordre de quelques centaines de Mbits/s alors que la solution non itérative permet d'atteindre quelques Gbits/s. Le mécanisme de détection de fautes permet la détection d'une faute durant l'exécution de l'algorithme AES sans toutefois pouvoir la corriger. Dans notre cas, nous avons considéré une technique basée sur la parité des données afin de détecter une faute [Wu 2004]. Le mécanisme de tolérance aux fautes offre une solution plus résistante puisque l'opération de cryptage peut être réalisée même en cas d'attaque. Une technique du type TMR (redondance triple avec vote) a été sélectionnée [Carmichael 2001].

Dans cette étude nous avons utilisé l'outil Xilinx ISE Foundation 6.3i pour effectuer les étapes de synthèse et de placement/routage. L'estimation de la consommation a été réalisée en utilisant l'outil Xilinx XPower 6.3i. Comme présenté dans la Table 3, chaque solution correspond à un niveau de performance en terme de surface, débit et consommation. Le débit le plus élevé (3151.1 Mbits/s) est obtenu pour l'exécution en mode non itératif. En effet, dans ce cas tous les rounds de l'algorithme sont calculés en parallèle et pipelinés. En contre partie la surface et l'énergie consommées sont aussi les plus élevées (respectivement 13689 slices et 1724 mW). Selon l'état du système, des débits plus faibles ou une plus haute fiabilité sont nécessaires. La solution tolérante aux fautes est la plus sécurisée mais les surcoûts en surface et en énergie sont également élevés (respectivement 6302 slices et 1673 mW). La solution à base de détection de fautes n'induit qu'une faible dégradation en surface

et en énergie, respectivement +2.1% de slices et -2.7% en énergie comparé à une solution non sécurisée en mode itératif. Pour ces trois dernières implémentations les débits sont très proches, autour de 400 Mbits/s.

Une autre métrique est intéressante pour effectuer la comparaison de ces implémentations, l'efficacité énergétique qui représente le débit atteint par énergie consommée (Gbits/J). La solution non itérative représente la solution la plus efficace mais son principal inconvénient provient de son manque de fiabilité, une faute injectée peut compromettre l'ensemble du système. Les solutions itérative et itérative avec détection de fautes offrent la même efficacité. La solution tolérante aux fautes garantie la sécurité mais son surcoût en consommation est élevé ce qui conduit à une faible efficacité énergétique. Ainsi, la solution permettant la détection de fautes correspond à un bon compromis en

Table 3 • Comparaison des performances des 4 implémentations de l'algorithme AES (chemin de données). Chaque implémentation correspond à un compromis particulier en terme de performance vs. sécurité.

Version AES	Slices (% du nombre total)	Slices (% comparé à FB)	Période (ns)	Fréquence (MHz)	Puissance (mW)	Puissance (% comparé à FB)	Energie (nJ)	Débit (Mbits/s)	Débit (% comparé à FB)	Efficacité énergétique (Gbits/J)
FB	2192 (16%)		26.4	37.8	996		316	403.7		0.4
FB_FD	2240 (16%)	+2.1	25.3	39.4	970	-2.7	295	420.9	+4	0.4
FB_FT	6302 (46%)	+65.2	25.2	39.6	1673	+40.5	507	422.2	+4.4	0.25
N_FB	13689 (99%)	+83.9	40.6	24.6	1724	+42.2	70	3151.1	+87.7	1.8

terme de performance vs. sécurité et une telle solution pourrait être choisie par défaut dans l'architecture de la primitive.

La Figure 22 permet la comparaison de l'efficacité énergétique entre les solutions à base de processeur, ASIC et FPGA. La solution ASIC est la meilleure en performance (deux décades en Gbits/J comparée à la solution détection de fautes sur FPGA). En revanche, ce type d'approche n'offre aucune flexibilité. L'implémentation doit être en permanence sécurisée ou non, et aucune évolution n'est possible en ce qui concerne l'algorithme. Les

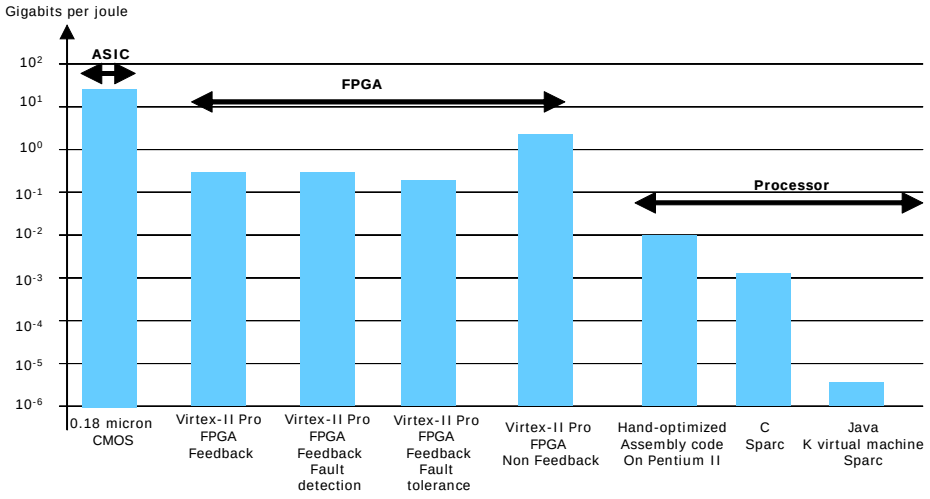


Figure 22 • Comparaison des efficacités énergétiques entre les solutions ASIC, processeur et FPGA pour l'algorithme AES 128 bits. Les valeurs pour les solutions ASIC et processeur sont obtenues de [Schaumont 2003]

solutions à base de processeurs offrent la flexibilité mais leurs performances restent faibles. Aussi, l'approche basée sur les technologies reconfigurables présente un compromis très intéressant puisque performance et sécurité peuvent être considérées simultanément.

Les trois solutions précédentes en mode itératif (FB, FB_FT, FB_FD) ont été considérées afin de réaliser une primitive de sécurité complète (Figure 23). La solution non itérative consomme trop de slices (99% de notre composant) pour envisager son intégration au sein de la primitive et surtout mettre en œuvre des mécanismes de reconfiguration dynamique. Nous avons définis trois modules reconfigurables qui sont le chemin de données, le SPC et le SSC. Une contrainte de surface a été associée à chaque module comme indiqué sur la Figure 23. Dans cette étude nous avons considéré une primitive de sécurité unique mais il n'y a pas de contrainte particulière concernant ce point (hormis la taille du composant). La communication entre les modules a été effectuée grâce à la mise en place de trois bus macro qui sont des IPs matériels prédéfinis et fournis par Xilinx [Xilinx 2004]. Ces bus macro permettent d'établir la communication entre les modules fixes et les modules reconfigurables. Un des bus macro est utilisé pour fournir le signal *fault* entre le chemin de données et le contrôleur SSC. Les deux autres bus macro permettent la connexion entre le chemin de données et le contrôleur SPC et véhiculent les signaux de contrôle (e.g. *start*, *reset*, *done*). La reconfiguration dynamique est réalisée par le SPC qui est connecté à l'interface ICAP qui gère l'auto reconfiguration dynamique partielle au sein du FPGA [Xilinx 2005]. L'interface ICAP est une IP matérielle de Xilinx pré placée/routée sur le composant. La reconfiguration est qualifiée de dynamique dans la mesure où elle s'effectue alors que le composant est en fonctionnement, ce qui est indispensable afin de pouvoir réagir en cas d'attaque ou afin d'adapter la politique de sécurité du système. Elle est également qualifiée de partielle car seul le chemin de données de la primitive est modifié. Enfin, il s'agit d'une auto reconfiguration car c'est le SPC qui reconfigure le chemin de

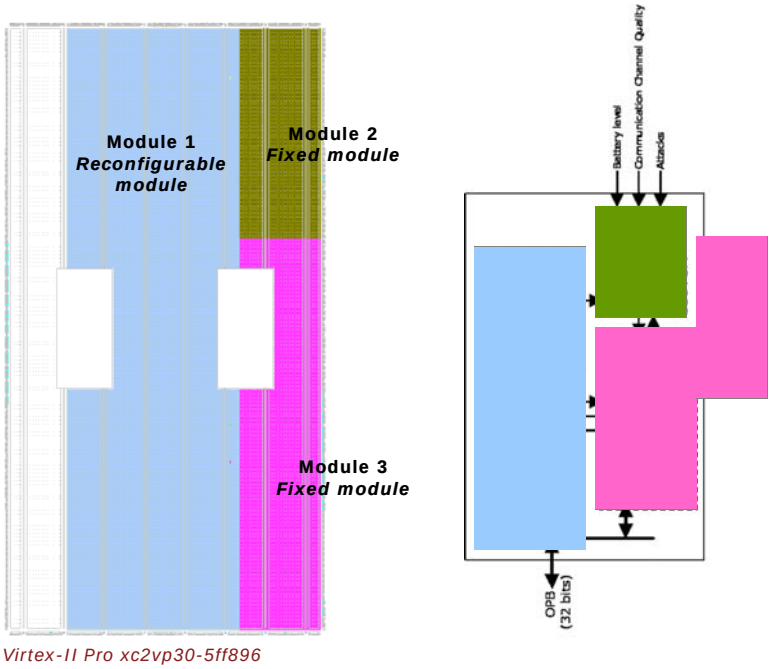


Figure 23 • Contrainte de synthèse de la partie reconfigurable : le chemin de données est reconfigurable dynamiquement, les deux contrôleurs sont fixes.

données. La Figure 24 présente les trois configurations possibles. Comme illustré le surcoût en surface de la solution tolérante aux fautes est important comparé aux deux autres solutions. Les contrôleurs SPC et SSC sont très petits et ne changent pas dans les trois cas (ils sont statiques dans cette étude). Leur complexité est faible comparée au chemin de données ce qui ne pénalise pas les performances du système. Pour cette étude nous avons considéré des politiques de performance et de sécurité très simples et principalement basées sur le dépassement de seuils critiques ou sur l'apparition d'un évènement. Pour des systèmes réels il est clair que les techniques à mettre en œuvre seront plus complexes et feront appel à des politiques avancées. Toutefois, les surcoûts induits par les contrôleurs devraient restés faibles par rapport au chemin de données.

Concernant les performances d'une solution de ce type, le temps de reconfiguration est directement lié à la taille du *bitstream*. Le *bitstream* complet qui est utilisé au moment de la mise sous tension représente 1415kB et les trois *bitstreams* correspondant aux solutions FB, FB_FD, FB_FT sont respectivement 356 kB, 356 kB et 463 kB. Dans notre cas, la fréquence d'horloge de l'interface ICAP est de 50 MHz ce qui conduit à un temps de reconfiguration moyen de 8 ms. A chaque fois qu'une reconfiguration est réalisée, il y a également un surcoût en consommation. Cependant, ce dernier est négligeable pour le cœur du FPGA et représente un accroissement d'environ 6% pour l'alimentation du FPGA (principalement les entrées/sorties) [Becker 2003].

Les résultats de cette étude, qui ont été obtenus en simulation avec les outils ISE Foundation et XPower de Xilinx, démontrent clairement l'intérêt d'une approche basée sur la reconfiguration dynamique afin d'adapter les niveaux de protection d'une primitive de sécurité et par extension d'une architecture. Toutefois, il est également important de valider l'étude à travers un prototype matériel. Pour cela, le laboratoire LESTER a travaillé en parallèle sur la réalisation d'un prototype basé sur une plateforme ML310 (contenant un Virtex-II Pro). L'étude a permis de mettre en œuvre l'auto reconfiguration au sein du FPGA en utilisant l'interface HW ICAP. Le processeur Power PC embarqué dans composant Virtex-II Pro ou un contrôleur matériel est donc capable de reconfigurer partiellement une zone du FPGA.

Les concepts liés à l'architecture SANES (primitives de sécurité et moniteurs) et les outils nécessaires à sa mise en œuvre ont donc été validés et ouvrent plusieurs perspectives de recherche intéressantes. Tout d'abord, il reste à prototyper l'architecture SANES dans sa

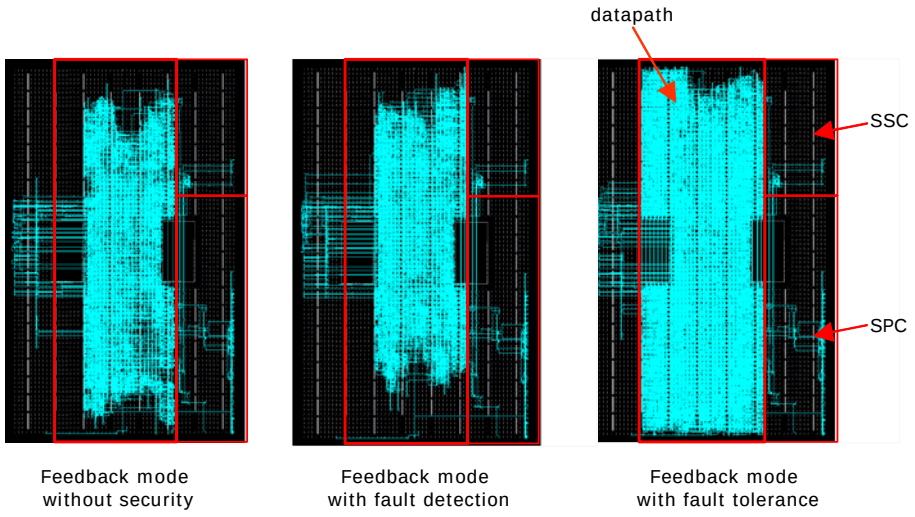


Figure 24 • Layout des trois configurations de la primitive de sécurité AES. Les trois modules (chemin de données, SPC et SSC) composent la primitive.

globalité en mettant en œuvre une ou plusieurs primitives de sécurité et un ensemble de moniteurs. La méthodologie de conception proposée doit être encore approfondie afin de mieux définir les différents niveaux de sécurité à mettre en place en fonction des contraintes sur le système. Les politiques de sécurité et de performances doivent être également raffinées afin de proposer plusieurs scénarios possibles et évaluer leurs coûts. Les moniteurs de sécurité jouent un rôle majeur dans l'architecture SANES, aujourd'hui plusieurs moniteurs potentiels ont été identifiés mais il reste à définir et à mettre en œuvre leur architecture afin de quantifier leur efficacité et leurs coûts. Un dernier point également sensible est relatif aux interactions entre les protections logicielles et matérielles. Cette dimension du problème n'a pas été encore analysée mais il est clair qu'il est important de mener une étude sur ce thème afin de renforcer la vision de la protection en profondeur du système.

Confidentialité et intégrité des données entre processeur et mémoire

Les solutions actuelles de sécurité ne permettent pas d'assurer la protection de l'ensemble du système [Vaslin 2006]. En effet certaines ressources se situent dans des zones non sécurisées. Un attaquant peut alors avoir un accès physique ou logique à ces ressources afin de mettre en œuvre une attaque ayant pour objectif de récupérer des informations ou de perturber le fonctionnement du système. Parmi les ressources en zone non sécurisée se trouve généralement une mémoire externe qui contient le code de l'application et certaines données pouvant être critiques. Il est alors indispensable de proposer des solutions permettant de garantir la confidentialité et l'intégrité de ces informations.

Ces travaux se situent dans ce contexte et visent à proposer une solution matérielle permettant d'assurer la protection des données présentes dans la mémoire. Plusieurs attaques peuvent être menées au niveau des bus reliant le processeur à la mémoire [Vaslin 2007]. En effet, un adversaire peut introduire des sondes sur les bus de données et d'adresses afin d'analyser et éventuellement perturber le trafic entre les deux composants [Elbaz 2006]. Si les données sont sensibles il est alors important de chiffrer ces dernières avec des algorithmes symétriques tels que 3DES [3DES 1995] ou AES [AES 2003]. Dans ce cas la confidentialité des données est garantie. L'attaquant peut observer les données mais ne peut les interpréter. Cependant ce type de protection n'est pas toujours suffisant et n'élimine pas les risques d'attaques. Le système est toujours sous la menace d'attaques du type : usurpation, réallocation, ou rejeu. Les attaques par usurpation (*spoofing* en anglais) consistent à placer sur le bus, lors d'un accès à la mémoire, une donnée ou une instruction non valide entraînant ainsi un fonctionnement incorrect du système. Les attaques par réallocation (*relocation* ou *splicing* en anglais) consistent, lors d'un accès à la mémoire à une adresse donnée, à placer sur le bus une donnée ou une instruction provenant d'une autre adresse mémoire. Si toute la mémoire est chiffrée avec la même clef, l'instruction sera exécutée mais cette dernière ne correspondra pas à celle souhaitée puisqu'elle aura été remplacée. Par exemple l'instruction de substitution pourrait entraîner un détournement du programme vers une zone contenant un programme visant à pirater le système. Les attaques par rejeu (*replay* en anglais) sont proches des attaques par réallocation dans la mesure où la donnée accédée est remplacée par une autre donnée. Toutefois dans ce cas la donnée de substitution correspond à la donnée présente à la même adresse mémoire mais dont la valeur n'est plus valide.

Afin de garantir une protection efficace contre ces attaques nous proposons une solution matérielle basée sur une technique de chiffrement OTP (*One Time Pad*). La difficulté principale n'est pas de protéger le système dans la mesure où plusieurs solutions existent (i.e. XOM [Lie 2000][Lie 2003], AEGIS [Suh 2003a][Suh 2003b][Suh 2005], PE-ICE [Elbaz 2006]) mais de minimiser l'impact de cette sécurité sur les performances du système. En effet il est essentiel pour les applications visées de réduire la latence résultant d'un accès mémoire et de minimiser la surface de l'unité de protection.

Le principe de l'algorithme de chiffrement OTP est d'utiliser une clef secrète unique pour protéger chaque donnée. Pour cela une opération du type ou-exclusif est réalisée entre chaque donnée et chaque clef secrète garantissant ainsi la confidentialité de l'ensemble des données présentes en mémoire. Le fonctionnement de l'approche OTP est le suivant :

- Lors d'une opération de lecture mémoire, pendant le temps d'accès à la donnée en mémoire, une clef unique est calculée. Une fois la donnée présente sur le bus l'opération ou-exclusif peut alors être effectuée. L'avantage de cette technique provient du recouvrement entre le calcul de la clef et l'accès aux données.
- Lors d'une opération d'écriture mémoire, aucun recouvrement n'est malheureusement possible puisqu'il faut d'abord générer la clef avant d'effectuer l'opération ou-exculsif.

Dans la plupart des systèmes, le temps d'accès mémoire prend un nombre de cycles important. Par conséquent, la lecture d'une ligne de cache peut être suffisamment longue afin de calculer complètement en parallèle la clef unique. Dans notre cas l'algorithme de chiffrement AES est utilisé afin de générer cette clef secrète. La Figure 25 illustre ce principe. L'exemple en haut de la figure (Figure 25.a) correspond aux approches classiques (e.g. XOM, PE-ICE) où l'opération de chiffrement est réalisée en série avec l'opération d'accès à la mémoire. Les exemples en bas de la figure (Figure 25.b et Figure 25.c) utilisent la technique OTP. Le gain en nombre de cycles est important et varie suivant l'utilisation ou non d'un pipeline. Afin de garantir l'intégrité des données nous ajoutons un mécanisme de détection d'erreur du type CRC (*Cyclic Redundancy Check*). Pour chaque ligne du cache écrite en mémoire externe un CRC est calculé et stocké dans le module de sécurité. Si une

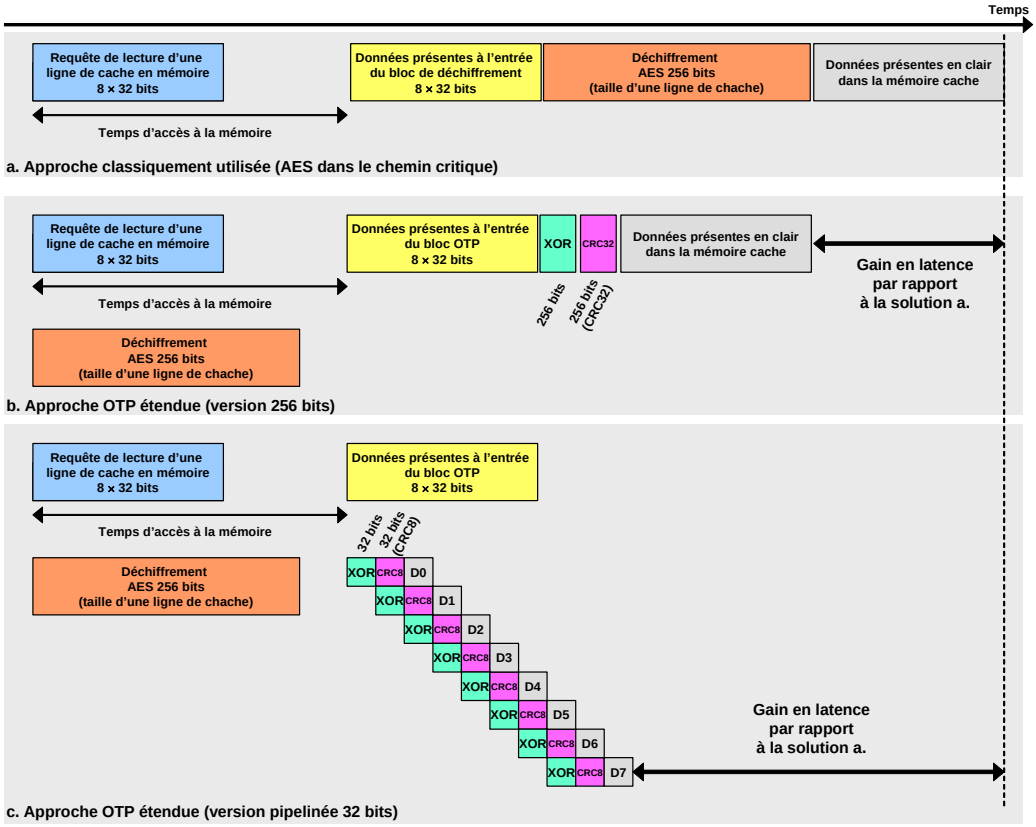


Figure 25 • Modèles d'exécution des solutions de protection classiques et de la solution OTP étendue en version pipelinée et non pipelinée.

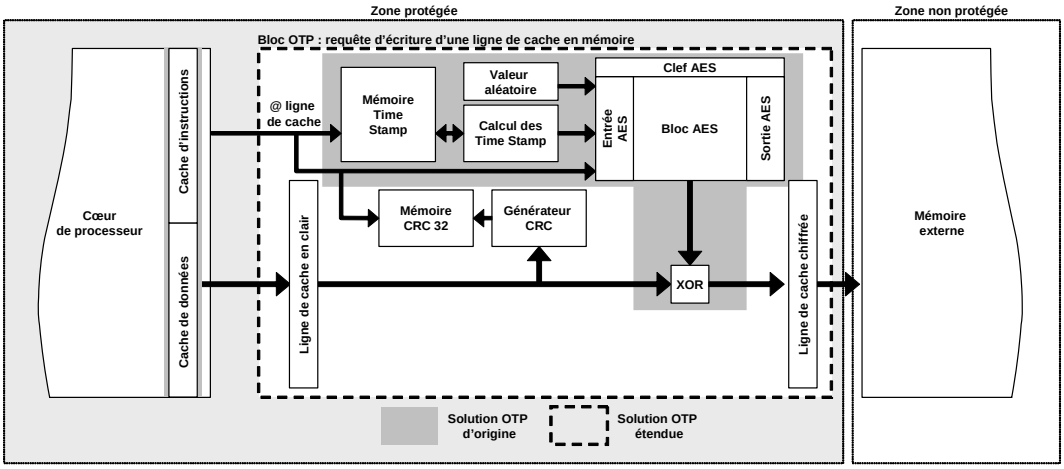


Figure 26 • Architecture OTP en mode écriture d'une ligne de cache en mémoire. La ligne de cache est chiffrée afin d'être protégée.

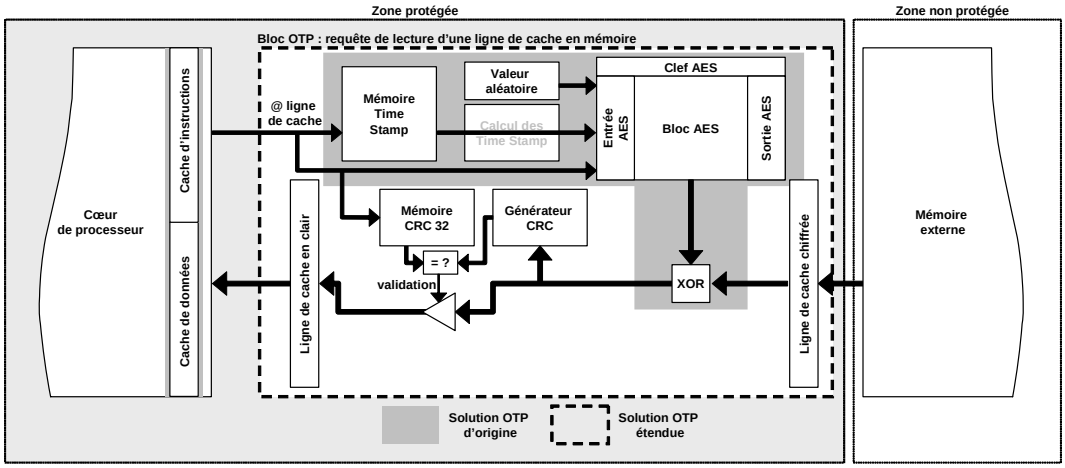


Figure 27 • Architecture OTP en mode lecture d'une ligne de cache en mémoire. La ligne de cache est déchiffrée puis l'intégrité est vérifiée.

donnée est modifiée dans la mémoire, le calcul du CRC permet de valider cette dernière avant d'effectuer l'opération d'écriture dans la ligne du cache.

La Figure 26 présente le fonctionnement du système lors d'une opération d'écriture d'une ligne de cache en mémoire externe. Le module de sécurité est composé d'un bloc AES qui génère la clef secrète pour chaque ligne de cache. Cette clef doit être unique afin de garantir la sécurité du système. Pour cela plusieurs données alimentent le bloc AES : une valeur aléatoire (initialisée à la mise sous tension du système), l'adresse de la ligne de cache en mémoire et un tampon temporel (*time stamp*). L'utilisation d'un tampon temporel permet de se protéger contre les attaques du type rejeu puisque chaque donnée écrite en mémoire possède un tampon qui lui est propre. L'utilisation de l'adresse permet de se protéger contre les attaques par réallocation. L'utilisation du CRC permet de se protéger contre les attaques du type usurpation puisque la donnée est signée. Lors d'une écriture en mémoire d'une ligne de cache, l'adresse mémoire est utilisée afin de stocker dans le module OTP le tampon temporel et le CRC de la donnée. Ces valeurs sont utilisées lors de la lecture de la ligne de

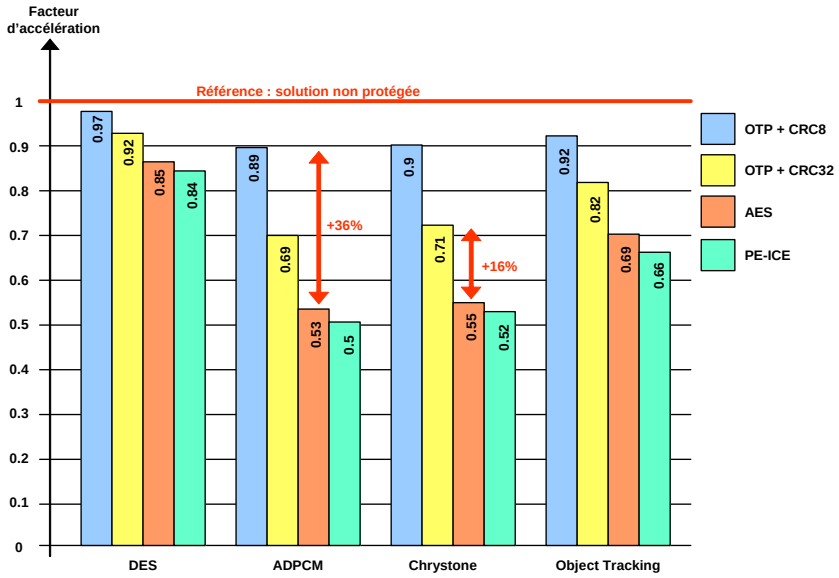


Figure 28 • Comparaison des performances d'exécution des solutions OTP, AES et PE-ICE pour différents benchmarks.

cache. Une fois la clef calculée, une opération ou-exclusif est réalisée entre cette dernière et la donnée afin de stocker une donnée chiffrée dans la mémoire externe.

Lors de l'opération de lecture (Figure 27) l'adresse de la ligne de cache permet de récupérer le tampon temporel afin de recalculer la clef secrète nécessaire pour l'opération ou-exclusif. Pendant que les données sont lues depuis la mémoire externe, le bloc AES calcul la clef secrète. Lorsque les données chiffrées sont présentes à l'entrée du bloc OTP le déchiffrement peut avoir lieu. Une fois les données déchiffrées, le CRC est calculé et comparé avec celui précédemment stocké. Si les deux valeurs sont identiques alors la donnée est valide et peut être transmise au cache, sinon la valeur est rejetée et le système peut réagir en conséquence.

La solution développée a été intégrée et validée pour un système à base du processeur NIOS sur une carte de prototypage ALTERA. En terme de mémoire notre approche implique un surcoût inférieur aux solutions existantes même si le stockage des tampons temporels et des CRC peut induire un accroissement de 32% de la taille mémoire pour une version pipelinée utilisant des CRC8. Concernant la latence notre approche permet un gain important comme illustré dans la Table 4.

Table 4 • Comparaison des surcoûts des différentes solutions par rapport à une solution élémentaire basée sur l'algorithme AES. Les latences présentées sont celles ajoutées par le chiffrement (le temps pour accéder aux données est également inclus, 8 cycles sont nécessaires).

	Solution AES (sans intégrité)	Solution OTP+CRC32 (version séquentielle)		Solution OTP+CRC8 (version pipelinée)		XOM AES+MAC		PE-ICE AES		AEGIS OTP+hash tree	
			overhead		overhead		overhead		overhead		overhead
Mémoire (koctets)	512	600	+18.75%	662	+31.25%	N/A	N/A	776	+50.7%	768	+50%
Latence en lecture (cycles)	22 (14+8)	11 (8+3)	-11	3 (0+3)	-19	22	0	25 (17+8)	+3	≈(SHA-1)	+4502/69
Latence en écriture (cycles)	22 (14+8)	12 (8+4)	-10	12 (8+4)	-10	22	0	26 (18+8)	+4	N/A	N/A

Enfin, la Figure 28 présente le facteur d'accélération de notre solution par rapport aux approches existantes et pour différentes applications. En utilisant la version la plus rapide du module OTP nous obtenons un gain d'environ 36% en terme de performance par rapport à une approche classique basée sur un algorithme de chiffrement AES (sans vérification de l'intégrité). Le surcoût temporel lié à la sécurité par rapport à une solution non sécurisée est de l'ordre de 10% et dépend du taux de miss de cache.

5.3 Conclusion

Le domaine de la sécurité des systèmes embarqués bien qu'assez récent est extrêmement actif dans la mesure où les enjeux associés sont très importants. Nous participons à ce mouvement à travers plusieurs études qui visent à renforcer la sécurité d'un système principalement implémenté sur FPGA. Pour cela nous avons adressé le problème de la protection des fichiers de configurations pour les technologies SRAM. Nous avons également imaginé et prototypé une approche de sécurité dynamique permettant d'adapter le niveau de sécurité en fonction des menaces. Nous avons développé plusieurs techniques permettant d'assurer la confidentialité et l'intégrité des échanges de données entre un processeur et sa mémoire tout en réduisant le surcoût temporel. Nous avons également proposé une solution permettant de garantir la sécurité des échanges de données dans les NoC. Nous avons développé plusieurs moniteurs matériels ayant pour bus d'analyser le comportement du système afin de détecter toute déviation anormale. Nous étendons actuellement nos travaux afin de prendre en compte les OS et ainsi adapter le niveau de protection des différentes tâches s'exécutant sur un processeur. L'objectif de ces travaux est d'accompagner les solutions logicielles actuelles par des unités matérielles spécifiques.

Afin de mener à bien ces travaux 1 doctorant [Vaslin 200X/T] et un post-doc [Wanderley 2007/P] ont participé ou participent actuellement au projet et 4 stagiaires de DEA ou de Master [Guillot 2004/D] [Dumérat 2005/D] [Zui 2007/D] [Ducloyer 2007/D].

Les travaux menés au sein de cet axe de recherche ont conduit à 14 publications scientifiques (1 revue, 13 conférences internationales) [Bossuet 2006a/R] [Vaslin 2007b/CI] [Vaslin 2007a/CI] [Wanderley 2007b/CI] [Wanderley 2007a/CI] [Diguet 2007/CI] [Wolf 2006/CI] [Wanderley 2006/CI] [Vaslin 2006b/CI] [Vaslin 2006a/CI] [Gogniat 2006b/CI] [Gogniat 2005b/CI] [Gogniat 2005a/CI] [Bossuet 2004/CI].

5.4 Fiche de synthèse des travaux

Co-encadrements de thèses et de Post. doc

[Vaslin 200X/T] Romain Vaslin 2005/2008 – Bourse MENRT

Sécurité des systèmes embarqués
Soutenance prévue en 2008, en Co-direction avec le CR CNRS Jean-Philippe Diguet (50%)

[Wanderley 2007/P] Eduardo Wanderley 2006/2007 – Bourse Contrat ANR

Réduction de l'overhead lié à la sécurité par une approche de compression
Post. doc – Juillet 2006/Juillet 2007

Encadrement de stages de DEA et de Master

[Guillot 2004/D] Jérémie Guillot

Cryptographie et auto reconfiguration dynamique sur FPGA
DEA Lorient, année 2003/2004

[Dumérat 2006/D] Arnaud Dumérat

Algorithme de chiffrement ECC : détection et tolérance aux fautes
Master Math/Info Vannes, année 2005/2006

[Zui 2007/D] Tao Zui

Algorithme de chiffrement : mise en oeuvre sur le Nios
Master Math/Info Vannes, année 2006/2007

[Ducloyer 2007/D] Ducloyer Sylvain

Architecture matérielle pour le hachage : application à MD5/SHA-1/SHA-2
Master Electronique Lorient, année 2006/2007

Collaborations scientifiques

Université du Massachusetts, Amherts, USA

[SecureNIOS 2007] Projet SecureNIOS

Trusted Computing with NIOS based systems
Type : Projet sur fond propre
Durée : 2006/2007
Partenaires : LESTER, VSPG

[SANES 2005] Projet SANES

Security Architecture for Embedded Systems
Type : DGA ERE (Etude et Recherches à l'Etranger)

Durée : 2004/2005
Partenaires : LESTER, VSPG

[SecureFPGA 2004] Projet SecureFPGA

Bitstream security for SRAM based FPGAs
Type : Projet sur fond propre
Durée : 2003/2004
Partenaires : LESTER, VSPG

[ICTeR 2008] Projet ICTeR

Les Technologies Reconfigurables : Intégrité et confidentialité des informations
Type : Projet ANR, projet Blanc
Durée : 2006/2008
Partenaires : LIRMM, ENST, LIST/Univ. St Etienne, NETHEOS

Publications scientifiques

[Bossuet 2006a/R] L. Bossuet, G. Gogniat, W. Burleson, **Dynamically Configurable Security for SRAM FPGA Bitstreams**, International Journal of Embedded Systems, IJES, From Inderscience Publishers Vol. 2, Nos. 1/2, 2006

[Gogniat XXXX/R] G. Gogniat, T. Wolf, W. Burleson, J-P. Diguet, L. Bossuet, R. Vaslin, **Reconfigurable hardware for high-security/high-performance embedded systems: The SANES perspective**, en révision à IEEE Transactions on VLSI Systems Special Section on Configurable Computing Design

[Vaslin 2007b/CI] R. Vaslin, G. Gogniat, J-P. Diguet, R. Tessier, W. Burleson, **High Efficiency Protection Solution for Off-Chip Memory in Embedded Systems**, *The International Conference on Engineering of Reconfigurable Systems and Algorithms*, June 25-28, 2007, Las Vegas, Nevada, USA

[Wanderley 2007b/CI] E. Wanderley, G. Gogniat, J-P. Diguet, **A Code Compression Method With Confidentiality and Integrity Checking**, *The 2007 International Conference on Embedded Systems and Applications*, June 25-28, 2007, Las Vegas, Nevada, USA

[Wanderley 2007a/CI] E. Wanderley, R. Elbaz, L. Torres, G. Sassatelli, R. Vaslin, G. Gogniat, J-P. Diguet **IBC-EI: An Instruction Based Compression method with Encryption and Integrity Checking**, 3rd International Workshop on Reconfigurable Communication Centric System-on-Chips (ReCoSoC'07), 18th-20th June 2007, Montpellier, France

[Vaslin 2007a/CI] R. Vaslin, G. Gogniat, E. Wanderley, R. Tessier, W. Burleson
Low latency solution for confidentiality and integrity checking in embedded systems with off-chip memory, 3rd International Workshop on Reconfigurable Communication Centric System-on-Chips (ReCoSoC'07), 18th-20th June 2007, Montpellier, France

[Diguet 2007/CI] J-P. Diguet, G. Gogniat, S. Evain, R. Vaslin, E. Juin, **NOC-centric security of reconfigurable SoC**, *The 1st ACM/IEEE International Symposium on Networks-on-Chip*, May 7-9, 2007, Princeton University, New Jersey, USA

[Wanderley 2006/CI] E. Wanderley, G. Gogniat, J-P. Diguet, **Bus Decryption Overhead Minimization with Code Compression**, *The 3rd III IEEE Southern Conference on Programmable Logic*, February 26-28, 2007, Mar del Plata, Argentina

[Vaslin 2006b/CI] R. Vaslin, G. Gogniat J-P. Diguet, **Secure architecture in embedded systems: an overview**, *Reconfigurable Communication-centric SoCs (ReCoSoc'06)*, July 3-5, 2006, Montpellier, France

[Vaslin 2006a/CI] R. Vaslin, G. Gogniat, J-P. Diguet, A. Pegatoquet, **Trusted Computing - A New Challenge for Embedded Systems**, *The 13th IEEE International Conference on Electronics, Circuits and Systems (ICECS 2006)*, December 10-13, 2006, Nice, France

[Gogniat 2006b/CI] G. Gogniat, T. Wolf, W. Burleson, **Reconfigurable security support for embedded systems**, *The 39th IEEE Hawaii International Conference on System Science (HICSS-39)*, January 2006, Poipu, HI, USA

[Wolf 2006/CI] T. Wolf, S. Mao, D. Kumar, B. Datta, W. Burleson, G. Gogniat, **Collaborative monitors for embedded system security**, in Proc. of First International Workshop on Embedded Systems Security in conjunction with 6th Annual ACM International Conference on Embedded Software (EMSOFT), Seoul, Korea, Oct. 2006

[Gogniat 2005b/CI] G. Gogniat, L. Bossuet, W. Burleson, **Configurable computing for high-security/high-performance ambient systems**, *5th International Workshop Embedded Computer Systems: Architectures, MOdeling, and Simulation SAMOS 2005, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, Volume 3553/2005*, July 18-20, 2005, Samos, Greece

[Gogniat 2005a/CI] G. Gogniat, T. Wolf, W. Burleson, **Reconfigurable Security Primitive for Embedded Systems**, *The IEEE International Symposium on System-on-Chip (SOC 2005)*, November 15-17, 2005, Tampere, Finland

[Bossuet 2004/CI] L. Bossuet, G. Gogniat, W. Burleson, **Dynamically Configurable Security for SRAM FPGA Bitstreams**, *11th Reconfigurable Architectures Workshop (RAW 2004)*,. April 26–27 2004, Santa Fé, USA

6. Conclusion et perspectives de recherches

Après cette présentation détaillée de mes activités de recherche depuis 1998 et avant de proposer des directions de recherches pour l'avenir, il est nécessaire de se poser la question du futur et de le mettre en perspective avec les évolutions que nous avons connues [Beiu 2007]. Quelles seront les architectures de demain ? Quelles applications seront mises en œuvre ? Quels seront les flots de conception ? La réponse est évidemment bien difficile et source de polémiques. Toutefois, certaines pistes se dessinent.

Il semble que les maîtres mots seront le parallélisme massif en nombre de processeurs embarqués et l'auto adaptabilité des systèmes. Ces deux concepts seront fondamentaux afin de faire face à la complexité des traitements à mettre en œuvre et à la mobilité des futurs systèmes qui continûment devront s'adapter à leur environnement dynamique tout en gérant de façon extrêmement fine leur consommation.

Par exemple la société picoChip propose des processeurs incorporant jusqu'à 430 cœurs de processeurs 16 bits sur une matrice de silicium [Pico 2007]. Intel a annoncé récemment le Teraflops Research Chip qui correspond au premier composant atteignant une puissance de calcul de 1 Teraflops tout en minimisant la consommation en puissance [Shekhar 2007]. Ce composant explore les voies prometteuses des futurs composants multi cœurs sur silicium ainsi que la problématique des interconnexions configurables en fonction des charges de calcul à un instant donné. Actuellement 80 cœurs, contenant chacun deux unités de calculs flottants, sont proposés mais déjà Intel annonce que l'avenir verra des composants intégrant jusqu'à plusieurs centaines de cœurs de calculs.

L'auto adaptabilité des systèmes va également devenir indispensable dans la mesure où il ne sera plus possible d'optimiser totalement hors ligne la conception de tels systèmes et qu'ainsi les systèmes devront s'auto gérer en fonction de l'application et de l'environnement auquel ils seront confrontés (nous entrons dans l'ère du *Self-X_ing : Self-defining, Self-optimazing, Self-healing, Self-protecting…).*

Par exemple le problème de la gestion des points chauds dans les circuits devient préoccupant dans la mesure où la dispersion de température dans les circuits peut devenir critique pour des raisons de fiabilité et de vieillissement prématurés des composants [Shekhar 2007][Rosing 2007]. Le système devra donc être capable d'analyser son architecture et de redistribuer les calculs sur ses ressources si des points chauds apparaissent.

Le réseau de communication devra également s'adapter dynamiquement afin de s'auto optimiser lors de l'exécution. Des solutions à base de réseaux sur silicium configurables semblent prometteuses [Diguet 2007][Hansson 2007]. La gestion dynamique de la consommation deviendra également un point très sensible du système, aussi il sera nécessaire de gérer dynamiquement les unités actives et non actives. Certaines solutions existent déjà mais il faudra atteindre des niveaux de finesse en terme de gestion dynamique des ressources très supérieures aux solutions actuelles.

Afin de mettre en œuvre de telles architectures aux caractéristiques mouvantes (flexibles) la mise en place de capteurs de surveillance semble incontournable. Ces capteurs permettront d'analyser les performances, les défaillances mais aussi les attaques contre le système [Arora 2005][Wolf 2006]. La sécurité devient un paramètre fondamental dans la mesure où la quantité d'informations stockées dans les systèmes mobiles s'accroît à chaque nouvelle génération de produits. Cette tendance va se confirmer et les utilisateurs seront particulièrement attentifs à cet aspect [ePaynews 2005]. Si la sécurité n'est pas garantie cela constituera un frein majeur au déploiement de ces futurs systèmes. La sécurité deviendra donc une des métriques majeures des systèmes de demain.

Ces évolutions à venir et déjà perceptibles vont impliquer des changements en profondeur des outils de conception. Il reste beaucoup de choses à inventer. L'efficacité énergétique (GOPS/mW voire TOPS/mW) devra être au cœur des préoccupations des outils de conception. La gestion du parallélisme sera également fondamentale, comment gérer la mise en œuvre des applications sur des centaines de processeurs potentiellement hétérogènes (crypto processeur, processeur réseau, processeur vidéo…) au sein d'un même composant ? Comment gérer l'hétérogénéité de ces futures architectures du point de vu applicatif ?

Le développement de nouvelles techniques permettant d'explorer massivement le parallélisme est indispensable [Fang 2007]. Ce dernier devra être analysé à tous les niveaux de conception afin de garantir les facteurs d'accélération requis pour les futures applications (audio, vidéo, graphique, communications). Les techniques de compilation devront également prendre en compte ce parallélisme mais également la reconfiguration dynamique des architectures des processeurs à l'image des solutions proposées par Stretch qui permettent d'identifier des séquences d'instructions et de les traduire sous la forme de contextes matériels optimisés [Stretch 2007].

La définition de nouvelles méthodologies de conception et les outils associés permettant la convergence des compétences requises pour concevoir ces systèmes sera également au cœur des préoccupations. Les tentatives autour des langages tels que UML devront se renforcées et devront s'accompagner de méthodologies de conception non ambiguës permettant la mise en œuvre de transformations systématiques et essentiellement automatiques [UML4SoC 2004].

La gestion des configurations des systèmes sera également un enjeu principal. Il faudra développer des techniques permettant de gérer efficacement les différents états du système, sachant que certains d'entre eux pourront apparaître durant le cycle de vie du produit. Des techniques d'apprentissages seront vraisemblablement nécessaires afin de permettre aux systèmes de s'adapter en fonction de contraintes et d'objectifs potentiellement contradictoires.

Les outils devront franchir un nouveau pas afin de permettre la spécification, la conception et la validation de systèmes à comportement dynamique. L'utilisation des OS devra se généraliser afin de simplifier la gestion des services requis pour la mise en œuvre de la reconfiguration et la connectivité des systèmes. En effet, l'abstraction sera fondamentale afin que les concepteurs puissent appréhender la complexité des futurs systèmes et garantir leur interopérabilité.

Ces évolutions sont ambitieuses et cela d'autant plus au regard des prévisions des experts Européens qui indiquent que les cycles de conception devront être réduit de 50% dans les dix années à venir afin de diminuer par 50% le coût de la conception. Par ailleurs, les concepteurs devront gérer un accroissement de l'ordre de 100% de la complexité des systèmes tout en assurant une réduction de 20% de l'effort de conception [ARTEMIS 2006].

Face à ces défis il est intéressant d'imaginer de nouvelles activités de recherches permettant de lever certains verrous et ainsi progressivement transformer ces défis en réalité. Cette aventure scientifique et humaine est riche et malgré les doutes auxquels nous sommes confrontés il est essentiel d'avancer afin de construire l'avenir. Plusieurs thèmes s'inscrivant dans les trois axes de recherches présentés précédemment peuvent être définis. Par ailleurs, les interactions entre les axes 1 (systèmes embarqués), 2 (architectures reconfigurables) et 3 (sécurité des systèmes embarqués) se trouvent renforcées comme nous le verrons.

Concernant la définition de nouvelles méthodologies de conception nous proposons d'étendre les concepts propres aux approches MDA afin de prendre en compte l'adaptabilité des systèmes. Pour cela le caractère dynamique aussi bien au niveau applicatif qu'au niveau architectural doit être considéré. L'approche que nous proposons consiste donc, à partir d'une description UML de la spécification, à mettre en œuvre plusieurs étapes

de transformations de modèles afin de progressivement passer d'une spécification purement fonctionnelle à une spécification au niveau RTL pour le matériel et à une spécification du type C pour l'embarqué au niveau logiciel [MOPCOM 2007]. La spécification fonctionnelle est basée sur un ensemble de *Use Cases* qui permettent de définir les fonctionnalités devant être réalisées. Ces *Use Cases* sont accompagnés d'un ensemble de diagrammes UML afin de raffiner les spécifications. Ensuite, des transformations sont mises en œuvre afin d'aboutir à des modèles dépendants tout d'abord d'une architecture logique puis d'une plateforme d'exécution. Ces différents niveaux de conception sont validés à travers des spécifications SystemC au niveau TLM qui permettent de simuler le système en utilisant des modèles de communications progressivement raffinés. Ce type d'approche repose sur l'utilisation de profils UML adaptés permettant de spécialiser le langage UML à un domaine d'application donné. Dans le cadre de ce projet l'extension du profil MARTE [MARTE 2007] est prévue afin de prendre en compte le caractère dynamique des applications et des plateformes d'exécution.

Les méthodologies de conception visant à coupler, au sein d'une architecture multiprocesseur, des clusters de traitements intégrant un processeur associé à des coprocesseurs et/ou des accélérateurs matériels sont particulièrement intéressantes, à l'image des solutions proposées par [Synfora 2007] et [Tensilica 2007]. Ce deuxième projet [MASTER 2007] vise donc à fournir un flot unifié, intégré, et optimisé pour la conception de systèmes sur puce hétérogènes sur cible FPGA. Le but est d'augmenter la productivité des équipes de conception en visant spécifiquement le domaine des systèmes embarqués. Ceci implique le développement de méthodes et d'outils permettant l'exploration efficace et automatique de l'espace de conception (cœurs de processeurs, coprocesseurs, IP matériels, mémoires, bus ou NoC) de systèmes embarqués complets, en tenant compte des différentes contraintes applicatives (principalement les contraintes de consommation et de débit) ainsi que des interactions généralement ignorées entre la synthèse d'architecture et les techniques d'implantation des cibles physiques et des outils de bas niveau associés (synthèse logique et compilation). Le flot imaginé permet, à partir de la spécification de haut niveau écrite en langage C, d'estimer, d'analyser, d'optimiser les performances et finalement d'implanter réellement une architecture après l'exploration rapide des différents choix architecturaux au sein d'un espace de conception maîtrisé. Les architectures visées sont du type multiprocesseur à base de processeurs Microblaze de Xilinx étendus selon les besoins avec des coprocesseurs et/ou des accélérateurs matériels sous la forme d'IP.

La conception de systèmes de télécommunication dédiés à la Radio Logicielle reste encore aujourd'hui un verrou important malgré les nombreuses études menées sur le sujet ces dernières années [A3S 2005][Delahaye 2007][SDR 2007]. L'objectif de la Radio Logicielle est d'aboutir à des récepteurs radio dont l'exploitation est pilotée par logiciel. La problématique visée dans le cadre de ce projet adresse deux verrous fondamentaux : l'abstraction et la communication. La mise en œuvre d'une couche d'abstraction est essentielle afin de permettre le déploiement rapide de nouvelles applications. Aujourd'hui les concepteurs sont confrontés à une complexité prohibitive lors de l'intégration de nouveaux composants au sein d'une plateforme existante. Ce type de verrou va à l'encontre du concept même de Radio Logicielle où l'adaptabilité est au centre du système. Le caractère dynamique du système passe donc forcément par la mise en œuvre d'une couche d'abstraction. Par ailleurs, l'intégration de tels systèmes induit une problématique critique lors du dimensionnement de l'architecture de communication. La réponse proposée dans ce projet consiste donc à positionner les communications au centre du développement où l'abstraction des communications repose sur une structure de communication flexible et performante pouvant s'adapter dynamiquement en fonction des besoins (forme d'onde considérée, données manipulées, traitements mis en œuvre). La solution imaginée vise donc à étudier les possibilités d'un NoC afin d'adapter dynamiquement les performances et les chemins de communications de la plateforme d'exécution et d'analyser les coûts induits par

l'utilisation des couches d'abstractions. Ces deux points sont essentiels afin d'aboutir à la définition d'une solution réaliste et effective.

La reconfiguration dynamique est au coeur de nombreux projets actuels [Delahaye 2007] et son rôle va clairement se renforcer dans l'avenir. Nous développons depuis plusieurs années des démonstrateurs afin de prototyper des applications utilisant la reconfiguration dynamique partielle. Le stockage des fichiers de configuration (i.e. bitstream) est un problème complexe car il contraint fortement le nombre de contextes possibles. Certaines solutions ont été proposées afin de limiter ce coût en utilisant des approches de compression [Huebner 2004]. Toutefois, ce type d'approche reste limité et ne permet pas le stockage d'un nombre élevé de contextes. De plus, il reste une contrainte majeure qui résulte du caractère statique de ce mécanisme, l'ensemble des contextes doit être préalablement intégré à la plateforme avant son déploiement dans son environnement d'exécution ce qui réduit le caractère adaptatif souhaité. Nous proposons donc d'anticiper le déploiement des futurs systèmes adaptatifs et de leurs connexions aux réseaux de communication en déportant le stockage des configurations sur des serveurs externes. Ce type d'approche est intéressant puisqu'il lève les deux limites présentées ci-dessus, à savoir le stockage des bitstreams et le caractère statique des contextes.

Pour cela nous développons un démonstrateur qui vise à mettre en œuvre l'auto reconfiguration dynamique partielle avec chargement de bitstream en utilisant un protocole Ethernet simplifié. Ce type d'approche ouvre de nombreuses perspectives intéressantes au niveau applicatif puisqu'il rend possible l'adaptation dynamique du système en fonction des besoins et des contraintes à un instant donné.

De façon connexe à cette étude nous nous intéressons dans le projet AETHER [AETHER 2006] au problème de la décision de la reconfiguration et au déploiement des applications sur les futures plateformes hétérogènes. Les systèmes considérés dans le projet AETHER s'appuient sur des entités auto-adaptatives en réseau (*Self-Adaptive Networked Entities*, SANE) basées sur des architectures reconfigurables. Nous nous intéressons plus particulièrement au problème critique du déploiement dynamique d'applications. Pour cela nous développons un système de négociation basé sur la coopération inter SANE permettant ainsi la négociation et la distribution d'applications entre différents SANE. Nous souhaitons développer une solution par modèles permettant d'estimer et de prédire le comportement du système de négociation afin d'anticiper les décisions de reconfiguration. Pour cela un démonstrateur SystemC va tout d'abord être développé afin de valider les concepts proposés avant d'imaginer la mise en œuvre d'un démonstrateur physique.

Comme expliqué précédemment la concrétisation de ces futurs systèmes adaptifs et nomades ne sera possible que si la problématique de la sécurité est correctement adressée. Ce point est fondamental et de nombreuses études doivent encore être menées. Un point qui nous semble particulièrement important est relatif au développement conjoint de solutions de sécurité logicielles et matérielles. Nous proposons d'associer des couches de protections matérielles aux solutions logicielles actuelles. L'objectif étant de garantir la sécurité du système tout en minimisant le surcoût engendré par les mécanismes de sécurité. Une étude intéressante que nous menons actuellement consiste à étendre les services de l'OS afin de pouvoir gérer des niveaux de sécurité flexibles en fonction des besoins. Une application n'est pas composée que de tâches critiques aussi il est important de fournir les mécanismes permettant de garantir soit l'intégrité et la confidentialité des données et des codes, soit des niveaux intermédiaires entre ce niveau de sécurité et l'absence de sécurité. L'utilisation de techniques de filtrage des adresses mémoires en relation avec les services de l'OS semble intéressante afin de proposer une solution flexible et à faible coût.

La sécurité des NoC pour les systèmes embarqués reconfigurables est également un sujet important dans la mesure où le rôle des réseaux sur silicium ne cesse de s'accroître et qu'aujourd'hui des solutions industrielles commencent à être intégrées dans des systèmes numériques tels que les set top box qui intègrent un grand nombre de composants et manipulent de nombreux flux de données. Ces solutions se déployant de plus en plus il est

nécessaire dès aujourd'hui d'anticiper les futures attaques auxquelles les systèmes numériques seront confrontés. Un des éléments critiques du système est l'architecture de communication qui supporte l'ensemble des transferts de données, aussi une attaque sur le réseau de communication peut être extrêmement dangereuse pour l'ensemble du système. Par exemple un attaquant pourrait récupérer des informations du type numéro de client, numéro de carte bleu, ou bien détourner un service. Le modèle de menaces bien que pouvant évoluer à l'avenir est dès à présent bien identifié. Les menaces à considérer sont : 1) le détournement qui consiste d'un point de vue du réseau de communication à écrire dans une zone non autorisée afin de modifier le comportement du système (ce type de menace inclut les attaques classiques du type buffer overflow), 2) l'extraction d'informations secrètes comme des clefs privées de chiffrement ou des informations personnelles et 3) le dénie de service qui vise à compromettre le fonctionnement du système afin d'empêcher l'accès à un service.

Face à ces menaces il est essentiel de proposer des contres mesures adaptées au domaine des systèmes embarqués. Une solution consiste à analyser le comportement du système et à identifier des comportements anormaux. Toutefois, les approches existantes à base de traces ne sont pas réalistes au sein d'un système embarqué où les ressources mémoires sont limitées. Il est donc important d'imaginer des solutions moins coûteuses en ressources. Nous avons déjà formalisé plusieurs solutions afin de protéger les architectures de communication du type NoC [Diguet 2007]. Toutefois nous souhaitons aller plus loin afin de quantifier précisément les surcoûts en surface et en latence des solutions proposées par rapport à une solution non protégée. Nous souhaitons également étendre notre approche en ajoutant des mécanismes de confidentialité et d'intégrité afin de garantir la sécurité des transferts de données. Ces besoins sont essentiels afin de permettre des communications entre des zones sécurisées et non sécurisées.

Ces quelques directions de recherches mettent en évidence un point particulièrement important auquel nous assistons, la converge des compétences. En effet, les futures applications adaptatives nécessitent à la fois la définition de nouveaux flots de conception, la mise en œuvre de mécanismes de reconfiguration dynamique et la prise en compte de mécanismes de sécurité. La complexité des futurs systèmes embarqués implique donc des niveaux d'expertise extrêmement élevés et variés.

J'aimerai finir ces perspectives sur le besoin et la richesse des échanges entre chercheurs de domaines scientifiques communs mais aussi et de façon croissante entre chercheurs de disciplines complémentaires. La mise en place de collaborations étroites entre chercheurs me semble indispensable afin de construire cet avenir hautement technologique, interactif et adaptatif.

7. Références

[3DES 1995] **3DES RFC 1851** [en ligne] ftp://ftp.rfc-editor.org/innotes/rfc1851.txt, Septembre 1995

[A3S 2005] **Projet A3S RNTL**, Adéquation Application Architecture Système, [en ligne] http://web.univ-ubs.fr/lester/www-lester/Projets/Codesign/A3S/English/A3Shome.htm

[AES 2003] **AES RFC 3565** [en ligne] ftp://ftp.rfc-editor.org/innotes/rfc3565.txt, July 2003

[AETHER 2006] **AETHER IST-FET project**, [en ligne] http://www.aether-ist.org/

[Ahmed00] E. Ahmed, J. Rose, **The Effect of LUT and Cluster Size on Depp-Submicron FPGA Performance and Density**, In International ACM Symposium on Field Programmable Gate Arrays, FPGA 00,. February 2000

[Anderson 1996] R. Anderson, M. Kuhn, **Tamper Resistance – a Cautionary Note**, Proceedings of the Second USENIX Workshop on Electronic Commerce Proceedings, 1996

[Anderson 1997] R. Anderson, M. Kuhn, **Low Cost Attacks on Tamper Resistant Devices**, M Lomas et al. (ed.), Security Protocols, 5th International Workshop, Paris, France, Proceedings, Springer LNCS 1361, 1997

[Anderson 2001] R. Anderson, **Security Engineering, A Guide to Building Dependable Distributed Systems**, Wiley Computer Publishing, ISBN 0-471-3892-6, 2001

[Arora 2005] D. Arora, S. Ravi, A. Raghunathan, N. K. Jha, **Secure Embedded Processing through Hardware-assisted Run-time Monitoring**, Proc. Design, Automation & Test in Europe, Mar. 2005

[ARTEMIS 2006] **Strategic Research Agenda, Design Methods and Tools Report 2006**, [en ligne] http://www.artemis-office.org/

[Atat 2007] Y. Atat, N-E. Zergainoh, **Simulink-based MPSoC Design: New Approach to Bridge the Gap between Algorithm and Architecture Design**, IEEE Computer Society Annual Symposium on VLSI, 2007

[ATSC 1995] **Advanced Television System Committee**, Digital audio compression standard (AC3), [en ligne] http://www.atsc.org/standards.html

[Auguin 2001] M. Auguin, L. Capella, F. Cuesta, E. Gresset, **CODEF: a system level design space exploration tool**, In Proceedings of the Acoustics, Speech, and Signal Processing, 2001

[Auguin 2003] M. Auguin, K. Ben Chehida, J.P. Diguet, X. Fornari, A.M. Fouilliart, C. Gamrat, G. Gogniat, P. Kajfasz, Y Le Moullec, **Partitionning and CoDesign Tools & Methodology for Reconfigurable Computing : the EPICURE Philosophy**, In Proceeding of the Third International Workshop on Systems, Architectures, Modeling Simulation, SAMOS 03, Samos, Greece, July 2003.

[Bautista 2007] J. Bautista, **Tera-scale Computing - the Role of interconnects in Volume Compute platforms**, International Interconnect Technology Conference, IEEE 2007

[Becker 2003] J. Becker, M. Hübner, and M. Ullmann, **Power Estimation and Power Measurement of Xilinx Virtex FPGAs: Trade-offs and Limitations**, Proceedings of 16th

Symposium on Integrated Circuits and System Design (SBCCI 2003), September 08-11, 2003, Sao Paulo, Brazil

[Beiu 2007] V. Beiu, **Grand Challenges of Nanoelectronics and Possible Architectural Solutions What Do Shannon, von Neumann, Kolmogorov, and Feynman Have to Do with Moore**, Proceedings of the 37th International Symposium on Multiple-Valued Logic (ISMVL'07), 2007

[Betz 1999] V. Betz, J. Rose, A. Marquart, **Architecture and CAD for Deep Submicron FPGAs**, Kluwer Academic Publishers, 1999

[Bilavarn 2002] S. Bilavarn, **Exploration Architecturale au Niveau Comportemental – Application aux FPGAs**, Thèse de Doctorat soutenue le 28 Février 2002, Université de Bretagne Sud

[Bilavarn 2006] S. Bilavarn, G. Gogniat, J-L. Philippe, L. Bossuet, **Low Complexity Design Space Exploration from Early Specifications**, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 25, No. 10, October 2006, pages 1950-1968

[Blodget 2003] 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

[Bondalapati 1999] K. Bondalapati et al., **Defacto: A design environment for adaptive computing technology**, In Proceedings of the 11 IPPS/SPDP'99 Workshops, 1999

[Bossuet 2002] L. Bossuet, G. Gogniat, J-P. Diguet, J-L. Philippe, **A Modeling Method for Reconfigurable Architectures**, International Workshop on System-on-Chip for Real-Time Applications, July 6-7, 2002, Banff, Canada

[Bossuet 2004] L. Bossuet, **Exploration de l'espace de conception des architectures reconfigurables**, Thèse de doctorat, Université de Bretagne Sud, Lorient, septembre 2004, [en ligne] http://www.lilianbossuet.com/fr/Doc/publications/These_Lilian_Bossuet.pdf

[Bossuet 2006a] L. Bossuet, G. Gogniat, W. Burleson, **Dynamically Configurable Security for SRAM FPGA Bitstreams**, International Journal of Embedded Systems, IJES, From Inderscience Publishers Vol. 2, Nos. 1/2, 2006

[Bossuet 2006b] L. Bossuet, **Architecture Conception et Utilisation des FPGA**, Cours de l'ENSEIRB 2006, [en ligne]
http://www.lilianbossuet.com/fr/Doc/documents_pedagogiques/Bossuet_cours_FPGA_ENSEIRB.pdf

[Bossuet 2007] L. Bossuet, G. Gogniat, J-L. Philippe, **Communication-Oriented Design Space Exploration for Reconfigurable Architectures**, EURASIP Journal on Embedded Systems, Volume 2007 (2007), Article ID 23496, 20 pages, doi:10.1155/2007/23496

[Burton 2006] M. Burton, A. Morawiec, **Platform Based Design at the Electronic System Level: Industry Perspectives and Experiences**, Springer, 2006

[Cambonie 2003] J. Cambonie, **Reconfigurable Architecture for WLAN Application**, STmicroelectronics, 2003

[Carmichael 2001] C. Carmichael, **Triple Module Redundancy Design Techniques for Virtex FPGAs**, Xilinx Application Note 197 (XAPP197) November 1, 2001

[Catapult 2004] Mentor Graphics Corporation, **Catapult C Synthesis User's and Reference Manual**, release 2004b edition, June 2004

[Chow 1999a] P. Chow, S. Ong Seo, J. Rose, G. Paez-Monzon, I. Rahardja, **The Design of an SRAM-Based Field-Programmable Gate Array - Part I: Circuit Design and Layout**, IEEE Transactions on VLSI Systems, June 1999 , Vol. 7, No.2, 1999

[Chow 1999b] P. Chow, S. Ong Seo, J. Rose, G. Paez-Monzon, I. Rahardja, **The Design of an SRAM-Based Field-Programmable Gate Array - Part II: Architecture**, IEEE Transactions on VLSI Systems, Septembre 1999 , Vol. 7, No.3, 1999

[Cidon 2007] I. Cidon, **NoC: Network or Chip?**, First International Symposium on Networks-on-Chip, 2007

[Ciordas 2007] C. Ciordas, A. Hansson, K. Goossens, T. Basten, **A Monitoring-Aware Network-on-Chip Design Flow**, EUROMICRO Conference on Digital System Design: Architectures, Methods and Tools, 2006

[Clifford 2000] P. Clifford, S. J.E.Wilton, **Architecture of Cluster-Based FPGAs with Memory**, IEEE Custom Integrated Circuits Conference, CICC 00, May 2000

[Compton 1999] Katherine Compton, **Programming Architectures For Run-Time Reconfigurable Systems**, Master's Thesis, Dept of ECE, Northwestern University, Evaston, IL USA. December 1999

[Compton 2000] K. Compton, S. Hauck, **Reconfigurable Computing: A Survey of Systems and Software**, ACM Computing Surveys, 2000

[Daemen 2002] J. Daemen and V. Rijmen, **The Design of Rijndael AES-The Advanced Encryption Standard**, Springer-Verlag 2002

[Dagon 2004] D. Dagon, T. Martin, and T. Staner, **Mobile Phones as Computing Devices: The Viruses are Coming!**, IEEE Pervasive Computing, October-December 2004

[Delahaye 2004] J-P. Delahaye, G. Gogniat, C. Roland, P. Bomel, **Software Radio and Dynamic Reconfiguration on a DSP/FPGA platform**, The 3rd Workshop on Software Radios, March 17-18, 2004, Karlsruhe, Germany

[Delahaye 2007] J-P. Delahaye, J. Palicot, C. Moy, P. Leray, **Partial Reconfiguration of FPGAs for Dynamical Reconfiguration of a Software Radio Platform**, 16th IST Mobile&Wireless Communications Summit 2007, 2007

[Diguet 2000] J-P. Diguet, G. Gogniat, P. Danielo , M. Auguin, J-L. Philippe, **The SPF model**, Forum on Design Langage, FDL 00, Tübingen, Germany, 2000.

[Diguet 2007] J-P. Diguet, G. Gogniat, S. Evain, R. Vaslin, E. Juin, **NOC-centric security of reconfigurable SoC**, The 1st ACM/IEEE International Symposium on Networks-on-Chip, May 7-9, 2007, Princeton University, New Jersey, USA

[Donghyun 2007] K. Donghyun, K. Kwanho, K. Joo-Young, L. Seung-Jin, Y. Hoi-Jim, **Solutions for Real Chip Implementation Issues of NoC and Their Application to Memory-Centric NoC**, First International Symposium on Networks-on-Chip, 2007

[Elbaz 2006] R. Elbaz, L. Torres, G. Sassatelli, P. Guillemin, M. Bardouillet, A. Martinez, **A parallelized way to provide data encryption and integrity checking on a processor-memory bus**, In DAC '06: Proceedings of the 43rd annual conference on Design automation, 2006

[ePaynews 2005] **ePaynews - eCommerce Statistics**, http://www.epaynews.com/statistics/index.html

[ESL 2005] **ESL now! Electronic System-Level Design**, [en ligne] http://www.esl-now.com/

[Evain 2006a] S. Evain, J-Ph. Diguet, D. Houzet, **NoC design flow for TDMA and QoS Management in a GALS context**, EURASIP Journal on Embedded Systems, Article ID 63656, 2006

[Evain 2006b] E. Samuel, **µSpider : Environnement de Conception de Réseau sur Puce**, Thèse de doctorat, Université de Bretagne Sud, Lorient, novembre 2006

[Fang 2007] J. Fang, **Parallel Programming Environment: A Key to Translating Tera-Scale Platforms into a Big Success**, International Symposium on Code Generation and Optimization, 2007

[Flake 2006] P. Flake, S. Davidmann, F. Schirrmeister, **System-level exploration tools for MPSoC designs**, Design Automation Conference, ACM/IEEE 2006

[Ghenassia 2005] F. Ghenassia, **Transaction Level Modeling With SystemC: TLM Concepts and Applications for Embedded Systems**, Springer, 2005

[Gogniat 1997] G. Gogniat, **Architecture générique et synthèse des communications pour la conception conjointe de systèmes embarqués logiciel/matériel**, Thèse de Doctorat Université de Nice – Sophia Antipolis, 1997

[Gogniat 2000] G. Gogniat, M. Auguin, L. Bianco, A. Pegatoquet, **A Codesign Back End Approach for Embedded System Design**, ACM Transactions on Design Automation of Electronic Systems, Vol. 5N. 3, july 2000

[Gogniat 2005a] G. Gogniat, L. Bossuet, W. Burleson, **Configurable computing for high-security/high-performance ambient systems**, 5th International Workshop Embedded Computer Systems: Architectures, MOdeling, and Simulation SAMOS 2005, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, Volume 3553/2005, July 18-20, 2005, Samos, Greece

[Gogniat 2005b] G. Gogniat, T. Wolf, W. Burleson, Reconfigurable Security Primitive for Embedded Systems, The IEEE International Symposium on System-on-Chip (SOC 2005), November 15-17, 2005, Tampere, Finland

[Gogniat 2006] G. Gogniat, T. Wolf, W. Burleson, **Reconfigurable security support for embedded systems**, The 39th IEEE Hawaii International Conference on System Science (HICSS-39), January 2006, Poipu, HI, USA

[Guilley 2004] S. Guilley and R. Pacalet, **SoC securiy: a war against side-channels** Annals of the Telecommunications, Système sur puce électronique pour les télécommunications Vol. 59, N° 7-8, Juillet-août 2004

[Gupta 1993] R. K. Gupta, G. D. Micheli, **Hardware-Software Cosynthesis for Digital Systems**, IEEE Design & Test of Computers (Sept. 1993), pp. 29 – 41

[Gupta 2002] S. Gupta et al., **Coordinated transformations for high-level synthesis of high performance microprocessor blocks**, Proceedings of the 39th conference on Design automation, 2002

[Ha 2007] S. Ha, **Model-based Programming Environment of Embedded Software for MPSoC**, Asia and South Pacific Design Automation Conference, 2007

[Hansson 2007] A. Hansson, K. Goossens, **Trade-offs in the Configuration of a Network on Chip for Multiple Use-Cases**, First International Symposium on Networks-on-Chip, 2007. NOCS 2007

[Hartenstein 2001] R. Hartenstein, **A Decade of Reconfigurable Computing: a Visionary Retrospective**, In Design Automation and Test in Europe, DATE, 2001

[Hu 2006] Y-l. Hu, Q. Ding, **Design of an architecture for multiprocessor system-on-chip (MPSoC)**, Conference on High Density Microsystem Design and Packaging and Component Failure Analysis, 2006

[Huebner 2004] M. Huebner, M. Ullmann, F. Weissel, J. Becker, **Real-time Configuration Code Decompression for Dynamic FPGA Self-Reconfiguration**, 18th International Parallel and Distributed Processing Symposium (IPDPS'04), 2004

[IEEE 2007] **Institute of Electrical and Electronics Engineers (IEEE)**, [en ligne] http://www.ieee.org/

[ITRS 2007] **International Technology Roadmap for Semiconductors**, Process Integration, Devices, and Structures, [en ligne] http://www.itrs.net/home.html

[Jerraya 2004] A. Jerraya, W. Wolf, **Multiprocessor Systems-on-Chips**, Morgan Kaufman, 2004

[Kim 2006] Y. Kyun Kim, R. Prasad, **4G Roadmap and Emerging Communication Technologies**, Technology & Industrial Arts, Artech House, 2006

[Kocher 2004] P. Kocher, R. Lee, G. McGraw, A. Raghunathan, S. Ravi, **Security as a New Dimension in Embedded System Design**, ACM/IEEE Design Automation Conference, 2004

[LeMoullec 2003a] Y. Le Moullec, **Aide à la conception des systèmes sur puce hétérogène par l'exploration paramétrable des solutions au niveau système**, PhD Thesis, Université de Bretagne Sud, Lorient, April 2003.

[LeMoullec 2003b] Y. Le Moullec, N. Ben Amor, J.P. Diguet, J.L. Philippe and M. Abid, **Multigranularity Metrics for the Era of Strongly Personalized SoCs**, In Design Automation and Test in Europe, 2003

[Lie 2000] D. Lie, C. Thekkath, M. Mitchell, P. Lincoln, D. Boneh, J. Mitchell, M. Horowitz, **Architectural support for copy and tamper resistant software**, In ASPLOS-IX: Proceedings of the ninth international conference on Architectural support for programming languages and operating systems, pages 168–177, 2000.

[Lie 2003] D. Lie, C. A. Thekkath, M. Horowitz, **Implementing an untrusted operating system on trusted hardware**, In SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles, pages 178–192, October 2003.

[Maalej 2004] I. Maalej, G. Gogniat, M. Abid, J-L. Philippe, **High level analysis of multiprocessor system on chip**, Embedded Real-Time Systems Implementation Workshop (ERTSI 2004), December 5-8, 2004, Lisbon, Portugal

[Maalej 2006] I. Maalej, G. Gogniat, J-L. Philippe, M. Abid, **Genetic algorithm for high level analysis and architecture exploration**, IP Based Design 2006 Workshop, December 2006, Grenoble, France

[MARTE 2007] **UML Profile for Modeling and Analysis of Real-Time and Embedded systems (MARTE)**, RFP OMG, [en ligne] http://www.omg.org/

[Martin 2004] T. Martin, M. Hsiao, D. Ha, and J. Krishnaswami, **Denial-of-Service Attacks on Battery-powered Mobile Computers**, Proceedings of the 2nd IEEE Pervasive Computing Conference, Orlando, Florida, March 2004, pp. 309-318

[Martin 2005] G. Martin, W. Mueller, **UML for SoC Design**, Springer Verlag, Berlin, 2005.

[Martin 2006] G. Martin, **Overview of the MPSoC design challenge**, Design Automation Conference, ACM/IEEE 2006

[Martin 2007] G. Martin, B. Bailey, A. Piziali, **ESL Design and Verification: A Prescription for Electronic System-Level Methodology**, Morgan Kaufmann Publishers, 2007

[MASTER 2007] **Multi-processor Analysis & Synthesis Tool for Embedded Reconfigurable Systems**, Soumission à l'appel à projet Architectures du futur, ANR 2007

[MDA 2003] **Model Driven Architecture**, [en ligne] http://www.omg.org/mda/

[Moore 1965] G. E. Moore, **Cramming more components onto integrated circuits**, Electronics, Volume 38, Number 8, April 19, 1965

[MOPCOM 2007] **Modeling and specialization of platform and components MDA**, Projet ANR RNTL, [en ligne] http://www.mopcom.fr/

[MPEG2 2000] **MPEG-2 Generic coding of moving pictures and associated audio information**, [en ligne] http://www.chiariglione.org/mpeg/

[Nash 2005] D. Nash, T. Martin, D. Ha, and M. Hsiao, **Towards an Intrusion Detection System for Battery Exhaustion Attacks on Mobile Computing Devices**, Proceedings of the 2nd International Workshop on Pervasive Computing and Communications Security, March 2005

[Objecteering 2007] **Objecteering Software**, Softeam, [en ligne] http://www.softeam.fr/produits.php

[OMG 2007] **Object Management Group (OMG)**, [en ligne] http://www.omg.org/

[OSCI 2007] **SystemC Users Group Survey Data Trends Report**, Open SystemC™ Initiative (OSCI), April 2007 [en ligne] http://www.systemc.org/

[Pico 2007] picoArray, **PicoChip**, [en ligne] http://www.picochip.com/

[Radumovic 1998] B. Radunovic, V. Milutinovic, **A Survey of Reconfigurable Computing Architectures**, In Field-Programmable Logic and Applications, R.W. Hartenstein and A. Keevallik (editorts), LCNS 1482, Springer, 1998

[Rosing 2007] T. S. Rosing, K. Mihic, G. De Micheli, **Power and Reliability Management of SoCs**, Very Large Scale Integration (VLSI) Systems, IEEE Transactions on Volume 15, Issue 4, April 2007

[Rouffet 2005] D. Rouffet, S. Kerboeuf, L. Cai, V. Capdevielle, **Universal Access: Connect me – 4G Mobile**, Alcatel-Lucent Telecom Review, 2005, [en ligne] http://www.alcatel-lucent.com/

[Rouxel 2002] S. Rouxel, **Caractérisation de l'impact du routage sur les performances (vitesse et consommation de puissance) d'un FPGA**, Master Thesis, université de Bretagne Sud, Lorient, Septembre 2002.

[Rouxel 2006a] S. Rouxel, G. Gogniat, J-P. Diguet, J-L. Philippe, C. Moy, **A3S Method and Tools for Analysis of Real-Time Embedded Systems**, International Workshop on Modeling and Analysis of Real-Time and Embedded Systems (MARTES'06), October 2006, Genova, Italy

[Rouxel 2006b] S. Rouxel, G. Gogniat, J-P. Diguet, J-L. Philippe, C. Moy, **System Level Design with UML: a Unified Approach**, IEEE Symposium on Industrial Embedded System (IES'06), October 2006, Antibes Juan-Les-Pins, France

[Rouxel 2006c] S. Rouxel, G. Gogniat, J-P. Diguet, J-L. Philippe and C. Moy, **Chapter 7. Schedulability Analysis and MDD**, From MDD Concepts to Experiments and Illustrations

Edited by: J-P. Babau, J. Champeau, S. Gérard International Scientific and Technical Encyclopedia, September 2006, pages 111 – 130

[Santarini 2000] M. Santarini, **Cadence adds system-level design tool to EDA flow**, EE Times, January 10, 2000 [en ligne] http://www.eetimes.com/story/OEG20000110S0006

[Sassatelli 2002] G. Sassatelli, L. Torres, P. Benoit, T. Gil, C. Diou, G. Cambon, J. Galy, **Highly Scalable Dynamically Reconfigurable Systolic Ring-Architecture for DSP Applications**, In IEEE Design Automation and Test in Europe, 2002

[Sassatelli 2007] G. Sassatelli, N. Saint-Jean, C. Woszezenki, I. Grehs, F. Moraes, **Architectural Issues in Homogeneous NoC-Based MPSoC**, International Workshop on Rapid System Prototyping, 2007

[Schaumont 2001] P. Schaumont, I. Verbauwhede, K. Keutzer, M. Sarrafzadeh, **A Quick Safari Through the Reconfiguration Jungle**, In Design and Automation Conference, DAC, 2001

[Schaumont 2003] P. Schaumont and I. Verbauwhede, **Domain-Specific Codesign for Embedded Security**, IEEE Computer, April 2003

[Schaumont 2006] P. Schaumont, D. Hwang, S. Yang, I. Verbauwhede, **Multi-level Design Validation in a Secure Embedded System**, IEEE Transactions on Computers, special issue HLDVT 2005, October 2006

[SDR 2007] **Software Defined Radio Forum**, SDR Forum [en ligne] http://www.sdrforum.org/

[Sentieys 1993] O. Sentieys, J-L. Philippe, E. Martin, **Gaut: an architecture synthesis tool for dedicated signal processors**, In Proceedings IEEE International European Design Automation Conference (Euro DAC), 1993

[Shekhar 2007] B. Shekhar, N.P. Jouppi, P. Stenstrom, **Microprocessors in the Era of Terascale Integration**, Design, Automation & Test in Europe Conference & Exhibition, 2007. DATE '07, 2007

[Stretch 2007] **Software Configurable Processors**, [en ligne] http://www.stretchinc.com/

[Suh 2003a] G. E. Suh, D. Clarke, B. Gassend, M. Van Dijk, S. Devadas, **AEGIS: architecture for tamper-evident and tamper-resistant processing**, In ICS '03: Proceedings of the 17th annual international conference on Supercomputing, pages 160–171, 2003.

[Suh 2003b] G. E. Suh, D. Clarke, B. Gassend, M. Van Dijk, S. Devadas, **Efficient memory integrity verification and encryption for secure processors**, In MICRO 36: Proceedings of the 36th annual IEEE/ACM International Symposium on Microarchitecture, page 339, 2003.

[Suh 2005] G. E. Suh, C. W. O'Donnell, I. Sachdev, S. Devadas, **Design and implementation of the AEGIS single-chip secure processor using physical random functions**, In ISCA '05: Proceedings of the 32nd Annual International Symposium on Computer Architecture, pages 25–36, 2005.

[Synfora 2007] Synfora Inc., **PICO Express**, [en ligne] http://www.synfora.com/

[SystemC 2005] **IEEE 1666™ Standard System C Language**, [en ligne] http://standards.ieee.org/

[Tensilica 2007] Tensilica Inc., **XPRES Compiler**, [en ligne] http://www.tensilica.com/

[Tera 2007] **Tera–scale Computing**, Intel [en ligne] http://www.intel.com/

[Tmar 2006] H. Tmar, J-Ph. Diguet, A. Azzedine, M. Abid, J-L. Philippe, **RTDT : a Static QoS Manager, RT Scheduling, HW/SW Partitioning CAD Tool**, MicroElectronics Journal , vol. 37 , 2006

[Torii 2005] S. Torii, J. Sakai, H. Inoue, T. Tokue, Y. Ito, **Asymmetric Multi-Processing Mobile Application Processor MP211**, NEC Journal of Advanced Technology, 2005

[Toshiba 2007] **Smartphones G500 et G900**, Toshiba [en ligne] http://www.toshiba-europe.com/mobile/

[Tredennick 2003] N. Tredennick, B. Shimamoto, **The Rise of Reconfigurable Systems**, In proceeding of Engineering of Reconfigurable Systems and Application, 2003

[Ulmann 2004] M. Ulmann, M. Hübner, B. Grimm, J. Becker, **An FPGA Run-Time System for Dynamical On-Demand Reconfiguration**, 11th IEEE Reconfigurable Architectures Workshop, RAW 2004, Santa Fé, New Mexico, USA, 26-17 avril 2004.

[UML 1997] **Unified Modeling Language**, [en ligne] http://www.uml.org/

[UML4SoC 2004] **UML for SOC Design**, [en ligne] http://jerry.c-lab.de/uml-soc/

[UMTS 1999] **Universal Mobile Telecommunication System**, [en ligne] http://www.umtsworld.com/technology/overview.htm

[Vaslin 2006] R. Vaslin, G. Gogniat J-P. Diguet, **Secure architecture in embedded systems: an overview**, Reconfigurable Communication-centric SoCs (ReCoSoc'06), July 3-5, 2006, Montpellier, France

[Vaslin 2007] R. Vaslin, G. Gogniat, J-P. Diguet, R. Tessier, W. Burleson, **High Efficiency Protection Solution for Off-Chip Memory in Embedded Systems**, The International Conference on Engineering of Reconfigurable Systems and Algorithms, June 25-28, 2007, Las Vegas, Nevada, USA

[Verbauwhede 2007] I. Verbauwhede, P. Schaumont, **Design Methods for Security and Trust**, Design Automation and Test Conference in Europe (DATE 2007), Nice, France, April 2007

[Walls 2005] Colin Walls, **Embedded Software: The Works**, Elsevier 2005

[Wanderley 2007] E. Wanderley, G. Gogniat, J-P. Diguet, **A Code Compression Method With Confidentiality and Integrity Checking**, The 2007 International Conference on Embedded Systems and Applications, June 25-28, 2007, Las Vegas, Nevada, USA

[Wang 2007] H. Wang, J-P. Delahaye, P. Leray, J. Palicot, **Managing dynamical reconfiguration on a MIMO Decoder**, 14th Reconfigurable Architecture Workshop, 26-27 March 2007

[Wilton 1999] S. J.E. Wilton, J. Rose, Z. G. Vranesic, **The Memory/Logic Interface in FPGA's With Large Embedded Memory Arrays**, IEEE Transactions on VLSI Systems, Vol. 7, No.1, March 1999.

[Wolf 2006] T. Wolf, S. Mao, D. Kumar, B. Datta, W. Burleson, G. Gogniat, **Collaborative monitors for embedded system security**, in Proc. of First International Workshop on Embedded Systems Security in conjunction with 6th Annual ACM International Conference on Embedded Software (EMSOFT), Seoul, Korea, Oct. 2006

[Wu 2004] K. Wu, R. Karri, G. Kuznetsov, M. Goessel, **Parity Based Concurrent Error Detection for the Advanced Encryption Standard**, International Test Conference 2004 (ITC), 2004, Charlotte, USA

[Xilinx 2004] **Two Flows for Partial Reconfiguration: Module Based or Difference Based**, Xilinx Application Note XAPP290, Xilinx, September 2004

[Xilinx 2005] **Xilinx ICAP component for auto reconfiguration**, [en ligne] http://www.xilinx.com/

[Xilinx 2007] **Virtex-4 Multi-Platform FPGA**, [en ligne] http://www.xilinx.com/

[Yoon 2007] S-R. Yoon, S-C. Park, **Case Study on Design Space Exploration of MPSoC architecture**, The 9th International Conference on Advanced Communication Technology, 2007