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

Parties 1 et 2

Par

Guy Gogniat

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

Avant Propos

Ce manuscrit présente une synthèse de mes travaux de recherche, d'enseignement et administratifs réalisés depuis septembre 1998, date de ma nomination en tant que Maître de Conférences à l'Université de Bretagne Sud. Mes travaux de recherche ont été effectués au sein de l'équipe SysRec (Systèmes Reconfigurables) du laboratoire LESTER[1] (FRE 2734 CNRS/Université de Bretagne Sud).

Ce manuscrit présente également, mais de façon plus succincte, mes travaux de recherche entre octobre 1994 et août 1998, période durant laquelle j'ai effectué mes recherches en tant que doctorant puis ATER au sein du laboratoire I3S dans le thème ALM (Architectures Logicielles et Matérielles) de l'Université de Nice – Sophia Antipolis.

Depuis maintenant plus de dix ans mes travaux de recherche s'intéressent au domaine de la conception des systèmes embarqués avec une évolution forte vers les architectures reconfigurables qui n'ont cessé de prendre de l'ampleur ces dernières années et s'installent aujourd'hui comme une solution incontournable du domaine de la conception des systèmes embarqués.

Ce document est organisé en trois parties afin de couvrir les différentes facettes de mes activités :

Partie 1 : Synthèse des travaux

Cette première partie présente de façon complète l'ensemble de mon parcours en mettant en avant toutes ses contributions et originalités. Elle permet d'appréhender mes réalisations et de comprendre mes motivations. Une réflexion sur les évolutions de notre métier et sur les enjeux à venir est également proposée.

Partie 2 : Annexes, Sélection des publications significatives

Cette deuxième partie, illustre les contributions menées en présentant plusieurs articles scientifiques.

Partie 3 : Travaux de recherche détaillés et perspectives (document séparé)

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 et 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] Laboratoire d'Electronique des Systèmes TEmps Réel

Table des matières

Partie 1 : Synthèse des travaux

Cette première partie présente de façon complète l'ensemble de mon parcours en mettant en avant toutes ses contributions et originalités. Elle permet d'appréhender mes réalisations et de comprendre mes motivations. Une réflexion sur les évolutions de notre métier et sur les enjeux à venir est également proposée.

Résumé du dossier

<table>
<tr><td>Curriculum Vitae</td><td>
Fonction actuelle
• Maître de Conférences à l'Université de Bretagne Sud
Enseignement à l'IUT de Lorient, Département Génie Industriel et Maintenance
Recherche au Laboratoire d'Electronique des Systèmes TEmps Réel (LESTER)
Diplômes
• Thèse de doctorat (sciences pour l'ingénieur) Université de Nice – Sophia Antipolis
Mention très honorable avec les félicitations du jury, obtenue le 27 novembre 1997
• DEA de Traitement du Signal et Architectures Électroniques Université de Paris Sud Orsay
Mention Bien, septembre 1994
• Diplôme d'Ingénieurs FIUPSO Formation d'Ingénieur de l'Université Paris Sud Orsay
Spécialité systèmes électroniques, septembre 1994
Expériences professionnelles
• Septembre 2005/Aujourd'hui Maître de Conférences à l'Université de Bretagne Sud
• Novembre 2004/Août 2005 Chercheur invité (CRCT – Contrat ERE DGA)
Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA, USA
• Septembre 1998/Octobre 2004 Maître de Conférences à l'Université de Bretagne Sud
Prime d'encadrement doctoral et de Recherche (PEDR 2003/2007)
• Septembre 1997/Août 1998 Attaché Temporaire d'Enseignement et de Recherche à l'Université de Nice – Sophia Antipolis
• Septembre 1994/Août 1997 Moniteur de l'Enseignement Supérieur à l'Université de Nice – Sophia Antipolis
• Septembre 1994/Août 1997 Doctorant (bourse MENRT)
Laboratoire I3S, CNRS – Université de Nice – Sophia Antipolis
</td></tr>
<tr><td>Enseignement</td><td>Electronique numérique et analogique, conception ASICs, FPGA/CPLD, langage VHDL, langage C, cryptographie, architectures de processeurs, microcontrôleur, codesign, architectures adaptatives, automatique, automates programmables, réseaux industriels, chaîne d'acquisition du signal, physique des semi-conducteurs, langages de spécification des systèmes embarqués (Esterel, SystemC, VHDL-AMS…)</td></tr>
<tr><td>Recherche</td><td>Codesign, architectures reconfigurables dynamiquement, auto-reconfiguration partielle, sécurité des systèmes embarqués, partitionnement fonctionnel, optimisation, méthodes de synthèse, architecture des systèmes embarqués, intégration logiciel/matériel, estimation système, logiciel et matériel, exploration de l'espace de conception, partitionnement, RTOS, application de télécommunication, multimédia et cryptographie, conception dirigée par les modèles (MDA)</td></tr>
<tr><td>Développement</td><td>Synthèse des communications pour le codesign (outil CODEF Philips/VLSI technology)
Environnement de codesign Design Trotter</td></tr>
<tr><td>Laboratoire</td><td>Laboratoire LESTER, Université de Bretagne Sud, CNRS FRE 2734</td></tr>
<tr><td>Responsabilités administratives</td><td>Membre de la commission de spécialistes 61ème et 63ème sections (2001/aujourd'hui)
Membre du Conseil de l'UFR Sciences et Sciences de l'Ingénieur (2003/2004)
Membre du conseil du département Sciences et Techniques de l'UFR SSI (2007/aujourd'hui)

Co-responsable du Master recherche Electronique de l'Université de Bretagne Sud (2006/aujourd'hui)
Participation à la mise en place des relations internationales pour l'enseignement et la recherche (Canada, Portugal)

Responsable de la promotion du département Génie Industriel et Maintenance (2000/2003)
Membre de la commission recherche à l'IUT de Lorient (2006/aujourd'hui)</td></tr>
<tr><td>Responsabilités scientifiques</td><td>Membre de l'AS n°28 Architecture Reconfigurable Dynamiquement (2002/2003), co-responsable du thème outils
Membre du GdR ISIS, Thème C et membre du GdR SoC-SiP

Comité de programme de conférences internationales ERSA 2005, ERSA 2006, ERSA 2007, DASIP 2007 (Program co-chair)
Comité de lecture de conférences ERSA 2002, ERSA 2005, ERSA 2006, ERSA 2007, FPL 2003, ICM 2004, ASAP 2005, IES 2006, GLVLSI 2007, GRETSI 2007, EUSIPCO 2007, SIPS 2007
Modérateur de sessions de conférences internationales ERSA 2006, ICECS 2006, ReCoSoC 2007, CryptArchi 2007
Comité de lecture de journaux nationaux et internationaux Revue scientifique francophone Traitement du signal (TS) Journal IEE Proceedings – Computers and Digital Techniques, IEEE Transactions on VLSI, IEEE Transactions on CAD, International Journal on Computers and Electric Engineering, The Journal of VLSI Signal Processing
Expertise scientifique nationale et internationale expert international NWO Computer Science Open Competition 2005 (Hollande), Agence Nationale de la Recherche – Appel Architectures du futur (2006, 2007)</td></tr>
</table>

Encadrements	7 co-encadrements de thèses effectués (dont 4 en cours) 1 encadrement de Post-Doc 12 encadrements de DEA ou Master	
Collaborations	**MOPCOM 2009**	2007/2009, Projet RNTL, *Thales, Thomson, Ensieta, Supelec, Irisa, Lester, Sodius*
	AETHER 2008	2006/2008, Projet Européen IST-FET (4th call ACA / FP6)
	ICTeR 2008	2006/2008, Projet ANR, *Lirmm, Enst, List, Lester, Netheos*
	SecureNIOS 2007	2006/2007, Projet sur fond propre, *Lester,Vspg (UMASS, USA)*
	PROSYR2006	2003/2006, CMCU, Lester, *Enis (Sfax, Tunisie)*
	SANES 2005	2004/2005, DGA ERE, *Lester,Vspg (UMASS, USA)*
	SecureFPGA 2004	2003/2004, Projet sur fond propre, *Lester,Vspg (UMASS, USA)*
	DARSoC 2003	2002/2003, Projet sur fond propre, *Lester,Vspg (UMASS, USA)*
	A3S 2005	2003/2005, Projet RNRT, *Thales Communications, Softeam, Mitsubishi, Lester*
	POMARD 2004	2003/2004, Équipe Projet CNRS, *R2d2, Lien, Lirmm Le2i, Etis, List, A&S, Lester*
	EPICURE 2003	2001/2003, Projet RNTL, *I3s, Lester, Cea/List, Thales Communications, Esterel-Technologies*
	MACGTT 2002	2000/2002, Projet CNRS, *I3s, Lasti, Lester*
	D2ASR 1999	1999, Contrat industriel, *Lester, Serpe-iesm*
	CODEF 2001	1998/2001, Contrat industriel, *I3s, Philips/Vlsi Technology*
Publications	7 revues scientifiques internationales 1 revue scientifique nationale 3 participations à des ouvrages scientifiques internationaux 53 publications en conférences internationales 12 publications en conférences nationales	

1. Curriculum Vitae

1.1 Etat civil

GOGNIAT Guy
Né le 17 juillet 1970 à Palaiseau (91)
Nationalité française
36 ans, marié, 2 enfants

Adresse personnelle
Kerplevert, 56700 Merlevenez

Adresse professionnelle
Laboratoire LESTER
CNRS FRE 2734, Université de Bretagne Sud
Centre de recherche, BP 92116 - 56321 LORIENT Cedex
tel : 02 97 87 45 41 – fax : 02 97 87 45 27
email : guy.gogniat@univ-ubs.fr
web : http://web.univ-ubs.fr/lester/~gogniat/gogniat.html

1.2 Grades et titres universitaires

Novembre 1997
Thèse de doctorat (sciences pour l'ingénieur)
Université de Nice – Sophia Antipolis
Architecture générique et synthèse des communications pour la conception conjointe de systèmes embarqués logiciel/matériel
Mention très honorable avec les félicitations du jury
Obtenue le 27 novembre 1997
Directeur de thèse Michel Auguin (directeur de recherches au CNRS)

Septembre 1994
DEA de Traitement du Signal et Architectures Électroniques
Université de Paris Sud Orsay
Option : Outils et Nouvelles Méthodologies de conception
Mention Bien

Septembre 1994
Diplôme d'Ingénieurs FIUPSO
Formation d'Ingénieur de l'Université Paris Sud Orsay
Spécialité systèmes électroniques

1.3 Situations successives

Septembre 2005/Aujourd'hui
Maître de Conférences à l'Université de Bretagne Sud
Enseignement à l'IUT de Lorient, Département Génie Industriel et Maintenance

Recherche au Laboratoire d'Electronique des Systèmes TEmps Réel (LESTER)

Novembre 2004/Août 2005
Chercheur invité (CRCT – Contrat ERE DGA)
Sécurité des systèmes électroniques embarqués et technologies reconfigurables
Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA, USA

Septembre 1998/Octobre 2004
Maître de Conférences à l'Université de Bretagne Sud
Enseignement à l'IUT de Lorient, Département Génie Industriel et Maintenance
Recherche au Laboratoire d'Electronique des Systèmes TEmps Réel (LESTER)
Prime d'encadrement doctoral et de Recherche (PEDR 2003/2007)

Septembre 1997/Août 1998
Attaché Temporaire d'Enseignement et de Recherche à l'Université de Nice – Sophia Antipolis
Enseignement au sein des écoles d'ingénieurs ESINSA et ESSI
Recherche au Laboratoire I3S, CNRS – Université de Nice – Sophia Antipolis

Septembre 1994/Août 1997
Moniteur de l'Enseignement Supérieur à l'Université de Nice – Sophia Antipolis
Enseignement à l'IUT de Nice, Département Génie Electrique et Informatique Industrielle
Enseignement à l'UNSA, Licence EEA et Maîtrise Informatique

Septembre 1994/Août 1997
Doctorant (bourse MENRT)
Recherche au Laboratoire I3S, CNRS – Université de Nice – Sophia Antipolis

1.4 Résumé

J'ai obtenu mon doctorat en sciences pour l'ingénieur en 1997 à l'Université de Nice – Sophia Antipolis, au sein du laboratoire d'Informatique Signaux et Systèmes (I3S). De 1997 à 1998 j'ai été ATER en poste à l'ESSI (École Supérieure en Sciences Informatiques) et à l'ESINSA (École Supérieure d'Ingénieurs de Nice – Sophia Antipolis). Depuis septembre 1998 je suis Maître de Conférences à l'Université de Bretagne Sud en poste à l'IUT de Lorient dans le département Génie Industriel et Maintenance. En 2004 j'ai obtenu un congé CRCT afin d'effectuer un séjour de 10 mois de recherche à l'Université du Massachusetts, Amherst, USA. Ce séjour a été soutenu par la DGA à travers un contrat ERE (séjour d'études de longue durée à l'étranger).

J'effectue mes activités de recherche au LESTER (Laboratoire d'Electronique des Systèmes TEmps Réel) au sein de l'équipe SysRec (Systèmes Reconfigurables). Mes travaux concernent la définition de nouvelles méthodologies de conception associées au développement d'outils de CAO, au prototypage d'applications sur systèmes reconfigurables dynamiquement et à la définition d'architectures embarquées sécurisées.

J'ai publié dans de nombreuses conférences majeures du domaine des systèmes embarqués et des outils de CAO : ISSS, CODES, ISCAS, RAW, FPGA, FPL, SAMOS, ERSA, ainsi que dans différents ouvrages et revues scientifiques (IEEE TCAD, Eurasip, ACM TODAES). Je participe actuellement à plusieurs comités de programmes de conférences internationales (ERSA, DASIP). J'ai également participé en tant qu'expert à différents appels à projets nationaux et internationaux.

Je participe à plusieurs projets européen et nationaux. Je participe activement à l'activité nationale dans le domaine de la conception des systèmes numériques embarqués (GDR ISIS, GDR SoC-SiP).

Je suis co-responsable du Master Recherche électronique à l'Université de Bretagne Sud depuis 2006.

1.5 Encadrements doctoraux et post-doctoraux

La liste ci-dessous résume les 7 co-encadrements de thèses effectués depuis 1998 (dont 4 en cours) ainsi que la situation professionnelle actuelle de ces étudiants. A cette liste s'ajoute le travail avec un étudiant en Post-Doc et l'encadrement de 12 étudiants de DEA ou Master.

Sébastien Bilavarn

Années de thèse : 1999/2002
Titre : **Exploration Architecturale au Niveau Comportementale – Application aux FPGAs**
Etat : soutenue (février 2002)
Financement: Région
Encadrement : Jean-Luc Philippe (50%), Guy Gogniat (50%)
Situation : Maître de Conférences à l'Ecole Polytechnique de l'Université de Nice – Sophia Antipolis

Lilian Bossuet

Années de thèse : 2001/2004
Titre : **Méthodologie d'exploration des architectures reconfigurables**
Etat : soutenue (septembre 2004)
Financement: MENRT
Encadrement : Jean-Luc Philippe (50%), Guy Gogniat (50%)
Situation : Maître de Conférences à l'Ecole nationale supérieure d'électronique, informatique & radiocommunications de Bordeaux (ENSEIRB)

Samuel Rouxel

Années de thèse : 2003/2006
Titre : **Modélisation et caractérisation de plates-formes SoC hétérogènes : Application à la Radio Logicielle**
Etat : soutenue (décembre 2006)
Financement: Contrat RNRT A3S
Encadrement : Jean-Luc Philippe (50%), Guy Gogniat (50%)
Situation : Ingénieur R&D CRESITT Industrie, Orléans, France

Issam Maalej

Années de thèse : 2002/2007
Titre : **Métriques au niveau système et partitionnement fonctionnel pour la conception des SoC**
Etat : en cours
Financement: CMCU – Cotutelle avec la Tunisie
Encadrement : Jean-Luc Philippe (25%), Mohamed Abid (25%), Guy Gogniat (50%)

Yassine Aoudni

Années de thèse : 2003/2007
Titre : **Mise en œuvre d'applications réactives sur SoC : proposition d'une démarche de validation**
Etat : en cours
Financement: CMCU – Cotutelle avec la Tunisie
Encadrement : Jean-Luc Philippe (25%), Mohamed Abid (25%), Guy Gogniat (50%)

Romain Vaslin

Années de thèse : 2005/2008
Titre : **Sécurité des systèmes embarqués**
Etat : en cours
Financement: MENRT
Encadrement : Jean-Philippe Diguet (50%), Guy Gogniat (50%)

Jorgiano Marcio Bruno Vidal

Années de thèse : 2007/2010
Titre : **Reconfiguration dynamique des systèmes : de la modélisation à la validation**
Etat : en cours
Financement: Contrat RNTL MOPCOM
Encadrement : Jean-Luc Philippe (50%), Guy Gogniat (50%)

1.6 Publications

Mon activité scientifique menée depuis 1994 a conduit aux publications suivantes : 7 revues scientifiques internationales, 1 revue scientifique nationale, 3 participations à des ouvrages scientifiques internationaux, 53 publications en conférences internationales et 12 publications en conférences nationales, ce qui représente 77 publications scientifiques.

2. Résumé des activités de recherches

2.1 Contexte

Ma première expérience de recherche a débuté en 1993/1994 durant mon année de DEA et notamment durant mon stage qui a porté sur le domaine alors émergent de la conception conjointe logiciel/matériel (i.e. codesign). Mes travaux ont traité du problème de la synthèse automatique des communications dans un système hétérogène. Il s'agissait des premiers pas vers la définition d'outils d'aide à la conception pour ce type de problème.

Ces travaux se sont poursuivis durant mes 3 années de doctorat (1994/1997) puis mon année d'ATER (1997/1998), période pendant laquelle j'ai continué mon activité autour des problématiques liées au codesign. 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. Chaque composant est connecté à l'architecture de communication à travers une interface générique. Je me suis également intéressé de façon approfondie aux étapes terminales du cycle de conception : synthèse des communications et intégration des composants dans l'architecture cible.

En septembre 1998, j'ai intégré le laboratoire LESTER de l'Université de Bretagne Sud à Lorient. J'ai poursuivi mon activité de recherche autour du thème du codesign mais je me suis également intéressé à de nouveaux domaines tels que le domaine des technologies reconfigurables et plus récemment le domaine de la sécurité des systèmes embarqués. En novembre 2004, j'ai effectué un séjour de recherche de 10 mois à l'Université du Massachusetts, Amherst, USA afin de travailler sur le thème de la sécurité des systèmes embarqués.

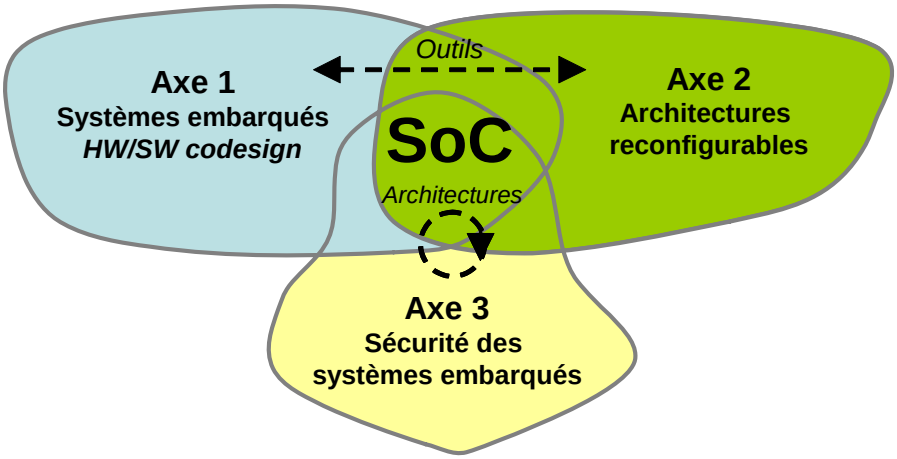


Figure 1 • Positionnement et interactions des 3 axes de recherche.

Toutes ces activités de recherche (codesign, reconfigurable, sécurité) interagissent à travers trois axes de recherche. Les travaux menés au sein de 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 extrêmement important et positif car il participe à l'éveil et à la maturité scientifique d'un chercheur. La Figure 1 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 présente succinctement mon activité de recherche selon ces trois axes.

Axe 1 : Systèmes embarqués (HW/SW codesign)

L'objectif de cet axe de recherche est 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/en cours)
- 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)

Axe 2 : Architectures reconfigurables

Cet axe de recherche s'intéresse à l'exploration de l'espace de conception des architectures reconfigurables gros grain et grain fin. 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. 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 été apportées au sein de 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)

Axe 3 : Sécurité des systèmes embarqués

Cet axe de recherche est le plus récent puisqu'il a débuté en 2004. Les problématiques adressées sont multiples mais visent 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 afin 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 de l'overhead lié à la sécurité par une approche de compression (2006/en cours)

2.2 Activités de recherches doctorales

DEA : Etude et modélisation de support de communication dans les systèmes mixtes logiciel/matériel

Mes travaux de DEA, dans le cadre du DEA Traitement du signal et architectures électroniques (Outils et nouvelles méthodologies de conception) de l'Université de Paris Sud Orsay (1993/1994), ont été effectué à l'Université de Nice – Sophia Antipolis au sein du projet MOSARTS (MOdélisation et Synthèse d'ARchitectures pour le Traitement du Signal) du Laboratoire I3S. La problématique adressée a été la synthèse des communications dans les systèmes hétérogènes logiciel/matériel. Une méthode permettant de générer automatiquement les protocoles de communication associés à une classe de support de type asynchrone (FIFO) a été développée. Cette méthode, qui est basée sur la modélisation de l'application à traiter par un graphe flot de données synchrone, détermine automatiquement les paramètres caractérisant le support de communication (profondeur, type de l'émetteur (bloquant ou non bloquant), type du récepteur).

Thèse de doctorat

J'ai continué mes activités de recherche à travers la préparation de mon doctorat (1994/1997), au sein du projet MOSART du laboratoire I3S de l'Université de Nice – Sophia Antipolis. Mes thèmes de recherche se sont articulés autour du domaine alors émergent du codesign. Je me suis intéressé à la définition d'une architecture générique et à la synthèse des communications.

Thèse de Doctorat soutenue le 27 novembre 1997 à l'Université de Nice – Sophia Antipolis
Architecture générique et synthèse des communications pour la conception conjointe de systèmes embarqués logiciel/matériel
Directeur de thèse : Michel Auguin (Directeur de Recherches au CNRS)

Composition du jury :

M. Auguin	Directeur de Recherches, UNSA,	Directeur de thèse
C. Belleudy	Maître de Conférences, UNSA,	Examinateur
F. Boéri	Professeur des Universités, UNSA,	Examinateur
E. Gresset	Ingénieur, VLSI Technology,	Invité
M. Israel	Professeur des Universités, UEVE,	Rapporteur, Président de jury
E. Martin	Professeur des Universités, UBS,	Rapporteur
F. Rousseau	Enseignant/chercheur, ESIM,	Examinateur

L'objectif de cette thèse était, à partir d'une application de traitement du signal à comportement statique partitionnée et ordonnancée, d'effectuer d'une part la synthèse des communications et d'autre part l'intégration logiciel/matériel sur une architecture cible. Aussi, pour atteindre cet objectif, la première étude a porté sur la caractérisation d'un modèle générique d'une architecture cible. Pour cela, l'analyse des principales caractéristiques des applications de traitement du signal a été effectuée. A partir des résultats obtenus et dans le souci de favoriser les principes de réutilisabilité et de

modularité, une architecture dirigée par les données utilisant des mécanismes de communication génériques a été définie.

La seconde étude a conduit à définir un modèle de communication selon lequel la synthèse des communications opère. Ce modèle directement lié à la structure d'interconnexion de l'architecture et des mécanismes de communication associés est basé sur des ressources du type FIFO et bus. Les communications entre les unités de l'architecture peuvent être synchrones ou asynchrones selon l'ordonnancement de l'application et peuvent cœxister dans l'architecture finale. L'étape suivante a permis de définir les schémas d'exécution associés au modèle de communication. Aussi, deux modèles d'évaluation ont été proposés : évaluations globale et fine. Le modèle d'évaluation globale considère les communications comme étant regroupées avec les traitements alors que dans le modèle d'évaluation fine ces deux opérations sont considérées séparément.

A partir de la modélisation des applications par un graphe flot de données et du modèle de communication avec les schémas d'exécution associés, l'étape de synthèse des communications a été déterminée. L'heuristique proposée vise à minimiser le surcoût entraîné par la mise en place des communications dans l'architecture cible. Pour cela, la méthode est décomposée en deux étapes successives : la caractérisation des communications et l'implémentation. L'étape de caractérisation conduit à fixer entièrement les mécanismes et les supports nécessaires à chaque communication de l'application. L'étape d'implémentation permet d'optimiser le nombre et la taille des ressources de communication implémentées dans l'architecture finale en regroupant les communications utilisant la même ressource et ayant des durées de vie disjointes.

Enfin, la dernière étape permet d'effectuer l'intégration logiciel/matériel de l'application sur l'architecture cible en ordonnançant les séquences de traitement à mettre en œuvre et en caractérisant le contrôle des communications selon les protocoles et les ressources utilisées.

La méthode proposée permet donc à partir de l'application partitionnée et ordonnancée d'aboutir à l'implémentation sur l'architecture finale. Ce travail constitue un apport original dans le domaine du codesign dans la mesure où ce problème était peu abordé dans sa globalité. En effet, les méthodes de synthèse des communications généralement proposées laissent à la charge du concepteur la détermination des protocoles à mettre en œuvre et s'occupent principalement de l'optimisation des ressources nécessaires. Par ailleurs, l'étude des communications est généralement déconnectée de l'étude de l'architecture ce qui ne permet pas de proposer une méthode aboutissant à l'intégration logiciel/matériel.

Mes travaux de thèse ont servi de support à la définition d'une méthode de synthèse des communications pour la conception de systèmes embarqués dédiés aux traitements du signal dans le cadre d'un projet contractuel avec la société Philips/VLSI Technology. Ce projet qui a duré trois ans (1998/2001) a abouti à l'outil CODEF (Codesign Framework of Heterogeneous System) qui est un outil propriétaire de la société Philips/VLSI Technology.

Avant d'être nommé Maître de Conférences à l'Université de Bretagne Sud j'ai été ATER pendant une année à l'Université de Nice – Sophia Antipolis. Durant cette année j'ai poursuivi mon activité de recherche au sein du projet MOSART et j'ai notamment participé au co-encadrement de thèse de Fernand Cuesta qui a également travaillé sur le problème de la synthèse des communications. J'ai également travaillé avec Luc Bianco et Alain Pégatoquet sur le partitionnement logiciel/matériel et sur l'estimation logicielle. Cette année de recherche m'a permis d'élargir mon champs de connaissance relatif au domaine du codesign puisque j'ai pu abordé la plupart des problématiques associées.

L'ensemble des travaux de recherche menés dans ce domaine ont conduit à 19 publications scientifiques (2 revues, 1 participation à un ouvrage scientifique, 10 conférences internationales, 4 conférences nationales) [Gogniat 2000/R] [Freund 1997/R] [Gogniat 1997/O] [Gogniat1998/CI] [Bianco 1998/CI] [Pegatoquet 1998/CI] [Gogniat 1997/CI] [Auguin 1997/CI] [Gogniat 1996/CI] [Auguin 1996/CI] [Freund 1996/CI] [Auguin 1995/CI] [Gogniat 1995/CI] [Freund 1997/CN] [Gogniat 1997/CN] [ASAR 1996/CN] [Auguin 1995/CN].

2.3 Activités de recherches en tant que Maître de Conférences

Depuis 1998, date de mon arrivée au laboratoire LESTER (Laboratoire d'Electronique des Systèmes TEmps Réel) à l'Université de Bretagne Sud, mon activité de recherche s'est inscrite dans le thème fédérateur du laboratoire en Adéquation Algorithme Architecture par une approche méthodologique de conception des circuits et systèmes sous contraintes. Je suis membre de l'équipe SysRec (systèmes reconfigurables, équipe composée de Jean Luc Philippe, Professeur des Universités et Jean Philippe Diguet, Chargé de recherche au CNRS pour ce qui est des membres permanents).

Les thèmes de recherche que j'ai développé ces dernières années se sont concentrés autour d'une problématique unique, la conception des systèmes embarqués (*System on Chip – SoC*), mais selon différents points de vues. Trois axes de recherche peuvent ainsi être dégagés : systèmes embarqués (HW/SW codesign), architectures reconfigurables, sécurité des systèmes embarqués. Douze stagiaires de DEA et Master et sept étudiants en thèse ont travaillé ou travaillent actuellement sur ces différents thèmes de recherche.

Ces travaux de recherche ont été ou sont validés sur différents types d'applications du domaine des télécommunications, du multimédia et aussi de la sécurité. Dans la suite je présente ces différents thèmes de recherche selon chacun des 3 axes.

Axe 1 : Systèmes embarqués (HW/SW codesign)

L'objectif de cet axe de recherche est 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.

Un point essentiel, lors de la définition d'une méthodologie de conception, est le choix des modèles et des langages de spécification. Afin de spécifier des applications complexes représentatives des systèmes de télécommunication et du multimédia (radio logicielle) les langages C et UML ont été retenus. A partir de ces langages, des modèles de graphes de tâches (modèle GT) et de fonctions (modèle HCDFG) ont été développés. Ces derniers associés à des modèles des plates formes d'exécution permettent l'exploration de l'espace de conception. Ces travaux ont conduit à la définition d'une méthodologie, de prototypage rapide pour des systèmes reconfigurables, dirigée par les modèles (MDA). Une thèse a été menée sur ce sujet [Rouxel 2006/T] et une thèse est en cours afin d'étendre les travaux précédents aux applications ayant un comportement dynamique [Vidal 20XX/T]. Dans la méthode de prototypage citée ci-dessus l'étape de partitionnement logiciel/matériel est manuelle, aussi afin d'aider le concepteur, des travaux sont actuellement menés sur la définition d'une méthodologie de partitionnement fonctionnel (avant la définition de la plate forme d'exécution); un étudiant effectue actuellement une thèse sur ce sujet [Maalej 20XX/T]. Par ailleurs, un point clef des méthodologies de conception actuelles, se situe au niveau de l'accélération des traitements implémentés sur des solutions à base de processeurs. L'objectif est alors d'augmenter les performances aussi bien en vitesse d'exécution qu'en consommation (efficacité énergétique). Pour cela des travaux sont menés actuellement sur l'extraction automatique, à partir d'une spécification décrite en langage C, de primitives de calcul pouvant être déportées sur des coprocesseurs ou des accélérateurs matériels. Une thèse est en cours sur ce sujet [Aoudni 20XX/T].

La Figure 2 illustre les travaux menés au sein de cet axe 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).

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].

Les travaux menés au sein de cet axe de recherche ont 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].

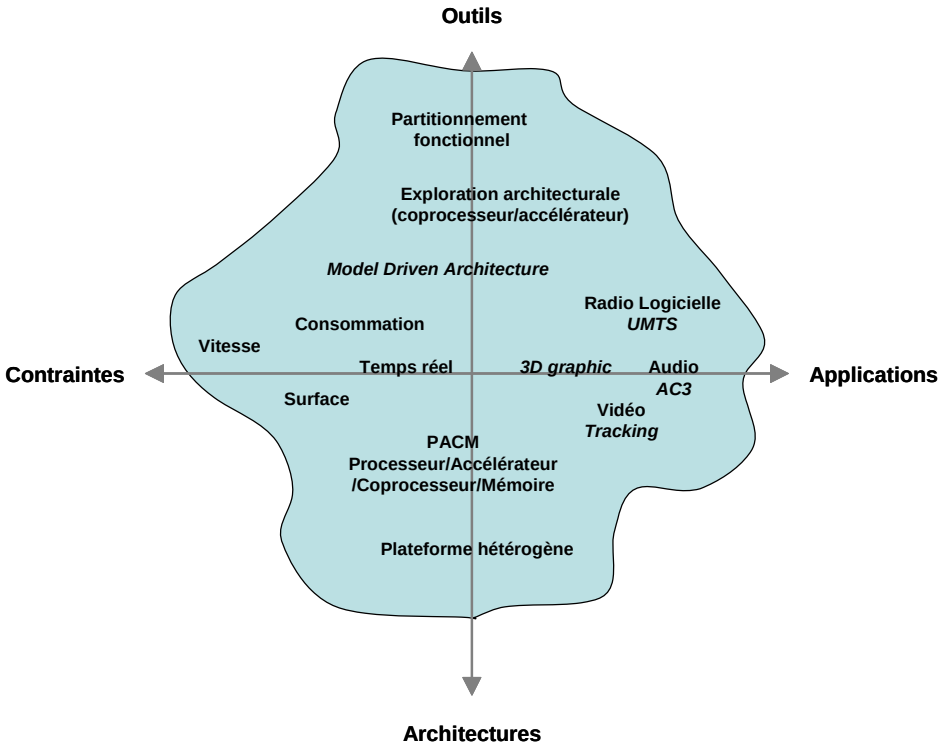


Figure 2 • Couverture de l'espace d'exploration (outils, contraintes, applications et architectures) de l'axe 1.

Axe 2 : Architectures reconfigurables

Les travaux de recherche menés au sein de cet axe correspondent à la définition d'une méthodologie d'exploration de l'espace de conception pour des architectures reconfigurables grain fin, gros grain ou hétérogènes. Deux thèses ont été menées sur ce sujet [Bilavarn 2002/T], [Bossuet 2004/T].

Un autre aspect essentiel est la maîtrise des nouvelles technologies reconfigurables dynamiquement. Un stagiaire de DEA a étudié l'impact du routage sur les performances temporelle et en consommation pour les composants FPGA [Rouxel 2002/D]. Deux stagiaires de DEA se sont concentrés sur la mise en œuvre de la reconfiguration dynamique

partielle et l'auto reconfiguration, technologie prometteuse étant données les nouvelles plates formes d'exécution du type SoC reconfigurable [Delahaye 2003/D], [Guillot 2004/D]. Ces travaux sont essentiels afin d'anticiper la définition des futures plates formes d'exécution.

La Figure 3 positionne les travaux menés dans cet axe selon les 4 dimensions définies précédemment. 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 oeuvre 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.

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].

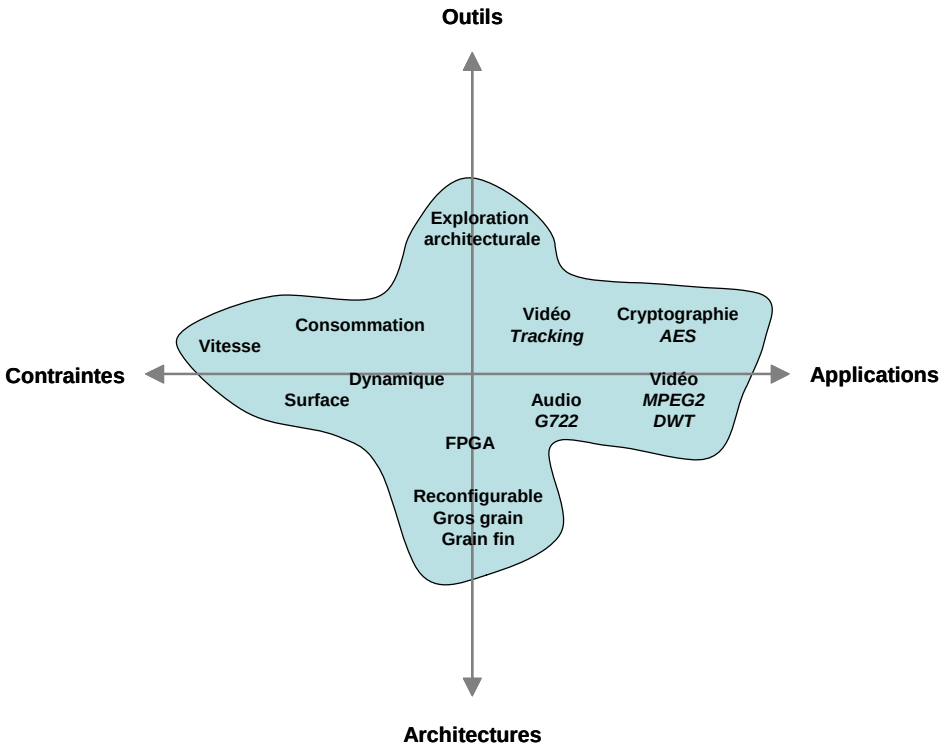


Figure 3 • Couverture de l'espace d'exploration (outils, contraintes, applications et architectures) de l'axe 2.

Axe 3 : Sécurité des systèmes embarqués

Cet axe de recherche est le plus récent puisqu'il a débuté en 2004. Les problématiques adressées sont multiples mais visent à renforcer la sécurité des systèmes embarqués. Ce thème prend un essor très important depuis plusieurs années. Les enjeux sont effectivement considérables étant donné le nombre de systèmes potentiellement visés (cartes à puce pour des transactions financières, le suivi médical, les télécommunications, set top box...).

Les études menées sur ce sujet concernent la protection des données et des systèmes à travers la mise en oeuvre de mécanismes dynamiques. Il est également important de soulager les solutions actuelles uniquement basées sur du logiciel (antivirus, pare feux...) en déportant des solutions de sécurité sur du matériel. Un étudiant en thèse travaille actuellement sur la mise en oeuvre de la confidentialité/intégrité des données entre un processeur et une mémoire [Vaslin 20XX/T]. Un post doctorant Brésilien travaille sur la réduction de l'overhead lié à la sécurité par une approche de compression [Wanderley 2007/P]. Trois étudiants en Master ont travaillé ou travaillent actuellement sur la mise en oeuvre de primitives de sécurité sur des technologies programmables ou reconfigurables [Dumérat 2006/D], [Zui 2007/D], [Ducloyer 2007/D]. Par ailleurs en novembre 2004, j'ai effectué un séjour de recherche de 10 mois à l'Université du Massachusetts, Amherst, USA afin de travailler sur ce thème. 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.

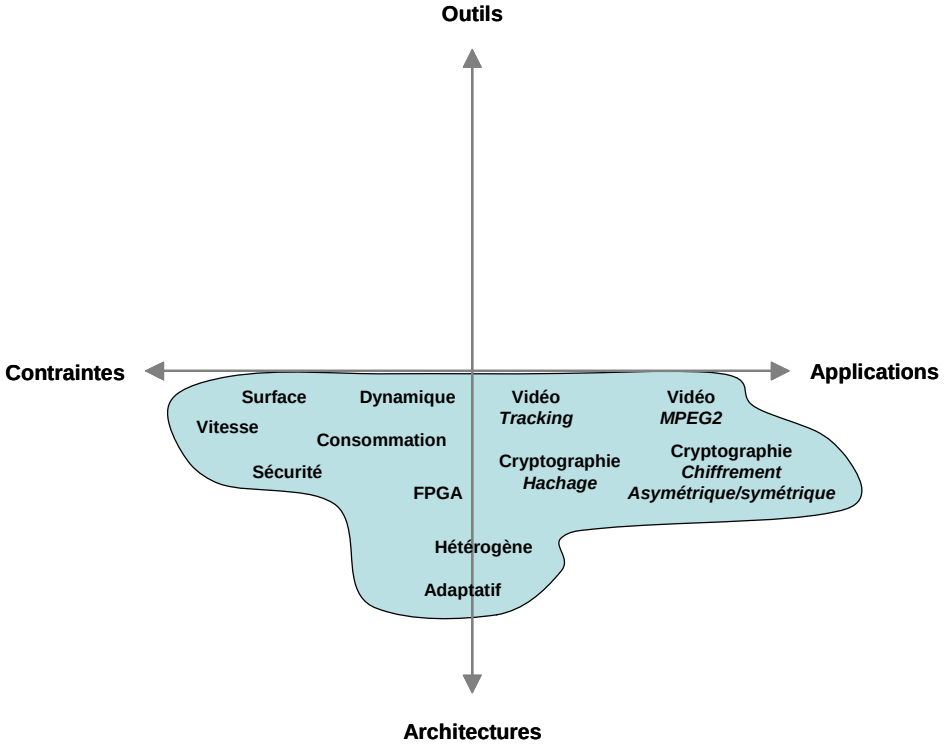


Figure 4 • Couverture de l'espace d'exploration (outils, contraintes, applications et architectures) de l'axe 4.

La Figure 4 positionne les travaux menés au sein de 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

mais ces derniers sont très insuffisants, aussi il indéniable qu'à l'avenir ce point deviendra critique.

Afin de mener à bien ces travaux 1 doctorant [Vaslin 200X/T] et un post-doc [Wanderley 2007/P] 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 2007a/CI] [Wanderley 2007a/CI] [Vaslin 2007b/CI] [Wanderley 2007b/CI] [Diguet 2007/CI] [Wanderley 2006/CI] [Vaslin 2006b/CI] [Vaslin 2006a/CI] [Gogniat 2006b/CI] [Wolf 2006/CI] [Gogniat 2005b/CI] [Gogniat 2005a/CI] [Bossuet 2004/CI].

2.4 Encadrement de travaux de recherches doctorales

Cette partie concerne les activités d'encadrement de travaux de recherches, à savoir :

- Encadrement de Thèses
- Encadrement de Post-Doc
- Encadrement de stages de DEA et de Master

2.4.1 Co-encadrements de thèses

Depuis 1998 j'ai co-encadré 7 thèses. Trois thèses ont été soutenues, respectivement en 2002, 2004 et 2006. Quatre thèses sont actuellement en cours dont deux en cotutelles avec la Tunisie.

[Bilavarn 2002/T] Sébastien Bilavarn 1998/2002 – Bourse Région

Exploration Architecturale au Niveau Comportemental – Application aux FPGAs

Thèse de Doctorat soutenue le 28 Février 2002

En Co-direction avec le Pr. Jean Luc Philippe (50%)

Résumé : *Un facteur important dans l'évolution des systèmes électroniques modernes est l'apparition de nouvelles architectures basées sur la programmation de circuits matériels tels que les composants programmables. Les récentes évolutions des différentes familles autorisent aujourd'hui l'intégration de systèmes de plus en plus complexes avec des contraintes de performances de plus en plus fortes. D'autre part, la flexibilité offerte par ce type de technologie fait des FPGAs (Field Programmable Gate Arrays) une cible architecturale promise à un bel avenir. L'évaluation des performances d'une application sur une technologie reconfigurable est un problème peu étudié à ce jour. Jusqu'à présent, les chercheurs ont principalement porté leurs efforts sur l'amélioration des architectures afin de les rendre plus performantes et ainsi constituer une réelle alternative aux ASICs (Application Specific Integrated Circuits). L'objectif du travail réalisé durant cette thèse consiste à proposer des techniques et les outils associés permettant l'évaluation rapide des performances (temps, surface) d'applications sur des architectures programmables. La méthode développée est générique (elle s'applique à plusieurs familles de FPGAs) et se situe au niveau comportemental. Elle permet l'exploration de plusieurs solutions architecturales et s'intègre dans un flot de conception conjointe logiciel / matériel.*

Composition du jury : Jean Luc Philippe *(Professeur des Universités, Université de Bretagne Sud, directeur de thèse)*, Michel Auguin *(Directeur de Recherches CNRS, I3S Nice – Sophia Antipolis, rapporteur)*, Dominique Lavenier *(Directeur de Recherches CNRS, IRISA Rennes, rapporteur)*, Michel Corazza *(Professeur des Universités, ENSSAT Lannion, président de jury)*, Thierry Collette *(Ingénieur de Recherches, CEA Saclay, examinateur)*, Guy Gogniat *(Maître de Conférences, Université de Bretagne Sud,*

examinateur), Jean Philippe Diguet *(Maître de Conférences, Université de Bretagne Sud, invité).*

Mention très honorable

Situation actuelle de Sébastien Bilavarn : Maître de Conférences à l'Ecole Polytechnique de l'Université de Nice – Sophia Antipolis

[Bossuet 2004/T] Lilian Bossuet 2001/2004 – Bourse MENRT

Méthodologie d'exploration des architectures reconfigurables

Thèse de Doctorat soutenue le 10 septembre 2004

En Co-direction avec le Pr. Jean Luc Philippe (50%)

Résumé : *Les travaux réalisés durant cette thèse concernent l'exploration de l'espace de conception des architectures reconfigurables pour des applications orientées traitement intensif à partir d'un haut niveau d'abstraction (niveau système). Dans cette thèse, nous proposons une méthode d'exploration de l'espace architectural de conception afin de converger rapidement vers la définition d'une architecture efficace pour une application donnée. Cette méthode intervient très tôt dans le flot de conception, ainsi dès les premières phases de spécification de l'application, les concepteurs peuvent définir une architecture adaptée pour leurs applications. Notre méthode s'appuie principalement sur l'estimation de la répartition des communications dans l'architecture ainsi que sur le taux d'utilisation des ressources de l'architecture. Ces métriques permettent en effet d'orienter le processus d'exploration afin de minimiser la consommation de puissance de l'architecture puisque cette dernière est directement corrélée à ces deux métriques. Ces travaux ont conduit au développement d'un outil qui s'inscrit dans un environnement logiciel plus large développé au LESTER ; Design Trotter. Nous avons appliqué notre méthode d'exploration architecturale à des applications du traitement des images et de la cryptographie. Les résultats obtenus montre que notre méthode permet de converger rapidement vers une architecture efficace en ce qui concerne la consommation de puissance. De plus le concepteur obtient de nombreuses informations sur l'architecture reconfigurable en adéquation avec l'application développée. Enfin, nos travaux nous ont permis de mettre en évidence des styles d'architectures reconfigurables adaptés à des domaines d'applications.*

Composition du jury : Joël Liénard *(Professeur des Universités, LIS, ENSIEG, Grenoble, président de jury),* Didier Demigny *(Professeur des Universités, LASTI, IUT Lannion, rapporteur),* Lionel Torres *(Maître de Conférences, HDR, LIRMM, Montpellier II, rapporteur),* Jean Luc Philippe *(Professeur des Universités, Université de Bretagne Sud, directeur de thèse),* Bernard Pottier *(Maître de Conférences, A&S, UBO, Brest, examinateur),* Guy Gogniat *(Maître de Conférences, Université de Bretagne Sud, examinateur),* Joël. Cambonie *(Ingénieur, St Microelectronics, Grenoble, invité).*

Mention très honorable avec les félicitations du jury

Situation : Maître de Conférences à l'Ecole nationale supérieure d'électronique, informatique & radiocommunications de Bordeaux (ENSEIRB)

[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-direction avec le Pr. Jean Luc Philippe (50%)

Résumé : *Les travaux de cette thèse se sont déroulés dans le cadre du projet RNRT A3S, et intègrent la notion de composants au sein d'une méthodologie de conception d'une plate-*

forme SoC (System on Chip), basée sur le langage de modélisation UML (Unified Modeling Language). Cette méthodologie propose un environnement de modélisation qui repose sur le profil UML A3S, appuyant l'approche d'architectures basées sur les modèles (MDA). Elle permet en outre de modéliser et de spécifier un système en décorrélant dans un premier temps l'application logicielle et l'architecture matérielle pouvant la supporter. Le modèle applicatif obtenu (PIM) est indépendant de la plate-forme matérielle (modèle architectural). C'est seulement dans un second temps, après avoir modélisé une ou plusieurs applications et plates-formes matérielles que le concepteur peut tester un ou plusieurs choix d'implémentation. Le modèle devient alors dépendant de la plate-forme matérielle (modèle PSM). La méthodologie inclut par ailleurs des règles de vérifications et de validation des systèmes (respect des contraintes structurelles et temps réel). L'outil développé, mettant en oeuvre cette méthodologie exploite le profil A3S, et permet ainsi de modéliser, de spécifier des systèmes complexes et de vérifier les modélisations effectuées en UML, et renseigne le concepteur sur la faisabilité du système développé (ordonnançabilité, taux d'occupation), après l'analyse de l'outil XAPAT intégré. Cette conception basée sur les modèles implique un niveau de représentation des éléments qui les compose, en adéquation avec le niveau de conception désiré. Elle implique également une manipulation aisée pour les architectes systèmes, en les assistant par une sémantique riche pour permettre l'analyse et la validation des systèmes. Le domaine applicatif retenu est celui des systèmes Radio Logicielle. Une chaîne UMTS a notamment permis la validation de l'outil en confrontant les résultats estimés de l'outil, à ceux mesurés sur une plate-forme temps réel hétérogène (multi-DSP, multi-FPGA).

Composition du jury : Fabienne Nouvel *(Maître de Conférences, HDR, IETR, Rennes, rapporteur)*, Charles André *(Professeur des Universités, I3S, Sophia–Antipolis, rapporteur, président de jury)*, Christophe Moy *(Enseignant/Chercheur, Supelec, Rennes, examinateur)*, Joël Champeau *(Maître de Conférences, ENSIETA, Brest, examinateur)*, Jean-Luc Philippe *(Professeur des Universités, Université de Bretagne Sud, directeur de thèse)*, Guy Gogniat *(Maître de Conférences, Université de Bretagne Sud, examinateur)*.

Mention très honorable

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-direction 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-direction avec les Pr. Jean-Luc Philippe (25%) et Pr. Mohamed Abid (25%)

[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%)

[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-direction avec le Pr. Jean Luc Philippe (50%)

Le Table 1 ci-dessous résume l'ensemble des thèses réalisées en co-encadrement. Différentes sources de financements ont été utilisées afin de financer les doctorants (bourse région, bourse MENRT, contrat RNRT A3S, CMCU – Cotutelle avec la Tunisie, contrat RNTL MOPCOM).
Nous avons notamment mis en place en 2002 un contrat CMCU avec l'école nationale d'ingénieur de Sfax (ENIS) qui a permis de financer partiellement deux doctorants (financements de missions en France). Il s'agissait pour nous d'une première expérience de cette nature qui faisait suite à différents stages de DEA et Master en communs. L'expérience n'est pas terminée mais comme nous pouvons le constater ce type de formule nous a conduit à des thèses se déroulant sur une période supérieure à 3 ans ce qui n'est pas satisfaisant. Le fonctionnement actuel ne permet pas d'atteindre le niveau de qualité requis par les écoles doctorales françaises; Il est important de tirer les enseignements de cette expérience pour construire une relation pérenne. A l'avenir il nous semble donc essentiel que les thèses en co-tutelles bénéficient d'un financement complet afin de donner aux doctorants les conditions nécessaires au bon déroulement de leurs travaux. Il me semble important de maintenir ce type d'échange mais il est essentiel de donner aux doctorants les moyens de produire une activité de recherche de qualité et motivante tout en respectant la durée des 3 ans.

Table 1 • Résumé des co-directions de thèses et du devenir des étudiants.

Année Doctorant	98/ 99	99/ 00	00/ 01	01/ 02	02/ 03	03/ 04	04/ 05	05/ 06	06/ 07	07/ 08	08/ 09	09/ 10
Sébastien Bilavarn Région		**Exploration Architecturale au Niveau Comportementale – Application aux FPGAs**			Maître de Conférences à l'ENSISA							
Lilian Bossuet MENRT				**Méthodologie d'exploration des architectures reconfigurables**			Maître de conférences à l'ENSEIRB					
Samuel Rouxel Contrat RNRT					**Modélisation et caractérisation de plates-formes SoC hétérogènes : Application à la Radio Logicielle**				Ingénieur R&D CRESITT Industrie, Orléans, France			
Issam Maalej CMCU					**Métriques au niveau système et partitionnement fonctionnel pour la conception des SoC**							
Yassine Aoudni CMCU						**Mise en œuvre d'applications réactives sur SOC : proposition d'une démarche de validation**						
Romain Vaslin MENRT								**Sécurité des systèmes embarqués**				
Jorgiano Marcio Bruno Vidal Contrat RNTL										**Reconfiguration dynamique des systèmes : de la modélisation à la validation**		

2.4.2 Travaux de recherche avec étudiant en Post-Doc.

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

Réduction de l'overhead lié à la sécurité par une approche de compression

Juillet 2006/Juillet 2007

2.4.3 Encadrements de stages de DEA et de Master

J'ai encadré 12 étudiants en DEA ou en Master sur des sujets relatifs à la conception des systèmes embarqués sur DSP et FPGA, à la mise en œuvre d'algorithmes de sécurité sur des architectures matérielle et logicielle et sur le thème des systèmes d'exploitation pour les systèmes embarqués temps réel.

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

[Bossuet 2001/D] Lilian Bossuet
Modélisation d'architectures reconfigurables embarquées
DEA Rennes, année 2000/2001

[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

[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

[Guillot 2004/D] Jérémie Guillot
Cryptographie et auto reconfiguration dynamique sur FPGA
DEA Lorient, année 2003/2004

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

[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

La Table 2 résume l'ensemble des stages de DEA et de Master dont j'ai assuré l'encadrement.

Table 2 • Résumé des encadrements des stages de DEA et Master.

Année Stagiaire	98/ 99	99/ 00	00/ 01	01/ 02	02/ 03	03/ 04	04/ 05	05/ 06	06/ 07
Said Chaboun ***DEA Rennes***		Etude et implémentation d'algorithmes de compression de signaux audio sur des cibles hétérogènes							
Lilian Bossuet ***DEA Rennes***				Modélisation d'architectures reconfigurables embarquées					
Issam Maalej ***DEA Sfax***					IP de communication générique à base de bus pour les systèmes embarqués				
Ismail Naoufel **Master Sfax**					Modélisation et intégration d'un accélérateur matériel sur le système à base du processeur LEON				
Samuel Rouxel ***DEA Lorient***					Caractérisation de l'impact du routage sur les performances (vitesse et consommation de puissance) d'un FPGA				
Erwan Piriou ***DEA Rennes***						Comparaison de performance entre DSP et FPGA pour des applications de traitement du signal et des images			
Jean Philippe Delahaye ***DEA Orsay***						Systèmes radio dynamiquement reconfigurables sur des architectures hétérogènes			
Jérémie Guillot ***DEA Lorient***			Cryptographie et auto reconfiguration dynamique sur FPGA						
Kais Loukil ***Master Sfax***					Estimation du temps d'exécution des systèmes temps réel sur puce				
Arnaud Dumérat ***Master Math/Info Vannes***					Algorithme de chiffrement ECC : détection et tolérance aux fautes				
Tao Zui ***Master Math & App. Vannes***						Algorithme de chiffrement : mise en oeuvre sur le Nios			
Sylvain Ducloyer ***Master Electronique Lorient***						Architecture matérielle pour le hachage : application à MD5/SHA-1/SHA-2			

2.5 Responsabilités scientifiques

Cette partie concerne les responsabilités scientifiques, à savoir :

- Participation à des jurys de thèse
- Participation à des jurys de *Master thesis*
- Participation à des comités de lectures et de programmes de conférences
- Participation à des comités de lecture de journaux nationaux et internationaux
- Expertise scientifique nationale et internationale

2.5.1 Participation à des jurys de thèse

J'ai participé à quatre jurys de thèse en tant qu'examinateur, un à l'Université de Nice – Sophia Antipolis et trois à l'Université de Bretagne Sud.

[Cuesta 2001/J] Fernand Cuesta

Synthèse des ressources de communication pour la conception de systèmes embarqués temps réel flots de données

Thèse de Doctorat soutenue le 26 octobre 2001, Université de Nice–Sophia Antipolis

Composition du jury : Daniel Litaize *(Professeur des Universités, IRIT, Université Paul Sabatier de Toulouse, rapporteur, président de jury)*, Bruno Rouzeyre *(Professeur des Universités, LIRMM, Université de Montpellier II, rapporteur)*, Guy Gogniat *(Maître de Conférences, Université de Bretagne Sud, examinateur)*, Laurent Kwiatkowski *(Maître de Conférences, Université de Nice – Sophia Antipolis, examinateur)*, Emmanuel Gresset *(Ingénieur, VLSI Technology, invité).*

Mention très honorable

[Bilavarn 2002/J] Sébastien Bilavarn

Exploration Architecturale au Niveau Comportemental – Application aux FPGAs

Thèse de Doctorat soutenue le 28 Février 2002, Université de Bretagne Sud

Composition du jury : Jean Luc Philippe *(Professeur des Universités, Université de Bretagne Sud, directeur de thèse)*, Michel Auguin *(Directeur de Recherches CNRS, I3S Nice – Sophia Antipolis, rapporteur)*, Dominique Lavenier *(Directeur de Recherches CNRS, IRISA Rennes, rapporteur)*, Michel Corazza *(Professeur des Universités, ENSSAT Lannion, président de jury)*, Thierry Collette *(Ingénieur de Recherches, CEA Saclay, examinateur)*, Guy Gogniat *(Maître de Conférences, Université de Bretagne Sud, examinateur)*, Jean Philippe Diguet *(Maître de Conférences, Université de Bretagne Sud, invité)*.

Mention très honorable

[Bossuet 2004/J] Lilian Bossuet

Méthodologie d'exploration des architectures reconfigurables

Thèse de Doctorat soutenue le 10 septembre 2004, Université de Bretagne Sud

Composition du jury : Joël Liénard *(Professeur des Universités, LIS, ENSIEG, Grenoble, président de jury)*, Didier Demigny *(Professeur des Universités, LASTI, IUT Lannion, rapporteur)*, Lionel Torres *(Maître de Conférences, HDR, LIRMM, Montpellier II, rapporteur)*, Jean Luc Philippe *(Professeur des Universités, Université de Bretagne Sud, directeur de thèse)*, Bernard Pottier *(Maître de Conférences, A&S, UBO, Brest, examinateur)*, Guy Gogniat *(Maître de Conférences, Université de Bretagne Sud, examinateur)*, Joël. Cambonie *(Ingénieur, St Microelectronics, Grenoble, invité)*.

Mention très honorable avec les félicitations du jury

[Rouxel 2006/J] Samuel Rouxel

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

Thèse de Doctorat soutenue le 5 décembre 2006, Université de Bretagne Sud

Composition du jury : Fabienne Nouvel *(Maître de Conférences, HDR, IETR, Rennes, rapporteur)*, Charles André *(Professeur des Universités, I3S, Sophia–Antipolis, rapporteur, président de jury)*, Christophe Moy *(Enseignant/Chercheur, Supelec, Rennes, examinateur)*, Joël Champeau *(Maître de Conférences, ENSIETA, Brest, examinateur)*, Jean-Luc Philippe *(Professeur des Universités, Université de Bretagne Sud, directeur de thèse)*, Guy Gogniat *(Maître de Conférences, Université de Bretagne Sud, examinateur).*

Mention très honorable

2.5.2 Participation à des jurys de Master thesis

J'ai participé à un jury de *Master thesis* à l'Université du Massachusetts, Amherts, USA suite à mon séjour de recherche au sein de cette Université.

[Gomez-Prado 2006/M] Daniel F. Gomez-Prado

Variable Ordering for Taylor Expansion Diagrams

Master thesis soutenue le 9 janvier 2006, Université du Massachusetts, Amherts, USA

Composition du jury : Maciej Ciesielski *(Professor, Université du Massachusetts, Amherst, directeur de thèse)*, Wayne Burleson *(Professor, Université du Massachusetts, Amherst, examinateur)*, Guy Gogniat *(Maître de Conférences, Université de Bretagne Sud, examinateur).*

2.5.3 Participation à des comités de lecture et de programmes de conférences

Comité de programme de conférences internationales

- International Conference on Engineering of Reconfigurable Systems and Algorithms (ERSA 2005, ERSA 2006, ERSA 2007)
- Workshop on Design and Architectures for Image and Signal Processing (DASIP 2007) Program co-chair

Comité de lecture de conférences

- International Conference on Engineering of Reconfigurable Systems and Algorithms (ERSA 2002, ERSA 2005, ERSA 2006, ERSA 2007).
- International Conference on Field Programmable Logic and Applications (FPL 2003).
- IEEE International Conference on Microelectronics (ICM 2004)
- IEEE 16th International Conference on Application-specific Systems, Architectures and Processors (ASAP 2005)
- IEEE Symposium on Industrial Embedded Systems IES (2006)
- ACM Great Lakes Symposium on VLSI (GLVLSI 2007)
- GRETSI 2007, European Signal Processing Conference (EUSIPCO 2007)
- IEEE Workshop on Signal Processing Systems (SiPS 2007)

Modérateur de sessions de conférences internationales

- International Workshop on Reconfigurable Communication Centric System-on-Chips (ReCoSoC 2007)
- International Workshop on Cryptographic Architectures Embedded in Reconfigurable Devices (CryptArchi 2007)

- IEEE International Conference on Electronics, Circuits and Systems (ICECS 2006)
- International Conference on Engineering of Reconfigurable Systems and Algorithms (ERSA 2006).

2.5.4 Participation à des comités de lecture de journaux nationaux et internationaux

- Revue scientifique francophone Traitement du signal (TS)

- Journal IEE Proceedings – Computers and Digital Techniques.
- IEEE Transactions on Very Large Scale Integration (VLSI)
- IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems
- International Journal on Computers and Electric Engineering
- The Journal of VLSI Signal Processing
- Integration, the VLSI Journal, Elsevier
- IEEE Design & Test of Computers

2.5.5 Expertise scientifique nationale et internationale

- Expert international NWO Computer Science Open Competition 2005 (Hollande)
- Agence Nationale de la Recherche – Appel Architectures du futur (2006, 2007)

2.6 Diffusion des connaissances et publications scientifiques

Mes activités de recherches ont donné lieu à un certain nombre de publications scientifiques dont la liste est répertoriée dans cette partie.

2.6.1 Thèse de doctorat

Thèse de Doctorat soutenue le 27 novembre 1997 à l'Université de Nice – Sophia Antipolis
Architecture générique et synthèse des communications pour la conception conjointe de systèmes embarqués logiciel/matériel
Directeur de thèse : Michel Auguin (Directeur de Recherche CNRS)

Composition du jury : Michel Auguin *(Directeur de Recherches CNRS, I3S, Nice – Sophia Antipolis, directeur de thèse)*, Cécile Belleudy *(Maître de Conférences, I3S, Nice – Sophia Antipolis, examinateur)*, Fernand Boéri *(Professeur des Universités, I3S, Nice – Sophia Antipolis, examinateur)*, Emmanuel Gresset *(Ingénieur, VLSI Technology, invité)*, Michel Israel *(Professeur des Universités, Université d'Évry Val d'Essonne, rapporteur, président de jury)*, Eric Martin *(Professeur des Universités, Université de Bretagne Sud, rapporteur)*, Frédéric Rousseau *(Enseignant/chercheur, ESIM, examinateur)*.

Résumé : *Les progrès technologiques constant dans les domaines des ASIC et des cœurs de processeurs permettent d'intégrer des systèmes embarqués de complexité croissante au sein d'un même circuit. Par ailleurs, depuis quelques années l'utilisation de systèmes embarqués devient régulière dans de nombreux domaines d'application. Leurs implémentations nécessitent généralement la mise en œuvre de composants hétérogènes et la vérification de contraintes de conception sévères (e.g. surface, performance, consommation). De plus, les exigences des utilisateurs entraînent la diminution des durées de vie de ces systèmes embarqués. Ainsi, l'importance de la conception conjointe*

logiciel/matériel croît fortement afin d'aider les concepteurs à respecter les contraintes de time to market. La conception des interfaces de communication entre les composants logiciels et matériels de l'architecture garantissant des transferts de données et de contrôle corrects est particulièrement longue et difficile. Ainsi, sur la base d'une architecture générique dédiée aux applications embarquées de télécommunication et de multimédia, nous proposons une méthode de synthèse des communications qui réalise la caractérisation et l'implémentation des communications dans l'architecture finale. Cette méthode prend place après les étapes de partitionnement et d'ordonnancement et peut constituer les fondements d'une approche de conception conduisant à l'intégration logiciel et matériel d'applications de traitement du signal.

2.6.2 Conférences invitées

Guy Gogniat, **Schedulability analysis and MDD**, Third Edition of the summer school MDD4DRES, 4th-8th September 2006, Brest, France
http://www.ensieta.fr/mda/ecoleMDA2006

Guy Gogniat, **Reconfigurable Security Architecture for Embedded Systems**, Cryptarchi 2006, Cryptographic Architectures Embedded in Reconfigurable Devices, 21/24 Juin 2006 Kosice, Slovaquie

Guy Gogniat, **Configurable computing for high-security/high-performance ambient systems**, Université du Massachusetts, Amherst, USA, 10 Mai 2005

Guy Gogniat, **UML Framework for PIM and PSM verification of SDR systems**, Journée scientifique radiocommunications haut débit, ENSTA Paris, 23 Novembre 2005
http://uei.ensta.fr/fr/JSHD/programme.htm

Guy Gogniat, **Software Radio and Dynamic Reconfiguration on a DSP/FPGA Platform**, Université du Massachusetts, Amherst, USA, 15 Avril 2004

2.6.3 Participation à des tables rondes

Panel Security and cryptography
Panélistes : Kris Gaj *(George Mason University, USA)*, Michael Hubner *(Karlsruhe University, Germany)*, Philippe Hoogvorst *(ENST Paris, France)*, Jean François Laporte *(ACTEL)*, Guy Gogniat *(UBS-CNRS, Lorient, modérateur)*.
3rd International Workshop on Reconfigurable Communication Centric System-on-Chips (ReCoSoC'07), 18th-20th June 2007, Montpellier, France
http://www.lirmm.fr/RECOSOC07/

Panel Model and Analysis
Panélistes : Sébastien Gérard *(CEA-List)*, Guy Gogniat *(UBS-CNRS, Lorient)*, Dorina Petriu *(Carleton University, modérateur)*, Bernard Rumpe *(TU Braunschweig)*, Bran Selic *(IBM)*, Jean-Luc Voirin *(THALES)*.
Third Edition of the summer school MDD4DRES 4th-8th September 2006, Brest, France
Schedulability analysis and MDD
http://www.ensieta.fr/mda/ecoleMDA2006

2.6.4 Revues 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

[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

[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

[Gogniat 2000/R] 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

[Freund 1997/R] L. Freund, M. Israel, F. Rousseau, J. M. Berge, M. Auguin, C. Belleudy, G. Gogniat, **A codesign experience in acoustic echo cancellation: GMDFα**, ACM Transactions on Design Automation of Electronic Systems, Vol. 2, N. 4, October 1997.

En cours de révision

[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**, soumise à IEEE Transactions on VLSI Systems Special Section on Configurable Computing Design

2.6.5 Participation à des ouvrages 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

[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

[Gogniat 1997/O] G. Gogniat, M. Auguin, C. Belleudy, **Interface synthesis in embedded HW/SW systems,** Hardware Description Languages and their Applications, Edited by C. Delgado Kloos and E. Cerny, Chapman & Hall, 1997, pages 80 – 82

2.6.6 Publications avec actes et comités de lecture international

[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

[Eustache 2007/CI] Y. Eustache, J-P. Diguet, G. Gogniat, **The Allele Search Lab to Improve Heterogeneous Reconfigurable Platform Design Skills**, *The 2nd International Workshop on Reconfigurable Computing Education*, May 12, 2007, Porto Allegre, Brasil

[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

[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

[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

[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

[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

[Gogniat 2006a/CI] G. Gogniat, W. Burleson, M. O'Malley, L. Bossuet, **IPSec Implementation Project using FPGA and Microcontroller**, *IEEE International Workshop on Reconfigurable Computing Education*, March 1, 2006 Karlsruhe, Germany

[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

[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

[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

[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

[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

[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

[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

[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

[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

[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

[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

[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

[Gogniat1998/CI] G. Gogniat, M. Auguin, L. Bianco, A. Pegatoquet, **Communication Synthesis and HW/SW Integration for Embedded System Design**, *6th International Workshop on Hardware/Software Codesign*, March 15-18, 1998, Seattle, WA, USA

[Bianco 1998/CI] L. Bianco, M. Auguin, G. Gogniat, A. Pegatoquet, **A Path Analysis Based Partitioning for Time Constrained Embedded Systems**, *6th International Workshop on Hardware/Software Codesign*, March 15-18, 1998, Seattle, WA, USA

[Pegatoquet 1998/CI] A.Pegatoquet, E. Gresset, M. Auguin, G. Gogniat, **Software Estimations: A guide to take decisions earlier in the design flow**, *International Conference on Signal Processing Applications and Technologies*, September 13-16, 1998, Toronto, Ontario, Canada

[Gogniat 1997/CI] G. Gogniat, M. Auguin, C. Belleudy, **A generic multi-unit architecture for codesign methodologies**, *5th International Workshop on Hardware/Software Codesign*, March 24-26, 1997, Brauschweig, Germany

[Auguin 1997/CI] M. Auguin, C. Belleudy, G. Gogniat, **Interface Synthesis In Embedded Hardware-Software Systems**, *CHDL'97*, April 20-25, 1997, Toledo, Spain

[Gogniat 1996/CI] G. Gogniat, M. Auguin, C. Belleudy, **ASP/Behavioral Hardware Synthesis of HW/SW Partitioned Systems**, *CESA'96 IMACS Multiconference*, July 9-12, 1996, Lille, France

[Auguin 1996/CI] M. Auguin, C. Belleudy, G. Gogniat, Y. Jegou, **A Multi-granumarity Data Synchronized Architecture for HW/SW Embedded DSP Systems**, *International Conference on Signal Processing Applications and Technologies*, October 7-10, 1996, Boston, MA, USA

[Freund 1996/CI] L. Freund, M. Israel, F. Rousseau, J-M. Bergé, M. Auguin, C. Belleudy, G. Gogniat, **A codesign experience in acoustic echo cancellation: GMDFα**, *9th International Symposium on System Synthesis*, November 6-8, 1996, La Jolla, CA, USA

[Auguin 1995/CI] M. Auguin, C. Belleudy, C. Kieffer, G Gogniat, **Software performance estimation of DSPs for HW/SW partitioning**, *IFIP workshop on Logic and Architecture Synthesis*. December 18-19, 1995, Grenoble, France

[Gogniat 1995/CI] G. Gogniat, M. Auguin, C. Belleudy, **Mixed specific processor/behavioral synthesis of hardware units in a HW/SW codesign environment**, *International Conference on Signal Processing Applications and Technologies*, October 24-26, 1995, Boston, MA, USA

2.6.7 Publications avec actes et comités de lecture national

[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

[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

[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

[Freund 1997/CN] L. Freund, M. Israel, F. Rousseau, J-M. Bergé, M. Auguin, C. Belleudy, G. Gogniat, **Etude de la conception logiciel/matériel d'une application d'annulation d'écho acoustique**, *Colloque CAO de circuits intégrés et systèmes*, 15-17 janvier, 1997, Grenoble

[Gogniat 1997/CN] G. Gogniat, M. Auguin, C. Belleudy, L. Bianco, **Méthode de Co-conception de Systèmes Logiciels/Matériels Embarqués**, *CNRIUT'97*, 14-16 mai, 1997, Blagnac

[ASAR 1996/CN] P. ASAR, **Vers un atelier générique pour la synthèse architecturale bâti autour de CENTAUR**, *4ème Symposium Architectures Nouvelles de Machines*, 7-8 février, 1996, Rennes

[Auguin 1995/CN] M. Auguin, C. Belleudy, F. Boeri, A. Giulieri, G. Gogniat, **Environnement de synthèse et d'applications de traitement du signal**, *GRETSI*, 18-22 septembre, 1995, Juan Les Pains

3. Activités d'enseignement

Depuis 1998, date de ma nomination, j'effectue l'intégralité de mon service d'enseignement au sein du département Génie Industriel et Maintenance de l'IUT de Lorient. La formation Génie Industriel et Maintenance a pour vocation de former des techniciens pluridisciplinaires de façon à leur permettre d'appréhender facilement les nombreuses technologies mises en œuvres dans les entreprises modernes (électricité, mécanique, thermique...). Dans ce cadre, j'interviens dans les enseignements relevant du domaine de l'EEA (Electronique, Electrotechnique et Automatique).

Toutefois, ces dernières années j'ai également enseigné à tous les niveaux de l'enseignement supérieur. J'ai eu de plus quelques expériences internationales, notamment aux Etats-Unis où j'ai participé à des enseignements durant un séjour de recherche de 10 mois à l'Université de Amherst, USA. Depuis plusieurs années je dispense également des séminaires de recherche à des étudiants de dernière année dans les domaines du codesign, des langages de spécification et de la sécurité. Dans la suite je présente chacune de mes expériences d'enseignement depuis mon doctorat. Pour plus de clarté, les volumes d'enseignement ne sont pas précisés pour chaque enseignement mais sont regroupés dans les Tables 3, 4, 5, 6 et 7 dans la suite du document.

3.1 Entre 1994 et 1997 en tant que Moniteur de l'Enseignement Supérieur

3.1.1 IUT GEII de Nice – Sophia Antipolis (1994/1997)

Dans le cadre d'un monitorat de l'enseignement supérieur j'ai enseigné durant les trois années de mon doctorat au sein du département de Génie Électrique et Informatique Industrielle (GEII) de l'IUT de Nice – Sophia Antipolis.

En 1ère année les étudiants de l'IUT GEII s'initiaient aux concepts élémentaires de l'électronique analogique, ils s'intéressaient notamment aux quadripôles passifs, aux transistors bipolaires et à effet de champ, au filtrage des systèmes du 1er ordre ou encore aux systèmes à amplificateurs opérationnels. Cette liste non exhaustive illustre le type d'enseignement dispensé aux étudiants aussi bien en TP qu'en TD. L'acquisition de ces connaissances permettait aux étudiants de réaliser des systèmes analogiques utilisant des composants discrets.

3.1.2 Licence EEA à la Faculté des sciences de l'Université de Nice – Sophia Antipolis (1996/1997)

J'ai enseigné l'électronique numérique en licence Électronique Électrotechnique Automatique (EEA). Les TP d'électronique numérique correspondaient à l'enseignement de la logique combinatoire et séquentielle. Ces TP regroupaient l'apprentissage aux fonctions combinatoires élémentaires (multiplexeur, codeur, unités arithmétiques) et l'étude des systèmes séquentiels synchrones et asynchrones (compteur, automate). Les étudiants disposaient également d'outils d'aide à la conception de circuits programmables du type GAL et FPGA. L'initiation à l'utilisation de ces outils (acquisition graphique des systèmes numériques) et la découverte des langages de programmation (PALASM) des composants programmables avaient pour objectif de permettre aux étudiants de pouvoir s'adapter rapidement aux nouvelles technologies de l'électronique numérique.

3.1.3 Maîtrise d'Informatique à la Faculté des sciences de l'Université de Nice – Sophia Antipolis (1996/1997) – Création de l'enseignement

En Maîtrise d'Informatique j'ai enseigné le langage VHDL. Les TP sur le langage VHDL, permettaient aux étudiants de s'initier aux langages de description du matériel, de découvrir un environnement d'aide à la conception des circuits programmables (FPGA) ainsi que d'approfondir leurs connaissances en architecture des machines. Les TP que j'ai développé portaient sur la modélisation et la simulation d'un processeur de type CISC en langage VHDL. Les étudiants devaient avoir une réflexion sur l'architecture (chemins de données et de contrôles) et sur les mécanismes de fonctionnement du processeur visé avant de programmer le modèle correspondant. La validation du résultat s'effectuait en exécutant un programme de calcul numérique (suite de Fibonnacci) par le processeur modélisé. Ce module d'enseignement permettait aux étudiants en informatique d'une part de mieux comprendre le lien étroit existant entre le jeu d'instructions d'un processeur et les mécanismes d'exécution sous-jacents et d'autre part de s'initier à un langage de description du matériel.

3.2 Entre 1997 et 1998 en tant qu'Attaché Temporaire d'Enseignement et de Recherche

3.2.1 École Supérieure en Sciences Informatiques (1997/1998) – Création partielle de l'enseignement

L'ESSI forme des ingénieurs en Informatique ou dans des domaines des sciences de l'ingénieur fortement utilisateurs de l'Informatique. Dans ce contexte de formation, le cours auquel j'ai participé présentait une vision assez complète du domaine de l'électronique numérique. Dans le cadre de ce cours, j'assurais les TD d'électronique numérique. Ces TD permettaient aux étudiants de maîtriser les fondements de l'électronique numérique et de s'initier aux outils de saisie graphique et de simulation (ALTERA) afin de spécifier et de valider une conception.

J'ai également participé à la création du module d'enseignement relatif à la conception de systèmes numériques où j'avais la charge de mettre en place les cours sur les méthodes et les technologies de conceptions des systèmes numériques ainsi que l'ensemble des TD. Ce cours présentait un panorama des technologies de conception (ASIC, FPGA, circuits standards) et les différentes étapes du flot de conception (descriptions algorithmique, architecturale, logique et physique). L'objectif était de donner aux étudiants les connaissances essentielles afin de pouvoir comprendre la conception et la réalisation d'un système numérique. Le langage VHDL servait de support pour la modélisation et la simulation des circuits à concevoir. Une partie des TD correspondant à ce cours s'effectuait sous la forme d'un mini projet. Il s'agissait de décrire en VHDL un processeur. Ces TD conduisaient les étudiants à évaluer les différentes étapes qui composent le cycle de conception (e.g. fonctionnelle, architecturale). L'outil utilisé (Vsystem de Model Technology) était basé sur la simulation afin de valider la conception réalisée.

La nécessité d'introduire ce module d'enseignement nous avait semblé indispensable afin de compléter la formation des élèves ingénieurs dans le domaine des nouvelles méthodologies de conception des systèmes numériques. La création de ce module s'était fait dans le souci d'apporter aux étudiants les connaissances indispensables afin de répondre dans de bonnes conditions aux exigences des industriels et surtout de pouvoir s'adapter aux nouvelles technologies informatiques qui dans de nombreux domaines ne peuvent être dissociées de l'électronique numérique.

3.2.2 École Supérieure d'Ingénieurs de Nice – Sophia Antipolis

L'ESINSA forme des ingénieurs généralistes en électronique avec comme principaux domaines d'applications, ceux des télécommunications et du traitement du signal. Le cycle de formation se déroule sur cinq années. Les enseignements dont j'avais la charge concernent la conception des systèmes numériques et l'étude des systèmes linéaires continus asservis. En 3ème année ESINSA, les enseignements d'électronique numérique visaient à donner aux étudiants les connaissances relatives aux systèmes synchrones basés sur une unité d'exécution pilotée par une unité de commande. Pour cela, les TP s'articulaient autour de l'apprentissage de la logique séquentielle, des unités de calculs logiques et arithmétiques et des composants programmables (ROM, RAM, PAL).

De plus, l'utilisation du GRAFCET ou du langage ABEL permettait de spécifier les applications traitées. Ces TP aboutissaient par la réalisation d'une machine microprogrammée avec des composants standards.

Les enseignements d'automatique dispensés en 3ème année concernaient les systèmes linéaires continus. Les TP permettaient aux étudiants de se familiariser avec la modélisation des systèmes asservis, la correction et l'étude de la stabilité. Ces TP s'effectuaient avec le logiciel MATLAB et également sur maquettes (e.g. asservissement de température, de position).

Enfin, les TP de VHDL en 4ème et 5ème année avaient pour objectif d'initier les étudiants aux langages de description du matériel et de leur faire sentir les avantages qu'ils apportent au niveau du cycle de conception. Les TP introduisaient les notions de description comportementale, structurelle et flot de donnée aux travers d'exemples. Un mini projet en 5ème année permettait d'approfondir l'enseignement dispensé et de faire la synthèse avec les cours sur les architectures des machines en réalisant un système de calculs à pile.

3.3 Depuis 1998 en tant que Maître de Conférences

3.3.1 IUT GIM de Lorient (1998/aujourd'hui) – Création partielle de l'enseignement

Depuis septembre 1998, je suis en poste à l'IUT de Lorient au sein du département Génie Industriel et Maintenance (GIM) où j'effectue mon service d'enseignement. J'interviens au niveau du département dans les enseignements relevant du domaine de l'EEA (Electronique, Electrotechnique et Automatique).

En électronique analogique et numérique, l'objectif des enseignements est de permettre aux étudiants de concevoir des systèmes de faible complexité mais surtout de pouvoir dialoguer efficacement avec un spécialiste du domaine. Aussi, durant le cours les étudiants voient les éléments fondamentaux du domaine (filtrage, amplification, contre réaction, systèmes combinatoire et séquentiel, technologie des composants et microprocesseur).

En électricité les étudiants découvrent le domaine des signaux continus et alternatifs. Ils découvrent également les théorèmes fondamentaux de l'électricité (Kirchhoff, Millman, Thévenin, Norton…).

En automatique, les étudiants apprennent de façon approfondie les asservissements et leurs mises en oeuvre au sein de systèmes automatisés (automates programmables). J'interviens dans ce cours, principalement sur les aspects concernant le développement de systèmes automatisés. Pour cela, plusieurs manipulations intégrant des automates programmables sont utilisées. En 1ère année les étudiants travaillent avec des automates Siemens et en 2ème année avec des automates Schneider-Electric. Ces manipulations

s'articulent autour de commande d'axe, d'asservissement de niveau de liquide, de commande de moteurs à courant continu ou encore de supervision.

En instrumentation et mesure les étudiants apprennent la mise en oeuvre d'une chaîne d'acquisition, depuis le capteur jusqu'au système d'acquisition (typiquement un automate ou un ordinateur) où le signal est traité. Dans ce cours, des domaines variés sont étudiés, puisque les étudiants abordent aussi bien les différentes technologies de capteurs que la théorie fondamentale de traitement du signal (échantillonnage, analyse fréquentielle ...).

3.3.2 Licence GEII à l'IUP de Lorient (2001/2002) – Création de l'enseignement

En 2001, j'ai assuré l'enseignement de physique des semi-conducteurs en licence GEII à l'IUP de Lorient, l'enseignant assurant habituellement ces cours ayant obtenu une mutation. Dans ce cours les étudiants abordent les points suivants :

- Les propriétés électroniques des matériaux semi-conducteurs (e.g. matériaux intrinsèques, extrinsèques, bande de valence, bande de conduction, matériaux de type P et de type N),
- La jonction PN (e.g. équilibre thermodynamique, polarisation inverse et directe),
- Le transistor bipolaire (e.g. NPN, PNP, régime normal direct, bloqué, saturé)
- Le transistor MOS (e.g. NMOS, PMOS, CMOS, transistor à enrichissement et à déplétion).

L'objectif de ce cours est de permettre aux étudiants de comprendre les phénomènes physiques intervenant dans les composants à semi-conducteurs. Afin d'illustrer les différentes notions abordées durant le cours, j'ai développé trois TP utilisant le logiciel libre Microwind2 (développé par Etienne Sicard à l'INSA de Toulouse). En 2002, je n'ai pas souhaité reprendre la responsabilité de ce module afin de ne pas réduire le temps consacré à mon activité de recherche.

3.3.3 DESS de Mécatronique de Lorient (2000/2004) – Création de l'enseignement

J'ai enseigné dans le DESS de Mécatronique de Lorient où j'effectuai un cours sur les langages de spécification des systèmes hétérogènes. Il s'agissait dans ce cours d'effectuer un panorama des langages permettant d'une part de spécifier les systèmes électroniques numériques mais aussi d'étendre ces aspects aux systèmes hétérogènes (i.e. intégrant des parties analogiques et mécaniques). En effet, durant leur cursus les étudiants apprennent plusieurs langages séparément et n'ont pas une vision d'ensemble qui leur permette d'identifier les langages utilisés au niveau industriel lors de la spécification de systèmes embarqués.

Pour combler ce manque et leur permettre de prendre du recul, j'ai choisi de décomposer le cours en deux parties : la première traite des langages de spécification des systèmes numériques embarqués et la deuxième traite des langages de spécification des systèmes hétérogènes embarqués. Dans la première partie les langages matériel (VHDL, Verilog), les langages logiciel (Assembleur, C, C++, RTOS), les langages flot de données (réseau de process Kahn, SDF) et les langages mixtes (Esterel, SDL, SystemC) sont étudiés. Pour chaque langage une attention particulière est portée sur la façon dont le système ciblé (processeur, ASIC), après compilation ou synthèse, exécute le langage et donc la façon dont la spécification est interprétée. Dans la deuxième partie, les langages VHDL-AMS et Rosetta sont présentés. Ces deux langages ont la particularité d'exprimer des comportements continus, ce qui permet de spécifier des systèmes analogiques et mécaniques mais également tous systèmes se modélisant à l'aide d'équations différentielles. Un point particulier concernant ces langages est leurs possibilités intrinsèques d'exprimer des contraintes non fonctionnelles (surface, consommation).

3.3.4 DEA d'électronique de Lorient (2000/2004) – Création partielle de l'enseignement

J'ai enseigné dans le DEA d'électronique de Lorient où j'ai effectué d'une part un cours sur les algorithmes pour la conception VLSI (entre 2002 et 2004) et d'autre part un séminaire concernant les méthodologies de conception conjointe logiciel/matériel (entre 2000 et 2004).

Le cours sur les algorithmes pour la conception VLSI vise à donner aux étudiants les concepts fondamentaux de modélisation d'un problème de conception et d'évaluation de la complexité de résolution. Ce cours propose un panorama des algorithmes classiques existant dans le domaine de la conception de circuits intégrés afin d'aboutir soit à une solution optimale soit à une solution respectant une fonction de coût. Les différents algorithmes sont détaillés et comparés afin de montrer leurs avantages respectifs. J'ai également dispensé ce cours à l'ENST de Brest et à Supelec à Rennes pendant la même période.

Le séminaire sur le codesign aborde chaque aspect intervenant dans le flot de conception de systèmes complexes (spécification, estimations logiciel/matériel, partitionnement, synthèse des communications, validation). Il présente également les travaux que nous développons au LESTER dans ce domaine ainsi que l'outil associé Design Trotter. Ce séminaire a été développé avec Jean Philippe Diguet à l'époque Maître de Conférences à l'Université de Bretagne Sud.

3.3.5 ENIS de Sfax, Tunisie (2002/2003) – Création de l'enseignement

Dans le cadre d'une collaboration entre l'Université de Bretagne Sud et l'Ecole Nationale d'Ingénieurs de Sfax (ENIS) j'ai dispensé un séminaire sur les langages de spécification des systèmes embarqués. Ce séminaire reprend l'enseignement effectué dans le cadre du DESS de Mécatronique mais en insistant davantage sur les aspects intégrations de composants et donc détaille les contraintes au niveau des interfaces.

3.3.6 Licence IMSA de l'IUT de Lorient (2006/2007) – Création de l'enseignement

En Licence Ingénierie et Maintenance des Systèmes Automatisés (IMSA) de l'IUT de Lorient j'ai effectué un module cours/TD sur les chaînes d'acquisision de mesure. Ce module présente les différents types de capteurs utilisés au niveau des systèmes automatisés puis détaille la chaîne d'acquisition (conditionneur, filtrage, amplification, échantillonnage, numérisation…). Ce module permet de donner aux étudiants un première approche de l'acquisition de données et des interfaces associées.

3.3.7 Master Recherche Electronique de Lorient (2006/2007) – Création de l'enseignement

J'ai dispensé un séminaire sur la sécurité des systèmes numériques aux étudiants du master recherche Electronique. Ce séminaire présente les notions essentielles en cryptographie avant d'adresser les besoins en sécurité des systèmes numériques. Les différents types d'attaques sont présentés afin de mettre en évidence les contres mesures actuelles aussi bien au niveau logique, qu'architectural et système. Les solutions à base de processeur (Trusted Computing) sont également présentées. Ce domaine de compétence, auparavant réservé aux informaticiens, est actuellement en pleine expansion au niveau numérique, aussi il est important de sensibiliser les étudiants à cette problématique.

3.3.8 Master Mathématiques et Applications de Vannes (2006/2007) – Création de l'enseignement

J'ai également dispensé un séminaire sur la sécurité aux étudiants du master Mathématiques et Applications de Vannes. Toutefois j'ai orienté la présentation sur les algorithmes de cryptographie afin d'expliquer aux étudiants les opérations mises en oeuvre dans les algorithmes de chiffrement symétriques (AES, DES) et asymétriques (RSA, ECC) et dans les algorithmes de hachage. Ce séminaire leur permet de découvrir un autre domaine d'application des mathématiques de plus en plus important.

3.3.9 ENSEIRB à Bordeaux (2006/2007) – Création de l'enseignement

Le séminaire à l'Ecole Nationale Supérieure d'Electronique, Informatique & Radiocommunications de Bordeaux (ENSEIRB) portait également sur la cryptographie et s'inscrivait dans un module d'enseignement dédié à la sécurité des systèmes embarqués. Dans ce séminaire après avoir présenté les algorithmes de cryptographie je me suis attaché à expliquer de façon approfondie leur mise en œuvre sur des composants matériel et logiciel. Les performances pouvant être obtenues ont également été présentées afin de voir les possibilités des technologies actuelles.

3.3.10 ENSIETA à Brest (2006/20007) – Création partielle de l'enseignement

Le module d'enseignement dispensé à l'Ecole Nationale Supérieure d'Ingénieurs de Brest (ENSIETA) concerne le codesign. Ce module propose aux étudiants de découvrir les flots de conception des systèmes embarqués hétérogènes. Ce module décrit également les nouvelles architectures de processeurs qui sont de plus en plus dédiés à des classes d'applications (e.g. processeur Tensilica). Ce module est accompagné de manipulations qui permettent aux étudiants d'explorer différentes solutions architecturales (processeur seul, processeur + coprocesseur, processeur + accélérateur). Pour cela une plateforme à base du processeur NIOS de chez Altera a été utilisée. Ces implémentations permettent aux étudiants de voir les différentes facettes de la conception d'un système hétérogène (langage C pour le processeur, langage VHDL pour les parties matérielles et intégration). La mise en place de ce module a été réalisée avec Jean Philippe Diguet, chercheur au laboratoire. La pédagogie de cet enseignement a été très appréciée des étudiants et a fait l'objet d'une publication en conférence internationale traitant de l'enseignement et des technologies reconfigurables [Eustache 2007/CI].

3.3.11 Université du Massachusetts, Amherst, USA (2004/2005) – Création partielle de l'enseignement

En 2005 durant un séjour de recherche aux Etats-Unis j'ai participé aux modules ECE354 et ECE559/659 au sein du Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA, U.S.A. Le module ECE354 correspond à des labs et permet aux étudiants de 3ème année (junior) de maîtriser les systèmes à base de microcontrôleur et de FPGA. Les modules ECE559/659 correspondent à des projets et permettent aux étudiants d'étudier les systèmes complexes embarqués. Plusieurs applications concernant le domaine de la sécurité ont été traitées.
J'ai également mis en place et suivi le projet Honor Section qui vise à proposer, aux meilleurs étudiants du module ECE354, de réaliser, en plus de leur travail, un projet supplémentaire. 7 étudiants avaient été sélectionnés pour participer au projet Honor Section. Le projet que j'ai proposé traitait de la mise en œuvre du protocole IPSec sur une

plateforme composée d'un microcontrôleur PIC, d'un PC et d'un composant reprogrammable de chez Altera. Les étudiants ont développé un démonstrateur validant l'ensemble de l'étude. Le travail a été très apprécié des étudiants et a fait l'objet d'une publication en conférence internationale traitant de l'enseignement et des technologies reconfigurables [Gogniat 2006a/CI].

Il est a noter que cette expérience d'enseignement a été très positive également pour moi dans la mesure où j'ai mis en place un suivi de projet proche des expériences que j'ai eu en recherche sur des projets nationaux. Cela m'a permis d'accompagner les étudiants tout en leur laissant une grande part d'autonomie et de proposition.

3.3.12 Cours multimédia (2004/2005) – Création de l'enseignement

En 2005 je me suis intéressé aux nouvelles technologies afin de proposer un module d'enseignement numérique. J'ai développé pendant mon séjour de recherche aux Etats-Unis une maquette d'enseignement de ce type où l'étudiant dispose d'un module d'enseignement entièrement numérique. Il a donc la possibilité de suivre un cours, des TD et de préparer des TP à travers une explication filmée du TP à réaliser et une démonstration des outils de développement associés. Cette maquette d'enseignement présente un module d'électronique numérique (bascules, compteurs).

Les Tables 3, 4, 5, 6 et 7 résument l'ensemble des enseignements effectués ces dernières années aux différents niveaux d'enseignement (de L1 à L3 et M1 à M2) et précisent les volumes annuels correspondants.

Table 3 • Résumé des enseignements effectués au niveau L1.

Année (niveau 1)	94/95	95/96	96/97	97/98	98/99	99/00	00/01	01/02	02/03	03/04	04/05	05/06	06/07
Electricité												**IUT GIM** *12h TD/an*	
Automatique						**IUT GIM** *12h TD/an 24h TP/an*						**IUT GIM** *12h TD/an*	
Electronique numérique				**ESSI** *14h TD*		**IUT GIM** *12h CM/an 12h TD/an 24h TP/an*							
Electronique analogique		**IUT GEII** *12h TD/an 48h TP/an*				**IUT GIM** *24h TD/an 24h TP/an*						**IUT GIM** *24h TD/an*	
Technologie des composants						**IUT GIM** *6h CM/an 6h TD/an 12h TP/an*							
Microprocesseurs											**IUT GIM** *4h CM/an 6h TD/an 8h TP/an*		

Table 4 • Résumé des enseignements effectués au niveau L2

Année (niveau 2)	94/95	95/96	96/97	97/98	98/99	99/00	00/01	01/02	02/03	03/04	04/05	05/06	06/07
Conception de systèmes numériques				**ESSI** *12h CM 12h TD*									
Instrumentation & mesure							**IUT GIM** *12h TD/an*						
Langage C							**IUT GIM** *24h TP/an*					**IUT GIM** *24h TP/an*	
Automatique							**IUT GIM** *24h TP/an*						**IUT GIM** *24h TP*

Table 5 • Résumé des enseignements effectués au niveau L3

Année (niveau 3)	94/95	95/96	96/97	97/98	98/99	99/00	00/01	01/02	02/03	03/04	04/05	05/06	06/07
Automatique				**ESINSA** *24h TP*									
Electronique numérique				**ESINSA** *24h TP*									
Electronique numérique			**Lic. EEA** *24h TP*										
Physique des semi-conducteurs								**Lic. EEA** *10h CM 10h TD 12h TP*					
Acquisition des données													**Lic. IMSA** *6h CM 6h TD*

Table 6 • Résumé des enseignements effectués au niveau M1

Année (niveau 4)	94/95	95/96	96/97	97/98	98/99	99/00	00/01	01/02	02/03	03/04	04/05	05/06	06/07
Langage VHDL			**M. Info.** *24h TP*	**ESINSA** *24h TP*									

Table 7 • Résumé des enseignements effectués au niveau M2

Année (niveau 5)	94/95	95/96	96/97	97/98	98/99	99/00	00/01	01/02	02/03	03/04	04/05	05/06	06/07
Langages de spécification des systèmes embarqués							DESS mécatronique *8h SM/an*						
Langages de spécification des systèmes embarqués									ENIS *4h SM*				
Codesign							DEA élec *4h SM/an*						
Codesign													ENSIETA *5h CM*
Algorithmes pour la conception VLSI								DEA élec *12h CM/an* UBS/ENST Bretagne/Supelec Rennes					
Sécurité des systèmes embarqués													M.R. élec *3h SM*
Cryptographie													ENSEIRB *4h SM*
Cryptographie													M.P. math *2h SM*
Langage VHDL				ESINSA *24h TP*									

3.4 Bilan et réflexion sur la fonction d'enseignant

Après plus de dix ans d'enseignement il est intéressant d'analyser la fonction d'enseignant au sein de l'enseignement supérieur. Son rôle premier est de transmettre des connaissances mais aussi et cela me semble de plus en plus important chaque année d'apprendre à apprendre tant le domaine des compétences s'élargit. Il devient en effet difficile aujourd'hui d'adresser l'ensemble des compétences durant les années d'études, aussi il est essentiel de préparer les étudiants à appréhender de façon autonome de nouveaux domaines et de nouvelles technologies. Cela peut sembler naturel mais sa mise en œuvre est loin d'être triviale. De nombreux paramètres interviennent dans ce processus d'apprentissage, la personnalité de l'étudiant, sa maturité, sa motivation et ses connaissances.

Mon activité d'enseignement s'est principalement déroulée avec des étudiants d'IUT et de Master Recherche. Ce grand écart est intéressant car la façon dont l'enseignement est appréhendé par les étudiants est très différente à ces deux extrémités. L'enseignement tel qu'il est construit aujourd'hui est principalement basé sur une transmission linéaire et essentiellement passive des connaissances. Ce schéma à mon sens a atteint ses limites aujourd'hui et il est important de repenser notre façon d'enseigner. Cela est d'autant plus marqué avec des étudiants à la sortie du bac pour qui la notion de connaissances reste encore assez floue.

Le schéma classique cours magistraux, travaux dirigés et travaux pratiques me semble dans de nombreux domaines peu efficace et peu motivant pour un public d'étudiants habitué à une certaine passivité. Il est essentiel d'arriver à impliquer l'étudiant dans son propre apprentissage afin de développer chez lui le sens du questionnement, de la remise en question mais aussi afin de favoriser son autonomie, sa motivation et encourager sa force de proposition. C'est un exercice difficile et il reste encore beaucoup de travail à faire dans ce domaine.

Les enseignements à partir de projets, qui se multiplient dans l'enseignement supérieur, correspondent à une réponse intéressante à cet enjeu. Cependant pour que les projets atteignent les objectifs pédagogiques fixés, ces derniers doivent être accompagnés d'un encadrement régulier. Le rôle de l'encadrant est alors d'amener les étudiants à analyser leurs solutions afin qu'ils développent leur sens critique, il est aussi de les accompagner et d'arbitrer leurs choix afin de progressivement converger vers une solution réaliste et satisfaisant les contraintes définies. Il me semble que nous n'avons pas encore fait l'analyse complète des possibilités offertes par les projets et que nous avons encore du chemin à parcourir afin de mieux organiser le processus d'apprentissage autour de ces derniers.

Je mène depuis quelques années des expériences pédagogiques à travers des projets et les résultats sont intéressants. Ces projets ont été principalement menés avec des étudiants de quatrième et cinquième années mais je suis convaincu que les principes peuvent être étendus à d'autres niveaux d'étude. Le principe de base que je me suis fixé, est de rencontrer les étudiants de façon très régulière, en général une fois par semaine, dans une réunion de groupe de deux heures maximum où tous les projets dont j'assure l'encadrement sont présents (en général 2 ou 3 projets). L'objectif de ces réunions est de demander aux étudiants de présenter l'avancement de leurs travaux et de mener une réflexion collective sur les difficultés rencontrées dans les différents projets. Cela impose aux étudiants de présenter, en utilisant des supports adaptés, leurs travaux, donc de prendre la parole, d'être précis et de recevoir de la part des autres étudiants des questions. Cet exercice difficile au début devient rapidement bénéfique pour tous les étudiants qui en plus de leur projet voient d'autres domaines et participent de façon indirecte à l'élaboration des différentes solutions. Ces réunions soulignent également que l'encadrant n'est pas omniscient mais que les solutions se construisent de façon collaborative. Afin d'étendre le bénéfice des projets je demande aux étudiants de mettre en œuvre un site WEB où ces derniers centralisent les

développements qu'ils ont fait et proposent un certain nombre de liens vers des domaines connexes à leur projet. La dissémination des résultats me semble également très importante. Lorsque cela est possible j'associe ces présentations aux réunions hebdomadaires de mon groupe de recherche afin que les étudiants participent à la vie et aux travaux d'un groupe complet. Cela est motivant et très enrichissant pour eux.

D'autres pistes très intéressantes doivent être explorées. L'environnement numérique de travail a fait son apparition depuis quelques années. Certaines universités utilisent de façon approfondie ce support. Durant une année sabbatique aux Etats-Unis j'ai eu l'opportunité de me familiariser avec ces nouvelles technologies. Le support multimédia me semble très intéressant et prometteur pour les étudiants afin de développer chez eux une certaine autonomie dans leur travail. Ce support permet aux étudiants de façon asynchrone d'accéder à un cours et de progresser à leur rythme. L'Université de Bretagne Sud commence à développer ce type de support et je suis convaincu que l'avenir verra son utilisation se renforcer afin de permettre un accès aux connaissances de façon plus large. J'ai eu l'occasion de développer avec des étudiants d'IUT et avec certains collègues des maquettes de support multimédia. Le résultat est intéressant et il apparaît clairement qu'une utilisation complémentaire du support multimédia avec un enseignement plus traditionnel serait très bénéfique. L'enseignement évolue ainsi que les étudiants, aussi il est essentiel dans notre métier de régulièrement repenser notre façon d'enseigner. Les évolutions technologiques nous ont ouverts et nous ouvrent des nouvelles possibilités quant à la transmission du savoir, il est important de s'y intéresser afin notamment de faire face aux défis de la mondialisation de l'enseignement.

L'évaluation des enseignements me semble également une voie importante. Comme indiqué précédemment il est essentiel que l'étudiant participe activement à son apprentissage. Une façon de l'impliquer davantage est de recueillir à la fin de chaque module d'enseignement une évaluation de l'enseignement par chaque étudiant. Ce retour doit être certes utilisé avec un certain recul mais permet de corriger et d'adapter l'enseignement en fonction de la perception des étudiants. Cette démarche de qualité ne peut être que bénéfique pour l'université et il me semble important d'initier une démarche dans ce sens. Depuis un an je suis co-responsable du Master recherche en électronique à l'Université de Bretagne Sud et j'ai initié l'évaluation des enseignements. Cette expérience intéressante est très bien accueillie par les étudiants et les enseignants car elle permet aux premiers d'exprimer leur analyse concernant les enseignements dispensés et aux seconds de mieux organiser leurs enseignements.

Une autre mission essentielle de l'enseignant est de participer au bon fonctionnement de l'université en s'impliquant dans les structures d'enseignement. Mon expérience dans ce domaine est plus limitée, toutefois depuis un an je suis co-responsable du Master Recherche en électronique à l'Université de Bretagne Sud. J'ai eu la volonté de prendre cette responsabilité pédagogique suite à mon séjour aux Etats-Unis où j'ai découvert une pédagogie différente de celle que je connaissais en France. Suite à cette expérience il m'est apparût plus clairement qu'il était possible de mettre en place des expériences originales au niveau de l'organisation des enseignements. J'ai donc initié une démarche auprès des étudiants et des collègues enseignants du Master Recherche afin de créer les conditions propices à l'apprentissage du métier de la recherche. Cette expérience a été basée sur la mise en place systématique de projets associés à chaque module d'enseignement. Ces projets en autonomie permettaient aux étudiants d'aller au delà des modules d'enseignement. Les étudiants ont également été associés au laboratoire LESTER afin qu'ils découvrent et participent à la vie d'une structure de recherche. L'expérience a été très riche et les étudiants ont développé une réelle force de proposition et une motivation forte dans leur travail. Certains ajustements ont été discutés avec les étudiants et les enseignants afin d'améliorer le fonctionnement pour l'année universitaire à venir. Cette démarche de qualité est importante et permet de mettre en place une structure d'enseignement efficace et répondant aux attentes des industriels et des besoins en recherche.

Pour conclure cette discussion concernant la fonction d'enseignant dans l'enseignement supérieur, il est clair que cette mission est riche et que de nombreuses pistes d'amélioration sont possibles. Il est important d'être à l'écoute des évolutions technologiques, des évolutions des métiers afin de permettre aux étudiants de progresser et d'avoir les connaissances pérennes et opérationnelles nécessaires afin de débuter leur carrière professionnelle. Cette fonction d'enseignant est un juste équilibre à trouver entre les deux missions d'un enseignant chercheur à savoir la recherche et l'enseignement. Il est important de trouver cet équilibre afin de pouvoir décupler l'efficacité de l'université, c'est par la formation que nous préparons les étudiants à devenir des professionnels de haut niveau et donc en particulier des chercheurs de haut niveau.

4. Responsabilités collectives, animations et projets scientifiques

Dans cette section je présente les différentes responsabilités collectives que j'ai eues depuis ma nomination à l'Université de Bretagne Sud. Je présente également ma participation à l'animation scientifique au niveau national et international. Enfin, je présente les différents projets de recherche auxquels j'ai participé.

4.1 Au niveau de l'Université de Bretagne Sud

4.1.1 Au sein du département Génie Industriel et Maintenance

J'ai été responsable de la promotion du département Génie Industriel et Maintenance de l'IUT de Lorient pendant la période 2000/2003. Je me suis occupé d'organiser la représentation du département dans les différents salons et lycées. Je me suis également occupé de définir les nouveaux supports et moyens de promotion afin de présenter le département GIM aux étudiants. Dans un contexte où le recrutement des étudiants dans le domaine scientifique devient préoccupant il est important de faire connaître nos formations et de présenter les perspectives qu'elles proposent. Bien que n'étant plus responsable de la promotion je continu à m'investir pour faire connaître le département lors des salons étudiants et des journées portes ouvertes.

Je suis membre de la commission recherche au sein de l'IUT qui à pour rôle de réfléchir sur les profils d'enseignement et de recherche des futurs enseignants/chercheurs. Elle mène également une réflexion sur le métier d'enseignant/chercheur au sein d'un IUT.

Je suis responsable des enseignements d'électronique numérique en 1ère année. A ce titre je m'occupe également des TP d'électroniques numériques.

4.1.2 Au sein du Master Recherche Electronique

Depuis 2006/2007 je suis co-responsable avec Emmanuel Boutillon du master recherche Electronique de l'Université de Bretagne Sud. Cette fonction me tient particulièrement à cœur car il me semble essentiel de bien former les étudiants aux métiers de la recherche afin d'attirer davantage de jeunes vers cette voie stratégique pour notre pays. En effet nous sommes confronté à un désintérêt des voies à vocation recherche par une méconnaissance forte des étudiants des possibilités offertes suite à ce type de diplôme.

Aussi mon objectif est de faire connaître aux étudiants cette formation et de mettre en avant la qualité des enseignements qu'elle offre. Il est également très important de montrer les richesses d'une telle formation qui ouvre bien plus de voies qu'uniquement le doctorat comme beaucoup d'étudiants le pensent. La promotion est donc très importante et je m'attache pour le moment à le faire en local mais il faut étendre cette démarche au niveau nationale et international.

Il est également essentiel de définir les bases d'une formation préparant aux métiers de la recherche, aussi cette année j'ai mis en place plusieurs points me semblant très importants afin de favoriser chez les étudiants l'acquisition d'une autonomie scientifique, afin de développer leur force de proposition et leur créativité, afin de développer leur regard critique et constructif sur des travaux existants, afin d'exercer leur capacité d'échange, de présentation et d'argumentation de leurs travaux face à un auditoire et afin de les préparer à écrire des articles scientifiques. Cette liste n'est pas exhaustive mais présente des notions à

mon sens fondamentales. Aussi en accord avec Emmanuel Boutillon nous avons considéré les étudiants du master comme membres du laboratoire, ils ont donc participé aux réunions de laboratoire ce qui leur a permis de mieux connaître la vie d'une structure de recherche. Ils ont également assisté à l'ensemble des soutenances de thèse afin d'appréhender la finalité d'un travail de recherche. Afin de développer leur aptitude à mener un travail de recherche j'ai demandé à chaque enseignant d'associer leur module à un projet de recherche ou de développement.

J'ai également mis en place des feuilles d'évaluation que remplissent les étudiants à la fin de chaque module d'enseignement afin d'améliorer l'interaction entre l'étudiant et le module enseigné. Cette démarche de qualité me semble importante afin de faire connaître et améliorer la façon dont nous dispensons nos enseignements.

Il existe encore beaucoup de pistes de réflexions possibles aussi bien au niveau de l'enseignement numérique (multimédia) que concernant une meilleure reconnaissance de ce type de formation auprès des étudiants car il me semble essentiel d'offrir aux étudiants des formations favorisant leur créativité, leur épanouissement mais aussi leur demandant de fournir un effort conséquent afin de pouvoir progresser et faire progresser notre système de formation. C'est à travers eux que nous construisons notre avenir.

Le bilan de cette année est très positif de la part des étudiants qui ont énormément appris et apprécié la politique pédagogique développée.

4.1.3 Au sein du LESTER

Depuis 2001, je suis membre de la commission de spécialistes 61éme et 63ème sections.

J'ai assuré la mise en place du site WEB du LESTER (avec l'aide de Jérome Guiban, ingénieur au LESTER en 1999 et 2000) et j'ai été responsable de sa mise à jour jusqu'en 2004. En effet la dissémination me semble important et je pense qu'aujourd'hui nous ne le faisons pas encore assez. Le portail numérique est notre vitrine et je suis convaincu qu'à travers lui nous transmettons une image de notre activité.

4.1.4 Au sein de l'Université de Bretagne Sud

J'ai été membre du Conseil de l'UFR Sciences et Sciences de l'Ingénieur en 2003/2004.
Je participe à la mise en place des relations internationales pour l'enseignement et la recherche (Canada, Portugal).
Je suis membre du conseil du département Sciences et Techniques de l'UFR Sciences et Sciences de l'Ingénieur (2007/aujourd'hui).

4.2 Au niveau national

Depuis ma nomination je me suis toujours attaché à participer aux réseaux scientifiques car cela me semble très important de connaître la communauté nationale. C'est grâce aux échanges que nous initions de nouvelles connaissances et de nouvelles collaborations.

J'ai participé au réseau thématique pluridisciplinaire du CNRS n°22 System-On-Chip au sein de l'action spécifique n°28 Architecture Reconfigurable Dynamiquement (AS ARD) de 2002 à 2003 : participation, présentations et responsable avec Bernard Pottier de la rédaction des aspects logiciels dans le document de synthèse.

J'ai participé à l'équipe projet multi-laboratoires n°47 du CNRS intitulé POMARD (Projet Outils Méthodes et Architectures pour la Reconfiguration Dynamique) animée par Didier Demigny de 2003 à 2004. J'ai animé le thème Outils de cette équipe projet.

Je suis membre du GdR ISIS (Information, Signal, Images et ViSion), et participe au Thème C, Adéquation Algorithme Architecture.

Je suis membre du GdR SoC-SiP (System On Chip - System In Package).

J'ai participé au Workshop du Réseau Thématique Pluridisciplinaire System On Chip du STIC CNRS, 22/25 Septembre 2002, Aussois. J'ai également participé au Workshop du Réseau Thématique Pluridisciplinaire System On Chip du STIC CNRS, 16/19 Mai 2004, La Londe les Maures

J'ai animé la mise en place de l'extension au niveau européen des conférences nationales JFAAA et READ. Le futur workshop DASIP 2007 est actuellement prévu en novembre 2007. Je suis *program co-chair* du workshop. 14 pays sont représentés à travers le comité de programme. Actuellement 6 sessions spéciales sont programmées : deux sur le thème méthodologie pour l'AAA, une sur les nouvelles architectures, une sur les capteurs intelligents, une sur les applications de l'AAA, une sur les projets européens et une sur les méthodologies et les perspectives des outils industriels. D'autres sessions plus générales sont également imaginées et 5 présentations invitées sont prévues. Plusieurs soutiens sont actuellement demandés (GDR ISIS, GDR SoC SiP, CNRS, Laboratoires, Industriels du domaine...). Enfin le workshop est en partenariat avec Eurasip et un numéro spécial suite à la conférence est prévu. D'autres journaux vont être contactés prochainement.

4.3 Participation à des collaborations scientifiques et à des contrats d'études

Depuis 1994, je me suis toujours impliqué dans des collaborations avec d'autres équipes universitaires et industrielles. Ces collaborations me semblent en effet essentielles afin d'enrichir et d'approfondir les réflexions scientifiques. Elles sont un moyen privilégié de rencontrer d'autres équipes, travaillant parfois sur des domaines connexes, et aussi d'élargir notre propre champ de connaissance. En 2002, mon désir d'établir des relations internationales se concrétise puisque plusieurs projets s'effectuent avec des universitaires étrangères (ENIS en Tunisie, University of Massachusetts aux Etats-Unis). Voici les projets où actions thématiques dans lesquels je me suis engagé ces dernières années.

4.3.1 Collaborations Académiques Internationales

Université de Sfax (Tunisie)

[PROSYR2006] 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, G. Gogniat, J-P. Diguet, J-L. Philippe (Lorient), ENIS, M. Abid, M. Ben Jemaa (Sfax)
Objet : Développement d'un environnement de conception pour les systèmes sur puce.
Contribution du LESTER : Définition d'une méthodologie de conception pour les systèmes sur puce à partir d'une spécification en langage C. Développement de l'outil Design Trotter.
Budget : 81K€ (LESTER) Accueil de doctorants et chercheurs étrangers. Accueils réguliers du Professeur Mohamed Abid sur des postes de Professeur Invité.

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, G. Gogniat, J-P Diguet (Lorient), VSPG, W. Burleson, R. Tessier (Amherst)
Objet : Sécurisation des applications basées sur le processeur NIOS.
Contribution du LESTER : Définition d'une technique permettant de garantir la confidentialité et l'intégrité des échanges entre la mémoire et le processeur NIOS. Extension de l'OS afin de mettre en œuvre des primitives de sécurité (sauvegarde de contexte sécurisée).
Budget : 4K€ (LESTER) Séjour d'un doctorant à l'Université du Massachusetts (durée du séjour de 3 mois). Accueil du Professeur Russell Tessier pendant un mois en 2007 sur un poste de Professeur Invité.

[SANES 2005] Projet SANES

Security Architecture for Embedded Systems
Type : DGA ERE (Etude et Recherches à l'Etranger)
Durée : 2004/2005
Partenaires : LESTER, G. Gogniat (Lorient), VSPG, W. Burleson (Amherst)
Objet : Définition d'une architecture reconfigurable dynamiquement permettant de renforcer la sécurité du système contre les attaques matérielles du type canaux cachés.
Contribution du LESTER : Proposition d'une architecture système et validation sur l'algorithme de chiffrement AES 128 bits.
Budget : 40 K€ (LESTER) Séjour de recherche de 10 mois à l'étranger (USA).

[SecureFPGA 2004] Projet SecureFPGA

Bitstream security for SRAM based FPGAs
Type : Projet sur fond propre
Durée : 2003/2004
Partenaires : LESTER, G. Gogniat (Lorient), VSPG, W. Burleson (Amherst)
Objet : Sécurisation du chargement du bitstream pour les FPGA de type SRAM.
Contribution du LESTER : Définition d'une technique de chargement de bitstreams basée sur la reconfiguration dynamique et utilisant des algorithmes de chiffrement afin de garantir la confidentialité. Proposition d'une approche de partitionnement d'IP orientée sécurité.
Budget : 4K€ (LESTER) Séjour d'un doctorant à l'Université du Massachusetts (deux séjours, un de deux mois et un de un mois).

[DARSoC 2003] Projet DARSoC

Dynamic Adaptative and Reconfigurable System on Chip
Type : Projet sur fond propre
Durée : 2002/2003
Partenaires : LESTER, G. Gogniat, J-L. Philippe (Lorient), VSPG, W. Burleson (Amherst)
Objet : Développement d'une méthodologie d'exploration pour les architectures reconfigurables. Application à l'architecture en tuile aSoC.
*Contribution du LESTER : Définition d'une méthodologie d'exploration générique pour les architectures reconfigurable*s

Budget : non contractuel, Accueil du Professeur Wayne Burleson pendant 15 jours en 2002.

4.3.2 Projets Européens

[AETHER 2008] Projet AETHER

Self-Adaptive Embedded Technologies for Pervasive Computing Architectures
Type : Projet Européen IST-FET (4th call ACA / FP6)
Durée : 2006/2008
Partenaires : CEA/LIST (FR, Leader), CNRS (I3S/LESTER), Univ. Amsterdam (UVA, NL), Univ. Karlsruhe (ITIV, DE), Imperial College of London (UK), Univ. Politècnica de Catalunya (UPC, ES), Thales Research (FR), VTT (FI), ATMEL (GR), INTRACOM (GR), Univ. Prague (UTIA, CZ), ACIES (administration, FR), Università della Svizzera Italiana (USI/ALARI, CH), University of Hertfordshire (UK).
Objet : Self-Adaptive Embedded Technologies for Pervasive Computing Architectures. Nouveaux concepts aux niveaux logiciel, architecture et RTOS pour les systèmes électroniques ambiants et reconfigurables à l'horizon 2030.
Contribution du LESTER : Définition d'une technique et d'une politique d'adaptation au niveau d'un système intégré au sein d'un OE (Operating Environment), il s'agit de répartir en ligne des applications sur un ensemble d'architectures reconfigurables dont le nombre et les propriétés sont variables. Le LESTER doit fournir un simulateur SystemC mettant en oeuvre les nouveaux concepts proposés en liaison avec les autres sous projets situés au niveau de la spécification des applications et des modèles d'architectures.
Budget : 4M€ (dont 251 K€ pour le CNRS réparti à parts égales entre le LESTER et I3S). Financement de thèse.

4.3.3 Contrats de Recherche Publique

[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 (Leader), THOMSON, ENSIETA, IETR/Supelec, IRISA, LESTER, SODIUS.
Objet : Modélisation et spécialisation de plates-formes et composants MDA (Modèle driven architecture) pour la mise en œuvre de systèmes sur puce.
Contribution du LESTER : Conception de RSOC à partir de technique MDA (poursuite du projet RNTL A3S).
Budget : 1,28 M€ (dont 111K€ pour le LESTER). Financement de thèse.

[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
Objet : Technologies Reconfigurables, Intégrité et confidentialité des informations. Sécurité au niveau logique contre les attaques par canaux cachés (logique sécurisée, générateur aléatoire). Sécurité au niveau architectural et système.

Contribution du LESTER : Conception d'une architecture reconfigurable capable de s'adapter aux besoins en sécurité. Protection des données entre le processeur et la mémoire. Minimisation de l'overhead dû à la sécurité.
Budget : 400K€ (dont 85 K€ pour le LESTER). Financement de Post-doc.

[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
Objet : Identifier les futurs enjeux liés à la conception de systèmes reconfigurables au niveau des outils de conception, des méthodologies de conception et des architectures reconfigurables gros grain et grain fin.
Contribution du LESTER : Identification des flots de conception et proposition de solutions possibles pour l'exploration et la synthèse.
Budget : 20K€ (dont 5K€ pour le LESTER).

[A3S 2005] Projet A3S

Adéquation Architecture - Application Système
Type : Projet RNRT
Durée : 2003/2005
Partenaires : THALES Communications, SOFTEAM, Mitsubishi Electric ITE, LESTER
Objet : Définition d'une méthodologie de conception MDA pour les applications radio logicielles.
Contribution du LESTER : Définition d'une méthodologie de conception et des outils associés ainsi que du profile UML pour la radio-logicielle (algorithme et architecture), interface et outil logiciels pour la validation non fonctionnel et fonctionnelle (temps réel).
Budget : 180K€ (LESTER). Financement de thèse.

[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
Objet : Définition d'une architecture reconfigurable et d'un environnement de conception pour le partitionnement logiciel/matériel reconfigurable.
Contribution du LESTER: Définition d'un outil de caractérisation au niveau algorithmique, d'exploration et d'estimation sur cible de type FPGA (Environnement Design Trotter).
Budget : 130 K€ (LESTER). Financement ingénieur.

[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
Objet : Définition d'une méthodologie de conception et des outils associés.
Contribution du LESTER : Définition d'une méthodologie d'estimation de performance pour les architectures reconfigurables grain fin.

Budget : 15 K€ (LESTER)

4.3.4 Contrat de Recherche Privée

[D2ASR 1999] Projet D2ASR

Démodulateur Aveugle pour les Applications de Surveillance de Réseau commuté
Type : Contrat industriel
Durée : 1999
Partenaires : LESTER, SERPE-IESM
Objet : Conception d'un prototype permettant l'observation d'une ligne téléphonique sur laquelle transite des échanges de données entre modems afin de localiser un éventuel défaut et de le diagnostiquer.
Contribution du LESTER : Analyse des différents normes V21, V22, V22bis, V32, V32bis, V34 et V90 afin de pouvoir discriminer la norme utilisée entre deux modems. Pour les normes V21, V22 et V22bis un démodulateur a été développé en langage C.

[CODEF 2001] Projet CODEF

Design Framework of Heterogeneous System
Type : Contrat industriel
Durée : 1998/2001
Partenaires : I3S, Philips/VLSI Technology
Objet : Etude et développement d'un environnement de prototypage rapide pour la conception de systèmes embarqués dédiés au traitement du signal.
Contribution du LESTER : Indirecte, travaux réalisés avant ma nomination au LESTER. Définition d'une méthode de synthèse des communications.

La Table 8 page suivante reprend l'ensemble des travaux depuis mon arrivée au LESTER.

Table 8 • Résumé des projets de recherche

Année / Projet	96/97	97/98	98/99	99/00	00/01	01/02	02/03	03/04	04/05	05/06	06/07	07/08	08/09	09/10
CODEF			**Contrat Industriel** *Codesign Framework of Heterogeneous System* I3S, Philips/VLSI Technology											
D2ASR					**Contrat Industriel** *Démodulateur Aveugle pour les Applications de Surveillance de Réseau commuté* LESTER, SERPE-IESM									
MACGTT								**Projet CNRS** *Méthode d'Aide à la Conception Globale des Terminaux de Télécommunications* I3S, LASTI, LESTER						
EPICURE									**Projet RNTL** *Environnement de Partitionnement et de Co-développement adapté aux processeurs à architectures REconfigurables* I3S, LESTER, CEA/List, THALES Communications, Esterel-Technologies					
PROSYR			**Projet CMCU** *PROtotypage de SYstèmes mixtes flot de données et/ou Réactifs, Application à la conception des systèmes sur puce* LESTER, ENIS (Tunisie)											
DARSoC								**Projet non contractuel** *Dynamic Adaptative and Reconfigurable System On Chip* LESTER, University of Massachusetts						
A3S				**Projet RNRT** *Adéquation Architecture - Application Système* THALES Communications, SOFTEAM, Mitsubishi Electric ITE, LESTER										
SecureFPGA									**Projet non contractuel** *Bistream protection for FPGA-based systems* LESTER, University of Massachusetts					
SANES										**Projet ERE DGA** *Security Architecture for Embedded Systems* LESTER, University of Massachusetts				
ICTeR											**Projet ANR** *Les Technologies Reconfigurables : Intégrité et confidentialité des informations* LIRMM, LESTER, ENST, NETHEOS, LIST			
AETHER											**Projet IST-FET** *Self-adaptive computing technologies for future embedded and pervasive applications* 14 partenaires Européens			
MOPCOM												**Projet RNTL** *Modélisation et spécialisatiOn de Plates-formes et COmposants MDA métiers* THALES Airborne systems, Thomson, ENSIETA, INRIA, Supelec, LESTER, Sodius		

4.4 Bilan et réflexion sur la fonction de chercheur

Une fois la thèse terminée et un poste de Maître de Conférences en poche, l'enseignant chercheur doit débuter une réflexion profonde sur son métier, savoir se remettre en question et accumuler les expériences afin de se préparer à devenir un chercheur autonome et actif. Je me suis toujours appliqué cette règle afin de progresser dans mon métier.

Durant ces dix dernières années j'ai eu l'occasion d'encadrer des doctorants et des étudiants en master, j'ai eu à animer et à monter des projets de recherche aussi bien nationaux qu'internationaux, j'ai débuté de nouveaux thèmes de recherche avec les étapes d'apprentissage que cela implique, j'ai participé et me suis impliqué dans des structures de recherche nationales, j'ai participé à la mise en place de conférences et effectué de nombreuses expertises d'articles scientifiques, j'ai participé à des expertises d'appels à projets nationaux et internationaux, j'ai favorisé la mobilité internationales des étudiants en thèse, j'ai moi-même effectué un séjour de recherche aux Etats-Unis pendant 10 mois, j'ai participé et animé des tables rondes au sein de conférences internationales, j'ai participé à de nombreuses collaborations de recherche… Cette liste qui ne se veut absolument pas être un catalogue d'activité illustre la richesse du métier de chercheur. Il m'apparaît clairement que c'est à travers ces différentes facettes du métier que le chercheur prend le recul et les repères nécessaires afin d'atteindre une maturité scientifique. Il me semble important d'ajouter ici que la qualité de l'activité est fondamentale et qu'une accumulation d'expérience n'est pas suffisante.

Certains thèmes me tiennent particulièrement à cœur, tout d'abord la collaboration qui me semble l'élément fondamental d'une activité de recherche. C'est à travers l'échange que les connaissances progressent aussi il est essentiel pour un chercheur d'initier des réflexions avec d'autres partenaires. Cela devient d'autant plus vrai aujourd'hui que les activités de recherche deviennent de plus en plus pluridisciplinaires. Ces collaborations peuvent être nationales mais il est également très important de considérer la dimension internationale de la recherche. La mobilité est une richesse qui ne se mesure totalement qu'une fois cette dernière réalisée. Il est donc important de pouvoir mettre en place ce type d'échange qui est très profitable pour les étudiants en thèse mais également pour les enseignants chercheurs. L'Université de Bretagne Sud soutien ce type d'approche et depuis plusieurs années il nous est possible d'accueillir des chercheurs étrangers et de mettre en place des séjours de recherche à l'étranger pour les étudiants en thèse. Ces expériences sont fondamentales et il est clair que l'avenir verra se renforcer la mobilité. La mondialisation passe par là et je pense que cette dimension internationale de la recherche est très intéressante et enrichissante.

Un autre point important concerne une difficulté, à laquelle nous sommes confrontés depuis plusieurs années, qui est la pénurie d'étudiants désirant effectuer une thèse. Il est fondamental de mener une réflexion sur ce thème afin de mieux faire reconnaître nos formations et nos métiers. Il me semble qu'une des raisons du blocage provient de la non reconnaissance forte des diplômes de doctorat au sein des entreprises. Certes les mentalités changent car l'université travaille de plus en plus conjointement avec le monde industriel mais il est encore difficile aujourd'hui de faire reconnaître la valeur ajouté qu'apporte une expérience en recherche. Nous avons encore des travaux à mener afin d'améliorer ce point mais l'université bouge et je pense que les liens étroits qui se créent progressivement entre le monde industriel et le monde universitaire sont positifs et encourageants pour l'avenir.

Comment parler recherche sans aborder le thème de l'animation scientifique. Ce point est fondamental que ce soit localement au sein d'un groupe de recherche ou d'un laboratoire qu'au niveau national. La maturité d'un chercheur se perçoit clairement à travers sa capacité à animer une communauté scientifique. Il est important pour un chercheur de réfléchir sur ce point afin de pouvoir construire et faire progresser un domaine d'activité. Je mène depuis

quelques années une réflexion sur l'animation scientifique de mon activité de recherche et après plusieurs expériences plus où moins satisfaisantes j'ai initié cette année la mise en place de réunions hebdomadaires de mon groupe de recherche travaillant sur la sécurité des systèmes embarqués. Ces réunions d'une durée de deux heures maximum sont très positives et permettent aux différents membres du groupe d'exposer leurs avancements et de discuter collectivement des difficultés rencontrées. Ces dernières permettent donc aux étudiants en thèse ou en projets, aux post doctorants et aux chercheurs de discuter ensemble de leurs problématiques et d'appréhender différents thèmes de recherche. Ce type d'expérience me semble très important afin de créer un groupe de recherche et de faire progresser collectivement un domaine d'activité.

Un point également important est la dissémination du résultat des travaux de recherche et le transfert des solutions élaborées vers l'industrie. Ces deux aspects bien que différents s'articulent autour de la notion commune de la valorisation du travail des chercheurs. Il me semble que nous sommes qu'au début de cette nouvelle aventure liée à la recherche universitaire car cela implique une évolution des mentalités. L'université se rapproche de l'industrie et développe en interne des structures de valorisation. Il est important de suivre et d'accompagner ces changements afin de pérenniser l'activité de recherche dans les universités. Il est également essentiel de sensibiliser les étudiants aussi bien pendant leur thèse qu'en amont durant le Master recherche ou professionnel afin qu'ils intègrent cette dimension liée à la recherche.

Pour conclure ces quelques réflexions il me semble fondamental de ne pas oublier que l'activité de recherche résulte d'une chaîne de transmission du savoir et d'expériences. Il est de notre devoir de préparer les étudiants en Master puis en thèse à un domaine très concurrentiel où les évolutions technologiques sont rapides et constantes. Pour cela il est essentiel de les impliquer dans les différentes activités de recherche afin de leur faire découvrir les multiples facettes du métier de chercheur. De la même manière il est fondamental d'accompagner les enseignants chercheurs dans leur processus de prise d'autonomie scientifique en multipliant leurs expériences liées au domaine de la recherche afin de créer les conditions nécessaires à l'excellence scientifique au sein des laboratoires de recherche.

Partie 2 : Annexes, Sélection des publications significatives

Cette deuxième partie, illustre les contributions menées en présentant plusieurs articles scientifiques.

1. Article concernant l'exploration de l'espace de conception pour les architectures reconfigurables

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

Hindawi Publishing Corporation
EURASIP Journal on Embedded Systems
Volume 2007, Article ID 23496, 20 pages
doi:10.1155/2007/23496

Research Article

Communication-Oriented Design Space Exploration for Reconfigurable Architectures

Lilian Bossuet,[1] Guy Gogniat,[2] and Jean-Luc Philippe[2]

[1] *Laboratoire de l'Intégration du Matériau au Système, Université de Bordeaux 1, CNRS UMR5218, 33405 Talence Cedex, France*
[2] *Laboratory of Electronic and Real Time Systems (LESTER), University of South Brittany, CNRS FRE2734, 56321 Lorient, Cedex, France*

Received 27 June 2006; Revised 21 December 2006; Accepted 16 January 2007

Recommended by Juergen Teich

Many academic works in computer engineering focus on reconfigurable architectures and associated tools. Fine-grain architectures, field programmable gate arrays (FPGAs), are the most well-known structures of reconfigurable hardware. Dedicated tools (generic or specific) allow for the exploration of their design space to choose the best architecture characteristics and/or to explore the application characteristics. The aim is to increase the synergy between the application and the architecture in order to get the best performance. However, there is no generic tool to perform such an exploration for coarse-grain or heterogeneous-grain architectures, just a small number of very specific tools are able to explore a limited set of architectures. To address this major lack, in this paper we propose a new design space exploration approach adapted to fine- and coarse-grain granularities. Our approach combines algorithmic and architecture explorations. It relies on an automatic estimation tool which computes the communication hierarchical distribution and the architectural processing resources use rate for the architecture under exploration. Such an approach forwards the rapid definition of efficient reconfigurable architectures dedicated to one or several applications.

1. INTRODUCTION

1.1. Context of design space exploration for reconfigurable architectures

Future applications like pervasive computing will require increasingly more flexibility. This major evolution will lead to imagine new execution platforms where flexibility, and also performances (speed, power consumption, throughput, etc.) will have to be guaranteed. Reconfigurable architectures correspond to an efficient solution to tackle this issue [1] as they are flexible and powerful, and represent a very attractive solution between software platform and dedicated hardware.

Many laboratories work on reconfigurable architectures [2] and propose different reconfigurable solutions. According to [3], the reconfigurable domain is a real jungle and it becomes mandatory to help the designer in order to increase the synergy between the application and the architecture. Applications will be efficiently implemented onto reconfigurable architectures only if several points are solved.

(i) Dynamic reconfiguration for efficient run-time adaptability: this point needs new techniques and tools like static and/or dynamic application partitioning tools, reconfiguration time estimation tools, control unit development, and operating systems to manage the reconfiguration steps.

(ii) Codesign of reconfigurable system on-chip: reconfigurable architectures are increasingly considered as a system on-chip, so they contain soft and dedicated hardware. Such systems need specific software and hardware design methodologies like codesign. These methodologies perform application partitioning and generally rely on performance estimation techniques to evaluate software and hardware implementations before covalidation and cosimulation of the design.

(iii) Design space exploration (DSE) for reconfigurable architectures: this last point focuses on exploring both the application and the architecture spaces. The aim is to find the most appropriate architecture for a single application or an applications family. In this case the architecture characteristics have to be defined according to reconfiguration issues. Such hardware architectures are not application-specific (like ASIC) and they

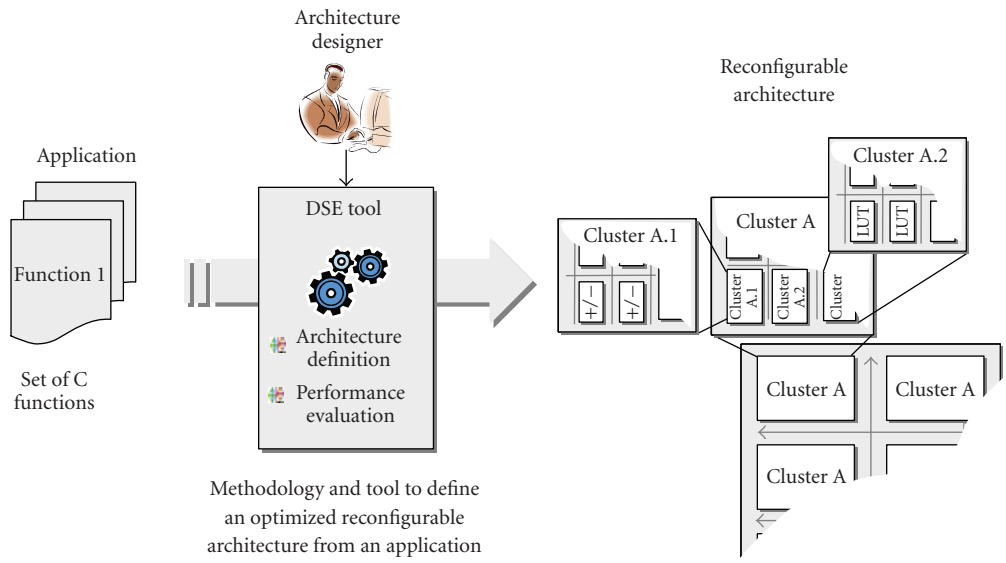


Figure 1: DSE process between application specification and architecture characterization.

provide large parallel structures that can be efficiently used within many applications. They embed coarse-grain and/or fine-grain operators and memories. New design techniques have to be developed in order to perform an efficient DSE for a better synergy between architectures and applications.

These points will have to be solved in the next few years in order to benefit from the huge potential provided by reconfigurable architectures. The challenges are developing operating systems to manage dynamic reconfiguration, developing codesign tools to build efficient reconfigurable SoC, and developing DSE tools to merge application space and architecture space. This paper focuses on the last challenge and demonstrates how to define an efficient architecture for an applications family.

1.2. Reconfigurable architecture DSE problematic

Although reconfigurable architectures correspond to hardware targets, their design is not the same as application-specific integrated circuit design (ASIC). Effectively, unlike ASICs that are designed to perform only one application with very tight performance constraints (area, latency, throughput, power consumption, etc.), reconfigurable architectures are designed to perform different applications relying on the same hardware capabilities. To be generic, reconfigurable architectures rely on massive parallel structures and use a dense reconfigurable routing network. They provide a set of low-level embedded elements (operator, logical function, memory block, etc.) organized into clusters. The DSE aim is to guide the designer to find some efficient clusters for an applications family. To address such an issue, it is essential to clearly define the architectural model under exploration.

(i) The template of the architecture for the exploration process is based on three hierarchical routing structures to propose the best communications scheduling inside the circuit and to take advantage of the application execution. Three levels of routing are generally admitted to represent an efficient solution. The low level of routing supports local communications between operators or logical elements and local variables storage, and the two high levels of routing support global communications.

(ii) The low level of the architecture relies on clusters, it is possible to use different clusters organized in a parallel structure. These clusters embed a range of coarse-grain arithmetic operators, coarse- and fine-grain logical operators, and memory blocks.

(iii) To be efficient, the architecture should take advantage of the application locality for treatment and storage. This last point is important to reach performance constraints.

As shown in Figure 1, DSE links the application specification (high-level specification) and the hierarchical clustered architecture. The DSE challenge is to take advantage of application execution graph to choose the best architecture parameters. Considering Figure 1, the exploration process will lead to the definition of the following:

(i) the type of resources within the low-level clusters (clusters A.2 and A.1),

(ii) the number of resources within the low-level clusters,

(iii) the number of low-level clusters within middle-level clusters (Cluster A),

(iv) the number of middle-level clusters within the whole reconfigurable architecture.

The exploration will thus enable designers to build their own architectures in order to be able to efficiently implement an application or an applications family. DSE gives designer information about application and architecture synergy, like

inside-communications cost during application execution and the total use rate of the architecture to perform the application.

1.3. Contribution

The contribution of this work is to provide a new DSE method based on communications distribution inside the reconfigurable architecture in order to define a power-efficient architecture under a time constraint in synergy with an application (or an applications domain). This work permits the consideration of fine-grain, coarse-grain, and heterogeneous architectures for the same application. The designer can explore a large domain in the reconfigurable design space. An important characteristic of the exploration method is to consider a high level of description for both applications and architectures. In spite of estimation accuracy, it is possible to quickly find a hierarchical clustering for the architecture. Following the exploration process, the designer describes the application and the architecture with low-level specifications in order to use more specific tools and to design the final system.

1.4. Paper organization

This paper is organized as follows. Section 2 presents several works dealing with reconfigurable architecture DSE, and it gives a comparison table of these works. Section 3 presents the contribution and position of our work. Section 4 describes the application and architecture specifications used within the exploration process. In Section 5, the algorithms to estimate the communication distribution are proposed. Section 6 gives several results of exploration in the case of image computing and cryptography. Finally, Section 7 concludes this paper.

2. RELATED WORK

2.1. Introduction to design space exploration for reconfigurable architecture

It is possible to perform DSE at different levels of abstraction in order to progressively reduce the number of solutions. The more the abstraction level is refined, the more accurate the results are since a lower number of solutions need to be considered [4]. In the case of hardware DSE, two main methods are generally considered [5].

Synthesize and compare

This method uses a full synthesis flow to synthesize the application for each type of architecture under exploration before comparing the overall performance results. Using this method, it is possible to obtain very accurate performance measures. Nevertheless, it is necessary to have a specific synthesis tool for each type of architecture (which is not always available in the case of coarse-grain architecture exploration) or to use generic synthesis tools. However, synthesis steps compute very complex algorithms, which lead to a limited and slow exploration process. Furthermore, when using generic synthesis tools, it is necessary to have very good knowledge of the target architectures since it is necessary to develop a model for them. Hence, this method is not really adapted for a large and rapid architecture exploration and is more relevant for architecture refinement steps.

Estimate and compare

The second method relies on performance estimations instead of synthesis. In that case, it is necessary to consider a generic architecture model to describe the different target architectures. The goal is to perform relative performance estimations (speed, power consumption, and area) in order to compare different architectures very quickly. Although the estimations do not give necessarily real and accurate performance results, it is enough to compare the architectures since the relevant point in that case is that estimations are faithful and an absolute error is not the major concern.

These two methods are complementary and can be used within the same design process (same application) but at different abstraction levels. At a high level of abstraction, there are few synthesis tools and the architectural design space is huge. Therefore, it is more efficient to use an *estimate and compare* method in order to reduce the design space. At a low level of abstraction, the architectural design space is reduced. In this case, the exploration must converge towards a reliable architectural solution. Therefore, the *synthesize and compare* method is more relevant for this case. Obviously, a design space exploration flow should use several methods according to the level of abstraction. Interested readers can find more information about DSE in [6]. The following paragraph gives some examples of DSE tools and methods for FPGAs and coarse-grain reconfigurable architectures.

2.2. FPGA place and route generic tools used for DSE

Generic place and route tools for FPGAs are generally used within the *synthesize and compare* method. When the architecture model allows for the physical description of the routing structure, the tools can provide accurate performance estimates (particularly for speed and area). Such techniques provide interesting results concerning the use of the routing resources and the ability to route the device.

The versatile place and route (VPR) tool, developed at the University of Toronto in Canada, is a very interesting approach that works on a physical model (P-Spice model) [7]. VPR is a place and route tool that works at the logic level and is oriented for island style fine-grain architecture (like Xilinx FPGA). The physical model forwards the description of the architecture physical parameters (technology, routing type and size, routing switch resources, clusters size, etc.). VPR has an automatic mode which tries to route a circuit for different numbers of routing wires. Using VPR, it is possible to explore several aspects of the architecture like LUT and cluster size [8] or embedded memory size [9].

Madeot-Bet, a generic place and route tool developed by the University of Brest in France, uses a functional description of the architecture [10]. As VPR, Madeot-Bet is

oriented for fine-grain reconfigurable architectures even if some extensions are currently under development to address coarse-grain architectures. The functional specification used to model the reconfigurable architecture enables the description of a large panel of architectures and is technological-independent. In fact, each element of the architecture is described by the functions it can execute. Although VPR and Madeot-Bet are generic place and route tools, they can be used for fine-grain architectural exploration, particularly for routing exploration.

2.3. FPGA exploration and estimation tools

According to the *estimate and compare* method, it is possible to perform DSE using estimation tools. The estimations can focus on one or several parameters (power consumption, speed, area, etc.) depending on the designer's expectations.

People of the University of Southern California in the USA present in [11] a power consumption estimation tool based on a parametric view. It provides a domain-specific modeling technique that exploits the knowledge of both the algorithm and the target architecture family for a given problem to develop a high-level model. This model captures architecture and algorithm features, parameters affecting the power performance, and several power estimation functions based on these parameters. However, the designer needs to have a great deal of knowledge in the domain (application and architecture) to be able to determine the parameters and the functions.

Enzler et al. [12] propose a high-level estimation methodology for area and speed developed within the Swiss Federal Institute of Technology. This methodology relies on the inputs and outputs, the control signals, the operators, the registers, the degree of parallelism, and the number of iterations within the application to characterize the application. With these characteristics, several parameters are computed to provide the delay and area performances. The target FPGA is characterized through the mapping of the operations. Therefore, a mapping model is specified for each type of operation from which the area and timing parameters are derived. The application and the target architecture are used to estimate the delay and area performances. After this estimation, several application parameters can be explored like the number of registers into the data path, the number of replications of a given block (parallelism), and the number of block decomposition into a sequence of identical subtasks (pipeline). This method mainly allows for the application exploration since the architecture exploration is rather limited.

2.4. Coarse-grain reconfigurable architectures DSE tools

Previous efforts focus on fine-grain reconfigurable architectures (FPGAs) even if it is possible to extend several techniques for coarse-grain architectures. Other works focus directly on coarse-grain architectures.

MIT researchers have developed a DSE framework for the raw microprocessor [13]. This reconfigurable architecture is reminiscent of coarse-grain FPGA and it comprises a replicated set of tiles coupled by a set of compiler orchestrated pipelined switches. Each tile contains an RISC processing core and SRAM memory for instructions and data. Several parameters, like the number of tiles, the memory size, or the communication bandwidth, characterize the architecture. The application is split into several subproblems, each of them is characterized by the number of architecture resources it consumes. Finally, some cost functions are used to estimate the performance (delay, area). This method is architecture-dedicated since the cost functions are only defined for one architecture. The result accuracy depends on the relevance of the architecture parameters.

In [14], the DSE flow targets a mesh architecture called KressArray [15], a fast reconfigurable ALU. The exploration tool Xplorer works at the algorithmic level and aims at assisting the designer to find a suitable architecture for a given set of applications. This tool is architecture-dependent, but the use of fuzzy logic to analyze the results of the exploration is a very attractive approach.

2.5. Comparative study

Table 1 gives a comparison between the related work presented above. The last line details our work characteristics in order to give the reader a comparison with previous efforts. The first two studies focus on simple FPGA (island style FPGA without embedded features like memory blocks or multipliers). Then, VPR and Madeot-Bet are really close since they use the same place and route algorithm and they mainly focus on the routing aspect. They can be used for exploration but it is necessary to first perform a high-level exploration to reduce the design space. The last studies are architecture-dependent and they are developed for coarse-grain reconfigurable architectures. Except for the second study, all of the studies explore the architecture according to one or several objectives. The second study explores the application algorithm and implementation. The two last studies have some automatic exploration steps since they are more specialized for a specific architecture. We can see on the last row that the different tools provide different results, depending on the starting design space (set of architectures or given architecture with a set of parameters). Therefore, they provide the best architecture in a set or the best configuration for a given architecture. Most of them give design information to help the designer to improve the application and the architecture definition.

VPR and MADEO can explore the largest design space since they are the most generic tools. The first two studies in Table 1 are specialized for an FPGA family, so the design space is limited. The last two studies are architecture-specific, so they can only explore a small design space. However, they provide very accurate estimations thanks to more accurate models. Therefore, the development of a DSE tool is a tradeoff between the design space size and the estimation accuracy.

Table 1: Characteristic comparison of the related work.

Tool	Architecture target	Applications specification	Architectures specification	Exploration	Results
Univ. Southern California [11]	Simple FPGA	Parameterization	Parameterization	Architectural objective: power manual and exhaustive	One architecture in a set (domain)
ETH [12]	Simple FPGA	Data flow graph	Characterization of simple operator	Algorithmic objectives: delay and area manual and exhaustive	Application design information (parallelism degree and level of pipeline)
VPR Univ. Toronto [7–9]	Complex FPGA	Netlist BLIF (or EDIF)	Structural model	Architectural objective: routing manual and exhaustive	FPGA architecture design information (clusters size, configurable element size, routing)
MADEO Univ. Brest [10]	Complex FPGA (multigrains)	Data flow graph	Structural model	Architectural objectives: routing, area and critical data-path manual and exhaustive	FPGA architecture design information (clusters size, configurable element size, routing)
Raw MIT [13]	Coarse-grain raw architecture	Parameterization	Parameterization	Architectural objective: execution time automatic and heuristic	Optimal architecture raw configuration
Xplorer Univ. Kaiserslautern [14, 15]	Coarse-grain KressArray architecture	ALE-X	Parameterization and structural model	Architectural objectives: power and routing automatic and heuristic	Optimal architecture KressArray configuration
Authors Univ. Bretagne Sud [16]	Heterogeneous architecture	HCDFG	Hierarchical functional model	Algorithmic and architectural objective: power automatic and heuristic	Architectural design information (cluster size, configurable element type, memory size, communication distribution, resource use rate)

This comparison helps us to define the characteristics of a new DSE method for reconfigurable architecture. It appears that all these methods are too specialized, technological-dependent, or architecture-dependent to explore a large design space with targets like fine-grain, coarse-grain or heterogeneous architectures. It is important to overcome this limitation to be able to compare fine-grain and coarse-grain architectures and to combine both in a single device (heterogeneous architecture). Furthermore, previous efforts require profound knowledge of the reconfigurable architecture and technology which can be difficult to handle for the designer. It would be helpful to provide a methodology where the designer can have an early estimation of his architecture performance without going through all the details of the architecture. All the above-presented tools do not take into account (static or dynamic) architecture reconfiguration during the performance estimation process. This aspect is becoming increasingly relevant and should be addressed within the exploration process.

Finally, the presentation of previous efforts relies on general benchmarks (which can also be used for ASIC implementation) to demonstrate their design flow and to validate the different concepts. In this paper, we use the same approach to validate our work as will be presented in Section 6.

3. A NEW VISION OF DSE BASED ON AN AUTOMATIC ESTIMATION TOOL FOR SOC DESIGN CALLED *DESIGN TROTTER*

3.1. *The Design Trotter tool*

The work presented in this paper is part of the Design Trotter project [17]. The Design Trotter framework is a computer-aided design (CAD) environment for reconfigurable system on a chip (RSoC). This environment is composed of several tools that work at different levels of abstraction and explore the design space in different ways. Figure 2 presents the interaction between the main tools of the Design Trotter framework. First, the application is specified using a subset of the C language, then the specification is translated into a hierarchical control data flow graph (HCDFG) [18]. For the present work, two tools of the *Design Trotter* framework are considered.

System estimation [19]

This tool aims at scheduling the application for several time constraints. The results are defined through "cost profiles," that is, scheduling for all the resources used by the

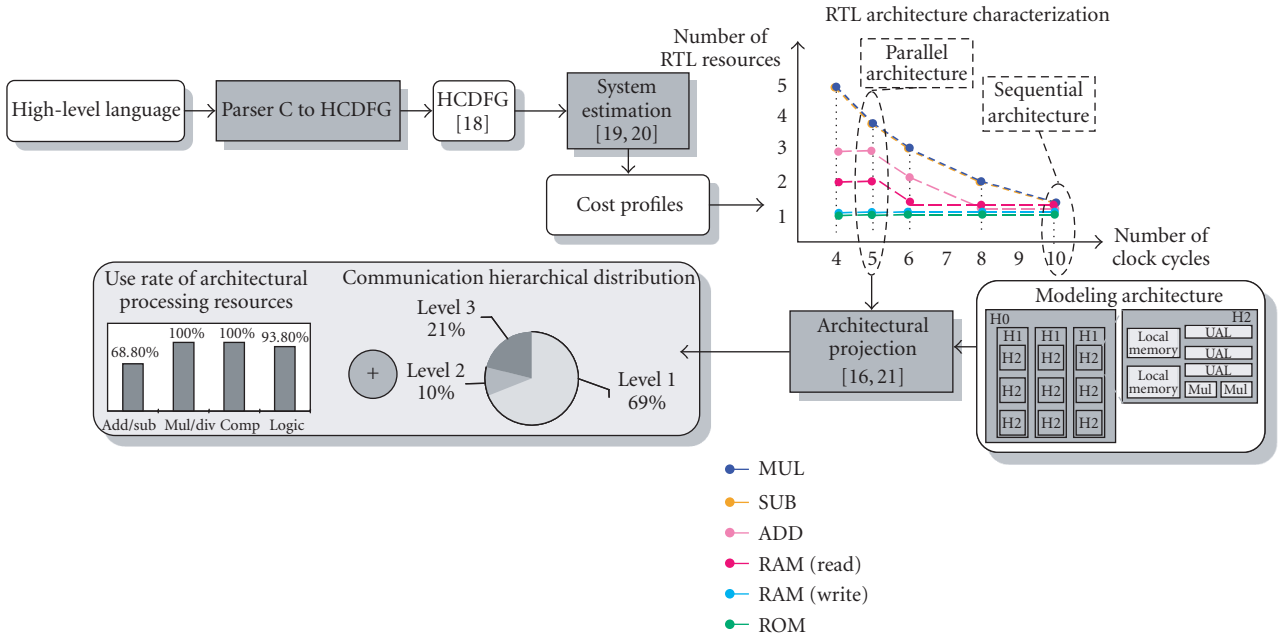


Figure 2: Design Trotter framework.

application. There is one cost profile for each time constraint. The processing and memory resources considered to compute a scheduling enable the definition of a logical architecture at the RTL level [20].

Architectural projection [16, 21]

This tool is used by the architectural exploration method presented in this paper. It computes performance estimations and enables the comparison between several architectures characterized by their power efficiency to implement an application. It provides design information that helps the designer to progressively improve the architecture definition through several iterations.

The system estimation tool performs an algorithmic exploration and the architectural projection tool drives the physical architecture exploration of the reconfigurable targets. So, the synergy between the application and the architecture is explored to reach the best couple application/architecture.

As shown in Figure 2, the cost profiles are the results of the system estimation tool and correspond to the inputs of the architectural projection tool. To launch the architectural projection tool, it is first necessary to select an RTL logical architecture to implement the application. The designer can consider a sequential RTL logical architecture with a high time constraint and a low number of resources (computing and memory resources) or the designer can consider a parallel RTL logical architecture with a low time constraint, so the number of resources is larger than for the sequential architecture. This last solution is adapted for hardware implementation (FPGA, coarse-grain reconfigurable architecture, or ASIC) as these technologies provide massive parallelism. The cost profile of the selected RTL logical architecture provides the number of computing and memory resources for a given time constraint. To perform the system estimation, several scheduling algorithms have been developed in order to explore various tradeoffs depending on the characteristics of the application. The system estimation method (application metrics and scheduling techniques) is presented in [20].

The architectural projection tool provides the designer with use rate estimates of the architecture computing resources and the communication distribution for the different hierarchical levels of the architecture. For that purpose, the designer describes the target architecture with a hierarchical functional model. He can also refine the architecture description during the exploration process in order to tune the architecture parameters depending on the architectural projection results. The designer aims at finding a power-efficient architecture for his application under a time constraint.

In the next section, we detail the definition of *efficiency* since this notion drives the exploration process in our case. We also present the hierarchical functional model and the algorithms used within the architectural projection tool. The methodology developed to explore the design space is also presented.

3.2. Strategy of coarse-grain DSE

In order to develop a DSE methodology, it is necessary to emphasize some criteria to compare the architectures for the same application (and so the same RTL logical architecture). Our approach is under a time constraint since we consider an RTL logical architecture for a given number of cycles to perform the application as shown in the upper right part of Figure 2. According to [22], power consumption is a major

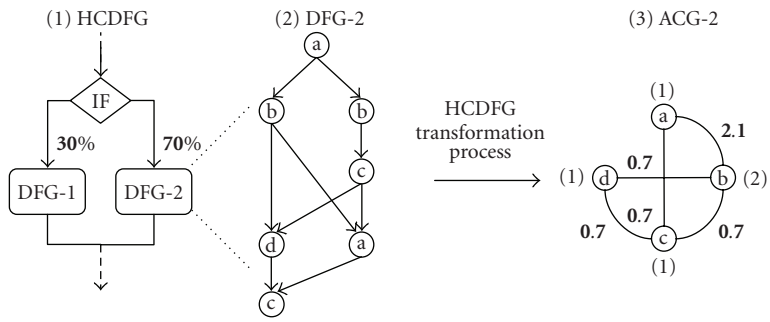


Figure 3: HCDFG transformation into ACG.

metric to compare the architecture efficiency under a time constraint. Power consumption reduction often becomes increasingly the main objective of a design flow, particularly for embedded systems. Power consumption is linked to the hardware time-life, the battery size, and the heat misbehavior.

In order to compare the power efficiency of different reconfigurable architectures, it is necessary to study the impact of the architectural resources on power consumption. We have studied this impact for fine-grain architecture (FPGA). According to our studies [23, 24] and according to other academic studies [25–29], several conclusions about power consumption of architectural resources for fine-grain architecture can be drawn.

(i) Routing resources are always the most consuming resources taking up from 50% to 80% of the total power consumption. However, the exact rate depends on design size, frequency, toggle rate of logical inputs, number of inputs/outputs, utilization rate of architecture resources, and synthesis option.
(ii) It is more power-efficient to use dedicated memory blocks instead of using distributed memories (e.g., with lookup tables).
(iii) A high use-rate for the architectural resources is better for power consumption since the free-resource static power leaks are lower.
(iv) It is very important to use local routing resources for intensive communicating resources (computing or memory). So the most communicating resources (which depend on the application specification) must be placed in a near neighborhood in order to decrease the routing power impact.

According to all these studies, computer-aided design tools must rely on a strategy of clustering for the most communicating resources within the architecture (for a given application or an applications family). So for a given application, a reconfigurable architecture has to be defined in order to promote an efficient clustering of the application resources. We extrapolate the previous conclusions of the fine-grain studies for coarse-grain and heterogeneous architectures since the internal routing structures are very similar, and the routing issues are still the same [27, 29].

The application and the architecture specifications have to emphasize the communications between the resources required by the application (according to the system estimation results) and the locality (from the routing point of view) of the architecture computing and memory resources. The next section presents these specifications and shows how they allow for the taking into account of communications and locality of the application and architecture resources.

4. APPLICATION AND ARCHITECTURE SPECIFICATIONS

4.1. Application specification

The application is first described using a subset of the C language [18]. This specification is then translated into a hierarchical control data flow graph (HCDFG) as an internal representation. This graph corresponds to a precise description of the application (computing, memory, and control) [18]. The system estimation tool provides information about the RTL logical architecture like the number of computing resources and memory resources needed for the application. As presented in the previous section, it is essential to obtain information about the communications between the application's resources since the most communicating resources have to be placed close within the architecture. To show this information, a new graph called average communication graph (ACG) has been developed. This particular graph highlights how each type of processing and memory resource communicates with each other. The edges in it represent the communications between two nodes and each node represents a type of processing resource or a memory resource (Figure 3).

Several differences exist between the HCDFG and the ACG graphs. The HCDFG describes the real control and data flow of the application independently of any implementation while the ACG corresponds to an approximation of the communications between operators and memories. The HCDFG is transformed into the ACG after having performed the scheduling of the operators and memories for a given time constraint. This scheduling as previously mentioned is performed during the system estimation step [20]. There are fewer nodes in the ACG than in the HCDFG since the ACG graph has only one node for one type of processing resources.

The ACG edges are not oriented and the communications are taken into account in all directions. Several attributes are added in the ACG to describe the internode communications.

Figure 3 shows an example of an HCDFG graph transformation into an ACG. In this example, the type of processing resource corresponds to a letter (a, b, c, or d). The number in brackets beside a node corresponds to the number of operators required for a given time constraint (result of the system estimation tool). The boldface number beside an edge is the total number of communications between two processing nodes. In order to define which pair of nodes communicates the most in the ACG, the relative number of communications between two processing types is computed. This value is obtained using the following equation:

$$\text{RelativeComm}_{\text{Op1}-\text{Op2}} = N(\text{Loop}) \times P(\text{Branch}) \times \frac{\text{TotalComm}_{\text{Op1}-\text{Op2}}}{\text{NumberOp1} + \text{NumberOp2}}, \quad (1)$$

where $\text{RelativeComm}_{\text{Op1}-\text{Op2}}$ is the relative number of communications, $\text{TotalComm}_{\text{Op1}-\text{Op2}}$ is the total number of communications, NumberOp1 and NumberOp2 are the numbers of allocated operators of each type. If the DFG is part of a hierarchical node with control nodes, $N(\text{Loop})$ is the loop number for a loop control node and $P(\text{Branch})$ is the branch probability for a conditional node as in Figure 3. For each control node, $N(\text{Loop})$ and $P(\text{Branch})$ are precomputed through a code profiling. For example, the ACG on the right-hand side in Figure 3 has two nodes, a and b, linked by one edge with a value equal to three. The node a describes one operator of a type and the node b describes two operators of b type. Therefore, the relative number on the edge, between these two nodes, is given by

$$\text{RelativeComm}_{a-b} = 0.7 \times \frac{3}{1+2} = 0.7. \quad (2)$$

4.2. Reconfigurable architectures specification

The reconfigurable architecture model is an important part of this contribution since it is a complex task to manage accuracy and high level of abstraction. According to Sections 1.2 and 3.2, the main characteristics of a model can be listed as follows.

(i) The model has to enable a large design space covering fine-grain, coarse-grain, and heterogeneous architectures.
(ii) The model has to describe the physical locality of computing and memory resources.
(iii) The model has to remain technologically-independent to be valid in spite of technological evolutions.
(iv) The model needs to be easily extended to take into account new architectural characteristics and possibilities.

To promote the architectural exploration, it is essential to mitigate the task of changing some architectural characteristics. The designer must be able to rapidly perform some manual evolutions of the architecture description.

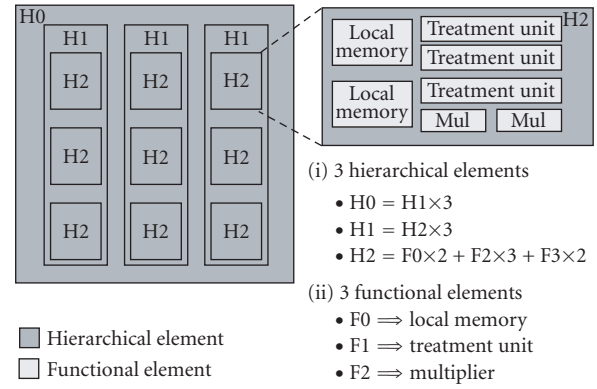


Figure 4: Example of coarse-grain reconfigurable architecture modeling with three hierarchical levels.

There are two possibilities to describe a reconfigurable architecture, using a physical description or a functional description. Using a physical description, as in [7], forwards the development of accurate estimations, but the model is technologically-dependent and cannot evolve easily. Moreover, the model can be complex (if the model describes all the details of the architecture) and it can be very tedious for the designer to manually change some architecture characteristics. Using the functional model, as in [10], leads to describe the architectural resources through the functions that they can realize (several functions if the resource is configurable). This type of model can easily evolve and the designer can quickly modify the architecture in order to explore the design space. Therefore, this kind of model is suitable for architectural specification.

According to Figure 1, in order to describe the architectural resources locality, we use a hierarchical view. For that, the proposed hierarchical functional model for reconfigurable architectures relies on two types of elements.

(i) *The hierarchical elements* are used to model the architectural hierarchy. They are containers; they embed other hierarchical elements or functional elements, and are described by their contents.
(ii) *The functional elements* describe the computing and memory resources. They are described by the list of functions that they can realize for a selected configuration.

Figure 4 shows an example of coarse-grain reconfigurable architecture modeling. In this figure it can be seen that according to the reconfigurable specification (see Section 1.2 and Figure 1), there are three hierarchical elements; H0 contains three H1, each H1 contains three H2. The high level of hierarchy is composed of one H0, the low level corresponds to the internal structure of H2. This last hierarchical element is composed of several functional elements. The architecture has three levels of hierarchy; the low level inside the H2 hierarchical elements contains only functional elements, the

middle level inside H1 elements contains H2 elements, and the high level of the hierarchy is represented by H0.

This model uses two important hypotheses concerning the communication costs in the hierarchical elements. They enable the routing resources to be taken into account without using an accurate physical description. These hypotheses are as follows.

(i) The communication costs inside a hierarchical element are homogeneous. If the designer wants to describe large hierarchical elements, he must guarantee that this hypothesis will be verified with the use of dedicated routing resources in the corresponding hierarchical level of the architecture.

(ii) The second hypothesis is that the communications are less power consuming in the low level of hierarchy than in the high level of hierarchy. That is to say, for the architecture example in Figure 4, the communications inside the hierarchical element H2 (the communications between the embedded functional elements) consume less power than the communications inside the hierarchical element H1 (the communications between the elements H2). These latter communications consume less power than the communications inside the hierarchical element H0 and so between H1 hierarchical elements.

5. COMMUNICATION ESTIMATION AND EXPLORATION METHODOLOGY

5.1. Introduction

Our exploration method is based on an estimation of the communications hierarchical distribution within the architecture and an estimation of the architectural resources use rate. The goal of the exploration is to define an architecture promoting the clustering of the most communicating resources. The exploration method is interactive and is based on the architectural projection tool.

The architectural projection tool proposes an algorithm to merge the application ACG nodes according to their communications and to allocate the application resources within the low level. Since our approach works at the algorithmic level, it does not target a specific synthesis tool and does not consider any accurate physical architecture model. Instead of giving designers a single communication cost value that may present a significant absolute error due to backend synthesis algorithms and architecture refinement steps, we compute two bounds and an intermediate value. This approach gives the designer the ability to define the architecture at the algorithmic level with the guarantee that the final performance will belong to the estimated performance interval (Figure 5).

It also provides designers with metrics on allocation algorithm impact. Figure 5 illustrates this point, the architecture C has a narrow performance interval, so allocation algorithms will have a small effect on the final performance and a low complexity algorithm can be considered. The architecture B has a large performance interval, so allocation heuristic will have a strong impact on final performance and it might be important to consider better allocation algorithms.

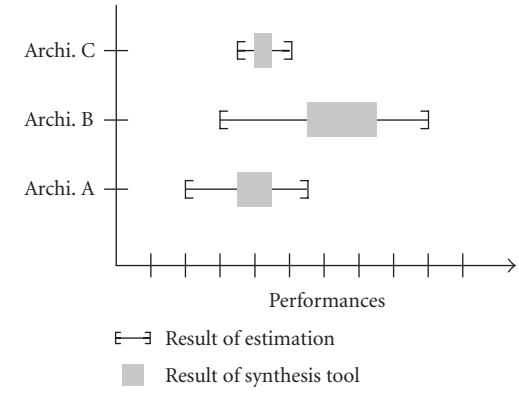


Figure 5: Bound performance results, example for three architectures.

To define such an approach, we have developed three algorithms that give three values of the communications hierarchical distribution estimation. The first algorithm, *the INTER algorithm*, gives distribution estimation with a maximum number of communications within the low hierarchical level H2, so the communication power cost for the application is minimal. The second algorithm, *the MIN algorithm*, gives distribution estimation with a minimum number of communications within the low hierarchical level H2. The last algorithm, *the INTER algorithm*, gives an intermediate value between the values given by the two other algorithms. Each algorithm is less complex and faster than an optimal algorithm.

The next sections present the different algorithms to perform the architectural projection and to obtain the communications hierarchical estimation. The tool deals with reconfigurable architectures composed of three levels of hierarchy since these levels are adequate in describing most current architectures.

5.2. The architectural projection

The architectural projection step makes the link between the required (application) and the available (architecture) resources with the challenge that the most communicating resources will be assigned in the same hierarchical element within the low hierarchical level.

The first step of the projection process is to search for the most communicating pair of nodes in the ACG as shown in Figure 6. Subsequently, the most communicating pair of nodes is merged if the two nodes are hierarchically compatible. To be compatible, nodes must be potentially embedded in the same hierarchical element. It means that the hierarchical element must have enough available resources (functional elements) to implement the processing or memory operation described by the two corresponding nodes. If nodes are compatible, a new node called "composite node" is created to describe the merging of the nodes. Since the ACG has a

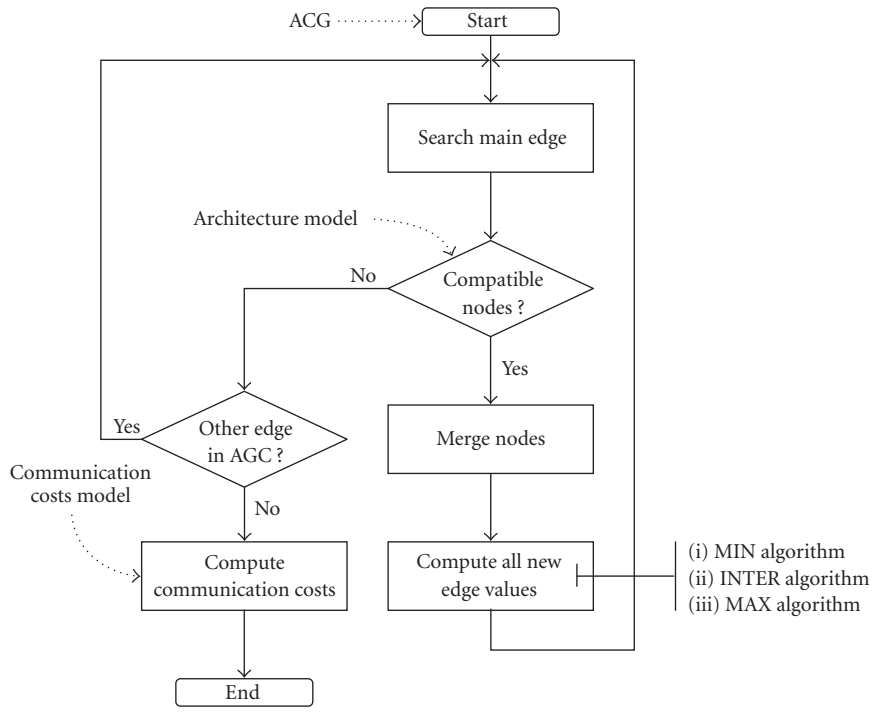


Figure 6: Architectural projection flow based on three algorithms: MIN, INTER, MAX.

new node, it is not the same graph, thus it is necessary to recompute all edge values and make several transformations due to the new composite node. The architectural projection stops when node merging is no longer possible and when all application resources are virtually associated to architecture functional elements. The complexity of the architectural projection algorithm is polynomial in $O(n^2)$, where n represents the number of edges within the ACG graph. This complexity does not represent a major issue as the number of edges is generally small. As explained previously, an edge represents the communications between two types of operations within the application.

Figure 7 shows the architectural projection process using the *INTER-algorithm*. For this example, the ACG and the architecture model are simple. The ACG has three processing nodes since the result of the system estimation tool allocated two multipliers, two subtracters, and one adder to respect the time constraint selected by the designer. The values on the ACG edges correspond to the number of required communications between the processing resources (corresponding to the processing nodes). For example, in Figure 7, twenty communications are required between the multipliers and the subtracters. The modeled architecture has two levels of hierarchy. In the hierarchical high level, one hierarchical element, H1, contains two elements H2. In the hierarchical low level, one hierarchical element, H2, contains three functional elements; two adders/subtracters and one multiplier.

In Figure 7, the process starts by merging the most communicating pair of nodes. The nodes multiplier and subtracter are the most communicating pair. As these two nodes are hierarchically compatible, in the second step a new composite node is created to describe that one subtracter and one multiplier are allocated in the same hierarchical element H2 in the architecture. The composite node has a number of internal communications; this number (ten in the case of Figure 7) depends on the algorithm (MIN, INTER, and MAX). In Figure 7, the process needs four steps to allocate all the required application resources in the modeled architecture. At the end, the ACG has two composite nodes; the number of communications in the low hierarchical level is computed. This number corresponds to the sum of the number of communications of the internal-composite nodes. For this example, there are 34.66 communications in the low hierarchical level (inside the hierarchical elements H2 between the functional elements) which corresponds to 82.5% of the total application communications. The next section will give the differences between the three algorithms.

5.3. Differences between the three architectural projection algorithms (MIN algorithm, INTER-algorithm, and MAX algorithm)

Three algorithms have been defined to merge the ACG nodes into composite nodes and to compute the ACG edge values for each architectural projection step. Figure 8 presents the first step of the hierarchical projection using the same example as Figure 7. The composite node is obtained from one subtracter and one multiplier merging.

The difference between the three algorithms is illustrated in Figure 8. The strategy of the *MIN algorithm* (top of Figure 8) is to consider that if two application resources (operator or memory) are assigned in the same hierarchical

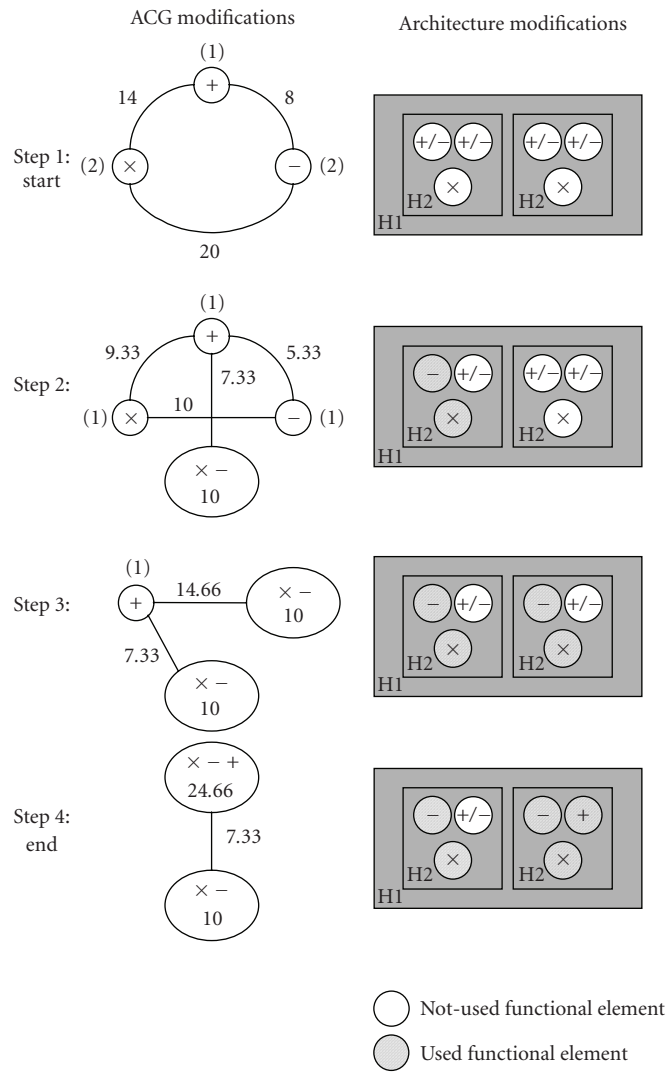


FIGURE 7: The architectural projection process, using inter algorithm with a simple three-node application ACG and two hierarchical-levels coarse-grain architectures.

element, all the communications between the two considered ACG nodes are allocated to the new composite node. Hence, when the composite node is created, the edge between the two initial nodes is deleted. Unlike *MIN algorithm*, the *MAX algorithm* does not take into account the creation of a new composite node, the communications between all the nodes and the composite node are distributed uniformly. In fact, the max algorithm corresponds to a greedy process to allocate the architectural resources. The idea of the *INTER-algorithm* is to consider that two operators in the same hierarchical element must communicate more than two operators in two different hierarchical elements; it is a tradeoff between min and max algorithms.

In order to have a better understanding of this approach, Figure 9 presents the three algorithms that estimate the communication costs. To understand the different algorithms, some notations are necessary:

(i) N_i shows node i of ACG,
(ii) t_i shows type of processing for the node N_i (processing such as adder, multiplier, etc.),
(iii) n_{ti} shows number of processings in the node N_i,
(iv) C_{ij} shows composite node with two processings t_i and t_j,
(v) $\mathrm{IC}_{C_{ij}}$ shows number of internal communications in the composite node C_{ij},
(vi) $E_{i,j}$ shows edge between nodes N_i and N_j,
(vii) $P_{i,j}$ number of communications between nodes N_i and N_j (i.e., value associated with edge $E_{i,j}$),
(viii) $E_{k,ij}$ shows edge between the node N_k and the composite node C_{ij},
(ix) $P_{k,ij}$ shows number of communications between the node N_k and the composite node C_{ij} (i.e., value associated with edge $E_{k,ij}$).

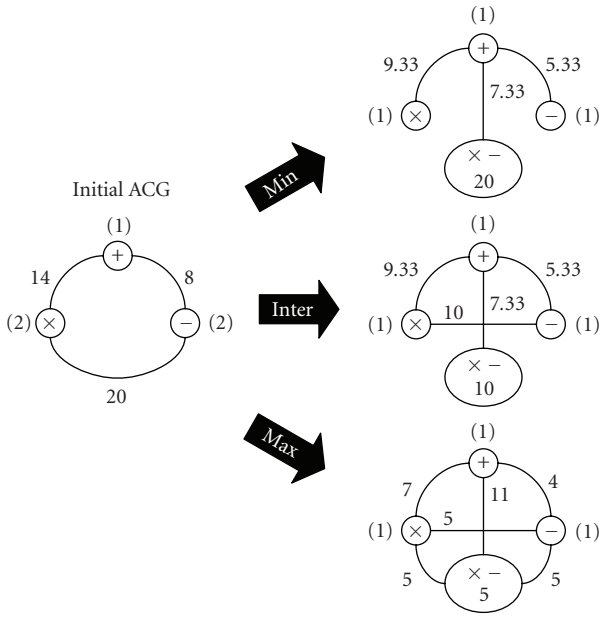


Figure 8: First step of the architectural projection for the three algorithms.

Some general functions are also used to describe the algorithms:

(i) CREATE_NEW_COMPOSITE(t_i, t_j) is the function that creates in the ACG a composite node with two processing types;
(ii) CREATE_EDGE(N_i, N_j) is the function that creates in the ACG an edge between nodes N_i and N_j;
(iii) DELETE_EDGE($E_{i,j}$) is the function that deletes in the ACG the edge $E_{i,j}$ between nodes N_i and N_j;
(iv) MIN(*float1*, *float2*) is the function that returns the smaller float between *float1* and *float2*.

At the end of the architectural projection process, the designer obtains three communications hierarchical distribution estimations and the estimation of the architecture resources use rate. Based on these results, the designer can modify the architecture model according to the DSE method in order to reach his performance constraints. The designer goal is to define a power-efficient reconfigurable architecture for an application under a given time constraint. The next section presents the DSE method.

5.4. DSE method

Before explaining the exploration process, it is important to depict how an application is described since our method works at a high level of abstraction. The application is split into several functions and the execution of these functions can be either sequential or pipeline depending on the performances to achieve. The reconfigurable architecture must be efficient for all the functions of the application. Finding an optimal architecture for all the functions can be very tedious and even intractable. Moreover, searching for an optimal efficient architecture for all the functions independently is not the best way to quickly obtain the most efficient architecture for the total application. Several experiments [5] have shown that it is more efficient to identify the application critical functions and to perform the exploration for these functions since they have the strongest impact on the application performances. If the architecture is efficient for these functions, the application performances will be higher. To find these critical functions (often one or two functions within an application), we have developed three metrics that emphasize the realization characteristics of each function. These metrics are computed for each function.

(i) *The execution parallelism degree* of a function is obtained from the system estimation tool [19]. This metric highlights if the selected scheduling is suited for the function. As we target hardware reconfigurable devices, the parallelism degree must be high in order to benefit from the large amount of available resources. If the parallelism degree is low and if there is no other scheduling possibility, the function can be considered as critical since the designer has limited freedom to implement the function.
(ii) *The potential of communications spatial locality* of a function corresponds to the ratio between the number of resources (processing and memory) and the number of communications to be performed during the execution of the function. If there are many communications for a small number of resources, it is important to consider the spatial locality of communications since it will have a large impact on power efficiency. A function with such a feature can be considered as critical.
(iii) *The potential of architecture routing resources temporal congestion* during the execution of a function. This metric corresponds to the ratio between the number of function execution cycles (or time constraint) and the number of communications to be performed during the execution of the function. If there are many communications for a small execution time, it certainly will be challenging to temporally distribute the communications onto the routing resources. Therefore, the function can be considered as critical.

The critical functions are critical for all the metrics or for two among three. Usually there are one or two critical functions per application, but this number depends on the application complexity. The architectural exploration process is only performed for the application critical functions and leads to the definition of a power-efficient architecture that supports these functions under a given time constraint (scheduling choice, see Figure 2). If the application has several critical functions, then the final architecture corresponds to a tradeoff between the dedicated architecture for each function [16].

The definition of an efficient architecture for an application critical function begins with the analysis of its ACG (Figure 10). This analysis provides information about communications like the communications repartition between

Algorithm 1: Algorithm MIN

C_{ij} = CREATE_NEW_COMPOSITE(t_i, t_j)
$\text{IC}_{c_{ij}} = p_{i,j}$
for each $N_k \in$ ACG
/ compute edge value*
*between node k, node i, and composite node ij */*
if ($N_k \neq N_i, N_k \neq N_j, N_k \neq C_{ij}$)
  if ($E_{k,i}$ exist)
    if ($E_{k,ij}$ exist)
      $p_{k,ij} = p_{k,ij} + \frac{p_{k,i}}{n_{t_k} + n_{t_i}}$
    else
      $E_{k,ij} = \text{CREATE\_EDGE}(N_k, C_{ij})$
      $p_{k,ij} = \frac{p_{k,i}}{n_{t_k} + n_{t_i}}$
    end if
  end if
  $p_{k,i} = \frac{p_{k,i} \times (n_{t_k} + n_{t_i} - 1)}{n_{t_k} + n_{t_i}}$
end if
/ compute edge value*
*between node k, node j, and composite node ij */*
/ idem that for node i but with node j */*
...
end for
$n_{t_i} = n_{t_i} - 1$
$n_{t_j} = n_{t_j} - 1$
DELETE_EDGE($E_{i,j}$)
end

Algorithm 2: Algorithm INTER

C_{ij} = CREATE_NEW_COMPOSITE(t_i, t_j)
$\text{IC}_{c_{ij}} = \text{MIN}\left(\frac{p_{i,j}}{n_{t_i}}, \frac{p_{i,j}}{n_{t_j}}\right)$
if ($n_{ti} > n_{tj}$)
  $E_{i,ij} = \text{CREATE\_EDGE}(N_i, C_{ij})$
  $p_{i,ij} = \frac{p_{i,j}}{n_{t_i}} + \frac{p_{i,j}}{n_{t_j}}$
else if ($n_{ti} < n_{tj}$)
  $E_{j,ij} = \text{CREATE\_EDGE}(N_j, C_{ij})$
  $p_{j,ij} = \frac{p_{i,j}}{n_{t_j}} + \frac{p_{i,j}}{n_{t_i}}$
end if
for each $N_k \in$ ACG
/ compute edge value*
*between node k, node i, and composite node ij */*
if ($N_k \neq N_i, N_k \neq N_j, N_k \neq C_{ij}$)
  if ($E_{k,i}$ exist)
    if ($E_{k,ij}$ exist)
      $p_{k,ij} = p_{k,ij} + \frac{p_{k,i}}{n_{t_k} + n_{t_i}}$
    else
      $E_{k,ij} = \text{CREATE\_EDGE}(N_k, C_{ij})$
      $p_{k,ij} = \frac{p_{k,i}}{n_{t_k} + n_{t_i}}$
    end if
    $p_{k,i} = \frac{p_{k,i} \times (n_{t_k} + n_{t_i} - 1)}{n_{t_k} + n_{t_i}}$
  end if
end if
/ compute edge value*
*between node k, node j, and composite node ij */*
/ idem that for node i but with node j */*
...
end for
$p_{i,j} = p_{i,j} - \text{IC}_{c_{ij}} - p_{i,ij} - p_{j,ij}$
$n_{t_i} = n_{t_i} - 1$
$n_{t_j} = n_{t_j} - 1$
end

Algorithm 3: Algorithm MAX

C_{ij} =CREATE_NEW_COMPOSITE(t_i, t_j)
$\text{IC}_{c_{ij}} = \frac{p_{i,j}}{n_{t_i} \times n_{t_j}}$
$E_{i,ij} = \text{CREATE\_EDGE}(N_i, C_{ij})$
$p_{i,ij} = \frac{p_{i,j} \times (n_{t_i} - 1)}{n_{t_i} \times n_{t_j}}$
$E_{j,ij} = \text{CREATE\_EDGE}(N_j, C_{ij})$
$p_{j,ij} = \frac{p_{i,j} \times (n_{t_j} - 1)}{n_{t_i} \times n_{t_j}}$
for each $N_k \in$ ACG
/ compute edge value*
*between node k, node i, and composite node ij */*
if ($N_k \neq N_i, N_k \neq N_j, N_k \neq C_{ij}$)
  if ($E_{k,i}$ exist)
    if ($E_{k,ij}$ exist)
      $p_{k,ij} = p_{k,ij} + \frac{p_{k,i}}{n_{t_i}}$
    else
      $E_{k,ij} = \text{CREATE\_EDGE}(N_k, C_{ij})$
      $p_{k,ij} = \frac{p_{k,i}}{n_{t_i}}$
    end if
    $p_{k,i} = \frac{p_{k,i} \times (n_{t_i} - 1)}{n_{t_i}}$
  end if
end if
/ compute edge value*
*between node k, node j, and composite node ij */*
/ idem that for node i but with node j */*
...
end for
$p_{i,j} = \frac{p_{i,j} \times (n_{t_i} - 1)(n_{t_j} - 1)}{n_{t_i} \times n_{t_j}}$
$n_{t_i} = n_{t_i} - 1$
$n_{t_j} = n_{t_j} - 1$
end

Figure 9: Description of the three algorithms.

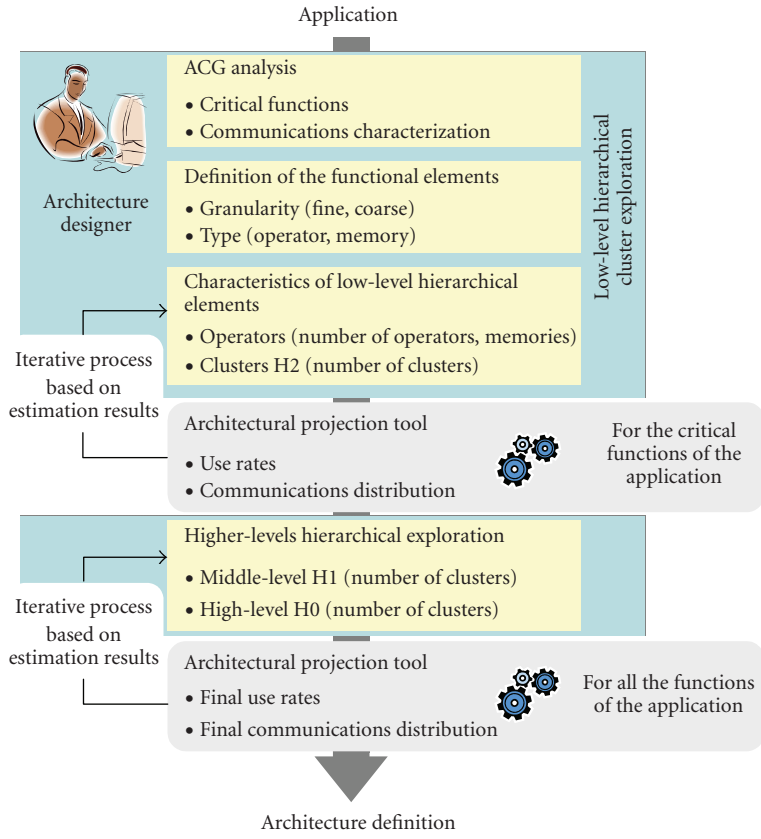


Figure 10: Exploration flow for each critical function and then for the whole application to converge towards a power-efficient architecture.

the different computing and memory resources. The designer uses this information to build the architectural low-level hierarchical elements. At that level, the main issue corresponds to the definition of the granularity and the type of resources of the different functional elements (processing or memory) within the low hierarchical elements.

Then, the designer needs to determine the size of each low-level hierarchical element. For that purpose, the architectural projection tool is used to find the memory size and the number of each functional element embedded within the low-level hierarchical elements. To explore the architecture low level (memory size, number of functional elements), the designer manually changes the architecture model characteristics and launches the architectural projection tool. Designer modifications are based on the results provided by the previous architectural projection runs (architecture processing resources use rate and communications hierarchical distribution). It is an interactive and iterative process between the architectural projection tool results and the designer model modifications. As mentioned previously, we have developed an architectural model that enables a fast modification of the architecture characteristics. Furthermore, the architectural projection algorithm even if in $O(n^2)$ is very fast to compute a performance estimation as the number of edges is small. These two points are essential to mitigate the cost of the exploration process and to enable the designer to rapidly evaluate several architectures. Once a good size is found for each low-level hierarchical element, an exploration of the architecture higher hierarchical levels can be performed in order to complete the exploration process. When an efficient architecture is obtained for each critical function, a trade-off between all the architectures is define-based on the designer analysis of the solutions. These successive architectural exploration steps are highlighted in the next section, which gives the architectural exploration results for several applications for the image computing domain and the cryptographic domain.

6. APPLICATIONS

Four applications from the image processing and cryptography domains are considered to illustrate our exploration process. For each application, a power-efficient architecture has been targeted. The exploration process has mainly focused on the granularity of the processing and memory architectural resources. Each application has been specified in C language before being automatically translated into an HCDFG description.

This section is organized as follows: first, main characteristics of the considered image processing and cryptographic applications are given. The results of the system estimation tool for each application are presented before defining a power-efficient architecture for the two application domains.

Table 2: System estimation results.

Application	Comm.	Cycles	ADD/SUB	MUL/DIV	Comp.	Logic	Memory
ICAM	29.086.835	373.872.291	512	125	516	328	3.1 Mbytes
MPEG-2 encoder	40.745.280	45.476.864	398	279	153	33	60 Kbytes
Matching pursuit	3.751.397	239.215	232	162	69	0	6.3 Mbytes
AES core	1120	471	11	16	16	15	1 Kbytes

For each application and architecture, the architectural communications distribution estimates are given. The architectural resource use rate estimates are also provided in order to perform a whole analysis of the results.

6.1. Image processing applications

Three image processing applications have been used within our framework.

(i) *ICAM* (intelligent camera) is a motion estimation by intensity difference and reference background update [30]. This camera is used for subway supervision and crowd motion management in an urban environment.
(ii) *Matching pursuit* is an image compression application [31]. matching pursuit encoder is based on a genetic algorithm and can be implemented onto different platforms. Therefore, we work only on the decoder.
(iii) *MPEG-2* is a compression standard [32], which allows for the coding of studio quality video for digital TV, high-density CD-ROMs, and TV-broadcasting. We study only the encoder part of the MPEG-2 system.

6.2. Cryptography application

In order to not only confront our method to image processing applications, we have tested the method with a cryptography algorithm. We have chosen the last international advanced encryption standard AES.

AES algorithm has been developed to replace the DSE standard with a 128-bit key [33]. We have chosen the AES specification with 10 rounds and have not taken into account the key generator. We have focused on finding an efficient architecture for the cryptographic core.

6.3. System estimation tool results

The system estimation tool is used to perform the first step of the exploration process. Table 2 provides the selected results for each benchmark among the solutions provided by the tool. Table 2 design characteristics are the estimated number of communications within the application and the number of cycles to perform the application using the allocated number of processing and memory resources. The designer uses Table 2 to define an RTL logical architecture able to support the application. For that point, two ways are possible depending on the execution model: a sequential execution model which requires dynamic reconfiguration of the architecture or a pipeline execution model [5]. In the first case, the designer considers that the architecture is reconfigured with the adequate function at each step of the application execution. Hence, the reconfigurable architecture supports just one function at a time, in which case, each function can be implemented with a high degree of parallelism since a single function is running at a time. However, the dynamic reconfiguration consumes some time and power. We do not consider the dynamic reconfiguration process in our exploration (like previous efforts presented in Section 2). We consider the second execution model; pipeline execution. In that case, all the functions are implemented onto the architecture and during the whole execution time (any dynamic or partial reconfiguration). The function realization must be less parallel than in the first execution model due to area limitation. With this assumption, Table 2 shows that image computing applications are more resource consuming than cryptographic application. ICAM is the most complex application concerning the number of processing resources.

6.4. Architectural exploration results

Table 3 provides the results of the ACG study for the critical functions only. This study provides the average percentage of communications in the ACG between the following:

(i) the processing coarse-grain resources (intercoarse grain, Table 3-column3),
(ii) the processing coarse-grain resources and the processing fine-grain resources (coarse grain/fine grain, Table 3-column4),
(iii) the processing coarse-grain resources and the memory resources (coarse grain/memory, Table 3-column5),
(iv) the processing fine-grain resources (interfine grain, Table 3-column6),
(v) the processing fine-grain resources and the memory resources (fine grain/memory, Table 3-column7),
(vi) the memory resources (intermemory, Table 3-column8).

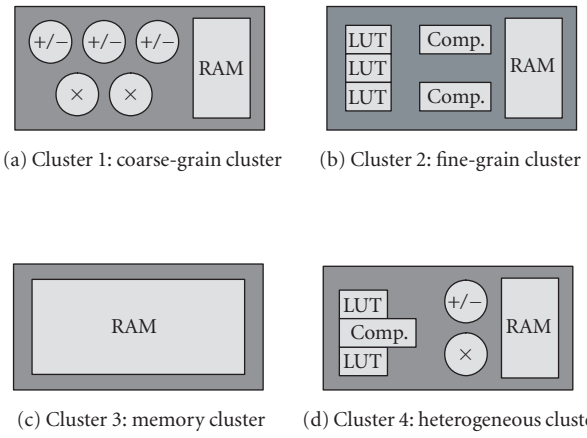
According to the flow presented in Figure 10, these results guide the designer to define a low-level cluster and particularly to identify if it is necessary or not to mix fine-grain and coarse-grain processing resources within the low-level clusters. These results show that it is efficient to build an architecture with separate fine-grain and coarse-grain clusters for the three first applications in order to have the maximum of fine grain processing resources in the same low-level hierarchical element (and the same thing for the coarse-grain processing resources). However, it is not the case for the last application, the AES core, since there is a large part of communications between coarse-grain and fine-grain resources and no intercommunications.

Table 3: Critical function ACG communication characterization.

Application	Number of critical functions	Intercoarse grain	Coarse-grain Fine grain	Coarse-grain memory	Interfine grain	Fine grain memory	Inter-memory
ICAM	2	2, 9%	0, 2%	15, 0%	19, 9%	36, 9%	25, 1%
MPEG-2 encoder	2	66, 9%	0, 7%	31, 1%	1, 1%	0, 2%	—
Matching pursuit	1	92, 3%	0, 1%	7.6%	—	—	—
AES core	1	—	15, 7%	28, 9%	—	25, 0%	30, 4%

According to these results, the designer can define four clusters that correspond to the atomic clusters of the final architecture for each application. Figure 11 provides a schematic representation of the four clusters. In this figure, the number of each functional element (processing or memory) is not relevant since this number is defined later in the exploration process.

(i) Cluster 1 has two coarse-grain processing functional element types, adder/subtracter and multiplier, and one memory functional element. It is a coarse-grain cluster (Figure 11(a)).
(ii) Cluster 2 has two fine-grain processing functional element types, comparator and lookup table, and one memory functional element. It is a fine-grain cluster (Figure 11(b)).
(iii) Cluster 3 only has one large memory functional element, often used to store a complete picture in the case of image computing application. It is a memory cluster (Figure 11(c)).
(iv) Cluster 4 has four processing functional element types, adder/subtracter, multiplier, comparator and lookup table, and one memory functional element. It is a heterogeneous cluster (Figure 11(d)).



(a) Cluster 1: coarse-grain cluster (b) Cluster 2: fine-grain cluster

(c) Cluster 3: memory cluster (d) Cluster 4: heterogeneous cluster

Figure 11: The four potential low-level clusters for the applications. Cluster 1, Cluster 2, and Cluster 3 are used for the three image processing applications. Cluster 4 is only used for the cryptographic core.

The architectural exploration leads to define the following:

(i) the number of processing functional elements for each type of functional element embedded in the low-level hierarchical cluster,
(ii) the size of the memory functional elements embedded in the low-level hierarchical cluster,
(iii) the number of low-level hierarchical clusters in the middle level hierarchical cluster,
(iv) and the number of middle-level hierarchical clusters in the high-level hierarchical cluster.

To perform the architectural exploration, the exploration rules based on the model hypothesis have to be satisfied; the communication costs inside a hierarchical element are homogeneous and the communications have less impact on the power consumption for low level of hierarchy than for high level of hierarchy.

Table 4 provides the number of processing functional elements embedded in each cluster (cluster 1, cluster 2, cluster 3, and cluster 4) for each application. According to the ACG study, cluster 4 is only used for the cryptography application. The image computing applications use the three other clusters. Concerning the ICAM application, two lines of Table 5 give the exploration results for two architectures (archi1 and archi2). The main difference between the two architectures is the cluster size. We use two different architectures in order to demonstrate that it is possible to obtain very good results with a nonrealistic architecture (archi2). This point will be discussed in the following section. Table 4 has to be jointly considered with Table 5 that gives the number of low-level hierarchical clusters. ICAM and MPEG-2 applications are the most complex applications, so they need more low-level clusters and larger clusters than the other two applications. AES application is less complex, so the cluster used for this application is smaller.

Once the designer has defined the low-level hierarchical clusters' size and number, he explores the middle level, and the high level of the hierarchy. The middle level embeds hierarchical elements like cluster 1, cluster 2, cluster 3, and cluster 4 (only for the AES application). The number of each cluster type in the middle-level for each application is given from row 2 to row 5 in Table 6. Row 6 provides the number of middle-level hierarchical elements embedded only in the high-level element.

Tables 4, 5, and 6 define the architectural exploration results for each application (with two possible architectures for ICAM application). These results provide the designer with an estimation of processing functional elements use rate and an estimation of the communications hierarchical distribution. The following section details these estimations.

Table 4: Processing functional element number embedded in the low hierarchical level cluster.

Application	Number of ADD/SUD in cluster 1	Number of MUL in cluster 1	Number of COMP in cluster 2	Number of LUT in cluster 2	Number of ADD/SUD in cluster 3	Number of MUL in cluster 3	Number of COMP in cluster 3	Number of LUT in cluster 3
ICAM archi1	4	1	5	2	—	—	—	—
ICAM archi2	20	10	21	13	—	—	—	—
MPEG-2 encoder	4	4	2	1	—	—	—	—
Matching pursuit	8	6	3	0	—	—	—	—
AES core	—	—	—	—	1	1	1	1

Table 5: Number of low hierarchical level clusters.

Application	Number of cluster 1	Number of cluster 2	Number of cluster 3	Number of cluster 4
ICAM archi1	130	234	26	0
ICAM archi2	26	26	26	0
MPEG-2 encoder	105	105	0	0
Matching pursuit	30	25	5	0
AES core	0	0	0	16

6.5. Estimation results

Table 7 provides the use rate estimations of each type of processing functional element for each application. The designer targets the highest use rate because unused resources reduce the power efficiency, particularly for coarse-grain processing resources. However, the problem is more complex because often the designer must choose a tradeoff between use rate and communications distribution (highest number of communications in the architecture hierarchical low level). For example, we have defined an architecture with a very high use rate for the matching pursuit application (Table 7 line 4) and an architecture with a lower use rate for MPEG-2 decoder application (Table 7 line 5). The issue is now to analyze the communications hierarchical distribution, since it also has a significant impact on the final performances. Table 8 provides the communications hierarchical distribution estimation. The number of communications estimated in the low level is higher for the MPEG-2 decoder application than for the matching pursuit application. Moreover, the number of communications estimated in the high level of hierarchy is lower for the MPEG-2 decoder application than for the matching pursuit application. The communications hierarchical distribution is better for the MPEG-2 decoder than for the matching pursuit application, but as we have seen, this is not the case for the use rate. Hence, the designer must choose the best solution in terms of tradeoff between use rate and communications hierarchical distribution according to the technological process used for his architecture.

To provide a schematic representation of the architecture dedicated for the MPEG-2 decoder, Figure 12 gives a representation of the three hierarchical levels: high level (Figure 12(a)), middle level (Figure 12(b)), and low level (Figure 12(c)).

Concerning the ICAM application, Table 8 shows that the estimation results obtained with the archi2 are better than with the archi1. Nevertheless, as seen in Table 5, the low-level clusters are five times larger on the average. With such an archi2 large cluster, it is difficult for the designer to find a solution that guarantees that the communication cost is homogeneous within the cluster. Therefore, the archi2 is not a realistic architecture except if a communication technology enables providing homogeneous cost within the cluster in terms of delay and power.

Concerning the AES application, it uses another type of architecture than the image computing applications. The architecture for the AES application has only one low-level cluster type with fine-grain and coarse processing functional element (heterogeneous cluster). As for the MPEG-2 dedicated architecture, Figure 13 presents a schematic representation of the three hierarchical levels of the architecture: high level (Figure 13(a)), middle level (Figure 13(b)), and low level (Figure 13(c)). Table 8 shows that the estimation results of the communication distribution are very good for this application with 69% of the communications in the architectural low level and only 21% in the high level. However, before concluding that this method leads to define an efficient architecture for several application domains, it is important to estimate the communications distribution with the architecture highlighted for image computing where the coarse-grain and fine-grain processing functional elements are separated. The last line in Table 8 provides the estimation results in this case (AES core ic-archi). The number of communications in the architectural low level is reduced by 19% and the number of communications in the architectural high level is increased by 15%. Therefore, the architecture defined for the image computing applications in not adapted for the cryptography application. It shows that according to the discussion in Section 1.2, this method enables the definition of dedicated reconfigurable architectures for different application domains.

7. CONCLUSION

Design space exploration for reconfigurable architectures combined with algorithmic exploration of applications is an important issue which has been insufficiently addressed till now. We propose in this paper an original approach based

Table 6: Exploration results of middle and high hierarchical levels.

Application	Middle-level hierarchical element				High-level hierarchical element
	Number of cluster 1	Number of cluster 2	Number of cluster 3	Number of cluster 4	
ICAM archi1	5	9	1	0	26
ICAM archi2	2	2	2	0	13
MPEG-2 encoder	7	7	0	0	15
Matching pursuit	6	5	1	0	5
AES core	0	0	0	4	4

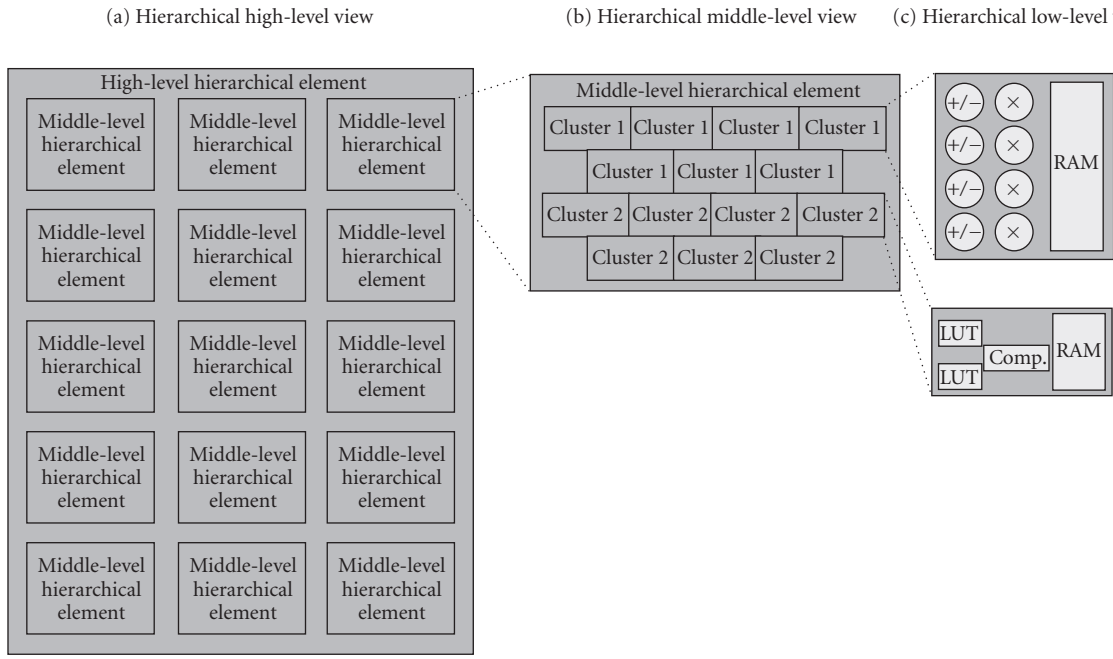


Figure 12: Schematic representation of special reconfigurable architecture for the MPEG-2 application.

Table 7: Use rate estimation of each processing functional element type.

Application	ADD/SUB	MUL	COMP	LUT
ICAM archi1	98,5%	96,1%	36,7%	70,1%
ICAM archi1	98,5%	48,0%	94,5%	97,0%
MPEG-2 encoder	67,0%	70,0%	13,0%	2,0%
Matching pursuit	97,0%	90,0%	92,0%	—
AES core	63,8%	100%	100%	93,8%

Table 8: Hierarchical distribution communication estimation.

Application	High level	Middle level	Low level
ICAM archi1	28%	35%	37%
ICAM archi2	13%	30%	57%
MPEG-2 encoder	29%	8%	63%
Matching pursuit	31%	32%	37%
AES core	21%	10%	69%
AES core ic_archi	36%	14%	50%

on a high-level representation of the application and on a hierarchical functional model for the architecture. Our approach targets fine-grain, coarse-grain, and heterogeneous architectures.

To perform the exploration of the architecture space, two metrics have been defined, the architectural processing use rate and the communications hierarchical distribution since we have shown (particularly with fine-grain architecture studies) that these metrics are significant in reducing the power consumption of an application under a given time constraint. The exploration process leads to the definition of a power-efficient hierarchical reconfigurable architecture for an application or an applications family. We have demonstrated the efficiency of our approach for image processing and cryptography applications. In order to provide the designers with estimates of the achievable performances, we have defined an estimation technique that computes an interval of performance. This point is important and more relevant than an optimal estimation technique considering the level of abstraction of our approach. The goal is to greatly

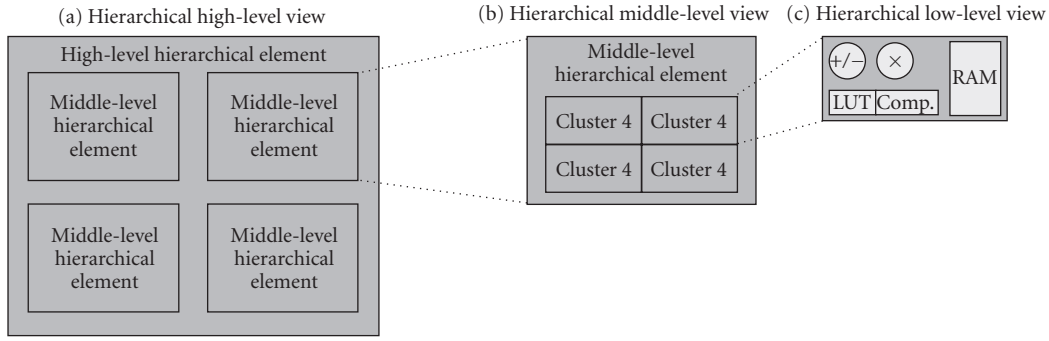


Figure 13: Schematic representation of dedicated reconfigurable architecture for the AES application.

prune the design space in order to shorten the design cycle and to rapidly converge towards the definition of a power-efficient reconfigurable architecture. The estimation results demonstrate that our approach rapidly leads to defining a power-efficient architecture for an applications domain. This point is essential since it is a current trend to specialize the reconfigurable architectures for a specific domain.

REFERENCES

[1] N. Tredennick and B. Shimamoto, "The rise of reconfigurable systems," in *Proceedings of the International Conference on Engineering of Reconfigurable Systems and Algorithms (ERSA '03)*, pp. 3–12, Las Vegas, Nev, USA, June 2003.

[2] R. Hartenstein, "A decade of reconfigurable computing: a visionary retrospective," in *Proceedings of Conference and Exhibition on Design, Automation and Test in Europe (DATE '01)*, pp. 642–649, Munich, Germany, March 2001.

[3] P. Schaumont, I. Verbauwhede, K. Keutzer, and M. Sarrafzadeh, "A quick safari through the reconfiguration jungle," in *Proceedings of the 38th Design Automation Conference (DAC '01)*, pp. 172–177, Las Vegas, Nev, USA, June 2001.

[4] A. D. Pimentel, L. O. Hertzberger, P. Lieverse, P. van der Wolf, and Ed. F. Deprettere, "Exploring embedded-systems architectures with artemis," *Computer*, vol. 34, no. 11, pp. 57–63, 2001.

[5] L. Bossuet, *Exploration de l'espace de conception des architectures reconfigurables*, Ph.D. thesis, Université de Bretagne Sud, Vannes, France, September 2004.

[6] M. Gries, "Methods for evaluating covering the design space during early design development," Technical Memorandum MO3/32, Electronics Research Laboratory, University of California, Berkeley, Calif, USA, August 2003.

[7] V. Betz and J. Rose, "VPR: a new packing, placement and routing tool for FPGA research," in *Proceedings of the 7th International Workshop on Field Programmable Logic (FPL '97)*, pp. 213–222, Oxford, UK, September 1997.

[8] E. Ahmed and J. Rose, "The effect of LUT and cluster size on deep-submicron FPGA performance and density," in *Proceedings of ACM/SIGDA International Symposium on Field Programmable Gate Arrays (FPGA '00)*, pp. 3–12, Moterey, Calif, USA, February 2000.

[9] S. J. E. Wilton, J. Rose, and Z. G. Vranesic, "The memory/logic interface in FPGA's with large embedded memory arrays," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 7, no. 1, pp. 80–91, 1999.

[10] L. Lagadec, *Abstraction, modélisation et outils de CAO pour les circuits intégrés reconfigurables*, Ph.D. thesis, Université de Rennes1, Rennes, France, 2000.

[11] S. Choi, J. W. Jang, S. Mohanty, and V. K. Prasanna, "Domain-specific modeling for rapid system-level energy estimation of reconfigurable architectures," in *Proceedings of International Conference of Engineering of Reconfigurable Systems and Algorithms (ERSA '02)*, Las Vegas, Nev, USA, June 2002.

[12] R. Enzler, T. Jeger, D. Cottet, and G. Tröster, "High-level area and performance estimation of hardware building blocks on FPGAs," in *Proceedings of the the Roadmap to Reconfigurable Computing, 10th International Workshop on Field-Programmable Logic and Applications (FPL '00)*, pp. 525–534, Villach, Austria, August 2000.

[13] C. A. Moritz, D. Yeung, and A. Agarwal, "Exploring optimal cost-performance designs for Raw microprocessors," in *Proceedings of IEEE Symposium on FPGAs for Custom Computing Machines (FCCM '98)*, pp. 12–27, Napa Valley, Calif, USA, April 1998.

[14] U. Nadelginder, *Coarse-grain reconfigurable architecture design space architecture exploration*, Ph.D. thesis, University of Kaiserslautern, Kaiserslautern, Germany, June 2001.

[15] R. Kress, *A fast reconfigurable ALU for xputers*, Ph.D. thesis, University of Kaiserslautern, Kaiserslautern, Germany, 1996.

[16] L. Bossuet, G. Gogniat, and J.-L. Philippe, "Generic design space exploration for reconfigurable architectures," in *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS '05)*, p. 163, Denver, Colo, USA, April 2005.

[17] Design Trotter Project: http://web.univ-ubs.fr/lester/~diguet/Design-TrotterPage.html.

[18] J. P. Diguet, G. Gogniat, P. Danielo, M. Auguin, and J.-L. Philippe, "The SPF model," in *Proceedings of Forum on Design Language (FDL '00)*, Tübingen, Germany, September 2000.

[19] Y. Le Moullec, P. Koch, J. P. Diguet, and J.-L. Philippe, "Design trotter: building and selecting architectures for embedded multimedia applications," in *Proceedings of IEEE International Symposium on Consumer Electronics (ISCE '03)*, Sydney, Australia, December 2003.

[20] Y. Le Moullec, J. P. Diguet, T. Gourdeaux, and J.-L. Philippe, "Design trotter: system-level dynamic estimation task a 1st step towards platform architecture selection," *Journal of Embedded Computing*, vol. 1, no. 4, pp. 565–586, 2005.

[21] L. Bossuet, G. Gogniat, and J.-L. Philippe, "Fast design space exploration method for reconfigurable architectures," in

Proceedings of the International Conference on Engineering of Reconfigurable Systems and Algorithms (ERSA '03), pp. 65–71, Las Vegas, Nev, USA, June 2003.

[22] T. Mudge, "Power: a first-class architectural design constraint," *Computer*, vol. 34, no. 4, pp. 52–58, 2001.

[23] S. Rouxel, "Caractérisation de l'impact du routage sur les performances (vitesse et consommation de puissance) d'un FPGA," M.S. thesis, Université de Bretagne Sud, Lorient, France, September 2003.

[24] D. Elleouet, "Caractérisation et modélisation de la consommation de puissance des mémoires sur FPGA," M.S. thesis, Université de Bretagne Sud, Lorient, France, September 2003.

[25] A. Garcia, W. Burleson, and J.-L. Danger, "Power modelling in field programmable gate arrays (FPGA)," in *Proceeding of the 9th International Workshop on Field Programmable Logic and Applications (FPL '99)*, pp. 396–404, Glasgow, Scotland, August-September 1999.

[26] V. George, H. Zhang, and J. Rabaey, "The design of a low energy FPGA," in *Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED '99)*, pp. 188–193, San Diego, Calif, USA, August 1999.

[27] E. Kusse and J. M. Rabaey, "Low-energy embedded FPGA structures," in *Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED '98)*, pp. 155–160, Monterey, Calif, USA, August 1998.

[28] L. Shang, A. S. Kaviani, and K. Bathala, "Dynamic power consumption in virtex^{TM}-II FPGA family," in *Proceedings of the 10th ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA '02)*, pp. 157–164, Monterey, Calif, USA, February 2002.

[29] K. K. W. Poon, A. Yan, and S. J. E. Wilton, "A flexible power model for FPGAs," in *Proceeding of the 12th International Conference on Field-Programmable Logic and Applications (FPL '02)*, pp. 312–321, Montpellier, France, September 2002.

[30] H. Zhang, M. Wan, V. George, and J. Rabaey, "Interconnect architecture exploration for low-energy reconfigurable single-chip DSPs," in *Proceedings of the IEEE Computer Society Workshop on VLSI (WVLSI '99)*, p. 2, Orlando, Fla, USA, April 1999.

[31] CEA. Intelligent Camera—3D Methodology. http://www-list.cea.fr/fr/programmes/systemes_embarques/docs/ICAM_internet_list_v0.pdf.

[32] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[33] MPEG2, http://www.mpeg2.de.

2. Article concernant la sécurité des composants FPGA

L. Bossuet, G. Gogniat, W. Burleson,

Dynamically Configurable Security for SRAM FPGA Bitstreams,

International Journal of Embedded Systems, IJES, From Inderscience Publishers 2006 - Vol. 2, No.1/2, pp. 73 - 85

Dynamically configurable security for SRAM FPGA bitstreams

Lilian Bossuet* and Guy Gogniat

LESTER laboratory of Université de Bretagne Sud,
56321 Lorient, France
Fax: 332 97 87 45 27 E-mail: bossuet@ixl.fr
E-mail: guy.gogniat@univ-ubs.fr
*Corresponding author

Wayne Burleson

Electrical and Computer Engineering Department,
University of Massachusetts, Amherst, MA 01003, USA
Fax: 413-545-1993 E-mail: burleson@ecs.umass.edu

Abstract: FPGAs are becoming increasingly attractive – thanks to the improvement of their capacities and their performances. Today, FPGAs represent an efficient design solution for numerous systems. Moreover, since FPGAs are important for electronic industry, it becomes necessary to improve their security, particularly for SRAM FPGAs, since they are more vulnerable than other FPGA technologies. This paper proposes a solution to improve the security of SRAM FPGAs through flexible bitstream encryption. This proposition is distinct from other works because it uses the latest capabilities of SRAM FPGAs like partial dynamic reconfiguration and self-reconfiguration. It does not need an external battery to store the secret key. It opens a new way of application partitioning oriented by the security policy.

Keywords: field programmable gate arrays; design security; bitstream encryption; partial reconfiguration and self-reconfiguration; reconfigurable architecture.

Reference to this paper should be made as follows: Bossuet, L., Gogniat, G. and Burleson, W. (xxxx) 'Dynamically configurable security for SRAM FPGA bitstreams', *Int. J. Embedded Systems*, Vol. x, No. x, pp.xxx–xxx.

Biographical notes: Lilian Bossuet is a PhD student in Electrical Engineering in LESTER laboratory of the Université de Bretagne Sud, Lorient France, where he has been since 2001. He has a BSEE from the ENSEA Cergy-Pontoise, France. He has a MSEE from the INSA-Université de Rennes, France. He has succeeded an Electrical Engineering competitive examination for teacher training from ENS Cachan. He has worked as a Tool Engineer for Autoliv Electronics. He was a Visitor (summer 2003) at the University of Massachusetts Amherst USA. His researches are in reconfigurable computing and particularly design space exploration for coarse-grained reconfigurable architecture. He also conducts research in high methodology and tools for SoC, FPGA utilisation and performances estimation, FPGA security.

Guy Gogniat is an Associate Professor of Electrical and Computer Science at the University of Bretagne Sud, Lorient, France, where he has been since 1998. He has a BSEE from the FIUPSO Orsay, France and a MSEE from the University of Paris Sud Orsay and a PhD in ECE from the University of Nice-Sophia Antipolis France. His researches are in the general area of CAD and reconfigurable computing, including codesign methodologies and software radio platform exploration with funding from national research projects (AS, RNTL, RNRT) and national and international companies and organisations (CNRS, CEA, THALES, MITSUBISHI …). He also conducts research in high-level methodologies and tools for FPGA utilisation, performance estimation and FPGA security.

Wayne Burleson is an Associate Professor of Electrical and Computer Engineering at the University of Massachusetts Amherst where he has been since 1990. He has a BSEE and MSEE from MIT and a PhD in ECE from the University of Colorado. His research is in the general area of VLSI and Signal Processing, including circuits for low-power, long interconnects, clocking and mixed signals with funding from NSF, SRC, Compaq/HP and Intel. He also conducts research in reconfigurable computing, content-adaptive signal processing, smart cards and multimedia instructional technologies. He is a member of the ACM, ASEE, Sigma Xi, a senior member of the IEEE Society.

1 Introduction

The FPGA (Field Programmable Gate Array) concept was born during the 80s, when the configuration point size (transistors or fuses) was too large in comparison with the chip size to have an interesting FPGA density. Therefore, these devices were just used to do prototyping or glue logic. For a long time, the FPGAs have not taken benefit from the best deep-submicronic technology, today the more advanced FPGAs use 90 nanometer technology with copper metallisation (best actual accessible technology). With the improvement of technological processes and since the FPGAs structure is very regular, it is possible to build some FPGAs with more than one million transistors. Thanks to these evolutions, FPGAs are increasingly attractive for numerous systems and to build efficient SoC (System on a Chip). The FPGAs market continues to increase and FPGAs are capturing the classical market share of ASIC (Application Specific Integrated Circuit) market. The cost crossover point, which permits to know the necessary number of systems built to choose an efficient ASIC solution, is increasingly far (Tredennick and Shimamoto, 2003). It is possible even for a large number of systems built to choose an economically efficient FPGA solution.

Since FPGAs are becoming so important for the electronic industry, it is necessary to think about the security of FPGA-based systems. It is possible to consider the FPGA-based systems' security problem in three ways.

1.1 Security system using FPGA

In this case, FPGA is used as a part of the security system. The FPGA dynamic reconfiguration improves the security system's flexibility. Therefore, it is possible to change the classical software update by hardware update in order to prevent attacks evolutions.

For example, internet-connected hosts are now frequently attacked by malicious machines located around the world. Hosts can be protected from remote machines by filtering the traffic through a firewall. Use of an FPGA can be very efficient for such application in order to build less static system. In Lockwood et al. (2003) a System-On-Programmable-Chip (SOPC), internet firewall has been implemented that protects high-speed networks from present and future threats. The high level of flexibility and extensibility required by such systems is guaranteed by the use of an FPGA (in Lockwood et al. (2003) authors use a Xilinx Virtex FPGA).

In the same way, in Dandalis and Prasanna (2000), the authors use Xilinx FPGA to develop an Adaptive Cryptographic Engine (ACE) for Internet Protocol Security (IPSec) architectures. Several FPGA configurations of cryptographic algorithms are stored in a memory in the form of cryptographic library. The FPGA is configured on-demand based on the cryptographic library and then performs the required encryption/decryption tasks.

We think that it is also possible to use the FPGA concept (e.g., reconfiguration, hardware update) for smart cards system or PAY-TV, for example. However, today, there is no published work on these applications.

1.2 Protecting FPGA data

In this case, it is necessary to protect the application that runs on FPGA. The data inside the circuit and the data transferred to/from the peripheral circuits during the communication must be protected. The main solution is to integrate data encryption scheme inside the FPGA. These circuits are attractive for executing the actual cryptographic algorithms and are of particular importance from security point of view. There has been a large amount of work done dealing with the algorithmic and computer architecture aspects of cryptographic schemes implemented on FPGA over the last five years. According to Wollinger et al. (2004) and Wollinger and Paar (2003), we can list the potential advantages of FPGA in cryptographic applications.

- *Algorithm agility.* This term refers to the cryptographic algorithms switching during operation of the targeted application. While algorithm agility is costly with traditional hardware, FPGA can be reprogrammed on the fly.
- *Algorithm upload.* It is perceivable that fielded devices are upgraded with a new encryption algorithm. FPGA-equipped encryption devices can upload the new configuration code.
- *Architecture efficiency.* In certain cases hardware architecture can be much more efficient if it is designed for a specific set of parameters. An example for the parameters for cryptographic algorithms can be the key. FPGA allows this type of devices and optimisations with a specific parameter set. Owing to the nature of FPGA, the application can be changed totally or partially.
- *Resource efficiency.* The majority of security protocols are hybrid protocols that need several algorithms. As they are not used simultaneously, the same FPGA device can be used for both through run-time reconfiguration.
- *Algorithm modification.* There are applications that require modification of standardised cryptographic algorithms.
- *Throughput.* General-purpose microprocessors are not optimised for fast execution. Although, typically slower than ASIC implementations, FPGA implementations have the potential of running substantially faster than software implementations (as with a processor).

- *Cost efficiency.* There are two cost factors, which have to be taken into consideration when analysing the cost efficiency of FPGAs: cost of development and unit price. The costs to develop an FPGA implementation of a given algorithm are much lower than that for an ASIC implementation. The unit prices are not significant when compared with the developmental costs. However, for high-volume applications (more than one million of circuit build) ASIC solution usually becomes the more cost-efficient choice.

1.3 FPGA design security

In this last case, the protection concerns the design against cloning and reverse engineering. It is custom intellectual property protection. Concerning the SRAM FPGAs, the design security corresponds to the way to protect the bitstream or the FPGA configuration.

This paper focuses on the latter case dealing with FPGA design security. If the FPGA design itself is not secure, the other security problems cannot be efficiently treated. Using an unsecured device embedded in a security system is not security-efficient. Many works already proposed solutions to protect the bitstream. However, the contribution of this paper relies on the utilisation of the latest improvements of SRAM FPGAs configuration techniques to answer the security problem.

This paper is organised as follows. Section 2 describes some aspects of the design security problem such as the classical hardware devices security level. Section 3 presents several works dealing with the protection of SRAM FPGA configuration. Section 4 describes the new capability of SRAM FPGA self-reconfiguration. In Section 5 a new SRAM FPGA bitstream protection solution is proposed. The drawbacks and advantages of the proposed solutions are given in Section 6. Section 7 compares the different solutions of design security for FPGA. Finally, Section 5 concludes this paper and exposes several future directions.

2 Design security

It is interesting, before investigating the different solutions to secure the configuration of SRAM FPGAs, to list what are the different attacks against an integrated circuit today, what is the protection level of some current circuits and why do they have this level of protection?

2.1 Need for design security

The problem of design security is simple; the designer does not want a competitor to be able to pirate his design. There are two sorts of piracy.

- *Cloning.* When a competitor makes an exact copy of a design including the board layout and chip, and when he is able to create a copy of the pirated system.
- *Reverse engineering*. When a competitor copies a design by reconstructing a 'schematic' or net list level representation. In this process, he analyses and understands how the design works and how to improve it, or to modify it with malicious intents. Reverse engineering generally consists of the following stages:
  - analysis of the product
  - generation of an intermediate level product description
  - human analysis of the product description to produce a specification
  - generation of a new product using the specification.

Therefore, reverse engineering is more serious than cloning. These two aspects correspond to different attacks, and the design security must protect the system against both attacks. To perform cloning or reverse engineering, two types of attack can be considered; the non-invasive and the invasive attacks.

The non-invasive attacks gather all the methods that use external means. For example, the attackers can use all the possibilities of the circuit inputs in order to obtain all the different outputs and draw the system truth table; this method is called 'Black Box Attack'.

In the case of SRAM FPGA, a simple attack method is intercepting the bitstream between the root ROM and the FPGA when the system power is switched on. More complex attacks can be brought into play; time, power and electromagnetic changes and measures like the simple or differential power analysis – interested readers can refer to the works on power analysis of FPGA in Standaert et al. (2003, 2004) and Örs et al. (2003).

The invasive attacks (or *physical attacks*) are characterised by the necessity to destroy the integrated circuit (component package) to study the chip (design inside the component) with some complex methods. For example, it is possible to use laser cutter microscope in order to split the chip in several slices and understand the chip layout. These attacks can use sophisticated tools like optical microscope, mechanical probes and even Focused Ion Beam (FIB). As these attacks use the weakness of the silicon technology, when they are possible, it is very hard to secure the system against them.

The paper Anderson and Kuhn (1996, 1997) give some information about these different attacks. It is possible to classify the integrated circuits according to their protection against the different types of attacks. The next section presents an example of security level classification.

2.2 Protection level of some circuits

The level of protection offered by actual integrated circuits is an interesting metric to identify works that must be carried out to improve the security level of one particular type of integrated circuit. In the IBM Systems Journal, a paper Abraham et al. (1991) defines the various security

levels for modern electronic systems and the corresponding taxonomy of attackers.

- *Level 0 (ZERO)*. No special security features added to the system. It is easy to comprise the system with low cost tools.
- *Level 1 (LOW)*. Some security features in place. They are relatively easily defeated with common laboratory or shop tools.
- *Level 2 (MODLOW)*. The system has some security against non-invasive attacks; it is protected for some invasive attacks. More expensive tools are required, as well as specialised knowledge.
- *Level 3 (MOD).* The system has some security against non-invasive and invasive attacks. Special tools and equipment are required, as well as some special skills and knowledge. The attack may become time-consuming but will eventually be successful.
- *Level 4 (MODH).* The system has strong security against attacks. Equipment is available but is expensive to buy and operate. Special skills and knowledge are required to use the equipment for an attack. More than one operation may be required so that several adversaries with complementary skills would have to work on the attack sequence. The attack could be unsuccessful.
- *Level 5 (HIGH).* The security features are very strong. All known attacks have been unsuccessful. Some research by a team of specialists is necessary. Highly specialised equipment is necessary, some of which might have to be designed and built. The success of the attack is uncertain.

According to this classification, it is possible to give a general security level for the current integrated circuits. Of course, these different levels are not fixed and depend of the factory and the type of circuit (in the same factory there are several families and some of them can be especially security-efficient like some military families). The authors have tried to give one level by classical integrated circuit and explain the reason of their choices. The security level of the classical integrated circuits is given in Table 1.

Table 1 Security level of classical integrated circuits

Integrated circuit	*Security level*
Conventional SRAM FPGA	0
ASIC gate array	3
Cell-based ASIC	3
SRAM FPGA with bitstream encryption	3
Flash FPGA	4
Antifuse FPGA	4

Conventional SRAM FPGAs have the lowest security level. These circuits need a bitstream transfer from the root ROM at power up (because the memory of configuration is a SRAM volatile memory). Therefore, it is easy for the pirate to read with a simple probe the bitstream during the transfer. The conventional SRAM FPGAs are inefficient for safe design. However, with a bitstream encryption it is possible to clearly improve the security level since the security weakness is secure. SRAM FPGAs have a good resistance against some attacks like power analysis (Standaert et al., 2003). Today few works present the results of attacks against SRAM FPGA (Örs et al., 2003 and Standaert et al., 2004).

Often considered like a secure technology, ASICs are actually relatively easy to reverse engineer. Because, unlike FPGAs, ASICs do not have switch. Therefore, it is possible to strip the chip to copy with certitude the complete layout in order to understand how it works. Methods to reverse engineer ASIC exist. The cost of reverse engineering is high since the tools required are expensive and the process is time consuming. Therefore, it is not a simple process and therefore the security level is 3 for such devices.

Contrary to the ASICs, the FPGAs, like antifuse or flash, are actually security-efficient since they are based on switches. With these FPGAs, no bitstream can be intercepted in the field (no bitstream transfer, no external configuration device). In the case of antifuse FPGAs, the attacker needs a Scanning Electron Microscope (SEM) in order to know the state of each antifuse. Nevertheless, the difference between a programming and a non-programming antifuse is very difficult to see. Moreover, such analysis is intractable in a device like Actel AX2000 that contains 53 million of antifuses and according to Actel (www.actel.com/products/rescenter/security/index.html) only 2–5% (average) of these antifuses are programmed. For flash FPGA, there is no optical difference after configuration, so the invasive attacks are very complex. The same advantages are given by QuickLogic to promote their flash FPGAs with the ViaLink technology (QuickLogic, 2002).

If the antifuse and the flash FPGAs are very security-efficient, they are just one time configurable (or one time programmable), so they are not reconfigurable devices. The system build with these devices, is not flexible. If the designer wants a reconfigurable device, he must target a SRAM FPGA. Moreover, the capacities of the SRAM FPGAs are the highest for FPGA devices. Actually, the SRAM FPGAs have a market share higher than 60% (just with the two leaders companies Xilinx (http://www.xilinx.com) and Altera (http://www.altera.com)). Therefore, the research to improve the security level of such FPGAs and particularly the improvement of bitstream encryption is necessary today.

Some works give efficient solutions to encrypt the SRAM FPGA bitstream. Nevertheless, there are some drawbacks and it is possible to improve them taking into account the latest innovations of these FPGAs. The following section presents some works about the bitstream encryption.

3 Related work

Two approaches are generally possible to address the design security problem. The first one considers that the

best solutions to protect the devices against piracy are legal solutions. The definition of efficient laws, the regulation and the management of intellectual properties are parts of this solution.

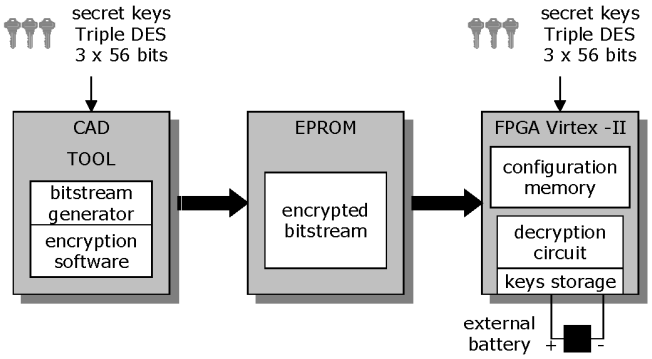
The second one, according to the last section, proposes to improve the security level of actual SRAM FPGAs by configuration protection (bitstream encryption). Even if the two solutions must be complementary, in the following, we only address the latter approach.

Xilinx proposes a security system (www.xilinx.com) based on a triple DES encryption scheme to protect the bitstream of the Virtex-II and Virtex-II Pro family device.

Xilinx CAD software tool encrypts the bitstream using the powerful Triple Data Encryption (DES) algorithm before downloading the configuration inside the FPGA. Triple DES is the standard used by many governments for safe communication and by banks around the world for money transfers. This algorithm uses three 56-bits public keys. The designer can use random keys or choose their own-keys.

Figure 1 shows the encryption/decryption system used by Xilinx to protect the configuration of Virtex-II devices.

Figure 1 Xilinx Virtex-II triple DES encryption scheme. The bitstream is encrypted by the CAD tool during the EPROM storage. When power is switched on, a DES decryption circuit embedded in the FPGA decrypts the configuration. Three 56-bits keys are embedded in the FPGA and stored in a volatile memory with an external battery



This system is relatively simple; it is just necessary to choose one option during the last step of the CAD process, the bitstream generation. First, a key file that describes the configuration of the three keys is programmed inside the FPGA. The customer chooses his own keys. Of course, it is not necessary to store the key file inside the configuration memory. It is not possible to encrypt two cores with different keys loaded into the same FPGA at the same time. The keys are stored in a dedicated SRAM memory inside the FPGA that can be backed up with a small battery (like a watch battery).

Next, the configuration step is performed like a classical configuration without the bitstream encryption. In fact, the configuration stored in the external EPROM is encrypted. The FPGA contains a decryption circuit that automatically detects when the bitstream is encrypted and it decrypts the configuration before the SRAM bits are programmed. Xilinx does not give information about the necessary extra-time to decrypt the configuration.

The Xilinx bitstream encryption scheme is efficient because without the correct key it is not possible to configure other chips with the encrypted bitstream. Nevertheless, when the device is configured, it is not possible to use partial reconfiguration or to do read-back and it is not possible to use bitstream compression.

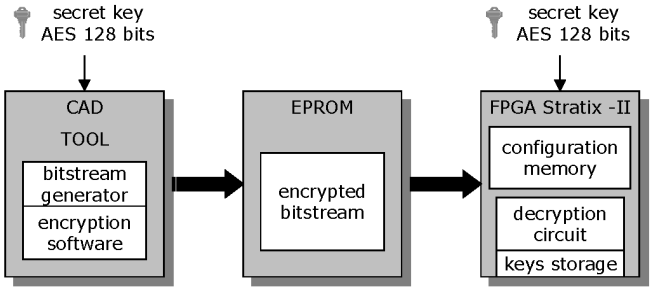
If the designer does not need security, the device can be configured with non-encrypted bitstream and the on-chip keys are simply ignored.

This method has a strong drawback; it uses an external battery to save the key. It is poor for several reasons. This solution costs a lot of area on the board and even if the used battery is small it is necessary to add a socket, and the board area is a critical issue for embedded system. Moreover, this solution increases the board cost (2–3$ per board (Trimberger, 2004)) and reduces the system lifetime (particularly bad for long-life hardware applied in space applications, for example).

It is necessary to improve the Xilinx solution by proposing a solution without the additional battery.

Not long ago, Altera proposed a solution of bitstream encryption for the new Stratix-II device (Altera Coporation, 2004). Figure 2 shows the encryption/decryption system used by Altera to protect the configuration of Stratix-II devices.

Figure 2 Altera Stratix-II AES encryption scheme. Like Xilinx solution, the bitstream is encrypted by the CAD tool during the EPROM storage. When power is switched on, an AES decryption circuit embedded in the FPGA decrypts the configuration. One 128-bits key is embedded in the FPGA and stored in a non-volatile memory without an external battery



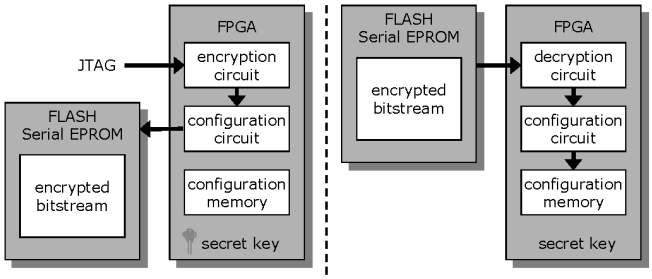
Design security in Stratix-II device is enabled by encrypting the configuration bitstream using 128-bit AES and a non-volatile key. AES is a standard for encryption, developed to replace the DES standard. The 128-bits AES key makes it much more secure than DES (56-bits key size) and triple DES (three 56-bits key). Unlike Xilinx solution, the non-volatile key retains its information when the power is off, eliminating the need for a backup battery.

Tom Kean of the Algotronix society proposes an attractive solution to answer the FPGA security problem (Kean, 2001; Kean et al., 2001). The first idea of Kean is to use a secret cryptographic key stored on an FPGA like Altera solution. He gives some ways to store this key as using a laser to program a set of links during manufacture.

As the secret key is only known by the FPGA, it must contain an encryption and a decryption circuit. However, contrary to Xilinx and Altera methods, the CAD does not change and just generates a classical bitstream.

Figure 3 shows the encryption/decryption system used by Kean to protect the configuration of SRAM FPGA. Figure 3(a) shows the initial configuration of secure FPGA and Figure 3(b) shows the normal configuration of secure FPGA.

Figure 3 Kean proposes encryption/decryption scheme embedded in the FPGA. (a) shows the initial configuration to encrypt the bitstream (inside the FPGA) and stores it in the EPROM and (b) shows the normal configuration of the FPGA when the power is switched on, the encrypted bitstream is decrypted inside the FPGA and configures it



This solution has many advantages; it does not affect system reliability, requires no additional components and it does not require support from CAD software. In this system, nobody (the designer or the CAD tool) needs knowledge of the key.

If Kean's and Altera solutions overcome the battery limitation of the Xilinx solution, all the solutions have the same important disadvantages. In all the cases, the decryption circuit is embedded inside the FPGA. These circuits take FPGA silicon area normally reserved for the developed application. Therefore, the total application dedicated-area is reduced by these solutions, particularly in the case of Algotronix solution, since the encryption and the decryption circuits are both embedded in the same FPGA.

Moreover, in all solutions the encryption and the decryption circuits are fixed, so it is not possible to upgrade them or to choose the encryption/decryption algorithm and architecture. It is a lack of flexibility for the system; it will be impossible to update it with new encryption algorithms, for example.

In all solutions, the entire design is encrypted with the same encryption algorithm. However, such approach is very restrictive since it does not consider any security policy. Actual designs (owing to the high degree of application complexity) are based on numerous heterogeneous parts that do not present the same 'security sensitivity'. Hence, the designer may want to partition his application in several parts and use different encryption/decryption algorithms to encrypt/decrypt these parts. For example, if the designer uses some free or very-easy-to-find IPs (Intellectual Property), it may be not necessary to encrypt these parts of the application. Other parts like interfaces, for example, do not need a high security level. On the other hand, the real designer's IPs need a high security level.

Finally, the three proposed solutions give only one fixed answer to the bitstream security problem and lack flexibility.

Other solutions are proposed; most of them can be found in recent US Patents for example, Kelen and Burnham. (2000), Erickson et al. (2001), Mason et al. (2001) and Pang et al. (2002). Nevertheless, these solutions are not very different from Xilinx (www.xilinx.com), Altera (www.altera.com) or Kean (2001; Kean et al., 2001) solutions.

If existing solutions are not very different one from another, it is mainly owing to the fact that they do not use the new features of SRAM FPGAs like partial reconfiguration, dynamic reconfiguration and self-reconfiguration.

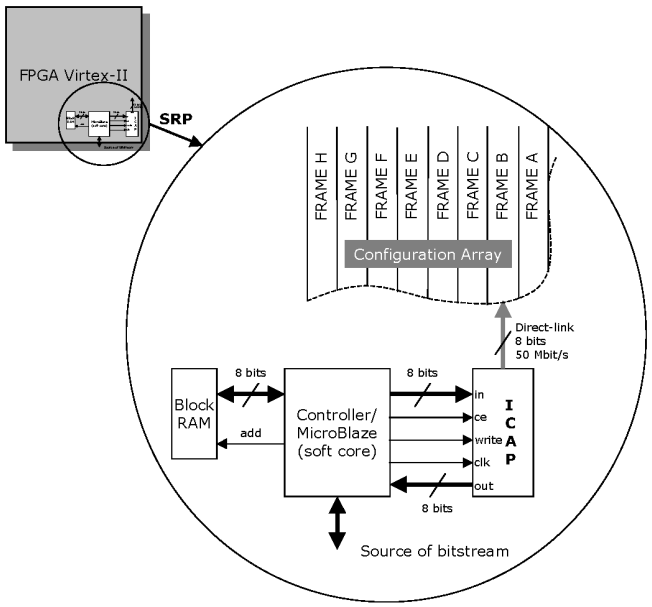
In the following section, we present the new self-reconfiguration capabilities of SRAM FPGA.

4 New self-reconfiguration technique for SRAM FPGA

According to previous sections, actual solutions to secure the SRAM FPGA bitstream are efficient, but lack flexibility. However flexibility, given by the reconfiguration capabilities, is the main advantage of the reconfigurable devices like SRAM FPGAs (particularly in comparison with other FPGAs or ASIC). This advantage is increasingly important with the new capabilities of SRAM FPGAs like partial reconfiguration, dynamical reconfiguration or self-reconfiguration.

In Blodget et al. (2003) and Blodget and McMillan (2003) Xilinx presents a Self-Reconfiguring Platform (SRP) for Xilinx Virtex-II and Xilinx Virtex-II Pro devices. Self-reconfiguration extends the concept of dynamic reconfiguration. It assumes that dedicated circuits within the FPGA are used to control the configuration of the other parts of the FPGA. In this case, the FPGA is able to dynamically reconfigure itself under the control of an embedded microprocessor or controller. This microprocessor can be a soft-core like Xilinx Micro Blaze (32-bit RISC) or a hard-core like IBM PowerPC (32-bit RISC) embedded on the Xilinx Virtex-II Pro. To perform the dynamical reconfiguration, the microprocessor or the controller use a specific interface called ICAD (Internal Configuration Access Port). When the bitstream is stored within the FPGA, the FPGA embedded RAM (called BlockRAM in Xilinx Virtex devices) are used like small configuration cache. Figure 4 presents a schematic view of the self-reconfigurable platform.

Figure 4 Schematic view of the self-reconfigurable platform SRP. The ICAP port is directly connected to the configuration array. It can partial reconfigure the different frame of configuration. The configuration controller can be a MicroBlaze soft core. The bitstream file can be provided from outside or inside the circuit. The BlockRAM can be used like configuration memory



The Virtex ICAP is a version of the Xilinx Select Map programming port that is internally accessible to the configure FPGA logic. According to Fong et al. (2003) the ICAP, interface is fairly simple, consisting of separate eight-bit datapaths for reads and writes, write and chip enables, a busy signal and a clock input. The ICAP interface is physically located in the lower right corner of the Virtex-II FPGA, and can be seen using the Xilinx FPGA editor tool. When using the Select Map express configuration mode (data available every clock cycle), ICAP can be loaded with data without the need for handshaking. The ICAP throughput is limited to 50 Mbit/s.

Xilinx proposes a tool to manage these new FPGA capabilities called XPART for Xilinx Partial Reconfiguration Toolkit.

Some applications of self-reconfiguration have been done in Fong et al. (2003), Ulmann et al. (2004) and Hübner et al. (2004). In Ulmann et al. (2004), self-reconfiguration is used for CAN-bus management, and in Hübner et al. (2004) the same authors use self-reconfiguration and bitstream compression.

In the following section, we present how the new solution to address the bitstream security problem takes advantage of the dynamic SRAM FPGA self-reconfiguration.

5 A new solution to protect the SRAM FPGA bitstream

5.1 Introduction

This solution takes benefit of the new possibilities of reconfiguration of SRAM FPGAs to improve their security level without the drawbacks highlighted previously.

The encryption and the decryption circuit must leave all the silicon area free for the developed application.

The solution must use an embedded key in order to work without an extra battery; to store the key, a model close to Kean's solution (Abraham et al., 1991; www.actel.com/products/rescenter/security/index.html) can be chosen. It is possible to use laser to engrave the key or use some antifuse elements to do a non-volatile key programming.

A very important feature is also to give the designer the opportunity to choose the encryption/decryption algorithms and architectures. In this way, it is possible to adapt the encryption/decryption scheme according to the requested security level for the developed application. Furthermore, this feature enables to easily upgrade the system if a new efficient encryption/decryption algorithm is available.

Finally, we address the security-sensitivity policy problem by allowing the designer to use different encryption algorithms for a single application. The Security-Critical Parts (SCP) of the application will only be encrypted.

For test, we use a Xilinx Virtex-II Pro XCV2VP20 FF1152 proto-board.

5.2 Application security policy

As the encryption/decryption scheme is costly owing to time, power consumption and takes silicon area, it is very interesting to adapt it according to the required security level of the application parts.

All the solutions presented in Section 3 use a complete bitstream encryption with a single encryption algorithm. Nevertheless, a security application analysis can show that some parts of the application do not need protection whereas other parts need strong protection. These last parts can be security-sensitive part (global system security) or they can be the custom intellectual properties with high development cost, for example. We call these application parts Security-Critical Parts (SCP) and the other parts, like some communication protocol IPs or easy-to-find IPs, the No-Critical Parts (NCP).

The designer must partition his application in function of the security level of the different parts. It is a security-oriented partitioning. He must choose the suitable encryption/decryption algorithm and architecture for the protection of the SCP bitstreams. The designer can choose

different encryption/decryption algorithms and architectures for several SCPs or he can choose the same for all. We think that it could be more security-efficient to choose different security features for the different SCPs.

To understand our approach, in the following, two examples are given; process during the initial configuration step and process during the normal configuration step of the FPGA. In the examples, the application is partitioned into three different parts; two SCPs that need high security level (they are encrypted with two different encryption algorithms) and one NCP that does not need encryption. For the examples, each SCP bitstream is encrypted with a different algorithm but a solution with a same algorithm can be considered. The case with two SCPs is just an example and other configurations can be considered.

5.3 Key management

One feature is very important in our solution; the key management. It is mandatory that a pirate cannot access the keys used by the different decryption/encryption circuits. To prevent spy configuration, we use bitstream authentication with checksum. The circuit used to control the bitstream authentication is embedded in the FPGA on the JTAG port.

Moreover, since in this solution, the decryption/ encryption algorithm is not fixed, it is necessary to store a large key. Indeed different algorithms do not use the same key size (for example the AES algorithm uses a 128-bits key, and the triple DES uses three 56-bits keys). In fact, among the *n*-key bits, the encryption/decryption circuits select *m* necessary bits. Since only the designer knows the position in the large key of the *m* chosen bits, it is a supplementary security barrier. With the large key knowledge necessary, the pirate must investigate to identify the effective key bits for the suitable algorithm.

5.4 Security configuration controller

Our solution uses the partial configuration and the dynamic self-reconfiguration of the FPGA. The management of such configuration process is complex, particularly for the self-reconfiguration. Moreover, several bitstreams are used while the application runs. In our system, there are three types of bitstream,

- encrypted bitstream of a SCP
- no-encrypted bitstream of a decryption circuit
- no-encrypted bitstream of a NCP.

The controller must be able to detect the different bitstreams. A bitstream signature (ID) gives the controller the bitstream characteristics (encrypted or not for example). These characteristics are used like processor instructions by the controller. According to the characteristics, the controller partial-configures directly the FPGA with the selected NCP bitstream or it partial-configures the FPGA with first the decryption circuit bitstream associated with an encrypted SCP bitstream before using self-reconfiguration to configure the FPGA with the decrypted SCP bitstream.

The security configuration controller is based on a finite state machine to perform the configuration management. To handle the configuration sequence, the controller needs the external EPROM memory partitioning (the memory mapping). We can notice that this mapping can be complex in order improve the system security. For example the designer can interleave the data stored in the memory and mix the several encrypted and no-encrypted configurations. A configuration address register stores the memory mapping.

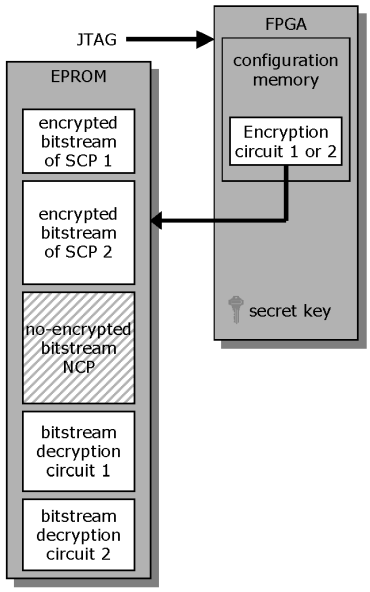
The security configuration controller can be external like a dedicated CPLD or a microprocessor. However, it is also possible that this controller is embedded inside the FPGA, like in the case of Xilinx self-configuration system (Fong et al., 2003). In this last case, the configuration controller can be a soft-core microprocessor (like Xilinx MicroBlaze) or a hard-core microprocessor (like IBM PowerPC for Xilinx VirtexII-Pro devices).

5.5 Initial FPGA configuration

The initial FPGA configuration is performed in the laboratory or manufactory in order to store all the different bitstreams in the EPROM memory. The CAD tool performs the initial configuration. If there are SCPs in the application, the bitstreams of each encryption circuits are generated to use these circuits to encrypt the SCPs bitstreams. In the same way, the bitstreams of the decryption circuits are generated to use these circuits to decrypt the encrypted SCPs bitstreams.

Figure 5 presents the encryption system when the FPGA is initially configured and the root configuration memory is programmed (initial configuration).

Figure 5 Encryption scheme during the initial FPGA configuration. The bitstreams are stored in the EPROM from the CAD tool through the FPGA JTAG port. For the SCPs bitstreams, the FPGA is configured with encryption circuits to encrypt the bitstreams before being stored it in the EPROM. The NCP bitstream are not encrypted.



During the initial FPGA configuration, the first step consists of programming the root configuration memory with the non-encrypted parts. First, the NCP bitstreams are stored; in the example shown in Figure 5, there is only one NCP. For the same example, two decryption circuits will be used to decrypt the encrypted SCPs bitstreams. Therefore, the bitstreams of the two decryption circuits are stored in the EPROM. In Figure 5, after the first step there are three no-encrypted bitstreams stored in the EPROM; the *NCP* bitstream, the *decryption circuit* 1 bitstream (associated with the *SCP* 1) and the *decryption circuit* 2 bitstream (associated with the *SCP* 2).

The second step is the storage of the encrypted bitstreams of the *SCP* 1 and *SCP* 2. First, it is necessary to configure the FPGA with the *encryption circuit* 1 in order to encrypt the bitstream of the *SCP* 1. Once the *SCP* 1 bitstream is encrypted, it is stored in the root external EPROM. Since the *SCP* 2 needs other encryption circuit, it is not necessary to keep the *encryption circuit* 1 in the device. The FPGA is partial configured with the *encryption circuit* 2; the *SCP* 2 bitstream is encrypted and stored in the EPROM.

Of course, it is necessary for the CAD to manage partial reconfiguration like in Xilinx proposition (Blodget and McMillan, 2003).

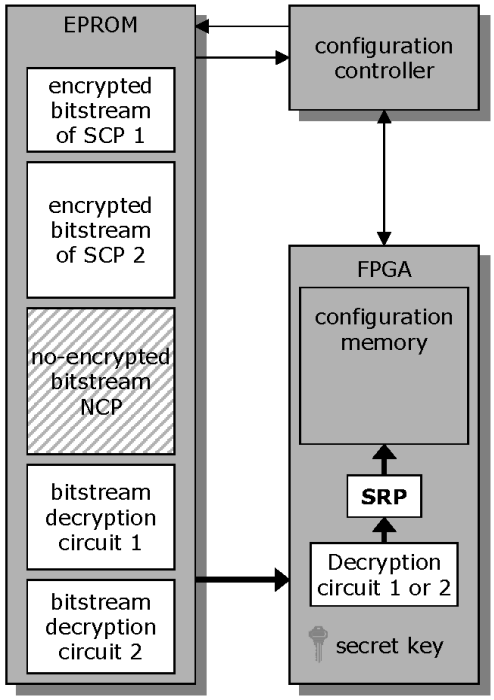
At the end of the initial configuration step, the root configuration memory contains the encrypted bitstreams of *SCP* 1 and *SCP* 2, the no-encrypted bitstream of the *NCP* and the no-encrypted bitstreams of the decryption circuits required to decrypt *SCP* 1 and *SCP* 2 (*decryption circuit* 1 and *decryption circuit* 2).

5.6 Normal FPGA configuration (when power is switched on)

When power is switched on, the SRAM FPGA must be configured since this inside configuration memory is volatile. Figure 6 shows the decryption-configuration system when the FPGA is configured from an external EPROM memory that stores the configuration (normal configuration). The configuration controller manages the configuration process.

The FPGA configuration process works as follows: First, the FPGA is configured with the *decryption circuit* 1 bitstream. Then the FPGA uses it to decrypt the encrypted *SCP* 1 bitstream and self-configures the *SCP* 1. As we can see on Figure 6, the SRP (Self-Reconfiguring Platform, see Section 4) is used to perform self-reconfiguration. Once the *SCP* 1 bitstream is decrypted and the FPGA is configured with the *SCP* 1 circuit, it is not necessary to keep the *decryption circuit* 1. The *decryption circuit* 2 replaces (with FPGA partial reconfiguration) it in order to decrypt the encrypted bitstream of *SCP* 2. In the same way, after the decryption and the self-configuration of the *SCP* 2 bitstream, it is not necessary to keep the *decryption circuit* 2.

Figure 6 Decryption and self-configuration scheme during the normal FPGA configuration. The FPGA is partial-configured by the decryption circuit 1 or 2 to decrypt the encrypted SCP 1 and SCP 2 bitstreams. The FPGA is self-configured with these decrypted bitstreams. The self-reconfiguration is performed by the SRP. At the end of the configuration process, the FPGA is configured with the NCP bitstream



After this first phase, the FPGA is configured with the *SCP* 1 and the *SCP* 2 circuits. The last step consists in configuring the FPGA free area with the other application parts that have not an encrypted configuration; so with the *NCP* bitstream.

Finally, the FPGA is configured with all the application parts; the *SCP* 1, the *SCP* 2 and the *NCP*. There can be any encryption or decryption circuit configured in the FPGA.

5.7 Configuration controller finite state machine

As described previously, the configuration controller is developed with a finite state machine. With the knowledge of the memory mapping, the configuration management finite state machine is relatively simple. The configuration controller is used only for normal FPGA configuration when power is switched on. The initial configuration is processed by the CAD tool.

Figure 7 shows the three-global-states used by the configuration controller. Table 2 describes the actions associated to the states of the configuration controller. The first state of this three-states FSM is an idle state. To change state the configuration controller waits for a start signal. This signal is the begin-signal of the normal configuration process.

Figure 7 Configuration controller finite state machine. It is a three-global-states machine. The states represent several actions. The active state depends if the bitstream is encrypted or not

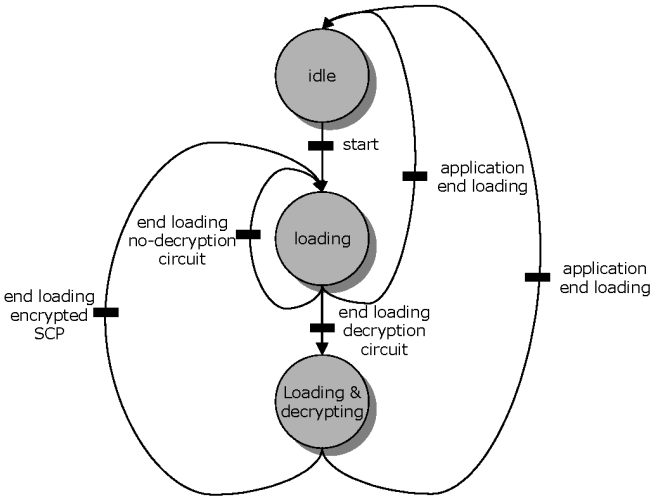


Table 2 States description of the configuration controller FSM

State name	*Actions*
Idle	Wait start
Loading	Configure the FPGA with selected bitstream* using partial-configuration
	Update the configuration address register
	*The selected bitstream can be the no-encrypted bitstream of a decryption circuit or a NCP
Loading and decrypting	Start the decryption algorithm and load the corresponding SCP bistream* on the FPGA
	Update the configuration address register
	*The selected bitstream is an encrypted bitstream

Once in the second state, the loading state, the configuration controller changes states according to the type of bitstream. If the bitstream is not encrypted the current state is the second state. In this state, the normal configuration of the FPGA is performed. If the bitstream is encrypted (so it is a SCP bitstream), the current state is the 'loading and decrypting' state. In this state, the configuration controller loads first the decryption circuit bitstream inside the FPGA before loading the encrypted bitstream of a SCP.

The machine returns to the idle state when all the application is loaded inside the FPGA.

This section has shown the main technological characteristics of our bitstream protection system for SRAM FPGA. The following sections give the drawbacks and advantages of our solution and compare it with the different solutions (presented in Section 3).

6 Drawbacks and advantages of the proposed solution

If this method permits to overcome the limitation of other proposed solutions, it has, however, some drawbacks.

The first drawback is the relative complexity of the method, since it is necessary to manage the partial reconfiguration and dynamic self-configuration. Most of the FPGA manufacturers do not have the technology and the CAD tools to manage these types of configurations but Xilinx, which proposes an efficient tool for such needs.

The decryption circuit can have several sizes according to the algorithm and the implementation. For example, several works give comparisons of the hardware performance of the different AES final candidates (MARS, RC6, Rijndael, Serpent or Twofish for example) using FPGA (Dandalis et al., 2000; Elbirt et al., 2000; Gaj and Chodowiec, 2000; Weaver and Wawrzynek, 2000). All these works use the Xilinx Virtex as the reconfigurable target. The Tables 3 and 4 compare the results of these studies for the area requirement (one Virtex slice corresponds to two four-inputs LUTs, two flip-flops and one carry chain) and time performance (throughput).

Table 3 Area requirement of FPGA implementations of AES final candidates

	No. of slices of the cryptographic core		
Algorithm	*Dandalis et al. (2000)*	*Elbirt et al. (2000)*	*Gaj and Chodowiec (2000)*
Rijndael	4312	5302	2902
Serpent	1250	7964	4438
RC6	1749	3189	1139
Twofish	2809	3053	1076
MARS	4621	–	2737

Table 4 Time performance of FPGA implementations of AES final candidates

	Throughput (Mbit/s)		
Algorithm	*Dandalis et al. (2000)*	*Elbirt et al. (2000)*	*Gaj and Chodowiec (2000)*
Rinjdael	353.0	300.1	331.5
Serpent	148.9	444.2	339.4
RC6	112.9	126.5	103.9
Twofish	173.1	119.6	177.3
MARS	101.9	–	39.8

The performances (time and area) showed in the two tables are different for each work. Because the architectures chosen, for the different studies, have different structures (loop unroll, pipeline and sub-pipeline). All these results are given only for an encryption core without the key-setup circuit. Nevertheless, this circuit must be considered because it can take area (slices). The Table 5 shows the number of slices for key-setup circuit of the five AES final candidates and the relative area percentage of the total area requirement (encryption core and key-setup circuits).

According to these results, it is significant to consider the key-setup circuit in the area requirement. Finally, the three tables show that a same decryption standard (AES in this example) can be performed with several

algorithms and each algorithm can have different implementations. Therefore, it is necessary to give all the possibilities to the designer, and our solution gives this flexibility. Moreover, the studies Dandalis et al. (2000), Elbirt et al. (2000), Gaj and Chodowiec (2000) and Weaver and Wawrzynek (2000) are throughput oriented, therefore, the area (or the number of used FPGA resources) is not the main constraint. In our system, the out data of the decryption circuit are used to self-reconfigure the FPGA. In the case of Xilinx technology, the ICAP interface limits the throughput to 50 Mbit/s. This throughput is widely inferior to most of the Table 4 results. Therefore, it is possible to develop decryption algorithm with area (used resources) constraint. Actually, the number of Virtex slices used for the cryptographic cores given in Table 3 must be reduced. For example, in our Xilinx proto-board, the Virtex-II Pro XC2VP20 contains 9280 slices, according to the Table 3, with such device, the Rijndael implementation of AES use from 31% to 57%. It is probably necessary to limit the number of used resources since the FPGA is not configured only with the decryption algorithm.

Table 5 Area requirement of FPGA implementations of AES final candidates

	No. of slices of the key-setup circuit		*Percent of the total area*	
Algorithm	*Dandalis et al. (2000)*	*Weaver and Wawrzynek (2000)*	*Dandalis et al. (2000)*	*Weaver and Wawrzynek (2000)*
Rijndael	1361	128	24	14
Serpent	1300	2060	51	35
RC6	901	290	34	15
Twofish	6554	1260	70	48
MARS	2275	50	33	3

The configuration controller can be complex. Its complexity depends on the number of SCPs in the application. This number is correlated to the application security partitioning. The costs of a larger root memory and a complex configuration controller are the hardware overhead costs of this method but they represent the origin of its flexibility. The system security has always costs that are necessary to evaluate in order to choose the best solution according to the required security level.

Since it is necessary to first configure the decryption circuit before the real configuration of each SCP, this method can spend time when the system is powered up. Nevertheless, today the SRAM FPGA configuration is increasingly faster (about 10 millisecond for a partial reconfiguration for a Xilinx Virtex 1000-E device (Delahaye et al., 2004)).

This method has many very interesting advantages. First, the encryption/decryption circuits do not take FPGA application-dedicated resources, since when a decryption circuit has been used it is removed from the FPGA. The FPGA resources initially used to perform the decryption circuit are free for other uses.

We choose, like Kean (2001; Kean et al 2001), to embed the key inside the FPGA in order to have non-external extra-battery.

One of the main advantages of this method is the increase of flexibility. The designer can partition the application according to the required security level. Therefore, if just a small part of the application needs a strong security, the system can be very simple (just one small SCP). The designer has the possibilities to choose the suitable algorithms and architectures for the encryption/ decryption circuits. It is possible to adjust the security level according to the application constraints.

Moreover, the designer can upgrade his application and the security scheme with the same reconfigurable hardware. In this way, it is possible to take advantage of the latest improvements of the security field.

7 Comparison of the different actual solutions

Section 3 of this paper has shown different actual solutions of FPGA protection against cloning and reverse engineering. It is interesting to compare these different solutions with our solution for several aspects; security level, encryption flexibility, reconfiguration flexibility and complexity. Table 6 presents the result of comparisons.

Table 6 Area requirement of FPGA implementations of AES final candidates

	Security	*Encryption flexibility*	*Reconfiguration flexibility*	*Complexity*
Actel Antifuse	High	–	Any	Easy
QuickLogic Flash	High	–	Any	Easy
Xilinx Triple DES	Middle	Any	Low	Easy
Altera AES	Middle	Any	Low	Easy
Algotronix T. Kean	Middle	Any	Low	Middle
UBS/UMASS L. Bossuet	Middle+	High	High	Complex

According to the table, we think that the security level is higher for antifuse or flash logic, but we think that it is necessary to better expertise the real security level of bitstream encryption system. The real advantage of our solution is the flexibility of encryption and reconfiguration. Moreover, with a real application security policy (i.e., security-oriented application partitioning), our solution proposes a higher security level that the other solution for SRAM FPGA. Nevertheless, our solution complexity is higher since it is necessary to manage the partial and self-reconfiguration.

8 Conclusion

Since the SRAM FPGAs are increasingly important for the electronic industry, it is necessary to improve the security level of such devices. Although some works have already proposed solutions to improve this security level, we think that is it possible to investigate more this domain.

In this paper, we propose a new solution to prevent piracy against SRAM FPGAs bitstream. Our contribution is to use the latest developments of configuration technique in order to improve the security system flexibility. The use of self-reconfiguration allows using the decryption circuit out data to configure the decrypted bitstream. Unlike the actual bitstream encryption scheme (Xilinx or Altera solution), our solution is flexible; the designer can choose the different encryption/decryption algorithms and architectures. He can easily update the system with new security feature. Moreover, we propose to the designer to apply a true security policy for the applications, by security-oriented partitioning.

We think that the security problem is a very important issue for FPGAs and for the reconfigurable systems on chip. Probably in the near future, there will be more and more works done about this subject.

Acknowledgement

Manuscript received June 28, 2004. This work was supported in part by the French Ministry for Education and Research.

References

Abraham, D.G., Dolan, G.M., Double, G.P. and Stevens, J.V. (1991) 'Transaction security system', *IBM Systems Journal*, Vol. 30, No. 2, pp.206–229.

Altera Coporation (2004) *Design Security in Stratix II Devices*, White paper, Available on www.altera.com.

Anderson, R. and Kuhn, M. (1996) 'Tamper resistance – a cautionary note', *Proceeding of the Second USENIX Workshop on Electronic Commerce*, November 18–21, Oakland, California, USA, pp.1–11.

Anderson, R. and Kuhn, M. (1997) 'Low cost attack on tamper resistant devices', *Proceeding of the 5th Workshop of Security Protocols*, April 7–9, Paris, France, pp.125–136.

Blodget, B. and McMillan, S. (2003) 'A lightweight approach for embedded reconfiguration of FPGAs', *Design, Automation and Test in Europe Conference and Exhibition*, DATE'03, Mach 3–7, Munich, Germany.

Blodget, B., James-Roxby, P., Keller, E., McMillan, S. and Sundararajan, P. (2003) 'A self-reconfiguration platform', *Proceeding of 13th International Conference on Field-Programmable Logic and Applications*, FPL'2003, September, Lisbon, Portugal, pp.565–574.

Dandalis, A. and Prasanna, V.K. (2000) 'An adaptive cryptographic for IPSec architectures', *Proceeding IEEE Symposium on Field-Programmable Custom Computing Machines*, FCCM'00, April, Napa, USA, pp.132–141.

Dandalis, A., Prasanna, K. and Rolim, J.D.P. (2000) 'A compartive study of performances of the AES final candidates using FPGA', *Workshop on Cryptographic Hardware and Embedded Systems*, August.

Delahaye, J.P., Gogniat, G., Roland, C. and Bomel, P. (2004) 'Software radio and dynamic reconfiguration on a DSp/FPGA platform', in Rykaczewski, P. and Schmidt, M. (Eds.): in special issue on *Software Defined Radio of Frequenz*, May-June, No. 58, pp.152–159.

Elbirt, A.J., Yip, W., Chetwynd, B. and Paar, C. (2000) 'An FPGA implementation and performance evaluation of the AES block cipher candidate algorithm finalists', *Proceeding of the third Advanced Encryption Standard Candidate Conference*, AES3, April 12–14, New York, USA, pp.12–27.

Erickson, C.R., Tovana, D. and Holen, V.A. (2001) *Encryption of Configuration Stream*, US Patent 6 212 639, April 3.

Fong, R., Harper, S. and Athanas, P. (2003) 'A versatile framework for FPGA field updates: an application of partial self-reconfiguration', *Proceeding of 14th IEEE International Workshop on Rapid System Prototyping*, RSP'03, 9–11 June, San Diego, California, USA, pp.117–123.

Gaj, K. and Chodowiec, P. (2000) 'Comparison of the hardware performance of the AES candidates using reconfigurable hardware', *Proceeding of the third Advanced Encryption Standard Candidate Conference*, AES3, April 12–14, New York, USA.

Hübner, M., Ulmann, M., Weissel, F. and Becker, J. (2004) 'Real-time configuration code decompression for dynamic FPGA self-reconfiguration', *11th IEEE Reconfigurable Architectures Workshop*, RAW 2004, Santa Fé, New Mexico, USA, 26–17 April.

Kean, T. (2001) 'Secure configuration of field programmable gate array', *Proceedings IEEE Symposium on Field Programmable Custom Computing Machines (FCCM)*, Rohnert Park CA.

Kean, T. (2001) 'Secure configuration of field programmable gate arrays', *Proceeding of 11th International Conference on Field-Programmable Logic and Applications*, FPL'2001. Belfast, UK, pp.142–152.

Kelen, S.H. and Burnham, J.L. (2000) *System and Method for PLD Bitstream Encryption*, US Patent 6 118 869, September 12.

Lockwood, J.W., Neely, C., Zuver, C., Moscola, J., Dharmapurikar, S. and Lim, D. (2003) 'An extensible, system-on-programmable-chip, content-aware internet firewall', *Proceeding of 13th International Conference on Field-Programmable Logic and Applications*, FPL'2003, September, Lisbon, Portugal, pp.859–868

Mason, M.T., Kunnari, N.D. and Kuo, H.H. (2001) *Secure Programmable Logic Device*, US Patent 6 331 784, December 18.

Örs, S.B., Oswald, E. and Preneel, B. (2003) 'Power-analysis attack on an FPGA – first experimental results', *CHES 2003, LNCS 2779*, pp.35–50.

Pang, R.C., Wong, J., Frake, S.O., Sowards, J.W., Kondapalli, V. M., Goetting, F.E., Trimberger, S.M. and Rao, K.K. (2002) *Non Volatile/Battery-Backed Key in PLD*, US Patent 6 336 117, April 2.

QuickLogic (2002) *Security in QuickLogic Devices*, White Paper, Available on http://www.quicklogic.com.

Standaert, F.X., Örs, S.B. and Preneel, B. (2004) 'Power-analysis on an FPGA implementation of AES', In Joye, M. and Quisquarter, J.J. (Eds.): *Proceedings of Cryptographic Hardware and Embedded Systems CHES'2004, Lecture Note in Computer Science (LNCS)*, Springer-Verlag, pp.30–44.

Standaert, F.X., van Oldeneel tot Oldenzeel, L., Samyde, D. and Quisquater, J.J. (2003) 'Power analysis of FPGAs: how practical is the attack', *Proceeding of 13th International Conference on Field-Programmable Logic and Applications*, FPL'2003, September, Lisbon, Portugal, pp.707–711.

Tredennick, N. and Shimamoto, B. (2003) 'The rise of reconfigurable systems', *Proceeding of Engineering of Reconfigurable Systems and Application*, ERSA'2003, June 23–26, Las Vegas, Nevada, USA, pp.3–9.

Trimberger, S. (2004) 'Virtex encrypted bitstreams', *2nd International Workshop on Cryptographic Architectures Embedded in Reconfigurable Devices*, CryptArchi 2004, Dijon, France, June 16–18.

Ulmann, M., Hübner, M., Grimm, B. and Becker, J. (2004) 'An FPGA run-time system for dynamical on-demand reconfiguration', *11th IEEE Reconfigurable Architectures Workshop*, RAW 2004, Santa Fé, New Mexico, USA, 26–17 April.

Weaver, N. and Wawrzynek, J. (2000) 'A comparison of the AES candidates amenability to FPGA implementation', *Proceeding of the third Advanced Encryption Standard Candidate Conference*, AES3, April 12–14, New York, USA.

Wollinger, T. and Paar, C. (2003) 'How secure are FPGAs in cryptographic applications', *Proceeding of 13th International Conference on Field-Programmable Logic and Applications*, FPL'2003, September, Lisbon, Portugal, pp.707–711.

Wollinger, T., Guajardo, J. and Paar, C. (2004) 'Security on FPGAs, state of the art implementations and attacks', *ACM Transactions in Embedded Computing Systems (TECS)*, Vol. 3, No. 3, pp.534–574.

Websites

Actel Coporation, Resource Center: Security, Available on www.actel.com/products/rescenter/security/index.html.

Altera Coporation, http://www.altera.com.

Xilinx Coporation, http://www.xilinx.com.

Xilinx Coporation, *Virtex-II platform FPGA Handbook*, Technical Documentation, Available on www.xilinx.com.

3. Article concernant l'exploration de l'espace de conception pour des architectures FPGA

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

Design Space Pruning through Early Estimations of Area / Delay Trade-offs for FPGA Implementations

Sebastien Bilavarn, Guy Gogniat, Jean-Luc Philippe and Lilian Bossuet

Abstract— **Early performance feedback and design space exploration of complete FPGA designs are still time consuming tasks. We propose an original methodology based on estimations to reduce the impact on design time. We promote a hierarchical exploration to mitigate the complexity of the exploration process. Therefore this work takes place before any design step, such as compilation or behavioral synthesis, where the specification is still provided as a C program. The goal is to provide early area and delay evaluations of many RTL implementations to prune the design space. Two main steps compose the flow: (1) a structural exploration step defines several RTL implementations, and (2) a physical mapping estimation step computes the mapping characteristics of these onto a given FPGA device. For the structural exploration, a simple yet realistic RTL model reduces the complexity and permits a fast definition of solutions. At this stage, we focus on the computation parallelism and memory bandwidth. Advanced optimizations using for instance loop tiling, scalar replacement or data layout are not considered. For the physical estimations, an analytical approach is used to provide fast and accurate area / delay trade-offs. We do not consider the impact of routing on critical paths or other optimizations. The reduction of the complexity allows the evaluation of key design alternatives, namely target device and parallelism that can also include the effect of resource allocation, bitwidth or clock period. Due to this, a designer can quickly identify a reliable subset of solutions for which further refinement can be applied to enhance the relevance of the final architecture and reach a better use of FPGA resources, i.e. an optimal level of performance. Experiments performed with Xilinx (VirtexE) and Altera (Apex20K) FPGAs for a 2D Discrete Wavelet Transform and a G722 speech coder lead to an average error of 10% for temporal values and 18% for area estimations.**

Index Terms— **Design Space Exploration, area and delay estimation, C specification, H/CDFG representation, graph scheduling, architectural synthesis, technology projection, FPGA device.**

I. INTRODUCTION

THE CONTINUOUS increase in the complexity of applications and architectures leads to prohibitively long design cycles. Usually, designers perform hardware exploration through several iterations of a synthesis flow to reach a good constraint compliant solution. As a result, exploring solutions able to match the parallelism potential of the application with the architecture is an uncertain and time consuming process often left to designer experience. To address such an issue, we present an automated exploration approach applicable from system specifications given in the form of a C program. FPGAs are considered for implementation because of their ability to cope with future design perspectives [1][2] and to exploit vast amounts of parallelism within new generations of high performance computing and reactive applications (adaptive video streaming, software radio, ...)[3][4].

Basically, the approach computes early area and delay values of RTL designs at a glance. So it is important to know what are its limits and the impact on the exploration space we analyze. There is obviously a trade-off between the exploration space coverage and the relevance of the architectural solutions, especially because of the abstraction level gap between a behavioral specification at the input and the FPGA layout characteristics at the output. The key point here has been to reduce complexity enough in order to allow fast exploration of a large space. For that, we defined a two-step process: (1) structural exploration performs automatic definition of several RTL solutions, and (2) physical mapping estimation computes the corresponding FPGA layout characteristics in the form of area / performance trade-offs.

The structural step considers a realistic architectural model including datapath, control and memory units. It focuses on the exploration of the parallelism potential (memory bandwidth and computation) but advanced optimizations such as loop tiling, scalar replacement, data layout, data reusing, memory sharing and memory pipelining are not considered. Also, a simplified execution model for loop unrolling and folding is used in the current state of the tool. Concerning the physical step, the analytical estimation of the mapping results is based on area and delay predictions. Advanced considerations at the logic level such as the impact of routing on critical path (that is on clock period) or others are not analyzed. In other words, we do not take into account all the possible optimization models in the exploration, we restrict its scope from the reliable parameters we can consider at a system level. The motivation is to prune the design space to point out a set of promising solutions for further refinement. We believe such a pragmatic approach permits mitigating the complexity of exploring the design space. Finally, the dependence of the estimations on a target device and low level synthesis tools is thus reported on the physical step, and simplified through the use of libraries. This way, application to several FPGA families (including recent devices) has been made possible. This approach has been integrated in a CAD framework for the codesign of heterogeneous SoCs called *Design Trotter*.

The remainder of the paper is organized as follows: Section

S. Bilavarn is with the Signal Processing Institute, Swiss Federal Institute of Technology (EPFL), Lausanne Switzerland (email:sebastien.bilavarn@epfl.ch).

G. Gogniat and J.L. Philippe are with the Laboratory of Electronic and REal Time Systems (LESTER), University of South Britanny (UBS) - CNRS FRE2734, Lorient, France (email: guy.gogniat@univ-ubs.fr; jean-luc.philippe@univ-ubs.fr).

L. Bossuet is with the IXL laboratory - ENSEIRB - University of Bordeaux 1 - CNRS UMR5818 France (email: bossuet@ixl.fr)

II reviews some prior contributions in the fields of design space exploration and area / delay estimators for FPGAs. Section III focuses on the exploration and estimation flow. This section provides the necessary information to understand the detailed description of the flow reported in Sections IV and V. Section IV exposes the structural exploration principles whereas Section V presents the physical estimations. Section VI illustrates the approach on several examples to stress the benefits of providing such system level estimations to a designer and the ability to enhance design space exploration compared to classical iterative methods. Section VII presents future work and concludes the paper.

II. Related Work

The Design Space Exploration (DSE) problem related to FPGA implementation is the task of exploring different RTL architectures, where the FPGA architecture is set and different implementation possibilities of an application are analyzed: computation parallelism, pipelining, replication, resource binding, clock value, ... Such an exploration is motivated by vast amounts of resources available within an FPGA that can speedup the execution of the algorithm until several orders of magnitude. Three types of exploration approaches can be considered: (1) synthesis [5][6]; (2) compilation [3][4][7]; and (3) estimation [8]. The third approach (namely *estimate and compare*) relies on estimations to perform DSE. In that case the synthesis or compilation steps are replaced by low complexity estimators. Once estimations are completed, each RTL architecture is characterized by an area and delay doublet corresponding to the system characteristics when the application is mapped onto the FPGA. As for the second approach, synthesis steps are still required to actually design the final RTL architecture. In the following section, a detailed presentation of some major contributions in the fields of DSE and area / delay estimators for FPGAs is presented. Table I summarizes their main characteristics.

A. DSE and area / delay estimators for FPGAs

A first technique proposed in [13] by Miller and Owyang is based on a library of benchmarks. A set of circuits is implemented and characterized for several FPGAs. Area and delay estimation is performed by partitioning the application into several circuits, that are then substituted by the most similar benchmark. The drawback of this is related to the difficult task of maintaining the library for different devices and applications. Another methodology described in [9][10] by Xu and Kurdahi computes area and delay values from an estimation of the mapping and place & route steps. Starting from a logic level description, they first build a netlist which is then used to compute the actual performance of the system. During the estimation task, wiring effects and logic optimizations are considered to obtain more accurate results. However, this method does not address the DSE problem and is very technological dependent since it is dedicated to the XC4000 family (CLB-based architecture).

The method defined by Enzler et al. [8] performs an estimation from higher abstraction levels (Data Flow Graph, namely DFG). Area and delay are estimated using a combination of both an algorithm characterization (e.g. number of operations, parallelism degree) and an FPGA mapping model (based on operation characteristics in terms of area and delay). The DSE is performed by analyzing the improvements when using pipelining, replication and decomposition of the DFG specification. Their estimator targets a XC4000E device (CLB-based architecture) and uses an analytical approach. Extension to other architectures is not obvious and may need further developments. Their approach is interesting but the limitation to DFG specifications does not allow considering the control and multidimensional data overhead.

The five next methods are based on the compilation of high level specifications and target loop transformations (except [11]). Nayak et al. [7] propose an estimation technique dealing with a MATLAB specification. Their method computes area and delay estimates for a XC4010 device through a two-step approach: first they use the MATCH compiler [14] to perform the DSE (e.g. code parallelization, loop unrolling) and generate an RTL code in VHDL. Then, estimators are used to compute area and delay: area estimation is processed through scheduling and register allocation (to define the number and the type of operators), delay estimation is based on IP characterization and considers the interconnection cost overhead. They also consider some synthesis optimizations (through a multiplicative factor) to obtain more accurate results. Another interesting feature of their approach is to take into account the datapath and control logic in the RTL implementation model. The main limitations are due to the memory unit left unconsidered and control implementation using CLBs, whereas recent FPGAs permit efficient integration of product terms and ROMs using dedicated resources (e.g. Apex Embedded System Blocks [15]).

Bjureus et al. [11] propose a simulation-based approach that computes area and delay from MATLAB specifications. During the simulation of the MATLAB code, a trace is generated to build an acyclic DFG that contains all the operations needed to execute the algorithm. Note that all loops are unfolded to build this DFG. Scheduling and binding are then applied using greedy algorithms. The FPGA architecture is represented through a performance model that is used to compute area and delay estimations. Each resource is modeled with a function that maps an operation to an area and delay tuple. DSE is iteratively processed and considers several design alternatives like the number of input channels, the bitwidth of the input stream, the device clock speed or the device area. Their approach is interesting but limited by the dataflow representation: it is dedicated to applications dealing with scalar variables since they do not consider memory and control units. In [11] the authors do not provide which CLB-based architecture has been used to build their performance model.

Kulkarni et al. [3] propose an iterative compilation-based method starting from an SA-C specification. Their approach expects the compiler to apply extensive transformations to achieve a code that exploits more efficiently the available parallelism. For each transformation, a DFG is generated to derive the impact on area. Their estimator is based on the mapping of the DFG nodes onto the FPGA architecture where

TABLE I
MOST RELEVANT ESTIMATOR TOOLS DEDICATED TO FPGAS

Authors	Estimator Input	Estimator Outputs	Design Space exploration	Architecture Model	FPGA Model	Complexity	Accuracy
Xu and Kurdahi 1996 [9][10]	Netlist	Area, Delay *Tool optimization*	No *Partitioning + analytical*	Datapath, Control logic Operator, Register	XC4000, LUT Interconnections	$O(n)$ $O(n^2 log(n))$ Min-cut	$\pm 10\%$
Enzler et al. 2000 [8]	DFG	Area, Delay	Yes, direct Pipelining Replication Decomposition *Analytical*	Datapath Operator, Register	XC4000E, LUT	$O(n)$	$\pm 20\%$
Nayak et al. 2002 [7]	MATLAB to RTL VHDL	Area, Delay *Tool optimization*	Yes, iterative compilation Loop unrolling, pipelining *Scheduling + analytical*	Datapath, Control logic Operator, Register	XC4010, LUT Interconnections	$O(n^2)$ FDS Left-edge	$\pm 15\%$
Bjureus et al. 2002 [11]	MATLAB to DFG	Area, Delay	Yes, iterative Stream data rate Device clock speed *Trace + analytical*	Datapath Operator, Register	?	$O(n)$	$\pm 10\%$
Kulkarni et al. 2002 [3]	SA-C to DFG	Area *Tool optimization*	Yes, iterative compilation Parallelization Loop unrolling *Analytical*	Datapath Operator, Register	XCV1000, LUT	$O(n)$	$\pm 5\%$
So et al. 2003 [4][12]	C to DFG (loop body)	Area, Delay	Yes, iterative compilation Parallelization Loop unrolling Memory bandwidth *Analytical*	Datapath, Memory bandwidth Operator, Register	XCV1000, LUT	$O(n)$	$\pm 20\%$
Authors This paper	C to HCDFG	Area, Delay	Yes, direct Parallelization Loop Unrolling Memory bandwidth *Scheduling + analytical*	Datapath, Control logic Operator Memory bandwidth	XCV400, EP20K200 LUT, BRAM DSP blocks	$O(n)$	$\pm 20\%$

each node is represented through an approximation formula. The method also takes into account some synthesis optimizations to enhance accuracy (e.g. shift operations instead of multiplication by a power of 2), but it is restricted to loop bodies and does not consider the memory and control overhead. Moreover like in [8], the estimation of area is restricted to one kind of FPGA resource (Configurable Logic Cells in [8] or number of Look Up Tables in [3]) and does not allow considering dedicated high performance resources like embedded operators and memories.

So et al. [4] also propose a compiler-based approach starting from a C specification. However, compared to [7] and [3] they introduce a key parameter that impacts greatly the DSE problem: memory bandwidth. Their approach is based on the DEFACTO compiler [12] that can successfully identify multiple accesses to the same array location across iterations of multi-dimensional loop nests. This analysis is used to identify opportunities for exploiting parallelism, eliminating unnecessary memory accesses and optimizing the mapping of data to external memories. Thus, their approach deals with advanced optimizations and provides behavioral VHDL codes for each transformation. Area and delay estimation is based on the synthesis results of HLS tools. The simplified performance model of the FPGA results in an estimation accuracy that depends on the complexity of the loop body.

Finally, Shayee et al. [16] propose a very accurate area and delay modeling approach using analytical and empirical techniques. Their method targets loop nests described as a C specification and evaluates the impact of multiple loop transformations on the datapath and memory interface resources. DSE is iterative and estimations include datapath, memory and control costs.

To summarize previous efforts, the following analysis can be done (Table I). Most studies deal with loop nests and derive a DFG to compute area and delay estimations. Most of them apply iterative DSE. Some studies consider design optimizations to enhance accuracy but they mainly rely on a corrective factor. Datapath estimation is always targeted and memory and / or control units are only considered in a few studies. Finally, most efforts consider a single RTL architecture except [7][8][16] that propose several implementation models (e.g. pipelining, replication). Concerning this point, Choi et al. [17] propose an interesting contribution based on the definition of efficient performance models for specific application domains. Due to this, they are able to analyze a large design space and to take into account the important features of the applications. The main limitation is related to the difficult task of defining the performance model which requires expertise of the application domain.

B. Discussion

The work presented in the paper differs from these efforts in several respects. First, we target a more global exploration

as we do not restrict our input specification to basic blocks or loop nests. We consider applications including control structures, arrays and computations (due to the use of the H/CDFG as will be described in the paper). We target a single FPGA-based system, where datapath, memory and control units are all implemented within the FPGA which differs from [4][16] who consider external memories and from other works as they do not consider any memories. We perform an exploration for an RTL architecture model composed of a control unit, a memory unit and a datapath (up to our knowledge, no current work consider such a complete characterization). Our automatic DSE methodology exhibits the memory and computation parallelism potential, this point is original since most efforts only consider the computation parallelism potential and perform iterative DSE (except [4][16]). However, as we aim to prune the design space using low complexity estimators, the RTL architecture does not capture all of the optimizations which can be considered as a limitation. Compared to [3][4][7][16] our RTL architecture model is simpler (but obviously realistic) since we do not consider loop tiling [16], scalar replacement [4][16], data layout [4][7][16], data reusing [4][16], memory sharing [4][16] and memory pipelining [16]. Furthermore, we only deal with a simplified model of loop unrolling and folding. However, the simplicity of our model corresponds to a trade-off to achieve a low complexity exploration: our approach requires only a few minutes to prune the design space for a complete application which enables designers to focus on the highlighted subset of the design space using for example some of the efforts presented above. Concerning the area and delay performance model, we take into account both dedicated resources (embedded memory, DSP block) and LUT. However, compared to [3][7] we do not take into account routing impact on critical path, that is on clock period and logic optimizations. Furthermore like most efforts, we need to maintain a library of operators for each new FPGA which can be considered as a limitation unless it is a first step toward tool and technology independence.

In conclusion, the main contribution of our approach compared to previous efforts consists of dealing with both a complete application, a complete RTL architecture model (even if relatively simple), and a complete FPGA model. We perform an automatic DSE of main design parameters to highlight most promising subsets of the design space which then have to be refined to actually implement the final solution.

III. Exploration and Estimation Flow

In this section we present a general overview of the exploration approach. It is based on two main steps: (1) Structural Exploration and (2) Physical Mapping Estimation [18][19]. The first step performs an exploration at the RTL level that leads to highlight several architectures and characterize them in terms of execution resources and execution cycles. To provide a reliable characterization, Processing, Control and Memory units are considered and modeled through the following parameters:

- For the processing unit (datapath):
  - number and type of each operator;
  - bitwidth of each operator;
  - number of registers;
- For the control unit:
  - number of control states;
  - number of control signals;
- For the memory unit:
  - total memory size;
  - number of simultaneous reads from the RAMs;
  - number of simultaneous writes to the RAMs;
  - number of simultaneous reads from the ROMs;

The goal of this is to provide the main characteristics of several RTL architectures in order to point a relevant subset of the design space; we do not actually build the RTL architectures but instead we define their main characteristics based on the RTL model presented in Section III-B. The upper right corner of Fig. 1 provides the 2D chart used for a convenient exhibition of the results: each solution corresponds to a number of execution cycles N_c on the horizontal axis and the previously described parameters on the vertical axis.

The second step is called Physical Mapping Estimation; it computes the expected physical values of area and delay corresponding to the previous RTL solutions mapped onto an FPGA. Device characteristics such as operator / memory area and delay are used to compute accurate estimations of:

- the execution time;
- the number of required FPGA resources, including specific resources like dedicated operators / embedded memories;
- the corresponding FPGA use rate;

FPGA resources considered are logic cells (e.g. slices, logic elements), dedicated cells (e.g. embedded memories, DSP operators), I/O pads and tristate buffers. The results are gathered in a 2D chart illustrated in the lower right corner of Fig. 1. Each solution corresponds to an execution time value (on the horizontal axis) and is characterized by the number of FPGA resources of each type used (on the vertical axis). At the end of the DSE process, several implementation alternatives are thus provided and characterized through area vs. delay trade-offs.

Additional information concerning the inputs of the methodology is introduced now. First, we describe the intermediate representation model (H/CDFG), then the underlying implementation model is detailed.

A. From C to the H/CDFG model

The input specification is given in a subset of the C language. The use of an high level software language for the purpose of hardware implementation imposes some restrictions: only a basic subset of the C language is used namely for the specification of control structures (conditional / iterative structures), basic data types, arrays, function calls. Complex constructs like pointers / records / dynamic memory size allocation are not accepted.

The C code is automatically parsed into a Hierarchical Control and Data Flow Graph (H/CDFG) that has been specifically defined for our exploration and estimation usage [20][21].

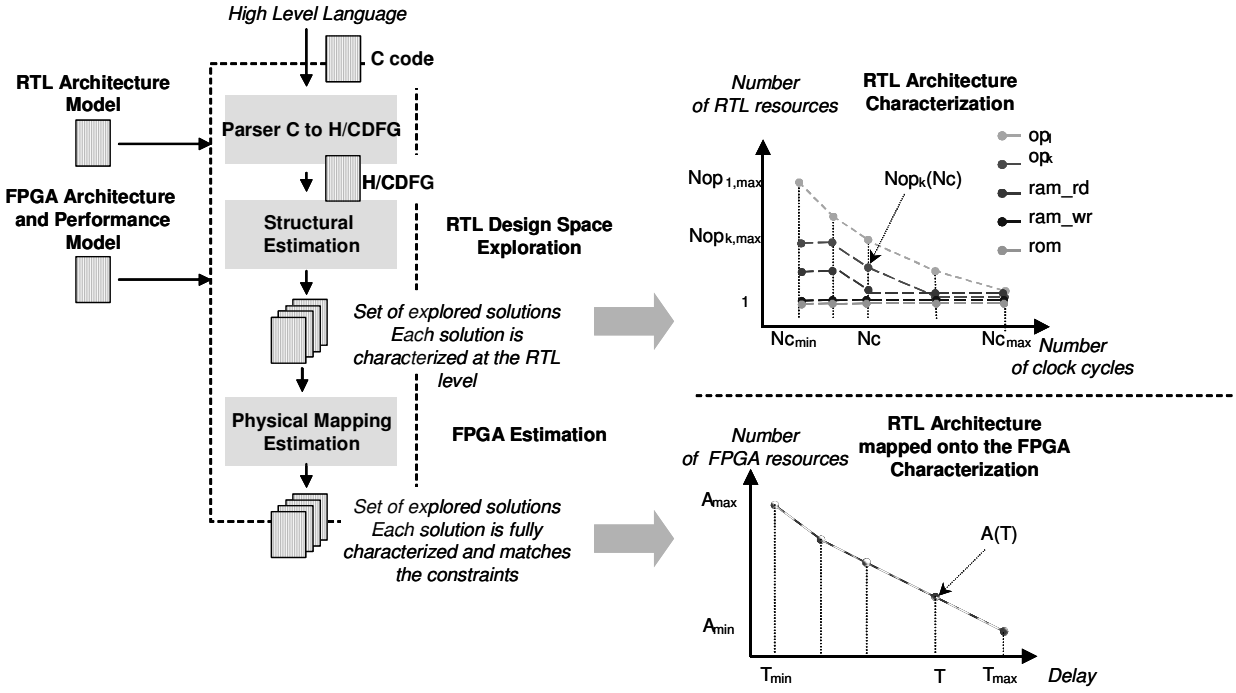


Fig. 1. Exploration / Estimation Flow: a two-step approach composed of 1) the structural definition of several solutions (i.e. at the RTL Level) and 2) the estimation of the physical characteristics of each solution (i.e. FPGA use rate vs. algorithm execution time).

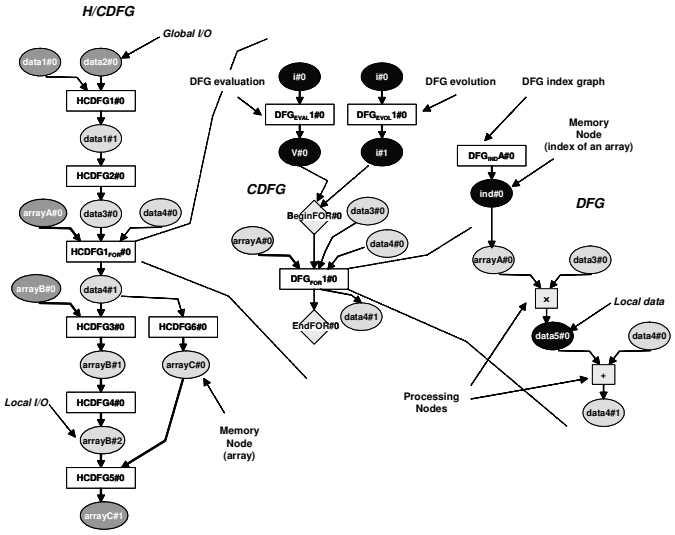


Fig. 2. Elements of a H/CDFG

This model is composed of three types of elementary (i.e. non hierarchical) nodes: processing, memory and conditional nodes. A processing node represents an arithmetic or logic operation. A memory node is a data access and a conditional node, a test / branch operation (e.g. *if*, *case*, *loops*). A Data Flow Graph (DFG) is a basic block, it contains only elementary memory and processing nodes. Namely it represents a sequence of non conditional instructions in the source code. For example the graph on the right of Fig. 2 is a DFG. A CDFG represents conditional or loop structures with their associated DFGs. The graph in the center of Fig. 2 is a CDFG. Finally, a H/CDFG is a composite graph that can include other H/CDFGs and / or CDFGs. It is used to encapsulate the entire application hierarchy, i.e. the nesting of control structures and graphs executed according to sequential or parallel patterns. The graph on the left of Fig. 2 is a H/CDFG.

A key point is the notion of I/O, local and global data. The parser automatically does a distinction between several types of memory nodes:

- $\eta^G_{I/O}$: global I/O which is an I/O data of the entire H/CDFG (whole application);
- $\eta^L_{I/O}$: local I/O which is an I/O data of a given subgraph. This data crosses the hierarchical levels of the H/CDFG representation;
- η^L_{data}: local data used to store internal processing results (temporary data within a DFG);
- $\eta_{constant}$: constant data;

The creation rules of a H/CDFG from a C code are based on a depth-first search algorithm [22]. A H/CDFG is created each time a conditional instruction (control or loop) is found in the original C code. When no conditional instructions remain in the current hierarchy level, a DFG is built.

B. RTL Architecture Model

As pointed out before, Processing, Control and Memory units have been considered to provide more realistic evaluations. Regarding this, some architectural assumptions have been made. We expose those choices that have been mainly defined to cope with FPGA specificities:

- The processing unit is composed of registers, operators and multiplexers. A general bus-based architecture has been preferred compared to a general register-based architecture since this solution minimizes the number of

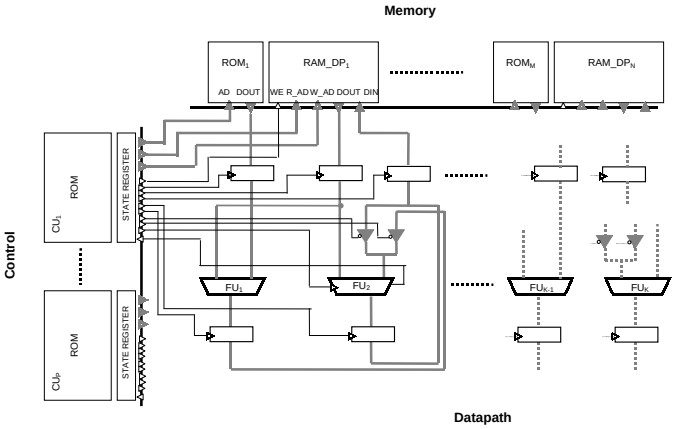


Fig. 3. RTL Architectural Model

interconnections between resources within the processing unit and promotes parallel execution. We assume that each output of an operator is connected to a register (within the same CLB).

- Concerning the control unit, Moore Finite State Machines (FSMs) are considered. We assume the use of microcode ROMs since recent devices permit efficient ROM integration using dedicated resources, way to keep the logic cells for the processing unit.
- The memory unit is composed of one or several RAM / ROM memories. ROMs are dedicated to constant storage and concerning RAMs, only dual port memories are considered since embedded memories are based on their use.

The choices presented above imply some design permissions / restrictions. However, this underlying implementation model is required to promote the definition of realistic solutions. It also allows reducing the gap between the estimation values of area and execution time (results of the physical mapping estimation step) and the actual design characteristics (results of the final physical synthesis). Other models could have been defined. In the following section, we will focus on the one presented in Fig. 3 and refer to it as our architecture template. Possibility to change / adapt the design template will be addressed in the Result Discussion section (Section VI-D).

IV. Structural Exploration

The structural definition step performs an exploration at the RTL level using the model of Fig. 3. Compared to a typical DSE flow, our approach is not iterative: an automated parallelism exploration is applied. Each RTL solution corresponds to a parallelism degree characterized for a cycle budget N_c by the number and the type of required resources (respectively $Nop_k(N_c)$ and op_k). N_c ranges from a sequential execution (when only one resource of each type is allocated) to the most parallel execution scenario (maximum number of resources is allocated; it corresponds to the critical path).

The definition of several RTL architectures starts with the scheduling of the DFGs (Section IV-C). Then analytical heuristics are used to deal with control patterns (section IV-D, CDFG estimation) and application hierarchy (Section IV-E.3, H/CDFG combination). Then a progressive bottom-up combination approach allows deriving an efficient schedule of the entire graph at low complexity costs. During the exploration step, both processing resources and memory bandwidth are considered through:

- The number and the type of execution units $N_{op_k}(N_c)$;
- The number of simultaneous reads (writes) from (to) the RAMs $N_{ram\_rd}(N_c)$ and $N_{ram\_wr}(N_c)$;
- The number of simultaneous reads from the ROMs $N_{rom\_rd}(N_c)$;
- The number of control states $N_s(N_c)$;

Each solution is thus characterized in terms of Processing (number of execution units) Control (number of control steps) and Memory (number of memory accesses). The computation is divided into five steps: (1) Pre-estimation, (2) Selection, (3) DFG scheduling, (4) CDFG Estimation and (5) H/CDFG Combination (Fig. 4).

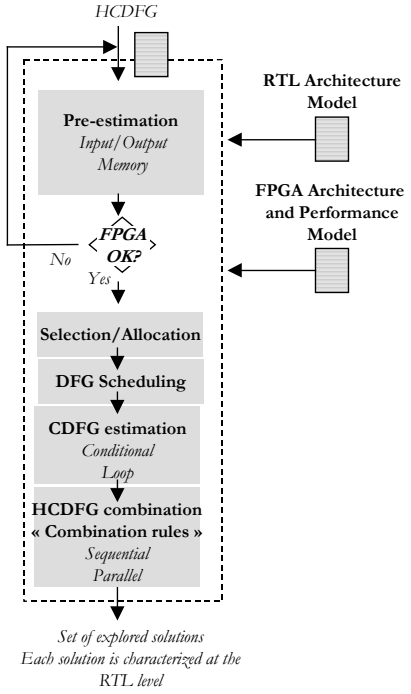


Fig. 4. Structural Estimation Flow

A. Pre-estimation

Pre-estimation checks whether a given FPGA device is suited or not in terms of I/O pads and memory resources for the application. During that step, the set of data nodes labeled $\eta^G_{I/O}$ are compared to the number of I/O pads in the target device. As stated before, $\eta^G_{I/O}$ is automatically derived from the number of formal parameters in the C code. Thus, from the number of data in the set $\eta^G_{I/O}$ and their corresponding bitwidth, the number of I/O pads required is computed and compared to the number of I/O pads available in the device. This information is described in a technology file called the FPGA Technology Architecture and Performance

Model (TAPM, Section V-A). Multiplexing techniques of I/O data are not considered.

The estimation of the total RAM size is computed from the set $\eta^L_{I/O}$. The technique used is the one proposed by Grun et al. [23] to compute a fast and reliable RAM size estimation from high level specifications. The estimation of the ROM size is derived from the number of elements in the set $\eta_{constant}$. Once computed, RAM and ROM sizes are compared to the amount of memory resources available in the FPGA.

If I/O pads and memory size conditions are satisfied, the next step of the flow is performed otherwise the designer has to select another device for evaluation.

B. Selection of Execution Units

First, a depth-first search algorithm [22] is used to list all the operations in the H/CDFG. The selection of the execution units is applied using a greedy algorithm. All available execution units are described and classified (speed decreasing order) in the TAPM file. For each operation, the first execution unit supporting the operation in the TAPM file is selected. Although, such an approach may not always give the best solution, it permits a first and rapid evaluation. However, regarding the greedy algorithm limitations, the designer has also the possibility to apply a manual selection in order to choose the most interesting execution resources, on the basis of his design experience. This approach can also be considered to refine the results obtained from a first greedy approach. Then, a clock value can be set. As previously, the designer has the possibility to manually choose a clock value, otherwise it is set to T_{clk_max} which is the propagation delay of the slowest execution unit. An example of clock period exploration and related limitations is given in Section VI-C.

C. DFG Scheduling

When the execution units have been selected and a clock value has been set, scheduling is applied to each DFG. A time-constrained *list-scheduling* heuristic extended to deal with both processing and memory nodes [24] is used with the goal to minimize the number of execution units and bandwidth requirements. Several time constraints (N_c) are considered and ranges from the most parallel solution (fastest execution) to the most sequential one (slowest execution) as illustrated in Fig. 5. Thus, the whole DFG parallelism is explored and each parallelism solution corresponds to a given N_c value (on the horizontal axis of the 2D chart).

In the example of Fig. 5, three solutions representing different number of resources / clock cycles trade-offs are provided. To implement the first solution (most parallel one), the processing unit must include 2 adders and 2 multipliers, and the memory unit must permit 2 simultaneous read accesses (and 1 write access). Note here that only I/Os of the graph and constants are mapped to the memories since internal data are assigned to registers (there is no register allocation as stated in section III-B). Finally, the control unit requires 5 states to manage the scheduling of this DFG.

Once scheduled, a DFG is replaced in the graph by its estimation results (resources vs. clock cycles curve). Then, progressive combination of the curves is applied using a bottom-up approach and leads to the estimation results of the entire specification. The combination rules that are used depend on the type of dependence between the parent graphs (Fig. 6). A H/CDFG may be composed of four types of dependencies: two of them correspond to control dependencies (conditional and loop structures) and the two others correspond to execution dependencies (sequential and parallel execution). For each type of dependency, the estimation curves are combined according to the equations presented below.

D. CDFG estimation

This section explains how control constructs (CDFGs) are combined. DFG cores within CDFGs are processed as explained previously. Then we have to deal with two types of control: (1) conditional structures (tests) and (2) loop structures (iterations).

1) CDFG conditional structures: Conditional structures correspond to CDFGs. They are equivalent to an "if" statement in a programming language. An "if" structure is composed of three subgraphs (DFGs): one for the evaluation of the condition and two for the true and false branches (respectively corresponding to the three subscript 0, 1 and 2 in the following equations). Branching probabilities (P_b) are used to balance the execution time of each branch. Application profiling is used to obtain the execution probabilities.

Each scheduled DFG is characterized by a set of solutions with different trade-offs representing the number of execution units vs. clock cycles. In our approach, we first compute the exhaustive combination of all these solutions and then we remove all non optimal results (i.e. we only keep the Pareto solutions [25]). Thus, the estimation of the solutions (characterized by a cycle budget N'_c) is obtained as follows:

$$N'_c = \lceil [N_{c_0} + (P_{b_1} * N_{c_1}) + (P_{b_2} * N_{c_2}) + 1] \rceil$$

$$N'_s(N'_c) = N_{s_0}(N_{c_0}) + N_{s_1}(N_{c_1}) + N_{s_2}(N_{c_2}) + 1$$

$$N'_{op_k}(N'_c) = MAX[N_{op_{k_0}}(N_{c_0}), N_{op_{k_1}}(N_{c_1}), N_{op_{k_2}}(N_{c_2})]$$

where $N_{s_i}(N_{c_i})$ and $N_{op_{k_i}}(N_{c_i})$ are respectively the number of states and the number of operators / memory accesses in the DFG labeled i for an execution time of N_{c_i} cycles. Execution units and memory accesses are estimated under the assumption of maximum sharing between the three DFGs by taking the maximum value (since they are never executed simultaneously). We assume data are always available in the expected memories. The total number of control states is the sum of the number of states of each DFG, plus one for the branching to the first state of the branch to execute. The equations above are computed for each possible combination of solutions, i.e. each possible combination of N_c (exhaustive approach) and results in a new resources vs. clock cycle curve.

2) CDFG loop structures: The most common scheme used to estimate a loop structure is based on a sequential execution. This scenario is considered when the number of iterations is not known statically or in case of data dependences between iterations. In that case, the estimation is computed like this: (1) first, the three subgraphs composing the loop (evaluation,

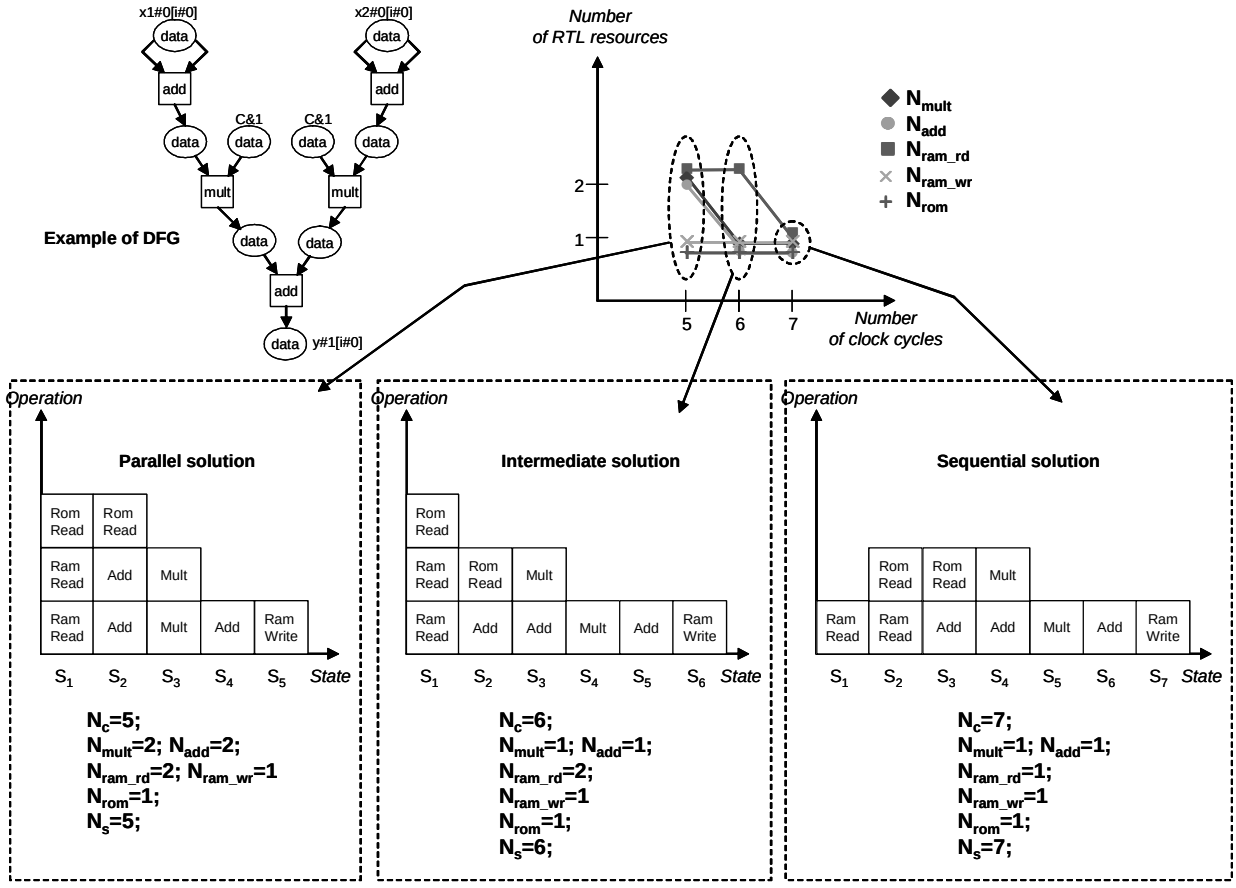


Fig. 5. DFG scheduling example

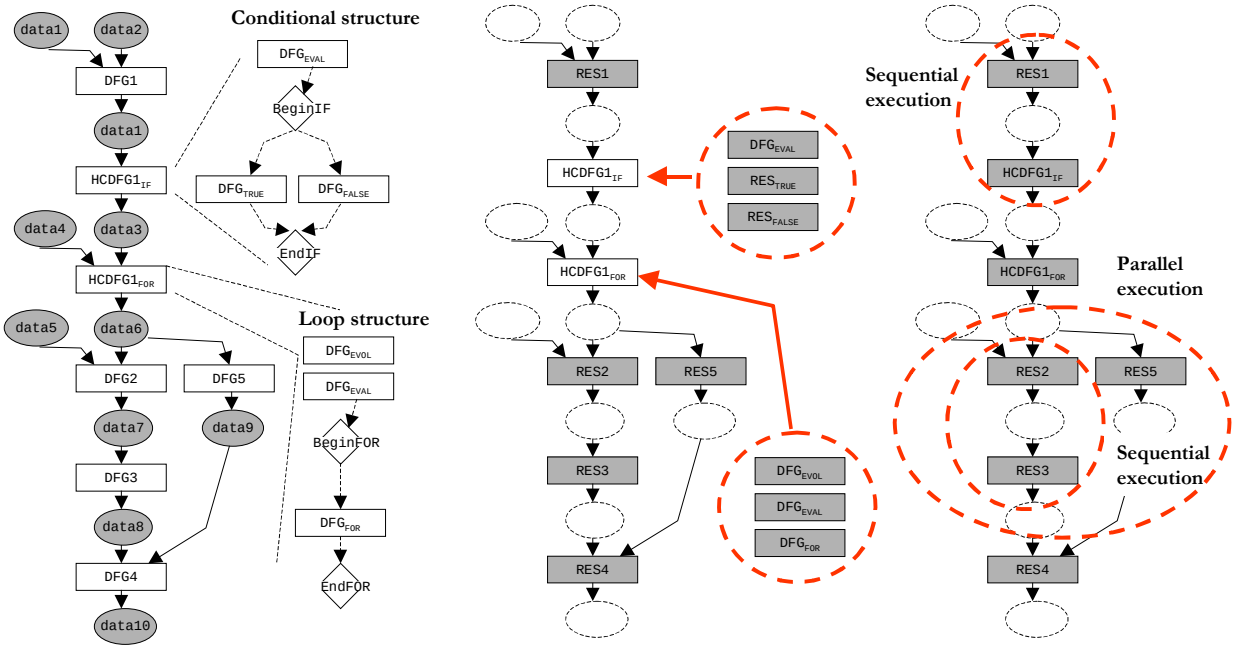


Fig. 6. Combination schemes

core and evolution) are processed through DFG scheduling. (2) then, a sequential combination of the three subgraphs is applied. (3) and finally, the entire loop structure is estimated by repeating N_{iter} times the loop pattern (with N_{iter} the number of iterations). This last step leads to the equations below.

- Sequential execution:

$$N'_c = N_{iter} * (N_c + 1)$$
$$N'_s(N'_c) = N_s(N_c) + 1$$
$$N'_{op_k}(N'_c) = N_{op_k}(N_c)$$

where N'_c, $N'_s(N'_c)$ and $N'_{op_k}(N'_c)$ correspond to the solution resulting from the combination.

In this case the execution model does not take into account memory and computation pipelining, the execution is totally sequential. However, we also propose a low complexity technique to unfold loops (partial unrolling and folding execution) in a way to explore more efficiently the available parallelism. Unfolding loops leads to the exploration of the memory and computation parallelism and to reduce the critical path. This technique is based on a simplified execution model since data sharing, data layout and advanced unrolling schemes are not considered. Several parallelism factors f_p are defined to help exploring this intra-loop parallelism.

- Partial unrolling and folding:

$$N'_c = N_c + (N_{iter}/f_p - 1)$$
$$N'_s(N'_c) = N_s(N_c) + (N_{iter}/f_p - 1)$$
$$N'_{op_k}(N'_c) = N_{op_k}(N_c) * f_p$$

where f_p corresponds to the number of parallel executions of the loop kernel (f_p takes a value between 1 and N_{iter}). A f_p value equal to one corresponds to a full pipeline execution of each iteration of the loop kernel. When f_p is set to N_{iter}, it corresponds to a full parallel execution of each iteration of the loop kernel. This estimation of pipelined and parallel executions reduces the total number of execution cycles. Consequently, the number of resources (thus area) increases according to the parallelism factor f_p.

This approach may not always be applicable especially in case of dependencies (resource, control and data). But it can be very helpful to exhibit the parallelism level that will allow respecting a given time constraint. Then, one can partially unfold the loop specification according to the parallelism factor and apply a full DFG schedule. This way of proceeding is close to optimality but it is at the expense of complexity. We are developing by now a new heuristic able to consider data dependencies based on the analysis of array index variations.

E. H/CDFG exploration

This section explains how we derive the entire estimation of a H/CDFG. When conditional and loop structures have been estimated (Section IV-D), the two following combination rules are applied to address sequential and parallel combinations of CDFGs.

1) H/CDFG sequential execution: The analytical equations for the combination of two CDFGs executed sequentially (labeled 1 and 2 in the following equations) are based on the assumption of a maximum reuse of the execution units. This means that the execution units are shared for the execution of both graphs:

$$N'_c = N_{c_1} + N_{c_2}$$
$$N'_s(N'_c) = N_{s_1}(N_{c_1}) + N_{s_2}(N_{c_2})$$
$$N'_{op_k}(N'_c) = MAX(N_{op_{k_1}}(N_{c_1}), N_{op_{k_2}}(N_{c_2}))$$

First, an exhaustive combination of the solutions between the two graphs is computed according to the equations above. Then, all non optimal solutions are rejected. The same combination scheme is repeated for each op_k and for the memory accesses.

2) H/CDFG parallel execution: The estimation results for a parallel execution of two graphs are computed as follows:

$$N'_c = MAX(N_{c_1}, N_{c_2})$$
$$N'_s(N'_c) = N_{s_1}(N_{c_1}) + N_{s_2}(N_{c_2})$$
$$N'_{op_k}(N'_c) = N_{op_{k_1}}(N_{c_1}) + N_{op_{k_2}}(N_{c_2})$$

The execution time is the maximum of both execution times. Execution units and memory accesses are estimated in the worst case by taking the maximum value. Resource sharing is not considered (unlike sequential execution) and may lead to an overhead in some cases. The reason of this is that such an analysis requires the storage and the analysis of each operation schedule. We believe the impact on the exploration algorithm complexity is too high regarding the improvement of the estimation accuracy.

Given this, the number of states $N'_s(N'_c)$ is the sum of the number of states of each graph since we do not merge FSMs, again for complexity reasons. Instead, we suppose each sub part of the architecture has its own FSM. This results in a hierarchical FSM composed of several microcode ROMs. Like in the case of sequential execution, this combination scheme is repeated for each possible combination of solutions N_c and each op_k, for both execution units and memory accesses.

3) H/CDFG global combination algorithm: The goal of that step is to correctly take into account the control and execution dependencies for the entire H/CDFG. At the beginning, all the nodes of the H/CDFG are analyzed with a depth-first search algorithm [22]. When a node is identified as a composite one (namely a hierarchical node containing subgraphs), a recursive function is called until a DFG is reached (Algorithm 1). All the DFGs are scheduled this way and replaced by their corresponding structural characterization. Then, conditional / loop structures are analyzed with respectively the $if\_combination$ and $for\_combination$ routines (implementing the heuristics of Sections IV-D.1 and IV-D.2) until there remains zero CDFG unprocessed. The $execution\_combination$ routine is invoked to process all the possible sequential / parallel execution of sub graphs (heuristics of Sections IV-E.1 and IV-E.2). And finally, the global combination algorithm combines recursively all the

Alg. 1 H/CDFG exploration algorithm

```
H/CDFG_ Expl(graph current_G)
{
nb_composite = 0
for  each node in current_G do
  if current node is a composite node then
    nb_composite = nb_composite + 1
    if composite node is not a result node then
      temp_G ← sub graph of the composite node
      H/CDFG_ Expl(graph temp_G)
    end if
  end if
end for

father_type = type of current_G father
if father_type = if then
  Apply if_combination for current_G
else if father_type = for then
  Apply for_combination for current_G
else if nb_composite > 1 then
  Apply execution_combination for current_G
end if
}


execution_combination(graph current_G)
{
Initialize the list of current node LCN with root
nodes in current_G
while LCN non-empty do
  LCN = successors(LCN)
  for each node ∈ LCN do
    Parallel combination of predecessor nodes
    Sequential combination of current node and
    predecessor node
  end for
end while
}
```

nodes until there exists only a single node corresponding to the entire application. These combinations are illustrated on the example of Fig. 6. They lead to the final characterization curve of the whole specification. The last step to complete the Exploration / Estimation flow is to compute the FPGA resources use vs. execution time estimates (Physical Mapping Estimation) for each RTL solution.

V. Physical Mapping Estimation

The physical characterization provides the FPGA resources use $A(T)$ vs. temporal constraint T (physical time unit, ns, μs). It is derived from previous structural characterization of each solution (Fig. 7). The resources of the FPGA considered to compute $A(T)$ are logic cells, dedicated cells, tristate buffers and I/O pads. An FPGA characterization file is used to describe a given device (Technology Architecture and Performance Model). It contains the following information:

- The number and the type of the FPGA resources (i.e number of LCs, DCs, multiplexers / tri-state buffers and I/O resources);
- The area and delay of operators;
- The area and access times of memories (dual port RAMs and ROMs);

This information is obtained from the data sheet of the target device and from the synthesis of basic arithmetic / logic operators and memories.

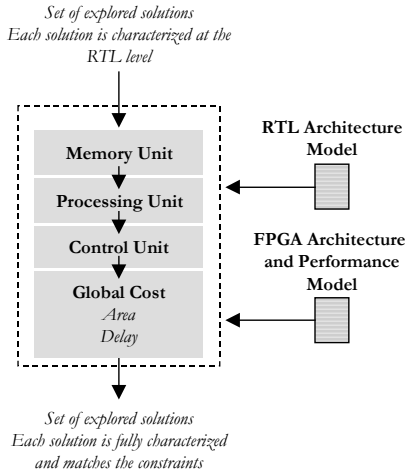


Fig. 7. Physical Estimation Flow

A. FPGA Technology Architecture and Performance Model

Table II gives a simplified example of characterization for a Virtex V400EPQ240-7 device [26]. This FPGA contains 4800 Logic Cells and 40 Dedicated Cells (Slices and BRAM respectively). Each operator is characterized for usual bitwidths (8, 16 and 32 bits) by the number of FPGA resources used, by the corresponding delay and the number of control signals. To give an example, an eight bits adder is characterized as follows: 4 slices, 4.9ns, 8 bits, 1 control line (corresponding to the signal to drive the output register of the adder). Those values are obtained after a logic synthesis step. Memories are characterized by the type of FPGA resources needed for implementation, the storage capacity and read / write delays. A generic characterization (i.e. independent from size and bitwidth) has been defined on the basis of the number of memory bits per cell and the worst case access time. Furthermore, two choices are left to implement a memory in both Xilinx and Altera devices. For example in an Apex EP20K200RC208-1, logic cells (*logic elements*) or dedicated cells (*ESBs*) are available for implementation. The respective characterizations are 16 bits / logic element, 11.2ns and 2048 bits / ESB, 4ns. As we can notice, dedicated cell implementation is faster, but it is also more area efficient. In our approach, dedicated cell implementation is always a preferable solution since it allows saving the logic cells for custom user defined functions.

B. Technology projection

From the characteristics of a given RTL solution and a target FPGA, we derive a simple and accurate estimation of the expected FPGA use rate and algorithm execution time. For better accuracy, a specific "projection" process has been defined for each unit of the architecture.

1) Memory unit: A simple approach is applied to evaluate the area of the memory unit. It is based on the total memory size, the number of simultaneous accesses, and the characteristics of the memory resources within the FPGA. As stated before, two types of implementation are considered: logic cells or dedicated cells. According to the type of cells used, the area of the memory unit is estimated as follows:

TABLE II
VIRTEX V400EPQ240-7 FPGA CHARACTERIZATION

Virtex V400EPQ240-7						
FPGA characteristics	LC	# LC	DC	# DC	# Tri	# I/O
	slice	4800	BRAM	40	4800	48
Execution units	op	impl	latency	area	bitwidth	ctl lines
adder	+	LC	4.9ns	4	8	1
sub	-	LC	4.9ns	4	8	1
mult	×	LC	12.3ns	36	8	1
comp	<,>,>=,<=	LC	4.8ns	6	8	1
equal	=	LC	5.0ns	4	8	1
shift reg	<<,>>	LC	4.9ns	4	8	1
reg		LC		4	8	1
mux		Tri		4	8	2
Memory	impl	access time	bit per cell	latency read	latency write	
RAM DP	LC	rw	32	13.4ns	13.4ns	
RAM DP	DC	rw	4096	7.2ns	7.2ns	
ROM	DC	r	4096	7.2ns		

- Logic cell implementation:

$$N_{lc}^{ram} = \lceil (MS_{RAM} * W_{ram}/N_{bits/lc}^{ram}) \rceil$$
$$N_{lc}^{rom} = \lceil (MS_{ROM} * W_{rom}/N_{bits/lc}^{rom}) \rceil$$

where MS_{RAM} and MS_{ROM} are respectively the total memory size for the RAM and the ROM memories. $N_{bits/lc}^{ram}$ and $N_{bits/lc}^{rom}$ are the number of memory bits per logic cell and W_{ram}, W_{rom}, the bitwidth of the data to be stored.

- Dedicated cell implementation:

In a first approach, the number of memories can be approximated to the maximum of the number of simultaneous read and write operations $N_{mem}^{ram} = MAX[N_{ram\_rd}, N_{ram\_wr}]$. However, the type of resource used is important because only one memory can be integrated in a single dedicated cell. So in this case, the number of embedded memories is computed by taking the maximum value between the number of dedicated cells needed to implement the total memory size and the number of simultaneous accesses:

$$N_{dc}^{ram} = MAX[\lceil (MS_{RAM} * W_{ram}/N_{bits/dc}^{ram}) \rceil, N_{ram\_rd}, N_{ram\_wr}]$$

$$N_{dc}^{rom} = MAX[\lceil (MS_{ROM} * W_{rom}/N_{bits/dc}^{rom}) \rceil, N_{rom}]$$

$N_{bits/dc}^{ram}$ and $N_{bits/dc}^{rom}$ correspond to the number of memory bits per dedicated cell and W_{ram}, W_{rom}, the bitwidth of the data to be stored.

Once the number of memories of each type is known, the number of control signals to drive the RAM (N_{cs}^{ram}) and ROM memories (N_{cs}^{rom}) are derived. They correspond to the address and *write enable* signals (Fig. 3):

$$N_{cs}^{ram} = (2 * W_{adr}^{ram} + 1) * MAX(N_{ram\_rd}, N_{ram\_wr})$$
$$N_{cs}^{rom} = W_{adr}^{rom} * N_{rom}$$

where the sizes of the address bus (W_{adr}^{ram} and W_{adr}^{rom}) are derived from the number of words in the memories:

$$W_{adr}^{ram} = \lceil log_2(MS_{RAM}/MAX(N_{ram\_rd}, N_{ram\_wr})) \rceil$$
$$W_{adr}^{rom} = \lceil log_2(MS_{ROM}/N_{rom}) \rceil$$

The current RTL model does not include a specific address generator. Hence, the control of the address signals of the RAMs and ROMs is left to the control unit. Thus, the expressions of N_{cs}^{ram} and N_{cs}^{rom} take respectively into account W_{adr}^{ram} and W_{adr}^{rom}. The total number of control signals for the memory unit is:

$$N_{cs}^{mu} = N_{cs}^{ram} + N_{cs}^{rom}$$

2) Processing unit (datapath): The area of the processing unit is computed by adding the contribution of each execution unit ($N_{lc}^{op_k}$ and $N_{dc}^{op_k}$):

$$N_{lc}^{pu} = \sum_{op_k} N_{op_k} * N_{lc}^{op_k}$$

$$N_{dc}^{pu} = \sum_{op_k} N_{op_k} * N_{dc}^{op_k}$$

For example, three eight bits adders require 12 slices in a Virtex V400E ($N_{lc}^{add_{8bit}} = 4\ slices$). Like in the case of the memory unit, the number of control signals required to drive the datapath is considered. There are four types of control signals, according to the current model of Fig. 3:

- signals to control the output register of each execution unit. The number of signals of this type is equal to the number of execution units N_{cs}^{op};
- signals to select the operation for multi-functional units $N_{cs}^{multi\_op}$;
- signals to control the registers of the memories read / write ports $N_{cs}^{reg} = N_{reg}^{ram} + N_{reg}^{rom}$;
- signals to control multiplexors / tristates N_{cs}^{mux};

The number of control signals for the processing unit is then

$$N_{cs}^{pu} = N_{cs}^{op} + N_{cs}^{multi\_op} + N_{cs}^{reg} + N_{cs}^{mux}$$

and the total number of control signals for the entire architecture is

$$N_{cs} = N_{cs}^{pu} + N_{cs}^{mu}$$

3) Control unit: The area of the control unit is derived from the number of states and the number of control lines [27]. Control logic is supposed to be integrated in a ROM memory. The area is computed from its number of words and data bitwidth:

$$N_{bits\_state\_reg} = \lceil log_2(N_s) \rceil$$

$$N_{bits\_rom} = N_{bits\_state\_reg} + N_{cs}$$

where N_s is the total number of states needed to schedule the entire graph. The area of the control logic is obtained from the area used by a $N_s * N_{bits\_rom}$ ROM. Then, the corresponding FPGA resource for a logic cell or a dedicated cell implementation is respectively:

$$N_{lc}^{rom} = \lceil (N_s * N_{bits\_rom} / N_{bits/lc}^{rom}) \rceil$$

$$N_{dc}^{rom} = \lceil (N_s * N_{bits\_rom} / N_{bits/dc}^{rom}) \rceil$$

4) Global cost characterization: The total area in term of FPGA use is computed for each type of FPGA resource by adding the contribution of each unit of the architecture:

$$N_{lc} = N_{lc}^{mu} + N_{lc}^{pu} + N_{lc}^{cu}$$

$$N_{dc} = N_{dc}^{mu} + N_{dc}^{pu} + N_{dc}^{cu}$$

$$N_{tristate} = N_{tristate}^{pu}$$

The physical value of the execution time is computed from the value of the clock period defined during the selection step (Section IV-B):

$$T = N_c * T_h$$

The above computation process is then iterated for each architectural solution resulting from the structural explorations and leads to the final cost vs. performance characterization (Fig. 1).

VI. Experiments and Results

This section is organized as follows: first, we apply the exploration methodology on several examples representative of different algorithmic complexities and processing characteristics (intense data processing, control dominated, high / low parallelism potential). Then, we address the problem of estimation accuracy. For this purpose, we compare the synthesis results of one architectural solution with the area and delay estimates given by our exploration tool. Three approaches have been considered for comparison: the first one is based on hand coded design, the second one on HLS design and the last one is based on pre-characterized IPs. Then, we apply the methodology on a 1D Discrete Wavelet Transform to illustrate how easy and independent from any design experience background the exploration process can be performed. Finally, a discussion is provided in order to analyze the drawbacks / benefits of the approach and stress the differences compared to existing works.

TABLE III
Number of solutions generated vs. exploration times for several DSP applications

	Virtex V400EPQ240		Apex EP20K200EFC484	
	sol.	expl. time	sol.	expl. time
FIR	5	0.04 sec	5	0.03 sec
Volterra	11	5.6 sec	11	4.2 sec
F22	40	2.3 sec	40	2.6 sec
Adaptive	4	2.6 sec	4	1.8 sec
FFT	56	0.4 sec	56	0.4 sec
DWT	342	8.7 min	342	9.4 min
DCT	14	1.5 sec	14	1.6 sec
G722	16	0.7 sec	16	0.4 sec
MPEG	2	3.1 sec	2	3.1 sec
Huffman	4	4.2 sec	4	4.5 sec

A. Applications

The methodology presented in this paper has been integrated in a framework for the codesign of SoCs called *Design Trotter* [21]. With the help of this tool, early exploration and design space pruning of applications from C specifications is fast and easy. Applications representative of several algorithmic complexities and processing characteristics have been used to analyze the design space coverage vs. exploration time tradeoffs. The first seven examples in Table III tackle typical DSP processing: filtering (FIR, Volterra, F22 and Adaptive), transforms (Fast Fourier Transform, 2D Discrete Wavelet Transform and Discrete Cosine Transform). These examples exhibit high amounts of memory and computation parallelism with intense data processing / storage, especially in the case of the 2D DWT. The last examples (G722 speech coding recommendation, MPEG and Huffman coding) are more control dominated systems with low parallelism potential.

Table III presents the number of solutions explored and the related exploration time obtained with a Pentium 3 processor running at 1.2 GHz. The results show the ability of the tool to define several architectural solutions in a reasonable amount of time, even in the case of complex specifications. In the 2D DWT for instance *Design Trotter* generates about 350 RTL solutions and the corresponding area / delay estimates within 10 minutes. For comparison, the time required to perform only the *logic synthesis* of *one single* solution is several orders of magnitude higher (about one day).

The results of Table III show the effectiveness of a global "low complexity" estimation framework operating from early specifications and exploring several RTL solutions characterized in terms of processing, control and memory. In the following, we address the problem of estimation accuracy in order to evaluate the relevance of the estimates provided.

B. Accuracy

To make a valuable evaluation of accuracy, three approaches have been carried out: compare estimations with hand coded

TABLE IV
ESTIMATION VS. SYNTHESIS FOR THE G722 PREDICTOR FUNCTION

	Virtex V400EPQ240-7				Apex EPK20K200EFC484-2X			
	Estimation		Synthesis		Estimation		Synthesis	
	slices	T_{ex} (ns)	slices	T_{ex} (ns)	lgc elt	T_{ex} (ns)	lgc elt	T_{ex} (ns)
Parrec	9	6	10	6	17	9	19	9
Recons	9	6	10	6	17	9	19	9
Upzero	217	1224	255	1171	608	1504	400	1272
Uppol2	257	292	303	300	857	358	718	302
Uppol1	216	230	275	254	589	282	612	236
Filtez	150	77	163	88	518	94	648	103
Filtep	181	593	177	511	484	728	497	515
Predic	9	6	10	6	17	9	49	9
G722Predictor	1166	1224	1263	1317	3132	1504	3027	1318

TABLE V
ESTIMATION VS. SYNTHESIS ERROR AND EXPLORATION VS. (LOGIC) SYNTHESIS TIME FOR G722 PREDICTOR FUNCTION AND DWT 2D FUNCTION

	Virtex V400EPQ240-7				Apex EPK20K200EFC484-2X			
	Accuracy (%)		Explo vs. lgc synth		Accuracy (%)		Explo vs. lgc synth	
	slices	T_{ex}	T_{expl} (sec)	T_{synth} (min)	lgc elt	T_{ex}	T_{expl} (sec)	T_{synth} (min)
Parrec	-10	0	0.05	1	-10.5	0	0.05	1
Recons	-10	0	0.05	1	-10.5	0	0.05	1
Upzero	-14.9	+4.5	0.22	5	+52	+18.2	0.06	5
Uppol2	-15.2	-2.7	0.11	5	+19.4	+18.2	0.11	5
Uppol1	-21.5	-9.4	0.11	5	-3.8	+19.5	0.11	5
Filtez	-8	-12.5	0.05	1	-20.1	-8.7	0.05	1
Filtep	+2.2	+16	0.05	2	-2.6	+41.4	0.05	2
Predic	-10	0	0.05	1	-10.5	0	0.06	1
G722Predictor	-7.7	-7.1	0.9	15	+3.4	+14.1	0.4	10
1stHLftStep	+6.9	+7.1	0.1	5	+1.4	+7.6	0.05	8
1stHDLftStep	+4	+0.6	0.05	5	+2.6	+1.9	0.06	8
2ndHLftStep	+5.1	+13.9	0.06	5	+2.8	+9.3	0.05	8
2ndHDLftStep	+2.5	+9.8	0.06	5	-0.2	+1.7	0.05	8
Hscaling	+2.7	+3.6	0.1	5	+4.9	+3.6	0.1	8
Hrearrange	+46.8	-25	0.06	5	+67	+9.1	0.05	8
1stVLftStep	+7.1	+25.5	0.1	5	-0.2	+2.9	0.05	8
1stVDLftStep	+5	+16.9	0.05	5	-0.6	+5.1	0.06	8
2ndVLftStep	+5.1	+18.5	0.06	5	+1.1	+2.9	0.05	8
2ndVDLftStep	+3.4	+18.3	0.06	5	-2.6	+7.7	0.05	8
Vscaling	+3.4	+5.5	0.1	5	+3.2	+3.8	0.1	8
Vrearrange	+50.9	-5.5	0.06	5	+61	+3.8	0.05	8
DWT 2D	+35.9	+18.2	5min	1.5days	+37	+3.1	5min	2days

designs, compare estimations with HLS design and compare estimations with pre-characterized IPs. The first approach is needed because the accuracy of the estimations is strongly related to the RTL model of Fig. 3. Thus, it highlights the accuracy of the physical mapping estimation which is important since it provides the designers with reference values allowing reliable comparison of area / performance trade-offs. The second and the third approaches can lead to more significant variations because the RTL models used are different. But on the other hand, a comparison with an automated architectural synthesis tool and with IP designs is required to discuss the relevance of the architectural solutions provided.

1) Comparison with hand coded designs: The two applications considered are a speech coder (G722) [28] and a 2D Discrete Wavelet Transform [29]. Concerning the G722 recommendation, we focused on the predictor which is the processing core of the application. The predictor is composed of eight sub-functions that are executed concurrently and represents an average of 260 lines of C codes. The DWT algorithm considered is based on a lifting scheme process composed of twelve filtering functions applied sequentially. Six loops are first executed to compute the horizontal transform and then six loops are executed to compute the vertical transform. Each loop is a second order nested loop. The complexity of the algorithm is about 250 lines of C codes that corresponds to almost 500 lines of H/CDGH grammar.

In the following, we compare exploration times vs. logic synthesis times (it does not include the time spent for architecture synthesis), for two representative devices of recent FPGA families: Virtex V400EPQ240-7 [26] and Altera Apex

TABLE VI
DESIGN TROTTER FIR 16 VS. HLS (GAUT) FIR 16

FIR 16 (Design Trotter)	FIR16 (GAUT)
1 add 16 bit	1 add 16 bit
1 mult 16 bit	1 mult 16 bit
3 reg 16 bit	6 reg 16 bit
1 RAM, 1 ROM	Mem unit not generated
19 control steps	19 control steps
0.3 sec to generate 5 solutions	1 min to generate this solution
+ area / time estimates	+ 5 minutes for logic synthesis

TABLE VII
DESIGN TROTTER FIR 16 VS. IP (XILINX) FIR 16

	Solutions			
	pipelined		sequential	
	Xilinx	DT	Xilinx	DT
$BRAM$	1	16	1	1
$Slice$	1542	2368	157	168
$Delay\ (ns)$	306	304	2394	3213
$Frequency(MHz)$	81.5	62.5	116	62.5

EP20K200EFC484-2X [15]. The ISE Foundation and Quartus synthesis tools have been used to target the respective device. In order to provide a significant evaluation, the entire applications have been synthesized in both cases (G722 Predictor and 2D DWT), as well as each sub-function independently. This way, the average accuracy has been computed on a total of 22 designs. Results are given in Table IV and Table V. Table IV reports the number of slices and logic elements after estimation and synthesis. Table V gives the average error in percent and reports the exploration vs. synthesis time. The average accuracy is about 13.5% and 10% respectively for area and execution time values. Some variations may be noticed locally, like in the case of function *upzero*, that are due to logic optimizations applied by the logic synthesis tool. In that case, the difference is due to the simplification of a multiplier with one of its operands remaining constant. Other variations like *horizontal* and *vertical rearrange* in the DWT are caused by the address generation model left to the control unit. This is especially sensitive in the case of this function where memory accesses are critical.

Concerning the processing times, only the logic synthesis times have been reported for comparison. The exploration times in the case of the G722 predictor and DWT are respectively 1 second and 5 minutes (to generate 16 and 342 solutions) while the respective logic synthesis times (for only *one* solution) are respectively 10 minutes and more than 1 day, plus the additional time needed to write the RTL description by hand (about 1 month). These results stress the benefits of the approach on the design times and the ability to rapidly select the most interesting solutions based on relatively accurate results.

2) Comparison to HLS design: In this section, we propose to compare our exploration / estimation methodology with a High Level Synthesis approach. Thus, we perform an exploration using both *Design Trotter* (DT) approach and an HLS tool (GAUT HLS [30]). The example is a 16-tap FIR filter for which DT generates 5 RTL solutions. For the purpose of our comparison, we selected one of those 5 solutions and constrained GAUT HLS tool with the corresponding time constraint ($304\ ns$) and clock period ($16\ ns$). The left row of Table VI reports the characteristics of the DT solution and gives the necessary information to design the RTL architecture with the HLS tool. As we can see, there are some slight differences which are mainly due to the different RTL models used in both tools:

- concerning the number of registers, the difference is due to the fact that each functional unit is assumed to be associated with an output register in our approach (Section III-B). So the actual number of registers is equal to 3 (originally estimated) plus 2 (one at the output of adder and multiplier).
- concerning memory, *Design Trotter* computes a basic estimation of the number of memories that is simply based on the analysis of simultaneous accesses (N_{ram_rd}, N_{ram_wr} and N_{rom}). GAUT HLS does not address the memory requirements in the current version used for this evaluation.

Finally, from the analysis of the processing times (bottom of Table VI) it arises the complementarity of the exploration methodology with High Level Synthesis more than an opposition: the use of an HLS tool may need several time consuming iterations before defining a suitable architecture. With our tool, several parallelism solutions are automatically generated with an average accuracy of 11.6%. Once a suitable solution has been selected, the designer can then constrain the HLS tool (using DT based directives about local / global time constraints, parallelism, scheduling, allocation, ...) to meet the solution. The confidence in finding more surely a suitable implementation is thus significantly enhanced.

3) Comparison with pre-characterized IPs: To compare our exploration approach with existing IPs, we also considered a 16-tap FIR filter obtained from the Xilinx core generator [31]. Two IPs have been used: the first one is a full pipelined filter whereas the second is a full sequential one. Design Trotter provides 5 solutions, we considered the fastest and the slowest ones for comparison with Xilinx' IPs in Table VII. For the fastest solution, 16 BRAMs are needed because a maximum of 16 simultaneous memory accesses have been estimated while Xilinx's solution uses only one single BRAM. The clock value we have set corresponds to the delay of the slowest execution unit (multiplier in our case). It keeps the same value for both solutions because we do not consider any logic optimization. Compared to Xilinx' solutions, the differences are due to the high optimization effort on the IP core. Unfortunately no information is available on this to permit comparison, for obvious confidentiality reasons. However, we believe the accuracy of estimations is reasonable regarding the abstraction level of the input specification. Further improvement on this point is possible and will be discussed in Section VI-D.

The last part of this section emphasizes the possibility of design analysis introduced by our methodology (analysis of design parameters, exploration / synthesis relation).

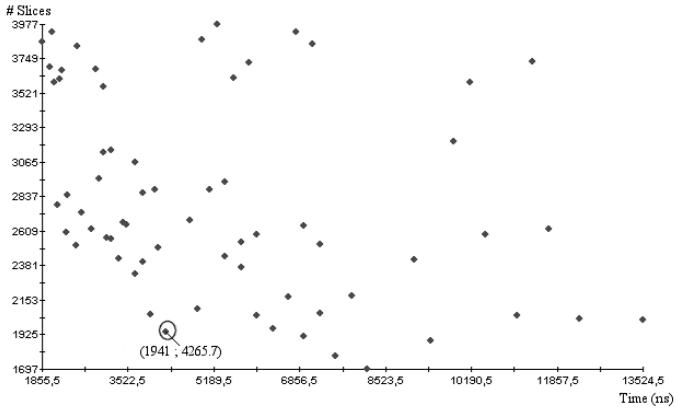


Fig. 8. Horizontal DWT exploration results (Virtex) - *slices* vs. time

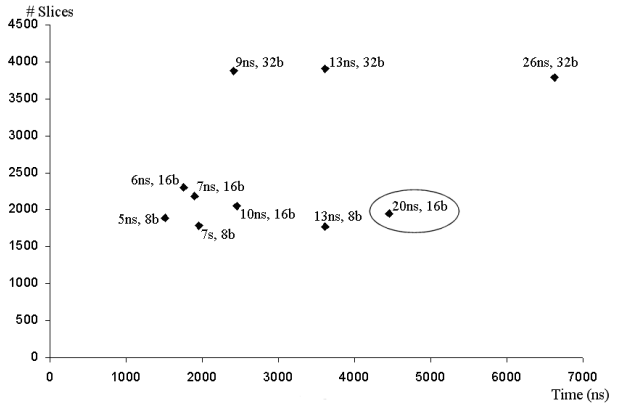


Fig. 9. Bitwidth and clock period exploration for a Virtex implementation

C. Exploration Approach

The application considered is a 1D Discrete Wavelet Transform, Fig. 8 presents the estimation trade-offs for a Virtex V400EPQ240-7 device (Xilinx). For this example, the tool evaluates a total of 342 architectures, each one corresponding to a particular parallelism solution.

If we consider the solution highlighted (respectively 1941 slices / $4265.7ns$) that corresponds to the most interesting area / delay trade-off, the designer can refine the exploration around that solution. In this example, the exploration of several clock values and data bitwidths have been performed, results have been reported in Fig. 9 (labels correspond to clock period - data bitwidth values).

When a solution has been pointed out (for example clock = 20ns, bitwidth = 16), analyzing the details of the RTL architecture results provides useful information. In the example of Table VIII, we are aware of that the solution is composed of 4 multipliers and 8 adders for an execution of 223 cycles. Mapped onto the FPGA, it corresponds to an occupation of 1941 ($/4000$) slices, 12 ($/40$) BRAMs and 256 ($/4960$) tristate buffers for a $4.3\mu s$ physical execution time. Due to the hierarchical approach of combinations, partial results are available at each level of the graph hierarchy. To refer our example, the For12_body sub-function is composed of 1 multiplier and 2 adders for the datapath and the RAM memory bandwidth is 1 write and 3 reads. Such partial results characterize each architectural solution and provide key design information. All this information can also be used to guide a HLS tool in the design of the solution.

D. Result Discussion

The experiments performed pointed out some benefits / limitations of the exploration methodology. Hereafter, we discuss the relevance of the results and propose some possible enhancements. In the next section, we will focus on the contribution over related work in the perspective of future complex / heterogeneous system designs.

The structural exploration represents the core of the methodology which efficiency strongly depends on the feasibility and the relevance of the different parallelism solutions. As a starting point, we defined a first approach based on DFG scheduling and analytical CDFG combinations using simple execution models. The main reason of doing this was to reach a complexity level that could permit a large coverage of the design space from early specifications. This has been shown possible as regard to the accuracy and processing times reported (Table V). Moreover, some enhancements are possible, of course at the expense of processing complexity, these are exposed below:

- Parallelism exploration: the weak point of out approach is the simplicity of the combination heuristics and the underlying execution model as we do not consider loop tiling and only consider a simplified model for loop unrolling and folding. On the other hand, an optimal schedule of the whole graph is something that would greatly impact on the estimation complexity. That's the reason we decided to schedule only the DFGs of the graph as a trade-off. A solution to enhance the quality of the solutions would be to apply a finer schedule of the critical processings, especially in the case of iterative structures with data dependencies. We believe this is something that can be done without too much penalty on the exploration times, regarding the current complexity of the exploration methodology. We are investigating new loop scheduling heuristics able to consider data dependencies.
- Simplicity of the memory model: in the current version, memory estimation is based on a basic load / store model and characterized through the estimation of memory size and bandwidth requirements. We do not consider scalar replacement, data layout, data reusing, memory sharing and memory pipelining. This is justified in part by the storage resources within an FPGA that is composed of several distinct memories. For complex memory structures and execution model (using caches, pipelined execution modes, resources sharing), suited models and heuristics must be defined. There exist relevant works in this field that could be efficiently used for this purpose [32][4][16].
- Another way to enhance the quality of the RTL solutions is to allow the characterization of IP cores in the FPGA characterization (TAPM) file. For example, a Discrete Cosine Transform implementation can be described in the TAPM file (in terms of FPGA resources and delay)

TABLE VIII
SOLUTION DETAILS FOR THE SELECTED RTL ARCHITECTURE LEADING TO THE FOLLOWING PERFORMANCE: 1941 SLICES, EXECUTION TIME $4265.7ns$

Graph	Cycles	States	Mul16	Add16	Reg16	RAM (wr)	RAM (rd)	ROM
For12_body	5	5	1	2		1	3	1
H1stLftStep	32	32	4	8		4	12	4
For22_body	5	5	1	2		1	3	1
H1stDLftStep	32	32	4	8		4	12	4
For32_body	5	5	1	2		1	3	1
H2ndLftStep	32	32	4	8		4	12	4
For42_body	5	5	1	2		1	3	1
H2ndDLftStep	32	32	4	8		4	12	4
For52_body	3	3	2			2	2	2
Hscaling	66	66	4			4	4	4
For62_body	2	2				2	2	
Hrearrange	33	33				8	8	
HDWT	223	223	4	8	28	8	12	4
	$T_{ex} = 4.3\mu s$		$slice = 1941$		$BRAM = 12$		$3state = 256$	

and used for estimation provided a specific DCT node is defined in the H/CDFG. This possibility is enabled by the H/CDFG representation and could greatly enhance the exploration reliability. As an extension, any processing sequence or common functionality can be associated with an actual implementation provided information on area and delay is available in the technology file (in the manner of [13]). This could greatly help the problem of software compilation including pre-characterized IPs or extraction of processing patterns associated with specific VLIW implementations.

Concerning the physical estimations, some variations have been noticed when comparing physical synthesis results with area and performance estimates (Table V). This is mainly due to some unconsidered low level optimizations.

- Control unit / address generation: variations may occur in case of high data processing / address generation. A solution is to consider a separate address generation unit from the control unit model.
- Logic optimizations: improvements are possible by considering some low level optimizations like operator area reduction when one operand remains constant, or a shift operation instead of a multiplication by a power of two.
- Clock period: clock value is defined before actually placing and routing the entire design and relies only on the characteristics described in the TAPM file. Thus, the actual critical path may be larger than the estimated one. This overhead might increase with the FPGA use rate since routing becomes more complex.

The impact of these low level effects is not critical since the average estimation accuracy is already around 15%. We believe the structural estimation step is a more important concern where additional processing complexity should focus on. Other scheduling strategies can be defined to better cope with the processing requirements of an application domain, a HLS tool procedure or an architectural implementation style. The architecture and scheduling models may have to be adapted in this case.

E. Comparison with existing approaches

Compared to the works presented in Table I, our approach can be summarized as follows:

- The estimator deals with a *behavioral* C description which is then parsed into a H/CDFG representation. So a complete characterization of the application is achieved in terms of Processing, Control and Memory unlike most approaches.
- Although it may be considered simplistic, the underlying implementation model is realistic since datapath, control and memory units are considered. It has been defined to cope with FPGA specificities and used in a way to enhance the complexity / accuracy trade-off in the exploration process. Other models are currently under study to refine the exploration of loop nests.
- The estimator provides area and delay values. No optimizations at the logic level are currently considered.
- The exploration of the design space is automatic. This means that several RTL solutions are defined for a given specification, without making several iterations of the exploration process.
- A variety of design parameters can be explored: implementation device, resource selection, clock period, data bitwidth, parallelism, ...
- The choice of an FPGA device is large and includes up to date devices. The TAPM characterization file includes logic cells as well as dedicated cells (DSP operators, embedded memories) which are important features in modern reconfigurable devices.
- The complexity of the algorithm is low ($O(n)$) since only a list-scheduling algorithm combined with a simple analytical method are applied. We are exploring finer scheduling heuristics that could be applied to better analyze critical parts of the application (complex loop processing with data dependencies).
- The average accuracy is 10% for delay estimations and 20% for area values.

The application of this work has shown the possibility of using accurate estimations from *behavioral* C specifications

to perform early design space pruning. The use of such low complexity estimations permits to compare very quickly different implementation alternatives and to make reliable choices that are derived directly from the processing characteristics of the specification. The main conditions that have enabled this are the use of (1) a complete representation model, (2) a realistic implementation model and (3) scheduling heuristics combined with analytical approaches. These 3 conditions can be changed / adapted to cope with a specific design flow / application domain.

Concerning this, the definition of new models suited to the design procedure of a specific High Level Synthesis flow (we are currently exploring the use of Celoxica DK suite) would greatly enhance a complete design process by providing the HLS tool with essential information, especially concerning the location of parallelism in the behavioral specification. Moreover, an extension to ASIC design is possible through the definition of an appropriate TAPM file. This could lead to a complete framework for fast hardware prototyping.

VII. Conclusion and Perspectives

The starting objective of this work was to define an accurate estimation methodology from early specifications to move more efficiently through a design space pruning process. The accuracy available in the current (first) version of the exploration tool is high considering the abstraction level at the input. This results from the use of a complete specification model (H/CDFG) and realistic implementation models for the datapath, control and memory units. The interest of a scheduling approach is to bring both confidence concerning the feasibility of the solutions (compared to a true estimation approach) and reliable information to guide their design. The simplification of the scheduling process is important to quickly explore a wide range of parallelism solutions. The drawback may be the relevance of the solutions defined in some cases, which is unavoidable given the abstraction level of the specification. But the complexity / design space coverage trade-off provided lets us expect easy adaptation to other implementation and execution models in a way to provide better quality to the final design.

The benefits of this methodology on a design process are multiple: it allows significantly reducing the design cycle (especially if used in complementarity with an HLS tool), to be less dependent from designer experience and synthesis tools (thus from technology evolution perspectives), and to converge faster toward an efficient and constraint compliant solution. Better application / architecture / device matching is reached thanks to the exploration coverage in terms of parallelism, devices, clock period, and functional unit allocation.

This approach has been integrated in a CAD framework for the codesign of heterogeneous SoCs called *Design Trotter*.

References

[1] R. Hartenstein, "Are we ready for the breakthrough ?" in *Proc. of International Parallel and Distributed Processing Symposium (IPDPS'03)*, Nice, France, Apr. 2003.

[2] N. Tredennick and B. Shimamoto, "The rise of reconfigurable systems," in *Proc. of Engineering of Reconfigurable Systems and Application Conference (ERSA'03)*, Las Vegas, Nevada, USA, June 2003.

[3] D. Kulkarni, W. A. Najjar, R. Rinker, and F. J. Kurdahi, "Fast area estimation to support compiler optimizations in fpga-based reconfigurable systems," in *Proc. of Symposium on Field-Programmable Custom Computing Machines (FCCM'02)*, Napa Valey, CA, USA, Apr. 2002.

[4] B. So, P. C. Diniz, and M. W. Hall, "Using estimates from behavioral synthesis tools in compiler-directed design space exploration," in *Proc. of Design Automation Conference (DAC'03)*, Anaheim, CA, USA, June 2003.

[5] (2004, May) Matlab-based ip for dsp design of fpgas and asics. [Online]. Available: http://www.accelchip.com/

[6] (2004, Mar.) The application of retiming to the synthesis of c based languages using the celoxica dk design suite. [Online]. Available: http://www.celoxica.com/

[7] A. Nayak, M. Haldar, A. Choudhary, and P. Banerjee, "Accurate area and delay estimators for fpgas," in *Proc. of International Conference on Design Automation and Test in Europe (DATE'02)*, Paris, France, Mar. 2002.

[8] R. Enzler, T. Jeger, D. Cottet, and G. Troster, "High-level area and performance estimation of hardware building blocks on fpgas," in *Proc. of International Conference on Field-Programmable Logic and Applications (FPL'00)*, ser. Lecture Notes in Computer Science, vol. 1896. Springer, 2000, pp. 525–534.

[9] M. Xu and F. J. Kurdahi, "Area and timing estimation for lookup table based fpgas," in *Proc. of the European Design and Test Conference (ED&TC'96)*, Mar. 1996.

[10] ——, "Layout driven rtl binding techniques for high-level synthesis using accurate estimators," *ACM Transactions on Design Automation of Electronic Systems*, vol. 2, no. 4, pp. 313–343, Oct. 1997.

[11] P. Bjureus, M. Millberg, and A. Jantsch, "Fpga resource and timing estimation from matlab execution traces," in *Proc. of International Symposium on Hardware/Software Codesign (CODES'02)*, Estes Parks, Colorado, USA, May 2002.

[12] P. Diniz, M. Hall, J. Park, B. So, and H. Ziegler, "Bridging the gap between compilation and synthesis in the defacto system," in *Proc. of Languages and Compilers for Parallel Computing Workshop (LCPC'01)*, Cumberland Falls, KY, USA, Aug. 2001.

[13] W. Miller and K. Owyang, "Designing a high performance fpga – using the prep benchmarks," in *Proc. of WESCON'93 Conference*, 1993, pp. 234–239.

[14] P. Banerjee, N. Shenoy, A. Choudhary, S. Hauck, A. Nayak, and S. Periyacheri, "A matlab compiler for distributed, heterogeneous, reconfigurable computing systems," in *Proc. of International Symposium on Field-Programmable Custom Computing Machines (FCCM'00)*, Napa Valey, CA, USA, Apr. 2000.

[15] (2001, Aug.) Altera apex 20k programmable logic device family. [Online]. Available: http://www.altera.com/

[16] K. R. S. Shayee, J. Park, and P. C. Diniz, "Performance and area modeling of complete fpga designs in the presence of loop transformations," in *Proc. of International Conference on Field-Programmable Logic and Applications (FPL'03)*, ser. Lecture Notes in Computer Science, vol. 2778. Springer, 2003, pp. 313–323.

[17] S. Choi, J. W. Jang, S. Mohanty, and V. K. Prasanna, "Domain-specific modelling for rapid system-wide energy estimation of reconfigurable architectures," in *Proc. of Engineering of Reconfigurable Systems and Algorithms (ERSA'02)*, June 2002.

[18] S. Bilavarn, "Exploration architecturale au niveau comportemental - application aux fpgas," Ph.D. dissertation, Univ. of South Britanny, Lorient, Feb. 2002.

[19] S. Bilavarn, G. Gogniat, and J. Philippe, "Fast prototyping of reconfigurable architectures: An estimation and exploration methodology from system-level specifications," in *Proc. of International Symposium on Field-Programmable Gate Arrays (FPGA'03)*, Monterey, CA, USA, Feb. 2003.

[20] J. P. Diguet, G. Gogniat, P. Danielo, M. Auguin, and J. L. Philippe, "The spf model," in *Proc. of International Forum on Design Languages (FDL'00)*, Tubingen, Germany, Sept. 2000.

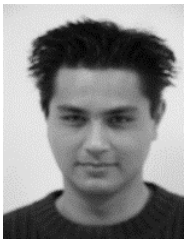
[21] Y. Moullec, J. P. Diguet, and J. L. Philippe, "Design-trotter: a multimedia embedded systems design space exploration tool," in *Proc. of International Workshop on Multimedia Signal Processing(MMSP'02)*, St. Thomas, US Virgin Islands, Dec. 2002.

[22] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, "Introduction to algorithms," in *Second Edition, MIT Press and McGraw-Hill*, Sept. 2001.

[23] G. Grun, N. Dutt, and F. Balasa, "System level memory size estimation," University of California, Tech. Rep., 1997.

[24] D. Gajski, N. Dutt, A. Wu, and S. Lin, *High-Level Synthesis: Introduction to Chip and System Design*. Kluwer Academic Publishers, 1992.

[25] S. A. Blythe and R. A. Walker, "Efficient optimal design space characterization methodologies," *ACM Transactions on Design Automation of Electronic Systems*, vol. 5, no. 3, July 2000.
[26] (2001, July) Xilinx virtex-ii 1.5 field programmable gate arrays. [Online]. Available: http://www.xilinx.com/
[27] S. Narayan and D. D. Gajski, "Area and performance estimation from system-level specifications," University of California, Tech. Rep., 1992.
[28] *7 kHz audio-coding within 64 kbit/s*, ITU -T Recommendation G722, 1988.
[29] I. Daubechies, "The wavelett transform, time-frequency localization and signal analysis," *IEEE Transactions on Information Theory*, vol. 36, no. 5, Sept. 1990.
[30] E. Martin, O. Sentieys, H. Dubois, and J. L. Philippe, "Gaut, an architecture synthesis tool for dedicated signal processors," in *Proc. of International European Design Automation Conference (Euro-DAC'93)*, 1993, pp. 14–19.
[31] Creating a core generator module, ise 6 in-depth tutorial. [Online]. Available: http://www.xilinx.com/
[32] F. Catthoor, S. Wuytack, E. D. Greef, F. Balasa, L. Nachtergaele, and A. Vandecappelle, *Custom Memory Management Methodology*. Kluwer Academic Publishers, 1998.



S. Bilavarn received the B.S. and M.S. degrees from the University of Rennes in 1998 and the PhD degree in Electrical Engineering from the University of South Brittany France in 2002. Then he joined the Signal Processing Institute of the Swiss Federal Institute of Technology (EPFL) to conduct research investigations with the System Technology Labs of Intel Corporation Santa Clara. Sebastien's research interests are in the fields of integrated circuits design, reconfigurable computing, programmable processors and compression algorithms.

G. Gogniat is an Associate Professor of Electrical and Computer engineering within the University of South Brittany, Lorient France where he has been since 1998. He has a BSEE (94) from the FIUPSO Orsay France and a MSEE (95) from the University of Paris Sud Orsay and a PhD (97) in ECE from the University of Nice-Sophia Antipolis France. In 2004, he spent one year as an invited researcher within the University of Massachusetts, Amherst, USA where he worked on embedded system security using reconfigurable technologies. His work focuses on managing the development of the EDA framework "Design Trotter" for design space exploration and combining the design space exploration with technology mapping over reconfigurable architectures. He also conducts research in high level methodologies and tools for FPGA utilization, performance estimation and FPGA security.

J. L. Philippe was born in 1958. Jean- Luc Philippe received the Ph.D. in signal processing from the University of Rennes in 1984. In 1992 he worked as an Associate Professor at ENSSAT, University of Rennes, conducting a research group in CAD for VLSI design. Since 1996 he is Professor at the UBS (University of South Brittany). His research interests include signal and image processing systems, and codesign.

L. Bossuet was born in France in 1975. He received the B.S. degree in electrical engineering from the ENSEA Cergy-Pontoise France, the M.S. degree in electrical engineering from INSA-University of Rennes France and a PhD in electrical engineering and computer sciences from the University of South Brittany France. Since 2005, he is an Associate Professor of electrical and computer science at the ENSEIRB, University of Bordeaux France. His main research activities at the IXL laboratory focus on digital systems design and hardware security for embedded systems. His interests also include reconfigurable computing, high level methodologies and tools for SoC, and FPGA performances estimation. He also works on A/D converter characterization.

4. Article concernant les approches de conception dirigées par les modèles (MDD)

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, pp. 111 – 130

Chapitre X

Schedulability analysis and MDD

X.1. Introduction

Complex system-on-a-chip (SOC) challenge is now achievable since both required hardware resources and integration technologies are available. The telecom domain is an interesting example where the SOC paradigm already enables the design of multi-standard chips (e.g. GSM, IEEE 802.11, IS-95). Such an evolution promotes the software radio concept for the management of multiple standards [MIT 95][SDR 06]. However, the design of these systems based on heterogeneous platforms (e.g. DSP, FPGA, GPP) and intensive-computation software applications (e.g. encryption, scrambling algorithm, service management) cannot anymore be addressed with traditional CAD tools. Actually higher levels of abstraction are required to cope with the design complexity and to provide the designers with an early feedback. Such co-design tools partly exist and are based on scalable hardware and software IPs reuse. Some of these tools can already meet the design constraints, like CoWare, that uses SystemC/C++ hardware language specifications, or Co-fluent studio, that is based on the MCSE methodology [BOL 97][CAL 90]. In this chapter we will stress how our solution offers a simple and more unified way to fill the gap between the specification and the prototyping phases, through an UMTS transceiver case study.

Major projects related to software radio are described in UML which enables modeling systems through a graphical approach. Furthermore UML continuously evolves to consider new specific characteristics from different activity domains thanks to the development of new profiles. A profile specializes the UML language for a work context, which offers scalability. It specifies all characteristics (e.g.

elements for real-time application) and relations between the UML elements. It allows model-based a priori verifications. A designer relies on the profile to analyze, generate code and specify various application and architecture constraints. Moreover, dependencies, inheritance, or groupings between profiles can be performed to promote the reuse of domain specific needs. Regarding the software radio application, three profiles are of interest: UML profile for software radio [SRP 05], UML profile for schedulability performance and time [SPP 03] and UML profile for QoS and fault tolerance [QFP 05]. Each profile provides some specific characteristics that are useful to perform the evaluation of the system performances. Dealing with these profiles, a system can theoretically be accurately specified by integrating various constraint types (e.g. power consumption, bounded execution time).

But the standardization of these profiles is not always completed and existing profiles do not cover all the parameters required for system prototyping. Our work proposes to improve these different profiles through the development of a new and specific one. Its purpose is to stress standard concepts required for prototyping and to add hardware attributes that are not currently taken into account. Furthermore the goal of our project (A3S project) is not limited to the definition of the A3S profile but also targets its implementation within a rapid-prototyping tool to evaluate the feasibility of complex applications over heterogeneous platforms (with DSP, FPGA components). Specification of dynamic reconfiguration is also investigated since this feature will be mandatory especially for software radio applications.

The remainder of this chapter is the following. Section 2 presents various high level specifications for system prototyping. Section 3 provides a global approach of system modeling as promoted within our project. Section 4 details the UML modeling by giving the set of parameters required to compute verifications and performance evaluation. Section 5 details the scheduling analysis techniques used to realize the performance evaluation. Section 6 gives an example of an UMTS application modeling. Section 7 concludes the chapter.

X.2. Related Work

Many tools aim at modeling systems, performing verifications, simulations, validations, and synthesis. Different modeling styles with different granularities are considered, different input specification languages as C, SystemC, VHDL, are also used to validate, verify, simulate or emulate a system [SHU 03][EDW 97]. First co-design tools, like VULCAN were using simple and limited hardware architecture models, others like COSYMA were based on dedicated hardware co-processors to speed up software execution [GUP 93][ERN 93]. COWARE and PTOLEMY

consider heterogeneous specifications to respectively design specific applications (embedded telecommunication) and co-simulate heterogeneous HW/SW systems [DAV 01].

However these approaches are limited as they require the use of different tools that must be kept updated. Actually the goal is to perform both modeling and design specification of hardware platforms and software applications within a single tool and through a common language to be less dependent of multiple software update [ARA 99]. The SoC Environment (SCE) developed within the University of California, Irvine provides such an approach as the design specification in each stage of the design flow is defined through a SpecC code [GAJ 04]. However, the use of a generic language, common to different domains, that is enough flexible to model all co-design aspects (e.g. architectural and application specifications, component properties, constraints specification) would be interesting. To target such a philosophy, the most recent rapid prototyping tools integrate methodology of hardware-software co-design into the concept of MDD (Model Driven Development) through UML. MOCCA and GASPARD v02 in the DART project describe their models in UML [FRO 04][DAR 03]. The MOCCA model compiler for reconfigurable architecture requires an action language to define low-level behavior. The DART project focuses on limited application domain (intensive signal processing application) and performs some transformations from the UML model to specific ISP UML. Indeed, UML provides elements to real time specification needs (e.g. parallelism, behavior, concurrency, communication modeling). Arguing that UML is incomplete [LAV 02], the SysML (System Modeling Language) project tries to add extensions of UML to integrate hardware aspects in response to OMG request for proposal [SYS 03].

We propose to use a unique language from specification to validation. Our approach relies on our UML A3S profile that inherits from other standardized profiles and completes them. This profile improves and offers more hardware specification possibilities that are essential for software radio or other electronic systems in order to specify hardware and software architecture systems. In addition our high abstraction level specification alleviates the modeling and the validation of applications that belong to other specific application domains. Moreover, as we consider applications as a set of IPs, components are only characterized by non-functional parameters instead of source codes (which depend on their implementation and need different tools).

X.3. Global Approach

A3S approach proposes a UML software framework where the designer can rapidly and easily prototype his system and check if constraints are met in terms of timing, memory, area, and power consumption. The main steps of our design flow for virtual prototyping are depicted in Fig. X.1.

X.3.1. Application specification (1st step)

With the MDA approach, software applications and hardware architectures can be specified independently, so 1st step and 2nd step (see Fig. X.1) can be exchanged. To manage complexity, an application is split into several functions that are represented by independent generic software (SW) components. It corresponds to a PIM (Platform Independent Model) since each function can be potentially mapped onto any hardware component. These SW components have specific non-functional parameters that correspond to specification constraints coming from the application or from designer requests. One example of these parameters is the periodicity of the SW component which is independent from any implementation. More information about these parameters is detailed in Section 4. At this stage of the design flow SW components can represent any function.

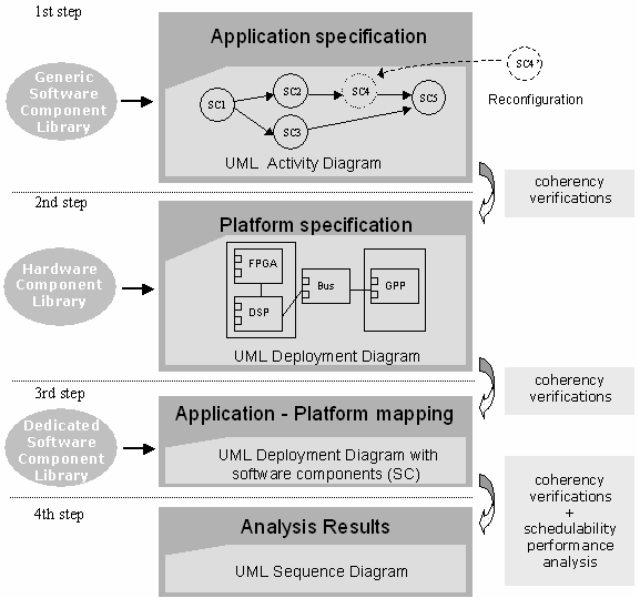


Figure X.1. *A3S design flow*

The application is modeled through a functional scheme based on the UML activity diagram which is composed of a set of action states (SW components) and transitions. Transitions correspond to dependency relations between functions and have specific parameters related to the exchanged data (e.g number, size). For each component, the designer specifies the corresponding parameters values.

An activity diagram example for an UMTS-FDD receiver is given Fig. X.2. This diagram also addresses the links between the different SW components to specify the system radio functionality. The black dot represents the input of the application which takes place at the propagation channel side. Each arrow corresponds to an edge (transition) and represents a data-flow dependency. The UMTS-FDD receiver is mainly a data-flow application with periodic and iterative functions (FrameProcessing, SlotProcessing, RadioProcessing, TransportBloc). The black dot in the circle is the output of the application; it corresponds to the exchanged data between the physical layer and the higher layers of the OSI model.

Through this model the designer can easily replace, add, move/remove a SW component, or modify some parameters to enhance the algorithm and thus test various configurations. By this way, he can analyze the impact of different reconfigurations, which is of major importance in a software radio context. Once the application model is completed, some coherency constraints verifications are performed. Among them, the tool verifies that all connections between SW components have been correctly done, through compatible data format and that all required parameters have been settled. These verifications have been implemented within the Objecteering case tool [OBJ 06].

X.3.2. Platform Specification (2nd step)

This step deals with the platform specification. Each hardware component is described in a hardware library (DSP, FPGA, GPP, memory, interconnect and ASIC) corresponding to an UML package. Each component has specific attributes defined through its stereotypes (this point is developed in Section 4). The designer builds his platform by assembling hardware components instantiations (in UML sense) through a UML deployment diagram. Many hardware platforms can be realized, especially heterogeneous platforms. This kind of architecture is essential for telecommunication applications like software radio that need flexibility (offered by FPGA and DSP components for hardware and software reconfiguration) and important computation resources (multi-processor).

X.3.3. Application - Platform Mapping (3rd step)

After the software application and hardware platform modeling steps completed, the designer chooses which dedicated SW component is implemented onto which hardware component. For each SW component, the designer selects the corresponding function in the software component library since a SW component corresponds to a processing element that is not dedicated to a specific target. Thus, the function represents an implementation of the SW component on a processor (e.g. DSP, GPP), a FPGA or an ASIC. The target hardware component selected to implement the SW component is obtained by defining an instance of a hardware component within the hardware platform in the UML deployment diagram.

A broad range of implementation solutions can then be tested for a specific platform (PSM – Platform Specific Model) due to all possible combinations. The example in Fig. X.3 depicts an hardware platform composed of two DSPs (DSP_A, DSP_C) on which different software components are implemented (e.g. scrambling function is implemented on DSP_A). Thus the deployment diagram is refined by a software component instantiation implemented into a hardware component instantiation. This partitioning is performed through links between the software components from the UML activity diagram and the hardware components from the UML deployment diagram. For example, the DSP_A that is connected to DSP_C via FIFO_AC handles four functions (SCR, SUM, SPRdpcch, DPCCHctrl).

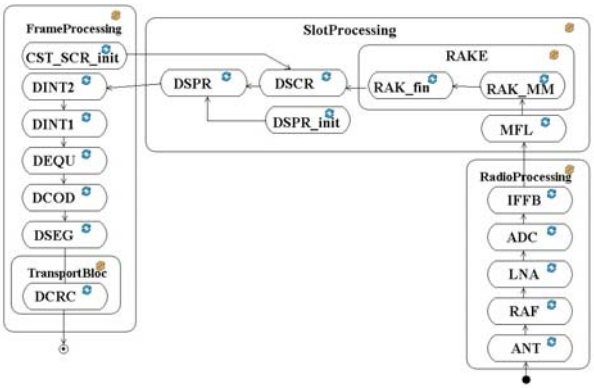


Figure.X.2. *UMTS-FDD Receiver Activity Diagram*

During the application specification step, non-functional verifications are automatically performed by the use of meta-model.

X.3.4. Analysis results (4th step)

Results are provided through a schematic view defined in a UML sequence diagram which is close to a Gantt diagram. The results emphasize the performances achieved by a heterogeneous platform with multi-processor resources to perform the application. For example, execution time, resources use rate, system evolution (scheduling), allocated memory resources are exhibited. Scheduling information is very important as if the system cannot be scheduled or if it does not reach the required timing constraints, the solution is not relevant.

If the solution built does not satisfy the constraints, it's easy to modify the implementation choices just by modifying the links between software and hardware components in the UML activity diagram without modifying the diagram. As several applications and platforms can be specified it enables testing an application on different platforms and with different implementations for a same platform. It also promotes testing different configurations and re-configurations of the system. It is also possible to modify some hardware characteristics by changing hardware component parameters values. Moreover, the A3S CAD tool returns results that help designer to make modifications according to identified critical functions.

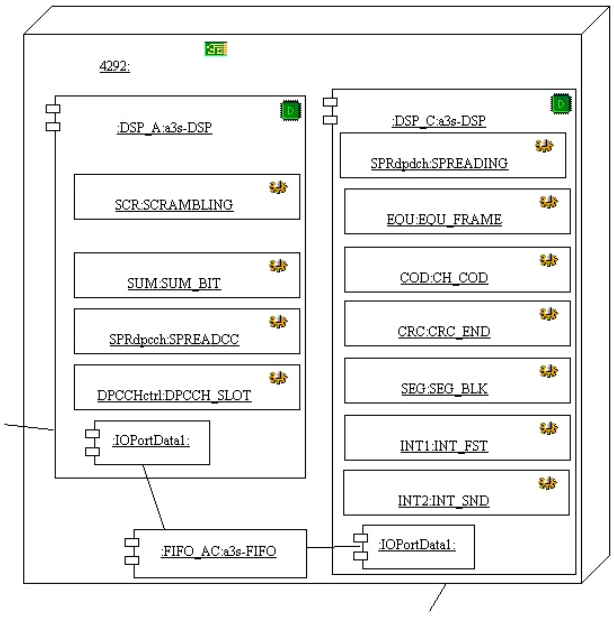


Figure. X.3. *Deployment diagram after mapping*

As the previous steps, coherency verifications are performed to check the solution. After this step, the system is completely specified and a functional analysis can be launched (presented in Section 4).

X.4. UML Modeling

X.4.1. Attributes identification

During the application and platform specification steps, the designer provides the values of the software and the hardware component attributes to perform the coherency verification and analysis of the system. Each component (software and hardware) can be characterized into three parts as described in Fig. X.4. The first part concerns the non-functional characteristics (attributes) of the component.

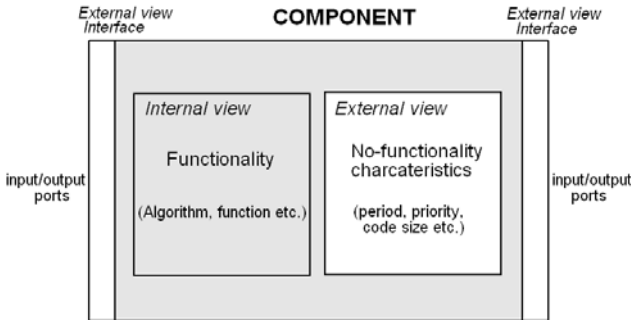


Figure. X.4. *Component Views*

For software components it represents the temporal aspects of the function (e.g. period) and the data characteristics. The second part describes its interface (its I/O port) with significant attributes relative to exchanged signal. The third part is relative to the functionality of the component. For hardware components it corresponds for example to the clock frequency, the type and quantity of internal/external memories. This view mainly corresponds to specification constraints. As our approach relies on IP cores, the internal view of the component is not explicitly represented since we assume that IP cores functional behavior (i.e. C, C++, SystemC, VHDL) is validated through other means that are not in the scope of this chapter. In our case attributes can be provided using the IP characteristics.

UML stereotypes permit to identify and characterize any elements by assigning different parameters called “attribute”. So each element of UML can be specialized by using different stereotypes that are used to define component parameters. Generic SW components which compose the UML activity diagram, HW components, ports

of HW components, dedicated SW component, ports of dedicated SW components have different stereotypes, which give them specific attributes. Basically, generic SW components which are not yet implemented have different attributes (e.g. a function is periodic or not, it has an initialization part or not) which correspond to a dedicated SW component which represents one implemented choice of one generic SW component. Each implementation choice brings some specifics constraints that are highlighted through the non-functional attributes. They deal with function periodicity, execution time, size code of IP cores, priority level if a RTOS is used, and other attributes like data and code localization, and access memory types. HW components have different stereotypes to differentiate HW processing components (e.g. DSP, ASIC, GPP), memory components (e.g. FIFO, RAM, ROM), reconfigurable components (FPGA) with their associated ports, and communication components (e.g. bus, wire). Specific performance parameters are considered according to the hardware components (e.g. frequency, data/program memory size, port type, data width, throughput).

All identified parameters are required to perform an analysis. They are used during the scheduling analysis step, to compute resources use rates, to perform constraints verification and to check the coherency of the system.

X.4.2. Analysis details

Once specification and mapping have been completed and coherency verifications have been performed (i.e. no error about HW/SW connections, all attributes settled), the A3S tool generates a XML file gathering the information about the system. The file contains the diagrams (activity, deployment), the hardware/software components allocated, and the attributes values. More precisely the UML activity diagram that represents the functional application scheme of the system is encompassed in the XML file. Thanks to an XML parser this diagram is converted into a task graph. Thus, the parsing of this file enables building a General Task Graph (GTG) based on the Radha Ratan model since we consider the corresponding method to perform period derivation [DAS 99]. This method computes the period of each task within the GTG even if some tasks are previously unknown. The GTG nodes represent tasks (functions), and the GTG oriented edges are channels from producers (tasks) to consumers (tasks). Each task can be triggered by a data or a control. Each edge contains producer and consumer information corresponding to data/control to be exchanged between functions. For applications that use multi-processor, the choice of a function implementation can lead to additional communication tasks (in case of two tasks communicating and implemented on different hardware devices). The period derivation step is

performed to compute the timing constraints (periods) that have not been settled by the designer during the specification steps. This point is important, since this kind of computation is very error prone and can be efficiently done with our CAD tool.

The GTG obtained from the XML processing is then used with the HW architecture characteristics within our real time analysis tool RTDT [TMA 04]. This tool performs automatically complex scheduling verification and provides performance analysis results which help the designer to drive his choices. Such automatic bridges and tools are essential to improve time to market and the quality of systems design.

X.5. Real time analysis tool (RTDT)

X.5.1. Real time scheduling strategy

X.5.1.1. Task classification

Usually, real-time embedded systems require a simple and safe scheduler which can guarantee that critical aperiodic or periodic tasks meet their deadlines. For these reasons, a static HPF (High Priority First) scheduling policy has been adopted, where the fixed priorities are computed as the inverse of the task period. The worst case response time is computed with an exact analysis [JOS 86].

In a first approach we consider two kinds of tasks. The first category is composed by the periodic tasks that are scheduled by means of hard real time constraints (RTC), and sporadic tasks with hard RTC. Like in [DAV 98], we consider the sporadic tasks as periodic tasks with a period equals to the minimum delay between two subsequent executions; this value is provided by the Radha Ratan tool [DAS 99]. The second category includes the non critical sporadic tasks which are handled by a server task with the lowest priority that can be fixed by the designer. The task priority is computed as the inverse of the task period.

The question of multi-rate dependencies is solved by shifting the release time computation as detailed in [AZZ 02].

X.5.1.2. Response time computation

The exact response time is computed iteratively with the following equation:

$$\forall T_j \in HP(i), \exists R_i \le D_i / R_i = \left(C_i + R_i\right) + \sum_{j \in HP(i)} \left\lceil \frac{R_i}{P_j} \right\rceil \times \left(C_j + C_{sw}\right) \quad [X.1]$$

Where:

– *HP(i)*: is the set of tasks with higher priority comparing to task *i*;
– R_i: is the worst case response time of task *i*;
– D_i: is the execution deadline for task *i*,
– C_i: is the execution time of task *i*;
– B_i: is the longest time that task *i* can be delayed by lower priority tasks,
– P_j: is the period of task *j*,
– C_{sw}: is the context switching; $C_{sw} = \delta_0 + \sum_k \delta(k)$ [X.2]

With:

– δ_0: is the context switching overhead without any coprocessor,
– $\delta(k)$: is the overhead due to the coprocessor *k*.

The context switching overhead is the delay between the preemption of a given task and the activation of another task. The difficulty is that C_{sw} depends not only on the target processor and on the RTOS and its configuration but also on the number of tasks in the system and on the number of coprocessors. The influence of the number of tasks is not insignificant but can be neglected compared with the coprocessor context saving influence. Moreover, without coprocessor, the available overhead metric is usually an average value estimated with different task sets. The influence of the coprocessor is obviously related to the number of data and status registers.

X.5.2. Design space exploration for HW/SW partitioning

X.5.2.1 Cost function

The cost function takes into account the global area of the SOC and its energy consumption. At a high level of abstraction, only relative estimations can be used for SW and HW IPs and the cost function is used to guide the selection of a reduced set of solutions. In order to eliminate solutions, relative costs are used to evaluate the cost value for a given schedulable solution *S*:

$$Cost(S) = \alpha \frac{Area(S) - MinArea}{MinArea} + \beta \frac{Pw(S) - MinPw}{MinPw} \qquad [X.3]$$

with $\alpha + \beta = 1$ and where *MinArea* is the schedulable solution with the minimal area without any power consideration and *MinPw* the schedulable solution with the minimal power without any area consideration. Note that the area cost influences the power consumption through the static power evaluation. So, the α parameter also acts on the power optimization.

X.5.2.2. Area Cost

The area cost includes the data and code memory size for software implementations, the area of coprocessors that can be shared by various tasks, the area of hardware accelerators and finally the area of memories added for communications.

X.5.2.3. Power Cost

The model for power evaluation is much more complex. Firstly, the dynamic power consumption depends on the SOC activity, which is strongly related to the task scheduling and switching. Secondly, the evolution of VLSI technology shows that static power consumption [BUT 00], especially in FPGAs, can no more be neglected. Finally, in mobile embedded systems the important metric is the system life span. It means that the energy used must be optimized. However, in our context of periodic tasks the energy optimization is equivalent to the average power minimization over the hyper period. Our power model for an implementation *S* is given by :

$$Pw(S) = Pw_d + Pw_s \qquad [X.4]$$

where: Pw_d is the average dynamic power dissipated during a hyper period T_G and Pw_s is the average static power.

X.5.2.4. Dynamic power/energy metric

Let Pw_d, the average dynamic power dissipated during a hyper period T_G.

$$Pw_d = \frac{E_d}{T_G} \qquad [X.5]$$

$E_d = E_d(sw) + E_d(hw)$ //E_d: energy consumed during a hyper period T_G [X.6]

$$E_d(sw) = E_d(idle) + E_d(switch) + E_d(exe) \quad [X.7]$$

$$E_d(exe) = T_G \sum_{i \in sw} P_{wd}(i) \frac{C_i}{P_i} \quad [X.8]$$

//$P_{wd}(i)$: average power for task i

$$E_d(switch) = P_{wd}(switch) T_G \sum_{i \in sw} \frac{C_{sw}}{P_i} \quad [X.9]$$

//$P_{wd}(switch)$: average task switching power

$$E_d(idle) = P_{wd}(idle) T_G \left(1 - \sum_{i \in sw} \frac{C_i + C_{sw}}{P_i} \right) \quad [X.10]$$

//$P_{wd}(idle)$: average processor idle power

$$E_d(hw) = T_G \sum_{i \in hw} P_{wd}(i) \frac{C_i}{P_i} \quad [X.11]$$

For flexibility and genericity concerns, the task average dynamic power values $P_{wd}(i)$ are normalized versus the supply voltage and clock frequency and the average task static power is expressed by area unit (W/gate or W/μm^2 as indicated in [ITR 03]).

X.5.2.5. Static power/energy metric

The available static power, usually given by means of mW/area, depends mainly of the leakage power, the supply voltage, the transistor count and a technology-dependent parameter:

$$Pw_s = f(N_{tr} K_{design} I_{leakage} V_{dd}) \quad [X.12]$$

Our model uses $Pw_s(sw)$ and $Pw_s(hw)$ for software and hardware parts respectively. A dynamic strategy can be adopted for static power management if hardware accelerator power supply can be switched off when unused. In such a case the average static power dissipation is given by:

$$Pw_s = Pw_{offsw} \times Area(sw) + Pw_{offhw} \times \sum_{i \in hw} Area(i) \frac{C_i}{P_i} \quad [X.13]$$

Without HW dynamic power supply management, we obtain:

$$Pw_s = Pw_{offsw} \times Area(sw) + Pw_{offhw} \times \sum_{i \in hw} Area(i) \quad [X.14]$$

X.5.2.6. Partitioning Algorithm

X.5.2.6.1. Solution evaluation

The main difficulties during the partitioning / RT scheduling algorithm are firstly the size of the design space, especially, since multiple granularity solutions can be considered for each hardware task implementation, and secondly the iterative scheduling of task worst case response time.

```
Boolean Schedulable (S){
        U = ProcUseRate(S) // Processor use rate
        IF (U+rs > 1) // rs: server task CPU ratio
                RETURN false;
        ELSE IF  U + rs ≤ n*(2^(1/n) − 1)  RETURN true;
        ELSE {
                FOR ALL Ti by Increasing Priority Order
                        Ri = ExactResponseTimeAnalysis(Ti);
                        IF Ri > Pi RETURN false;
                RETURN true;
        }
}
```

Figure. X.5. *Schedulability test*

A solution is valid if firstly all tasks meet their deadlines and secondly if the current cost belongs to the N first best costs. Contrary to the response time computation, the cost is not iterative and must be evaluated first. Thus the schedulability is computed in a three-step approach (see Fig. X.5) in order to restrict the use of iterative response time computations. The algorithm first tests if the processor rate is lower than 1. As a second test, the fast rate monotonic analysis

(RMA) is performed, it gives a sufficient but not necessary condition for schedulability. Finally if the first tests are valid an exact analysis is performed. Note that the designer can specify the CPU ratio *rs* to be guarantied for the server task.

X.5.2.6.2. Design space exploration

Two methods are currently available, the first one is exact and based on the Branch & Bound (B&B) algorithm, the second one is heuristic and uses a simulated Annealing approach (SA). The B&B starts with a left edge branch representing a complete software solution and progresses towards a complete hardware. Schedulability tests solutions with the finest granularity degree. the Tasks are ranked in a branch according to the priority order. On a given branch, for each task added, the cost is first evaluated; if the cost is lower than the best current solution then the task schedulability is computed according to the method described in Fig. X.5. When the cost is larger than the best value or when the solution is not schedulable then a new task implementation is evaluated. If no more implementation is available, another implementation is considered for the previous task in the current branch and so on. The main difficulty occurs when a hardware solution with a fine granularity implies the insertion of a communication task with a shorter period than its predecessor in the branch. In such a case, the schedulability of previous tasks with a lower priority must be computed again. The B&B is efficient even for large graphs (100 tasks) when there are few schedulable solutions, but its computation is prohibitive when numerous solutions are proposed for each task. When the response time computation dramatically slows down the design space exploration, the SA heuristic can efficiently relay the B&B.

X.6. UMTS FDD Case Study

The A3S profile has been created to specify the software defined radio physical layer in the UML 2.0 meta-model. An UMTS FDD channel in uplink mode has been chosen as a first reference to determine which software and hardware components could be included in the software and hardware components library from the A3S UML profile. This application has also been tested to validate the A3S tool. The UMTS transmitter and receiver applications have been modeled through two different activity diagrams (UMTS receiver modeled in Fig. X.2). The chosen hardware platform corresponds to a standard board composed of multiple DSPs connected to FPGAs via FIFO and SDRAM memories. The deployment diagram represents this Pentek board (4292) described thanks to the hardware components from the A3S hardware components library (an overview is given on Fig. X.3). For this case study the partitioning has been determined manually by the designer. After having specified the hardware components attributes using the Pentek board

components characteristics, and the software components attributes using software IP core characteristics, the validation step and the schedulability performance analysis are performed.

Non-functional attributes are first checked to verify the system coherency. Then the A3S tool generates the GTG file (.gtg) and provides the HW architecture characteristics to perform the schedulability and the power consumption analysis. Our real time analysis tool has been first designed for mono-processor architectures with hardware accelerators, it has been modified to support schedulability analysis in the context of multi-DSP/Processor architectures.

To prototype the UMTS FDD transmitter and receiver we have considered a hardware platform composed of four TMS320 C6203 DSP running at 300Mhz. Each DSP is connected to a XILINX Virtex XC2V3000 FPGA running at 100Mhz. Each DSP is also connected to an external shared SDRAM memory. The two UMTS FDD software applications, transmitter (SW 1) and receiver (SW 2) are implemented into the hardware platform described above. SW 1 is composed of 11 functions (pulse shaping, scrambling, coding, spreading, integrating, etc.) and SW 2 is composed of 14 functions (matched_filter, rake, descrambling, despreading, decoding, etc.).

	Data rate			**Data rate**		
	117 kbits/s			**950 kbits/s**		
	DSP_A	DSP_C	Time (ms)	DSP_A	DSP_C	Time (ms)
Transmitter (SW1)						
1st experience (DSPs)	96.6%	3.4%	9.99	96.6%	5.1%	10.33
2nd experience (DSP + FPGA)	11.4%	3.4%	7.96	11.4%	5.1%	8.26
Receiver (SW2)						
1st experience (DSPs)	185.5%	4.6%	19.27	185.5%	5.0%	19.33
2nd experience (DSP + FPGA)	17.1%	4.6%	9.44	17.2%	5.0%	9.49

Table X.1. *Hardware component utilization rate*

Different implementations are considered to verify and validate the efficiency of the A3S tool. The first experience consists in implementing all the functions for SW 1 and SW 2 into the DSPs (software solution), and then in modifying the data rate frequency to see the limits of such a solution. The second experience consists in partitioning the functions implementation between DSPs (DSP_A, DSP_C) and their respective associated FPGAs (FPGA_A, FPGA_C). The critical functions within each application are implemented into the FPGAs and the remainder into the DSPs. For both experiences, the two different data rates (117 kbits/s and 950 kbits/s) are tested. The UMTS design under consideration is not a complete fully realistic UMTS system but comprise enough processing element to evaluate the tool.

For each experience, a possible scheduling was determined and the corresponding hardware components utilization rates were computed. The results for one radio frame are given in Table X.1. An overall 100% means that all the processing power of all HW devices is necessary to run the application in real time. Less than 100% means that real-time is also reached. When the workload is more than 100% it means that a single HW device (DSP or FPGA) is not enough to run the application.

In the UMTS standard, a radio frame must be computed every 10 ms. In this first experience, we only consider the execution time issue. It shows that software-only solution is adequate for SW 1 as the rate is correct (<100%) for the two configurations (117kbits/950kbits). Actually this solution is not correct for the 950kbits configuration since the timing constraint is not respected as the execution time exceeds 10 ms (10.33>10). In the case of SW 2, the software-only solution cannot be realized because of the DSP overload (185%). Thus to respect both the DSP load and the timing constraint we have to define a new implementation.

The result analysis helps to identify function and/or data exchanges that affect the global system performance. Changing the implementation is straightforward with the A3S tool since it just requires to modify some links (corresponding to the critical functions) and not to rebuild the whole system. Thus only two critical functions (PSH for SW1, MFL for SW2) that were previously implemented onto the DSPs are implemented onto the FPGAs (hardware solution corresponding to the 2nd experience). The remainder functions are still implemented onto the same DSPs. The new results show that for each case (SW 1, SW 2), the DSP load was reduced (e.g. from 96% to 11% for SW 1 and from 185% to 17% for SW 2). This implementation also reduces the execution time, and the timing constraint (<10ms) is respected in each case. Thanks to the tool, the designer performs a fast analysis

and is able to compare the most appropriate implementations satisfying the application and architecture constraints.

X.7. Conclusion

In this chapter we have presented the UML compliant rapid prototyping A3S tool based on a UML A3S profile. The UML A3S framework provides designers with a unified, fast and easy method to specify software applications and hardware architectures. Such an approach significantly decreases the prototyping time and enhances system reuse, which is a major metric for software radio applications. It also provides to the designer an exploration tool for rapidly testing various HW/SW mappings and performing the corresponding schedulability analysis. Furthermore, the A3S project proposes more than a design framework. It also provides a design methodology to validate complex systems with a step by step design approach from PIM to PSM. This CAD tool simplifies designers job by making automatic usually heavy tasks, like coherency verifications, period task and timing computations as well as scheduling analysis and verification.

X.8. Acknowledgements

This research was sponsored by French National Research and Innovation Program for Telecommunication within the framework of A3S project. We particularly thank Jean-Etienne Goubard from Thales, Nicolas Bulteau and Philippe Desfray from Softeam, and Mickaël Raulet from Mitsubishi Electric ITE-TCL.

X.9. References

[ARA 99] ARAKI D., ISHII T., GAJSKI D. D., *Rapid prototyping with HW/SW codesign tool*, Proceedings. Engineering of Computer-Based Systems (ECBS), pp. 114-121, 1999.

[AZZ 02] AZZEDINE A., DIGUET J-P., PHILIPPE J-L., *Large exploration for HW/SW partitioning of multirate and aperiodic real-time systems*, in 10th Int. Symp. on HW/SW Codesign, Estes Park, USA, 2002.

[BOL 97] BOLSEN I., DE MAN H., LIN B., VAN ROMPAEY K., VERCAUTEREN S., VERKEST D., *Hardware/software co-design of digital telecommunication systems*, Proceedings of IEEE, vol. 85, no. 3, pp. 391-418, 1997.

[BUT 00] BUTTS J.A., SOHI G., *A static power model for architects*, in: 33rd ACM/IEEE Int. Symp. on Microarchitecture, 2000.

[CAL 90] CALVEZ J-P., *MCSE : Spécification et conception des systèmes : une méthodologie*, Masson, 1990.

[DAR 03] Team DaRT 2003, *Dataparallelism for Real-Time*, Activity Report 2003 http://www.inria.fr/rapportsactivite/RA2003/dart2003/dart_tf.html

[DAS 99] DASDAN A., *Timing Analysis of Embedded Real-Time Systems*, Ph.D. dissertation, University of Illinois, 1999.

[DAV 01] DAVIS J., HYLANDS C., KIENHUIS B., LEE E.A., LIU J., LIU X., MULIADIS L., NEUENDORFFER S., TSAY J., VOGEL B., XIONG Y., *Ptolemy II: Heterogeneous Concurrent Modeling and Design in Java*, Technical Memorandum, UCB/ERL M01/12, EECS, University of California, Berkeley, CA 94720, March, 2001.

[DAV 98] DAVE P., JHA N.K., *Casper: Concurrent hardware-software co-synthesis of hard real-time aperiodic specification of embedded system architectures*, in: Design, Automation & Test in Europe Conf., Paris, France, 1998.

[EDW 97] EDWARDS S. A., LAVAGNO L., LEE E. A., SANGIOVANNI-VINCENTELLI A., *Design of Embedded Systems: Formal Models, Validation and Synthesis*, Proceedings of IEEE, Vol. 85,N°3, pp. 366-390, 1997.

[ERN 93] ERNST R., HENKEL J., BENNER T., *Hardware-Software Cosynthesis for Microcontrollers*, IEEE Journal Design and Test of Computers, pp. 64-75, 1993.

[FRO 04] FROHLICH D., BEIERLEIN T., STEINBACH B., *Object-Oriented Co-Design for Run-Time Reconfigurable Architectures with UML™*, 5th International Conference on Computer Aided Design of Discrete Devices (CAD DD'04), 2004.

[GAJ 04] GAJSKI D. D., *System-Level Design Methodology*, ASP-DAC 2004 Pacifico Yokohama, Yokohama, Japan, January 27, 2004.

[GUP 93] GUPTA R., DE MICHELI G., *Hardware-Software Cosynthesis for Digital Systems*, IEEE Design and Test of Computers, pp. 29-41, 1993.

[ITR 03] S. I. Association, *International technology roadmap for semiconductors*, http://public.itrs.net/Files/2003ITRS/Home2003.htm (2003).

[JOS 86] JOSEPH M., PANDYA P., *Finding response time in a real-time system*, IEEE Design and Test of Computers 29 (5) (1986) 390-395.

[LAV 02] LAVAGNO L., MARTIN G., SELIC B.V., *UML for Real : Design of Embedded Real-Time Systems*, Kluwer Academic Publishers, 2002.

[MIT 95] MITOLA J., *The Software Radio Architecture*, IEEE Communications Magazine, vol.. 33, no. 5, pp. 26-38, 1995.

[OBJ 06] Objecteering Software, http://www.objecteering.com/

[QFP 05] *UML profile for QoS and Fault Tolerance Characteristics and Mechanisms*, May 2005, (http://www.omg.org/docs/ptc/05-05-02.pdf).

[SDR 06] SOFTWARE DEFINED RADIO FORUM, http://www.sdrforum.org

[SHU 03] SHULAS. S. K., TALPIN J-P., EDWARDS S. A., GUPTA R. K., *High Level Modeling and Validation Methodologies for Embedded Systems: Bridging the Productivity Gap*, 16th International Conference on VLSI design, pp. 9-14, 2003.

[SPP 03] *UML profile for Schedulability, Performance and Time Specification*, September 2003, (http://www.omg.org/docs/formal/03-09-01.pdf)

[SRP 05] *UML profile for Software Radio*, May 2005, (http://www.omg.org/docs/dtc/05-09-05.pdf).

[SYS 03] SysML Object Management Group, *UML for System Engineering Request for Proposal*, 2003.

[TMA 04] TMAR H., DIGUET J-P., AZZEDINE A., ABID M., PHILIPPE J-L., *RTDT : a Static QoS Manager, RT Scheduling, HW/SW Partitioning CAD Tool*, ICM, 2004.