



HAL
open science

Synthèse d'architectures pour les applications de traitement du signal et de l'image

Philippe Coussy

► **To cite this version:**

Philippe Coussy. Synthèse d'architectures pour les applications de traitement du signal et de l'image. Electronique. Université de Bretagne Sud, 2011. tel-00663253

HAL Id: tel-00663253

<https://theses.hal.science/tel-00663253>

Submitted on 26 Jan 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Université de Bretagne Sud

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

Synthèse d'architecture pour les applications de traitement du signal et de l'image

Parties 1 et 2

Soutenue le 15 septembre 2011 devant la commission d'examen composée de :

F. Pétrot	Professeur des Universités	INP Grenoble	Rapporteur
M. Renovell	Directeur de recherche	CNRS, LIRMM	Rapporteur
O. Sentieys	Professeur des Universités	Université de Rennes 1	Rapporteur
D. Etiemble	Professeur des Universités	Université de Paris Sud	Examineur
A. Greiner	Professeur des Universités	Université Pierre et Marie Curie	Examineur
L. Lavagno	Professeur des Universités	Ecole Polytechnique de Turin, Italie	Examineur
E. Martin	Professeur des Universités	Université de Bretagne Sud	Examineur
M. Paindavoine	Professeur des Universités	Université de Bourgogne	Examineur

Par

Philippe Coussy

Laboratoire Lab-STICC
Université de Bretagne Sud – CNRS UMR 3192
Centre de Recherche Christiaan Huygens
56321 Lorient Cedex

Avant Propos

Ce manuscrit présente une synthèse de mes travaux de recherche, mes activités d'enseignement et administratives réalisés depuis septembre 2004, date de ma nomination en tant que Maître de Conférences à l'UFR Sciences et Sciences de l'Ingénieur de l'Université de Bretagne-Sud. Mes travaux de recherche ont été effectués au laboratoire LESTER jusqu'en 2008 puis au laboratoire Lab-STICC au sein de l'équipe Méthodes et Outils pour les Circuits et Systèmes (MOCS) jusqu'à ce jour.

Ce manuscrit présente également, mais de façon plus succincte, mes travaux de recherche entre septembre 2000 et août 2004, période durant laquelle j'ai effectué en tant que doctorant moniteur, puis ATER, mes travaux de recherche au laboratoire LESTER et mes enseignements à l'IUP GEII (Génie Electrique et Informatique Industrielle) de l'Université de Bretagne-Sud – Lorient.

Depuis maintenant 11 ans, mes travaux de recherche s'intéressent au domaine des systèmes numériques avec une forte dominante pour les méthodes et en particulier pour la synthèse automatique d'architecture. Les approches de conception modernes dites de niveau système (« ESL Design » pour « Electronic System Level Design») n'ont cessé ces dernières années de prendre de l'ampleur pour apparaître aujourd'hui comme une solution incontournable dans le domaine de la conception des systèmes numériques.

Parallèlement je me suis impliqué dans des travaux pédagogiques en proposant des enseignements qui tentent de répondre aux besoins de forte qualification demandée aujourd'hui par le tissu industriel dans le domaine de l'électronique numérique embarqué. Au-delà de l'aspect purement pédagogique, j'ai participé à l'évolution de l'offre de formation. En particulier, j'ai participé à la mise en œuvre et au pilotage du parcours Electronique et Informatique Industrielle (EII) de la Licence Physique, Chimie et Sciences de l'Ingénieur (PCSI) mais aussi de l'option Systèmes et Informatique embarqués pour les Applications Mobiles (SIAM) du Master Math-STIC de l'Université de Bretagne-Sud.

Ce document est organisé en trois parties :

- 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.

- Partie 2, 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 trois axes de recherche autour desquels s'articulent mes travaux. Chaque axe est ensuite 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.

Table des matières

PARTIE 1 : SYNTHÈSE DES TRAVAUX - 7 -

1. Curriculum Vitae	- 11 -
1.1. Etat civil	- 11 -
1.2. Grades et titres universitaires	- 11 -
1.3. Situations successives.....	- 12 -
1.4. Résumé.....	- 12 -
1.5. Encadrement doctoraux et post-doctoraux	- 13 -
1.6. Publications	- 15 -
2. Résumé des activités de recherche	- 17 -
2.1. Contexte	- 17 -
2.2. Activités de recherche « doctorale »	- 18 -
2.3. Activités de recherche en tant que Maître de Conférences.....	- 20 -
2.4. Encadrement de travaux de recherche doctorale	- 22 -
2.4.1. Encadrement de doctorants.....	- 22 -
2.4.2. Encadrement de post-doctorants.....	- 27 -
2.4.3. Encadrement d'ingénieurs	- 27 -
2.4.4. Encadrement de stagiaires de DEA et de Master recherche	- 27 -
2.5. Responsabilités scientifiques.....	- 29 -
2.5.1. Participation à des jurys de thèse.....	- 29 -
2.5.2. Direction / édition d'ouvrages scientifiques	- 32 -
2.5.3. Participation à des comités techniques internationaux	- 32 -
2.5.4. Participation à des comités de programmes de conférences	- 32 -
2.5.5. Participation à des comités de lecture de conférences	- 33 -
2.5.6. Participation à des comités de lecture de journaux internationaux	- 33 -
2.5.7. Participation à des comités d'organisation	- 34 -
2.5.8. Modération dans des conférences internationales	- 34 -
2.5.9. Expertise scientifique	- 35 -
2.6. Diffusion des connaissances et publications scientifiques	- 35 -
2.6.1. Thèse de doctorat et rapport de DEA	- 35 -
2.6.2. Brevets.....	- 35 -
2.6.3. Articles dans des revues internationales avec comité de lecture	- 35 -
2.6.4. Articles dans des revues internationales sans comité de lecture.....	- 36 -
2.6.5. Chapitres de livre.....	- 36 -
2.6.6. Communications internationales avec comité de lecture et actes.....	- 36 -
2.6.7. Communications internationales sans comité de lecture avec actes	- 39 -
2.6.8. Communications internationales invitées sans actes	- 39 -
2.6.9. Communications francophones avec comité de lecture et actes	- 39 -
2.6.10. Communications nationales invitées sans actes	- 40 -
2.6.11. Participation à la rédaction de documents publics	- 40 -
2.6.12. Communications par affiches / Démonstrations (sélection).....	- 41 -
2.6.13. Logiciels libres.....	- 41 -
3. Activités d'enseignement.....	- 43 -
3.1. De 2000 à 2003 en tant que Moniteur de l'Enseignement Supérieur.....	- 43 -
3.1.1. Maîtrise GEII (2000/2001).....	- 43 -
3.1.2. Licence GEII (2000/2002).....	- 44 -

3.1.3.	DEUG GEII (2002/2003)	- 44 -
3.1.4.	Licence GSI (2000/2003)	- 44 -
3.1.5.	DEUG GSI (2001/2003).....	- 44 -
3.1.6.	Licence STI (2000/2001).....	- 45 -
3.1.7.	DEUG STI (2002/2003)	- 45 -
3.1.8.	DEUG GMP (2000/2002).....	- 45 -
3.1.9.	Licence GMP (2001/2003)	- 45 -
3.1.10.	Licence Pluridisciplinaire (préparation au CAPET) (2002/2003).....	- 45 -
3.2.	De 2003 à 2004 en tant que ½ Attaché Temporaire d'Enseignement et de Recherche	- 46 -
3.2.1.	Licence GEII	- 46 -
3.2.2.	DEUG GEII	- 46 -
3.3.	Depuis 2004 en tant que Maître de Conférences	- 47 -
3.3.1.	Master 2 Math-STIC, parcours SIAM (2006/2011)	- 47 -
3.3.2.	Master 1 Math-STIC, parcours SIAM (2004/2011)	- 47 -
3.3.3.	Licence 3 PCSI, parcours EII (2004/2011).....	- 47 -
3.3.4.	Licence 1 Physique Chimie Sciences de l'Ingénieur (2008/2011)	- 48 -
3.3.5.	Master EII (Institut du Génie Appliqué, Maroc) (2008/2010).....	- 49 -
3.3.6.	Formation doctorale (Université de Mondragon, Espagne) (2010)	- 49 -
3.3.7.	Ecole de printemps, ENS Lyon (2010).....	- 49 -
3.3.8.	Master ENSIETA (2008/2011).....	- 49 -
3.4.	Bilan de l'activité d'enseignement depuis 2000	- 49 -
3.4.1.	De 2000 à 2004 (moniteur et ½ ATER)	- 49 -
3.4.2.	De 2004 à aujourd'hui	- 50 -
4.	Responsabilités collectives, animations et projets scientifiques.....	- 53 -
4.1.	Au niveau de l'Université de Bretagne Sud	- 53 -
4.1.1.	Au sein de l'UFR Sciences et Sciences de l'Ingénieur.....	- 53 -
4.1.2.	Au sein du laboratoire	- 53 -
4.1.3.	Au sein des conseils de l'établissement.....	- 54 -
4.2.	Au niveau national	- 54 -
4.3.	Au niveau international	- 54 -
4.4.	Participation à des collaborations scientifiques et à des contrats d'études... -	- 55 -
4.4.1.	Collaborations Académiques Internationales	- 55 -
4.4.2.	Contrats publiques de recherche.....	- 56 -
4.4.3.	Contrats privés de recherche	- 57 -

PARTIE 2 : ANNEXES, SÉLECTION DES PUBLICATIONS SIGNIFICATIVES - 59 -

5.	Article concernant la synthèse d'architectures multi-modes.....	- 61 -
6.	Article concernant la synthèse de haut niveau.....	- 63 -
7.	Article concernant la synthèse d'architecture multi-horloges.....	- 65 -
8.	Article concernant la modélisation comportementale.....	- 67 -
9.	Article concernant la synthèse d'architecture d'unité mémoire gros grain -	69
10.	Article concernant la synthèse d'architecture d'unité mémoire grain fin -	71 -

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.

Résumé du dossier

<p>Curriculum Vitae</p>	<p>Fonction actuelle</p> <ul style="list-style-type: none"> • Maître de Conférences à l'Université de Bretagne Sud <i>Enseignement à l'UFR Sciences et Sciences de l'Ingénieur-SSI, Département Sciences et Techniques</i> <i>Recherche au Laboratoire en Sciences et Technologies de l'Information, de la Communication et de la Connaissance (Lab-STICC), UMR CNRS 3192</i> <p>Diplômes</p> <ul style="list-style-type: none"> • Thèse de doctorat (Sciences de l'ingénieur, option électronique) Université de Bretagne-Sud – Lorient / Laboratoire LESTER <i>Mention très honorable avec les félicitations du jury,</i> <i>obtenue le 13 décembre 2003</i> • DEA Architecture des Systèmes Intégrés et Micro Electronique (ASIME), <i>Option Conception de Systèmes Intégrés, Université Pierre et Marie Curie, Paris 6</i> <i>obtenue en septembre 1999</i> • Maitrise d'Informatique, <i>Option Architecture, Université Pierre et Marie Curie, Paris 6</i> <i>obtenue en juin 1998</i> <p>Expériences professionnelles</p> <ul style="list-style-type: none"> • Septembre 2004/Aujourd'hui Maître de Conférences à l'Université de Bretagne Sud <i>Prime d'encadrement doctoral et de Recherche (PEDR 2008/2012)</i> • Septembre 2003/Août 2004 Attaché Temporaire d'Enseignement et de Recherche ATER à l'Université de Bretagne Sud • Septembre 2000/Août 2003 Moniteur de l'Enseignement Supérieur à l'Université de Bretagne Sud • Septembre 2000/Août 2003 Doctorant (bourse MESR) <i>Laboratoire LESTER– Université de Bretagne-Sud – Lorient</i>
<p>Enseignement</p>	<p>Système d'exploitation, programmation, langage C, microcontrôleur, réseaux locaux industriels, technologies des composants, électronique numérique, méthodologie, logique combinatoire et séquentielle, synthèse de haut-niveau, ESL design, SoC, co-conception, architecture numérique</p>
<p>Recherche</p>	<p>Synthèse de haut-niveau, ESL Design, FPGA, Co-design, optimisation, architecture des systèmes embarqués, MPSoC, intégration logiciel/matériel, prototypage virtuel, exploration de l'espace de conception, partitionnement, ordonnancement, assignation, conception d'entrelaceur parallèle, application de télécommunication, multimédia</p>
<p>Développement</p>	<p>Synthèse de haut niveau (GAUT), synthèse d'adaptateur spatio-temporel (STAR system), Exploration de l'espace de conception (DSxPlore), générateur d'entrelaceur parallèle, prototypage virtuel (SocLib)</p>
<p>Laboratoire</p>	<p>Laboratoire Lab-STICC, Université de Bretagne Sud, CNRS UMR 3192 (2008 – aujourd'hui) Laboratoire LESTER, Université de Bretagne Sud, CNRS FRE 2734 (2000 – 2008)</p>
<p>Responsabilités administratives</p>	<p>Membre de la commission de sélection (2008/aujourd'hui) Membre élu de la commission de spécialistes 61^{ème} et 63^{ème} sections (2006/2008) Membre élu au Conseil Scientifique de l'Université de Bretagne-Sud (2009/aujourd'hui) Membre élu au Conseil de laboratoire du Lab-STICC (2008/aujourd'hui) Membre élu au Conseil de laboratoire du LESTER (2006/2008) Directeur des études du parcours Electronique et Informatique Industrielle de la Licence Physique Chimie Science de l'Ingénieur (2004/aujourd'hui) Membre du comité de pilotage de la Licence Science de l'Ingénieur (2007/aujourd'hui) Membre de la commission de perfectionnement de la Licence Science de l'Ingénieur (2010/aujourd'hui) et du Master STIC (2011/aujourd'hui) Membre de commission recherche et personnel de l'UFR (2011/aujourd'hui) Membre de la commission formation de l'UFR (2011/aujourd'hui) Membre élu au Conseil de l'Ecole Doctorale pluridisciplinaire de l'Université de Bretagne-Sud (2001/2003)</p>
<p>Responsabilités scientifiques</p>	<p>Membre élu au comité technique IEEE Signal Processing Society, <i>Design and Implementation of Signal Processing Systems (DISPS)</i> (2011 – 2014) Comité de programme de conférences internationales ASAP 2011, DATE 2010-2012 (co-chair track HLS), ISCAS 2005 à 2011, GLSVLSI 2007 à 2011, SIPS 2011, ERDIAP 2011, ESLsyn 2011 (program chair) Comité de lecture de conférences internationales DATE 2010-2011, ASAP 2011, ICASSP 2011, SIPS 2011, ICECS 2010, ISCAS 2005 à 2011, GLSVLSI 2007 à 2011, DAC 2005 Modérateur de session de conférences internationales ICASSP 2011, DATE 2010, DATE 2011,</p>

	<p>ISCAS 2010, GLSVLSI 2007, ISCAS 2005, tutorial DAC 2010, tutorial DAC 2009, workshop ASP-DAC 2009, workshop DATE 2008, workshop DAC 2008, workshop FDL 2006</p> <p>Direction/Édition d'ouvrages Journal of Electrical and Computer Engineering, special issue on ESL Design Methodology, (Hindawi) 2012, IEEE Design and Test of Computer, Special Issue on High-Level Synthesis 2009 "High-Level Synthesis: From Algorithm to Digital Circuit" (Springer) 2008.</p> <p>Comité de lecture de journaux internationaux IEEE Design and Test of Computers, IEEE Transactions on Computer Aided-Design of Integrated Circuits and Systems (TCAD), IEEE Journal on Emerging and Selected Topics in Circuits and Systems (JETCAS), ACM Transactions on Design Automation of Electronic System (TODAES), ACM Transactions on Embedded Computing System (TECS), Integration, the VLSI Journal - Elsevier Science, Journal of Embedded Computing (JEC), International Journal of Circuits, Systems, and Computers (JCSC), Journal on Embedded Systems, EURASIP (JES), Embedded Hardware Design (Microprocessors and Microsystems MICPRO) – Elsevier, Microelectronics Journal – Elsevier</p> <p>Comité d'organisation de tutoriaux internationaux CODES-ISSS 2010, DAC 2010, DAC 2009</p> <p>Comité d'organisation de workshops internationaux ASP-DAC 2009, DAC 2008, DATE 2008, FDL 2006</p> <p>Comité de programme de conférences nationales SYMPA 2011, Doctoriales 2002</p> <p>Comité de lecture de conférences nationales GRETSI 2011, SYMPA 2011, MajecSTIC 2006 et 2007, JFAAA 2002</p> <p>Comité d'organisation d'événements nationaux / régionaux tutorial SYMPA 2011, journée GIS SysCom 2010, journée inter-GDRs 2010, barcamp inter-GDRs 2011</p> <p>Expertise scientifique <i>expert scientifique externe</i> MESR CIR et JEI (2010-aujourd'hui)</p> <p>Expertise scientifique <i>expert scientifique ANR « Ingénierie Numérique & Sécurité » - INS 2011</i></p> <p>Membre du groupe de travail/standardisation SWG OSCI (Key Contributor 2009, Affiliate member 2010)</p> <p>Membre du comité de rédaction Roadmap MEDEA+/Catrene 2009</p> <p>Membre du réseau d'excellence européen HiPEAC (depuis 2009)</p> <p>Membre du comité de rédaction Roadmap HiPEAC 2011</p> <p>Membre du GdR ISIS, Thème C (depuis 2005 et responsable des relations inter GDR depuis 2010)</p> <p>Membre du GdR SoC-SiP (depuis 2005)</p>																		
Encadrements	<p>8 co-encadrements de doctorants (4 thèses soutenues, 2 thèses en cours de rédaction et 2 thèses en cours)</p> <p>2 encadrements de post-doctorant</p> <p>1 encadrement d'ingénieur</p> <p>8 encadrements de stagiaire en Master recherche ou DEA</p>																		
Collaborations	<table border="0"> <tr> <td>Astrium ST</td> <td>2011/2012, Contrat privé, <i>Lab-STICC, Astrium Space Transportation</i></td> </tr> <tr> <td>Projet P</td> <td>2011-2013, projet FUI, <i>Aboard Engineering, AGC Solutions, AdaCore, Airbus, Altair, ASTRIUM, ATOS Origin, Continental, Ecole des Ponts ParisTech, INRIA/Aoste-Espresso-Metalau, IRT/INPT/ENSEEIH, Lab-STICC, ONERA, Rockwell Collins, Sagem Défense Sécurité, Sailab, ST Informatique Services, Thales Alenia Space, Thales Avionics</i></td> </tr> <tr> <td>SoCKET</td> <td>2008 – 2011, projet FUI, <i>Airbus, Astrium, CNES, STMicroelectronics, Thales R&T, Schneider Electric Industries, PSI-S, CEA-LETI, Magilem Design Services, INPG-TIMA, UPS-IRIT, UBS-LabSTICC</i></td> </tr> <tr> <td>SocLib</td> <td>2006-2009, plate-forme ANR/RNTL, <i>STMicroelectronics, Thales Communications, Thomson Silicon Components, Prasilog, TurboConcept, Silicomp, UPMC/LIP6, ENST, UPMC/LISIF, CEA LIST, INRIA Futurs, IRISA, LESTER/Lab-STICC, IETR INSA, TIMA, CEA LIST, CITI</i></td> </tr> <tr> <td>FLASH</td> <td>2006-2007, ARC INRIA, <i>IRISA, LIFL, LESTER, INSERM U694</i></td> </tr> <tr> <td>Onagre</td> <td>2005-2008, CRE France Telecom, <i>France Telecom, LESTER/Lab-STICC</i></td> </tr> <tr> <td>ANTELOP</td> <td>2005, PRIRE, <i>Lab-STICC</i></td> </tr> <tr> <td>SystemC'Mantic</td> <td>2003-2005, projet RNTL, <i>Thales communications, TIMA, CEA LIST, TIMA, LESTER</i></td> </tr> <tr> <td>ALIPTA</td> <td>2002-2004, projet RNRT, <i>Valiosys, Thales, Sacet, Turbo-Concept, ENST Bretagne</i></td> </tr> </table>	Astrium ST	2011/2012, Contrat privé, <i>Lab-STICC, Astrium Space Transportation</i>	Projet P	2011-2013, projet FUI, <i>Aboard Engineering, AGC Solutions, AdaCore, Airbus, Altair, ASTRIUM, ATOS Origin, Continental, Ecole des Ponts ParisTech, INRIA/Aoste-Espresso-Metalau, IRT/INPT/ENSEEIH, Lab-STICC, ONERA, Rockwell Collins, Sagem Défense Sécurité, Sailab, ST Informatique Services, Thales Alenia Space, Thales Avionics</i>	SoCKET	2008 – 2011, projet FUI, <i>Airbus, Astrium, CNES, STMicroelectronics, Thales R&T, Schneider Electric Industries, PSI-S, CEA-LETI, Magilem Design Services, INPG-TIMA, UPS-IRIT, UBS-LabSTICC</i>	SocLib	2006-2009, plate-forme ANR/RNTL, <i>STMicroelectronics, Thales Communications, Thomson Silicon Components, Prasilog, TurboConcept, Silicomp, UPMC/LIP6, ENST, UPMC/LISIF, CEA LIST, INRIA Futurs, IRISA, LESTER/Lab-STICC, IETR INSA, TIMA, CEA LIST, CITI</i>	FLASH	2006-2007, ARC INRIA, <i>IRISA, LIFL, LESTER, INSERM U694</i>	Onagre	2005-2008, CRE France Telecom, <i>France Telecom, LESTER/Lab-STICC</i>	ANTELOP	2005, PRIRE, <i>Lab-STICC</i>	SystemC'Mantic	2003-2005, projet RNTL, <i>Thales communications, TIMA, CEA LIST, TIMA, LESTER</i>	ALIPTA	2002-2004, projet RNRT, <i>Valiosys, Thales, Sacet, Turbo-Concept, ENST Bretagne</i>
Astrium ST	2011/2012, Contrat privé, <i>Lab-STICC, Astrium Space Transportation</i>																		
Projet P	2011-2013, projet FUI, <i>Aboard Engineering, AGC Solutions, AdaCore, Airbus, Altair, ASTRIUM, ATOS Origin, Continental, Ecole des Ponts ParisTech, INRIA/Aoste-Espresso-Metalau, IRT/INPT/ENSEEIH, Lab-STICC, ONERA, Rockwell Collins, Sagem Défense Sécurité, Sailab, ST Informatique Services, Thales Alenia Space, Thales Avionics</i>																		
SoCKET	2008 – 2011, projet FUI, <i>Airbus, Astrium, CNES, STMicroelectronics, Thales R&T, Schneider Electric Industries, PSI-S, CEA-LETI, Magilem Design Services, INPG-TIMA, UPS-IRIT, UBS-LabSTICC</i>																		
SocLib	2006-2009, plate-forme ANR/RNTL, <i>STMicroelectronics, Thales Communications, Thomson Silicon Components, Prasilog, TurboConcept, Silicomp, UPMC/LIP6, ENST, UPMC/LISIF, CEA LIST, INRIA Futurs, IRISA, LESTER/Lab-STICC, IETR INSA, TIMA, CEA LIST, CITI</i>																		
FLASH	2006-2007, ARC INRIA, <i>IRISA, LIFL, LESTER, INSERM U694</i>																		
Onagre	2005-2008, CRE France Telecom, <i>France Telecom, LESTER/Lab-STICC</i>																		
ANTELOP	2005, PRIRE, <i>Lab-STICC</i>																		
SystemC'Mantic	2003-2005, projet RNTL, <i>Thales communications, TIMA, CEA LIST, TIMA, LESTER</i>																		
ALIPTA	2002-2004, projet RNRT, <i>Valiosys, Thales, Sacet, Turbo-Concept, ENST Bretagne</i>																		
Publications	<p>7 revues scientifiques internationales</p> <p>2 chapitres d'ouvrages scientifiques internationaux</p> <p>29 publications en conférences internationales</p> <p>9 publications en conférences nationales</p> <p>1 brevet Europe-USA (et 2 actuellement en cours de dépôt)</p> <p>3 outils de CAO diffusés dont 1 en open source</p>																		

1. Curriculum Vitae

1.1. Etat civil

COUSSY Philippe
Né le 4 septembre 1974 à Montreuil (93)
Nationalité française
36 ans, marié

Adresse professionnelle

Laboratoire Lab-STICC
CNRS UMR 3192, Université de Bretagne Sud
Centre de recherche, BP 92116 - 56321 LORIENT Cedex
Tel : 02 97 87 45 65 – fax : 02 97 87 45 27
Email : philippe.coussy@univ-ubs.fr
Web : <http://www-labsticc.univ-ubs.fr/~coussy/>

1.2. Grades et titres universitaires

Décembre 2003

Doctorat en Sciences pour l'Ingénieur, option électronique

Université de Bretagne Sud, Lorient

Laboratoire LESTER (laboratoire d'électronique des systèmes temps réel)

Titre : Synthèse d'Interface de Communication pour les Composants Virtuels

Mention : Très honorable avec les félicitations du jury

Septembre 1999

DEA Architecture des Systèmes Intégrés et Micro électronique (ASIME)

Option : Conception de Systèmes Intégrés

Université Pierre et Marie Curie, Paris 6

Juin 1998

Maîtrise d'Informatique

Option : Architecture des Systèmes Intégrés

Université Pierre et Marie Curie, Paris 6

Juin 1997

Licence d'Informatique

Université Pierre et Marie Curie, Paris 6

Septembre 1996

DEUG A, Sciences des structures et de la matière

Université Pierre et Marie Curie, Paris 6

Juin 1993

Baccalauréat série E

Lycée Diderot, Paris 19

1.3. Situations successives

Septembre 2004 – Aujourd’hui

Maître de Conférences (PEDR depuis octobre 2008)

Université de Bretagne-Sud

UFR Sciences et Sciences de l’Ingénieur (SSI)

Département Sciences et Techniques (SET)

Filière Electronique et Informatique Industrielle (EII)

Section CNU 61 (qualifié CNU 61 et CNU 27 en mars 2004)

Laboratoire en Sciences et Technologies de l’Information, de la Communication et de la Connaissance (Lab-STICC), UMR CNRS 3192

Septembre 2003 – Septembre 2004

Attaché Temporaire d’Enseignement et de Recherche (ATER)

Université de Bretagne-Sud

Département GEII, IUP de Lorient

Laboratoire LESTER (Laboratoire d’Electronique des Systèmes Temps Réel)

Septembre 2000 – Septembre 2003

Allocataire bourse MESR / Moniteur CIES Grand Ouest

Université de Bretagne-Sud

Laboratoire LESTER (Laboratoire d’Electronique des Systèmes Temps Réel)

Septembre 1999 – Juin 2000

Scientifique du Contingent (Service Militaire Actif)

Enseignant - Formateur en informatique et bureautique

Cellule formation de l’Etat Major de l’Armée de Terre (EMAT), Paris

1.4. Résumé

Après une thèse de trois années financée par une bourse MNERT et encadrée par le Professeur Eric Martin, j’ai obtenu en 2003 mon Doctorat en Sciences de l’Ingénieur à l’Université de Bretagne Sud. De 2003 à 2004, j’ai été ATER à l’IUP GEII de l’Université de Bretagne Sud et au laboratoire LESTER.

Depuis 2004, je suis Maître de Conférences à l’UFR Sciences et Sciences de l’Ingénieur de l’Université de Bretagne-Sud. J’effectue principalement des enseignements d’informatique industrielle (système d’exploitation, programmation, réseaux locaux) et d’électronique numérique (SoC, Synthèse de haut-niveau, architecture numérique). Depuis 2004 j’ai en charge la direction des études du parcours Electronique et Informatique Industrielle (EII) de la Licence Physique, Chimie et Sciences de l’Ingénieur (PCSI).

Je réalise mes activités de recherche dans l’équipe Méthodes et Outils pour les Circuits et Systèmes (MOCS) du laboratoire Lab-STICC. Mes travaux concernent les méthodes de conception des systèmes numériques avec une forte dominante pour les méthodologies de conception dites de niveau système (« ESL Design » pour « Electronique System Level Design») qui n’ont cessées ces dernières années de prendre de l’ampleur : synthèse de haut-niveau, prototypage virtuel...

J’ai publié une quarantaine d’articles dans des conférences majeures du domaine des systèmes embarqués et de l’architecture des systèmes numériques : ICCAD, FPL, ISCAS, ICASSP, GLSVLSI, etc., ainsi que dans différents ouvrages et revues scientifiques (IEEE

TCAD, IEEE D&T of Computers, ACM TECS, Eurasip, Springer). Je contribue régulièrement à des conférences internationales en organisant des tutoriaux et des workshops (IEEE DAC, DATE, ASP-DAC, CODES+ISSS) et en modérant des sessions techniques (DATE, ISCAS, ICASSP...). Actuellement et depuis plusieurs années, je participe à de nombreux comités de programmes de conférences internationales (DATE (track co-chair), ISCAS, GLSVLSI, ASAP, ICASSP...). J'ai édité deux ouvrages internationaux : un livre aux éditions Springer et un numéro spécial du magazine IEEE Design and Test of Computers. Je suis éditeur invité d'un numéro spécial du journal of Electrical and Computer Engineering, special issue on ESL Design Methodology, Hindawi (janvier 2012).

Je suis membre élu au comité technique de l'IEEE Signal Processing Society, Design and Implementation of Signal Processing Systems (DISPS). Je suis membre du réseau d'excellence Européen HiPEAC et membre du groupe de travail de l'OSCI (Open SystemC Initiative) définissant le sous-ensemble synthétisable du langage SystemC. J'ai participé à la rédaction de la « roadmap » Medea/Catrene Européenne sur l'EDA et participe actuellement à la « roadmap » HiPEAC.

Je participe à plusieurs projets de recherche (FUI, ANR, privé...). Je participe aussi activement à l'activité nationale dans le domaine de la conception de systèmes numériques embarqués (GDR ISIS, GDR SoC-SiP).

1.5. Encadrement doctoraux et post-doctoraux

La liste ci-dessous résume les huit co-encadrements de thèses effectuées depuis 2004 (dont quatre en cours) ainsi que la situation professionnelle actuelle de ces étudiants. A cette liste s'ajoute le travail avec deux étudiants en Post-Doc, un ingénieur et l'encadrement de neuf étudiants de DEA ou Master recherche.

Thèses en cours

- **Paolo BURGIO**, (Co-direction Luca Benini, Univ. Bologne, 50%)
Utilisation de la mémoire partagée dans le contexte d'un processeur multi-cœurs embarqué : exploration de la technologie et ses limites.
Co-tutelle : Université de Bologne
Financement : contrat
Début : janvier 2011
Soutenance prévue : décembre 2013
- **Awais Hussain SANI**, (Co-encadrement Eric Martin, 50%)
Approches de conception d'entrelaceurs parallèles pour les applications de codes correcteurs d'erreurs
Financement : allocation MESR
Début : octobre 2008
Soutenance prévue : décembre 2011
- **Aroua BRIKI**, (Co-encadrement Cyrille Chavet 50%)
Conception d'unités mémoires complexes pour les applications de traitement du signal
Financement : Allocation MESR
Début : octobre 2009
Soutenance prévue : décembre 2012

- **Ghizlane LHAIRECH-LEBRETON**, (Co-encadrement Eric Martin, 50%)
Synthèse de haut niveau pour la conception d'architecture faible consommation
Financement : contrat
Début : octobre 2007
Soutenance prévue : décembre 2011

Thèses soutenues

- **Kods TRABELSI**, (Co-encadrement Marc Sevaux, 50%)
Méthodes d'optimisation pour la conception sous contraintes de systèmes et de circuits électroniques
Financement : Allocation MESR
Début : octobre 2006
Soutenue en 2009
Situation actuelle : *Ing. R&D* Commissariat à l'Energie Atomique CEA-LIST
- **Caaliph ANDRIAMISAINA**, (Co-encadrement Emmanuel Casseau, 50%)
Flot de conception dédié aux architectures multi-modes pour les applications de traitement du signal et de l'image
Financement : Contrat
Début : octobre 2004
Soutenue en 2008
Situation actuelle : *Ing. R&D* Commissariat à l'Energie Atomique CEA-LIST
- **Cyrille CHAVET**, (Co-encadrement Eric Martin, 50%)
Synthèse automatique d'interfaces de communication matérielles pour la conception d'applications du domaine du traitement du signal
Financement : CIFRE STMicroelectronics
Début : octobre 2004
Soutenue en 2007
Situation actuelle : *Maître de Conférences* à l'Université de Bretagne-Sud, Lorient.
- **Farhat THABET**, (Co-encadrement Eric Martin, 50%)
Modélisation Comportementale Unifiée pour la Simulation et la Synthèse Haut-Niveau de composants Virtuels Algorithmiques
Financement : Contrat laboratoire
Début : octobre 2003
Soutenue en 2007
Situation actuelle : *Ing. R&D* Commissariat à l'Energie Atomique CEA-LIST

1.6. Publications

Mon activité scientifique menée depuis 2000 a conduit aux publications suivantes : sept revues scientifiques internationales, deux participations à des ouvrages scientifiques internationaux, vingt neuf publications en conférences internationales et neuf publications en conférences nationales. Cela représente quarante sept publications scientifiques.

Publication en revues (7)	<ul style="list-style-type: none"> - IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems (TCAD) - IEEE Design and Test of Computers - ACM Transactions on Embedded Computing Systems - Journal of Signal Processing Systems, Springer - Integration, the VLSI Journal, Elsevier Science - EURASIP Journal on Embedded Systems
Chapitre de livre (2)	<ul style="list-style-type: none"> - Springer , “High-Level Synthesis: From Algorithm to Digital Circuits” - Springer Metaheuristics
Publications en conférences (38)	<p>29 publications lors de conférences internationales avec comité de lecture dont 22 IEEE : ICASSP 2011, ISCAS 2011, ICECS 2010, SIES 2010, FPL 2010, ICASSP 2010, ISCAS 2010, MIC 2009, ICCAD 2007, EUSIPCO 2007, ISCAS 2007, AHS 2007, GLSVLSI 2007, SIPS 2006, FDL 2005, ICASSP 2005, ISCAS 2005, ICASSP 2004, ICM 2004, VLSI-SOC 2003, ICECS 2002, EUSIPCO 2002, ICCAS 2002, ISCAS 2002, CICC 2002, FDL 2001, VLSI-SOC 2001</p> <p>10 publications lors de conférences francophones avec comité de lecture : GRETSI 2011, SYMPA 2011, ROADEF 2008, FTFC 2007, JFAAA 2005, GRETSI 2005, MAJECTIC 2005, GRETSI 2004, JFAAA 2002</p> <p>1 conférence invitée avec actes : CODES+ISSS 2010</p>

2. Résumé des activités de recherche

2.1. Contexte

Ma prise de contact avec la recherche a débuté lors de mon stage de Maîtrise d'informatique que j'ai effectué au sein du département ASIM du laboratoire LIP6 sous la direction du Professeur Alain Greiner (janvier – juin 1998). Les travaux concernaient la conception et la réalisation d'un contrôleur de bus générique pour systèmes intégrés.

J'ai prolongé cette première expérience en architecture des systèmes intégrés par un stage de DEA effectué au sein du département « ComElec » de l'Ecole Nationale Supérieure des Télécommunications (ENST-Paris) sous la direction de Jean-Luc Danger (janvier – juin 1999). Les travaux visaient la conception et la réalisation d'un accélérateur matériel, couplé à et piloté par un processeur DSP, qui effectuait une Transformée en Cosinus Discrets et une Quantification, toutes deux directes et inverses, pour un codeur MPEG-2 temps réel. Ce fut mon premier contact avec le traitement des images, les approches de conception conjointe matérielle/logicielle, les « ASIP », les accélérateurs matériels et les architectures dites de nos jours « MPSoC ».

Ces travaux sur la conception d'accélérateur matériel se sont poursuivis durant mes trois années de thèse, puis mon année d'ATER, que j'ai effectuées sous la direction du Professeur Eric Martin, au sein du laboratoire LESTER à l'Université de Bretagne-Sud à Lorient (septembre 2000 – août 2004). Durant cette période, j'ai travaillé sur une approche de réutilisation et de conception automatisée de coprocesseurs dédiés dans les applications orientées traitement du signal, de l'image et des télécommunications. L'approche proposée reposait sur la notion de composants virtuels de niveau algorithmique définie dans le cadre des projets RNRT MILPAT et ALIPTA. C'est dans ce contexte que nous avons définie une approche de modélisation formelle des contraintes d'intégration des composants virtuels algorithmiques. Les travaux permettaient grâce à l'utilisation de la synthèse haut niveau de générer des composants « sur mesures », vérifiant chaque fois que cela était possible, les contraintes temporelles d'entrées/sorties (chronogramme) définies par le concepteur. Ces travaux ont été intégrés dans l'outil GAUT.

Durant mon année d'ATER, j'ai prolongé cet axe en travaillant sur la modélisation comportementale à l'aide du langage SystemC en vue de la simulation et de la synthèse. Cette approche a été explorée et validée durant le projet RNTL SystemC'Mantic ainsi que dans le cadre des travaux de thèse de Farhat Thabet. La prise en compte simultanée des contraintes temporelles et des contraintes mémoire a été abordée en collaboration avec Gwenolé Corre durant sa troisième année de thèse. L'année d'ATER s'est aussi vue, en partie, consacrée à une collaboration avec l'équipe « Communications Numériques » du laboratoire sur la synthèse d'architecture de l'algorithme Maximum A Posteriori (MAP) pour des applications de turbo décodage temps réel dans le cadre de la thèse de David Gnaedig.

En septembre 2004, j'ai été recruté à l'Université de Bretagne Sud en tant que Maître de Conférences au sein de l'UFR Sciences et Sciences de l'Ingénieur et au laboratoire LESTER. Depuis, j'ai poursuivi, intensifié et élargi mon domaine de recherche autour de la synthèse d'architecture. Mes activités, qui se déroulent maintenant au sein de l'équipe Méthodes et Outils pour les Circuits et Systèmes (MOCS) du laboratoire Lab-STICC,

peuvent ainsi se décomposer en trois grands axes : *la synthèse d'architecture d'unité de calcul, la modélisation et le prototypage ainsi que la synthèse d'architecture d'unité mémoire*. Ces trois axes visent à proposer des méthodes de conception de niveau « système » et d'optimisation permettant au concepteur de circuits et de systèmes numériques de faire abstraction des détails en rehaussant le niveau des spécifications, de réduire le temps de conception en générant automatiquement des architectures optimisées, d'explorer plus rapidement et efficacement l'espace de conception, d'obtenir rapidement une solution correcte par construction, de réaliser des architectures complexes répondant à un objectif ou une combinaison d'objectifs...

Ces travaux ont été abordés dans le cadre de différents projets collaboratifs de recherche (ANR SocLib, FUI/DGE SoCKeT, CRE Onagre, ARC Flash...) et de travaux pré et post doctoraux (2 post-docs, 8 thèses, 9 stages de DEA/Master recherche). Les différents travaux réalisés dans ces trois axes sont détaillés dans la partie II du rapport. Nous donnons ici (partie I) un aperçu rapide des activités de recherche ainsi que des travaux menés dans ces thématiques.

2.2. Activités de recherche « doctorale »

Maîtrise : Conception d'un contrôleur de bus pour systèmes intégrés

Mon expérience dans le domaine de la conception de système numérique a débuté lors de mon stage de maîtrise (1998) effectué au département *Architecture des Systèmes Intégrés et Micro-électronique ASIM* du Laboratoire d'Informatique de Paris 6 (LIP6) de l'Université Pierre et Marie Curie (UPMC) sous la direction du Professeur A. Greiner et l'encadrement de Frédérique Pétrot (Maître de conférences) et Denis Homais (Doctorant). Le but de ce stage était de concevoir une interface de communication contenant un cache à correspondance directe (Direct Mapping), un contrôleur de bus PIBUS ainsi qu'un tampon d'écriture postée (FIFO). Une partie du travail consistait à écrire un modèle comportemental du composant en langage C (équivalent au langage actuel SystemC). Cette spécification fonctionnelle était utilisée dans l'environnement de simulation PISIM (actuellement nommé SystemCASS et utilisé dans le projet plateforme ANR SoCLib) qui permet la simulation des systèmes intégrés mixtes logiciel/matériel. Il s'agissait des premiers travaux de recherche autour de la modélisation de niveau système.

DEA : Codeur/décodeur VLSI pour DVD RAM

Cette expérience s'est poursuivie durant le stage de DEA (1999) que j'ai effectué au sein du département Communication et Electronique de *l'Ecole Nationale Supérieure des Télécommunications* (ENST) de Paris sous la direction de Jean-Luc Danger et l'encadrement de Lirida Naviner. Ces travaux furent réalisés et intégrés en partenariat étroit avec la société STMicroelectronics (Division Digital Video, Grenoble) dans le cadre de l'étude et du prototypage d'un premier circuit codec pour magnétoscope numérique (DVD-RAM). Le modèle du système sur puce, qui reposait sur une architecture multiprocesseurs, intégrait plusieurs DSP950 ayant chacun un accélérateur matériel piloté par des instructions dédiées. Dans ce contexte de conception mixte logiciel/matériel, j'ai conçu et réalisé les blocs d'accélération DCT/IDCT et Q/IQ (Transformée en Cosinus Discrète et Inverse, Quantification et Quantification Inverse) qui ont été intégrés dans le codeur - décodeur vidéo MPEG2 temps réel.

Thèse de doctorat

Mes activités de recherche se sont poursuivies et accélérées à travers la préparation de mon doctorat (2000-2003), au sein du laboratoire LESTER de l'Université de Bretagne Sud à Lorient. Mes thèmes de recherche se sont articulés autour de la synthèse de haut niveau. Je me suis intéressé à la modélisation de contraintes temporelles en vue d'une synthèse optimisée des composants virtuels de niveau algorithmique.

Thèse de doctorat soutenue le 13 décembre 2003 à l'Université de Bretagne Sud – Lorient

Synthèse d'Interface de Communication pour les Composants Virtuels

Directeur de thèse : Pr. Eric Martin

Mention : *très honorable avec félicitations du jury*

Composition du jury :

A.A. Jerraya	TIMA, INPG de Grenoble	Président
F. Petrot	ASIM, Université Paris 6	Rapporteur
O. Sentieys	ENSSAT, Université de Rennes 1	Rapporteur
J.P. Calvez	IRESTE, Ecole Polytechnique de Nantes	Examineur
E. Martin	LESTER, Université de Bretagne Sud	Directeur de thèse
A. Baganne	LESTER, Université de Bretagne Sud	Examineur
H. Michel	STMicronics	Invité

Nous avons proposé dans ces travaux une approche de réutilisation et de conception des IPs dans les applications orientées traitement du signal, de l'image et des télécommunications. Pour cela, nous avons basé notre approche sur la notion de composants virtuels de niveau algorithmique, définie dans le cadre du projet *RNRT MILPAT (Méthodologie et Développement pour les Intellectual Properties pour Applications Telecom)*. Le flot de conception proposé s'inscrit dans la démarche Adéquation Algorithme Architecture du projet *RNRT ALITPA (Définition et Application d'une méthodologie de développement pour les (IP) intellectual property de niveau comportemental dans les applications de télécommunication)* et est basé sur l'utilisation de techniques de synthèse haut niveau sous contraintes d'intégration. Les unités fonctionnelles constituant l'architecture cible du composant étaient (re)conçues en fonction des caractéristiques de l'architecture de communication du système et de la spécificité de l'application.

Dans ce contexte, la spécification algorithmique de l'IP est modélisée par un Graphe Flot de Signaux (*SFG*) qui, couplé aux temps de propagations des opérateurs et à la cadence d'itération, permet la génération d'un graphe de contrainte algorithmique *ACG*. Nous avons développé une méthode d'analyse formelle des contraintes, qui repose sur les calculs de cycles, et permet de vérifier la cohérence entre la cadence, les dépendances de données de l'algorithme et les contraintes technologiques.

Les contraintes d'intégration, spécifiées pour chacun des bus (ports) reliant l'IP aux autres composants du système, sont modélisées par un graphe de contraintes d'Entrée/Sortie *IOCG (IO Constraint Graph)*. Ce formalisation supporte, entre autre, la modélisation (1) du type de transferts, (2) des variations temporelles des dates d'arrivées des données, (3) du séquençement des données échangées (4) des mécanismes liés aux protocoles (synchrone / asynchrone)... Les contraintes d'intégration et les contraintes algorithmiques de l'IP sont fusionnées pour fournir un graphe détaillé des contraintes *GCG (Global Constraint Graph)* exhibant les points de synchronisation entre l'environnement et le composant. Des optimisations pour l'implémentation ont été proposées à partir de transformations formelles du graphe.

La synthèse de l'unité de traitement (*UT*) est réalisée à l'aide de l'outil *GAUT* dont l'ordonnancement est contraint par les paramètres temporels de l'intégrateur du composant

virtuel. La synthèse de l'*UT* génère un ensemble de contraintes au E/S, modélisées sous la forme d'un *IPERM* (*IP Execution Requirement Model*). Ce dernier représente formellement (1) les dates de production et de consommation des données dans l'*UT* et (2) les bus sur lesquels elles transitent entre l'unité de communication et l'unité de traitement. Les modèles *IPERM* et *IOCG* sont finalement utilisés pour synthétiser l'unité de communication.

L'ensemble des travaux de recherche menés dans ce domaine durant ma thèse entre 2000 et 2003 ont conduit directement ou indirectement à dix neuf publications scientifiques : une thèse de doctorat [COU 2003a], deux revues internationales [COU 2007a] [COU 2006a], onze conférences internationales [PIL 2001a] [SAV 2001a] [COU 2002a] [COU 2002b] [COU 2002c] [COU 2002d] [COU 2002e] [COU 2003b] [ABB 2004a] [COU 2005a] [COU 2005b] et cinq conférences nationales [COU 2002f] [COU 2002g] [COU 2004b] [COU 2005c] [LAU 2007a].

2.3. Activités de recherche en tant que Maître de Conférences

Depuis 2000, date de mon arrivée au laboratoire LESTER 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. De 2000 à 2006, j'ai été membre de l'équipe « composant virtuel ». Cette équipe était composée, pour les membres permanents, d'Eric Martin, Professeur des Universités, élu en décembre 2004 Président de l'Université de Bretagne Sud et d'Emmanuel Casseau, Maître de conférences promu en 2006 Professeur des Universités à l'Université de Rennes 1, ENSSAT, Lannion après un CRCT de six mois en 2005 à Edinbourg en Ecosse. En 2006, j'ai donc eu la responsabilité scientifique du groupe alors renommé «High-Level Synthesis». Depuis 2008, année de création du Lab-STICC, je suis responsable de l'axe « Conception d'architectures avancées » dans l'équipe thématique « Méthodes et Outils pour les Circuits et Systèmes » (MOCS) du pôle « Communication, Architectures et Circuits » (CACS).

Les thèmes de recherche que j'ai développés ces dernières années se sont concentrés autour d'un thème principal : les approches de conception de haut niveau des systèmes embarqués (*System on Chip – SoC*). Trois axes de recherche peuvent ainsi être dégagés : *la synthèse d'architecture d'unité de calcul, la modélisation et le prototypage ainsi que la synthèse d'architecture d'unité mémoire*. Huit étudiants en thèse, neuf stagiaires de DEA ou Master recherche, deux post-doctorants et un ingénieur ont travaillé ou travaillent actuellement sur ces différents thèmes de recherche.

Ces travaux ont été ou sont validés sur différents types d'applications du domaine du traitement du signal ou de l'image. Ces différents thèmes de recherche, présentés ci-dessous selon chacun des trois axes, seront détaillés dans la troisième partie de ce document.

Axe 1 : Synthèse d'architecture d'unité de calcul

Cet axe de recherche cible la génération automatique d'accélérateurs matériels ou coprocesseurs dédiés. Les architectures de niveau transfert de registres (RTL) sont générées sous contraintes (temps, ressources...) en utilisant des techniques de synthèse de haut niveau et visent à optimiser un ou plusieurs objectifs (consommation, temps, surface...). Les travaux de recherche de cet axe ont été développés dans le cadre de différents projets. Ainsi :

- la conception d'architecture multi-modes par la synthèse de haut niveau a été abordée dans le cadre des travaux de thèse de Caaliph Andriamisaina (2005-2008),
- l'optimisation de la surface à l'aide de méta-heuristiques a été explorée dans le cadre de la synthèse de haut niveau durant la thèse de Kods Trabelsi (2006-2009),

- la génération par la synthèse de haut niveau hiérarchique d'architecture à multiples domaines d'horloge pour la conception faible consommation sur FPGA est étudiée dans les travaux de thèse de Ghizlane Lhairech-Lebreton (2007-2011),
- l'optimisation des architectures en surface et consommation par la synthèse de haut niveau de spécifications bit-près a été traitée dans le cadre du projet de recherche privée ONAGRE en partenariat avec France Telecom (2005-2008),
- l'introduction de la prédiction de branchement dans la synthèse comportementale a été évaluée durant le stage de Master recherche de Vianney Lapotre (2010).

Les travaux menés au sein de cet axe de recherche (hors publications réalisées durant la thèse) ont conduit à quinze communications scientifiques réparties comme suit : trois revues internationales [AND 2010a] [COU 2009a] [COU 2009b], deux chapitres de livre [TRA 2010a] [COU 2008b], six conférences internationales [LHA 2010b] [COU 2009c] [Andrai07a] [CHA 2007a] [COU 2005b] [COU 2004a], quatre conférences nationales [TRA 2008a] [LAP 2011a] [LHA 2011a] [LAU 2007a].

Axe 2 : Modélisation et prototypage

Cet axe de recherche porte sur les approches de conception de niveau « système » (ESLD) et sur l'intégration des outils de synthèse d'architecture dans ces flots de conception : raffinement algorithmique, prototypage FPGA, conception d'un system-on-chip... Les travaux de recherche de cet axe ont été menés dans le cadre de différents projets. Ainsi,

- la modélisation formelle des contraintes temporelles aux entrées / sorties (comportement aux interfaces) pour la synthèse de composants virtuels algorithmiques a tout d'abord été abordée durant ma thèse. Les prises en compte simultanées des contraintes mémoire et d'E/S a été étudiée durant mon année d'ATER (2000-2004),
- la modélisation, à l'aide du langage SystemC, en vue d'une simulation avant synthèse des composants a été explorée durant le projet RNTL SystemC'Mantic ainsi que dans le cadre des travaux de thèse de Farhat Thabet (2004-2007),
- la modélisation d'application de traitement du signal et la synthèse d'architecture prenant en compte des spécifications bit-près ont été abordées dans le cadre du contrat de recherche externe (CRE) France Telecom ONAGRE (2005-2008),
- la modélisation et la génération de modèles de simulation cycle-près ou transactionnel par des outils de synthèse de haut niveau a été adressée plus récemment dans le cadre du projet plateforme ANR SocLib (2006-2009) et dans le cadre du post-doctorat de Caaliph Andriamisaina (2008-2009),
- l'utilisation conjointe de la compilation et de la synthèse de haut niveau dans un flot de conception automatisé visant les architectures MPSoC est étudiée dans la thèse en co-tutelle de Paolo Burgio (2010-2013).

Les travaux menés au sein de cet axe de recherche (hors publications réalisées durant la thèse) ont conduit à treize communications scientifiques réparties comme suit : trois revues internationales [THA 2009a] [COU 2008a] [COU 2010a], huit conférences internationales [LEF 2010a] [LHA 2010a] [THA 2007a] [THA 2006a] [KRI 2005a] [Coussy5a] [ABB 2004a] [THA 2004a], deux conférences nationales [THA 2005a] [Coussy5c].

Axe 3 : Synthèse d'architecture d'unité mémoire

Ce troisième axe de recherche se concentre sur la génération automatique d'unité mémoire. Il s'agit de trouver un placement des données dans des éléments de stockage permettant des accès parallèles sans conflit. Ce problème complexe est fréquemment

rencontré durant la conception des entrelaceurs utilisés dans les circuits correcteurs d'erreur type LDPC ou turbo codes utilisés dans des applications telles que Wifi, Wimax, DVB-S, DVB-T, 3GPP-LTE. Ce problème apparaît aussi durant la conception d'architectures de traitement d'image ou dans la synthèse de haut niveau lorsque les données sont placées dans des bancs mémoires et que l'application doit être paralléliser pour respecter les contraintes de débit.

Les travaux de recherche de cet axe ont été développés dans le cadre de différents projets. Ainsi :

- le placement des données dans des éléments mémorisant grain fin à sémantique forte (FIFO, LIFO, registre...) a été traité dans le cadre de la thèse CIFRE STMicroelectronics de Cyrille Chavet (2004-2007) ;
- le placement des données dans des éléments mémorisant gros grain (bancs mémoires)
 - o a été initié dans le cadre de la thèse CIFRE STMicroelectronics de Cyrille Chavet qui a donné lieu à un dépôt de brevet en Europe et aux USA (2004-2007),
 - o a été ensuite poursuivi sur des aspects méthodologiques par les travaux de thèse d'Awais Sani Hussain (2008-2011) ;
- le placement des données dans des éléments mémorisant gros grain et grain fin par relaxation de contraintes dans le cadre des travaux de thèse d'Aroua Briki (2009-2012).

Les travaux menés au sein de cet axe de recherche ont conduit à dix communications scientifiques réparties comme suit : un brevet [CHA 2009a] [CHA 2008a], huit conférences internationales [SAN 2011a] [SAN 2011b] [SAN 2010a] [CHA 2010a] [CHA 2010b] [CHA 2007b] [CHA 2007c] [CHA 2007d], une conférence nationale [CHA 2005a].

2.4. Encadrement de travaux de recherche doctorale

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

- Encadrement de doctorants
- Encadrement de post-doctorants
- Encadrement d'ingénieurs
- Encadrement de stagiaires de DEA et de Master recherche

2.4.1. Encadrement de doctorants

Depuis 2004 j'ai co-encadré huit thèses. Quatre thèses ont été soutenues, respectivement en mai et octobre 2007, novembre 2008 et décembre 2009. Quatre thèses sont actuellement en cours dont une en première année et en cotutelle avec l'Université de Bologne, une en deuxième année et deux en phase de rédaction.

- Farhat Thabet 2003-2007 – Contrat

Modélisation Comportementale Unifiée pour la Simulation et la Synthèse Haut-Niveau de composants Virtuels Algorithmiques

Thèse de Doctorat soutenue le 4 mai 2007

En Co-encadrement avec le Pr. Eric Martin (50%)

Résumé : *Actuellement, des systèmes complets, contenant une partie logicielle et une partie matérielle, sont intégrés sur une même puce nommée Système-sur-Silicium (SoC). L'utilisation du niveau d'abstraction RTL (Register Transfer Level) comme point d'entrée dans les flots de conception, d'exploration et d'intégration d'un SoC ne permet plus de respecter les délais de mise sur le marché ni les contraintes de coût de conception. De nouvelles approches permettant de rehausser le niveau d'abstraction de la conception sont nécessaires. Ainsi, pour faire face à l'évolution continue du fossé qui existe entre la capacité de production et la capacité d'intégration, les équipes de recherche proposent de nouvelles méthodologies de conception qui reposent sur (1) la réutilisation de blocs logiciels ou matériels préconçus (Composant Virtuel), sur (2) l'élévation du niveau de description des applications (Electronic System Level), sur (3) l'orthogonalisation de différents aspects (fonctionnalité / architecture, calculs / communication, besoins / contraintes / choix d'implantation) et sur (4) l'automatisation des flots de conception.*

Les travaux de cette thèse ont abouti à une approche de modélisation unifiée allant de la spécification fonctionnelle jusqu'au niveau RTL/CA (pour Cycle Accurate). Ces travaux permettent l'exploration et la conception semi-automatisées d'accélérateurs matériels du domaine du Traitement Du Signal et de l'Image (TDSI). L'approche proposée repose à la fois sur (1) un modèle de spécification comportemental, nommé Behavioral Description Model BDM, et (2) sur la synthèse de haut niveau HLS (pour High Level Synthesis). Le modèle BDM, utilisé pour la simulation, permet de modéliser les architectures de communication (interface) puis d'explorer aisément pour chaque architecture différents comportements temporels. Il permet enfin de générer automatiquement les contraintes temporelles utilisées lors de la synthèse haut-niveau de la fonction de traitement.

Ces travaux ont été réalisés et validés sur des applications complexes dans le cadre des projets RTNL SystemC'MAntic et ANR SocLib.

Composition du jury : Patrice Quinton (Professeur des Universités, Université de Rennes 1, Président), Michel Auguin (Directeur de Recherche CNRS, I3S Nice – Sophia Antipolis, rapporteur), Jean-Paul Calvez (Professeur des Universités, Polytech'Nantes, rapporteur), Frédéric Blanc (Docteur Ingénieur, CEA-LIST, examinateur), Eric Martin (Professeur des Universités, Université de Bretagne Sud, directeur de thèse), Philippe Coussy (Maître de Conférences, Université de Bretagne Sud, examinateur).

Mention très honorable

Situation actuelle : Ingénieur R&D, CEA LIST, Saclay

- Cyrille Chavet 2004-2007 – CIFRE STMicroelectronics

Synthèse automatique d'interfaces de communication matérielles pour la conception d'applications du domaine du traitement du signal

Thèse de doctorat soutenue le 26 octobre 2007

En co-encadrement avec le Pr. Eric Martin (50%)

Résumé : *La complexité croissante des algorithmes de traitement du signal et de l'image (TDSI) implémentés, et l'augmentation continue des volumes de données et des débits applicatifs, requièrent souvent la conception d'accélérateurs matériels dédiés. Typiquement, l'architecture d'un tel composant utilise des éléments de calculs de plus en plus complexes, des mémoires et des modules de brassage de données*

(entrelaceur/désentrelaceur pour les Turbo-Codes, blocs de redondance spatio-temporelle dans les systèmes OFDM/MIMO, ...), privilégie des connexions point à point pour la communication inter éléments de calcul et demande d'intégrer dans une même architecture plusieurs configurations et/ou algorithmes (systèmes (re)configurables). Aujourd'hui, le coût de ces systèmes en termes d'éléments mémorisant est très élevé; les concepteurs cherchent donc à minimiser la taille de ces tampons afin de réduire la consommation et la surface total du circuit, tout en cherchant à en optimiser les performances.

Les travaux de cette thèse ont abouti à une méthodologie de conception permettant de générer automatiquement un adaptateur de communication (interface) nommé Space-Time Adapter (STAR). Le flot de conception prend en entrée (1) des diagrammes temporels (fichier de contraintes) ou (2) une description en langage C de la règle de brassage des données (par exemple une règle d'entrelacement pour Turbo-Codes) et des contraintes utilisateur (débit, latence, parallélisme...) ou (3) en ensemble de CDFGs ordonnés et assignés. Ce flot formalise ensuite ces contraintes de communication sous la forme d'un Graphe de Compatibilité des Ressources Multi-Modes (MMRCG) qui permet une exploration efficace de l'espace des solutions architecturales afin de générer un composant STAR en VHDL de niveau transfert de registre (RTL) utilisé pour la synthèse logique. L'architecture STAR se compose d'un chemin de données (utilisant des FIFOs, des LIFOs et/ou des registres) et de machines d'état finis permettant de contrôler le système.

Les travaux ont été utilisés et validés dans des applications industrielles sur trois familles de problèmes (1) interfaces pour l'intégration de composants virtuels (IP cores), (2) composants de brassage de données (type entrelaceur) pouvant avoir plusieurs modes de fonctionnements, et (3) chemins de données, potentiellement configurables, dans des flots de synthèse de haut niveau.

Composition du jury : Michel Paindavoine (Professeur des Universités, Université de Bourgogne , Président), Daniel Etiemble (Professeur des Universités, Université de Paris Sud, Rapporteur), Yves Mathieu (Professeur HDR, ENST Paris, rapporteur), Eric Martin (Professeur des Universités, Université de Bretagne Sud, directeur de thèse), Philippe Coussy (Maître de Conférences, Université de Bretagne Sud, examinateur), Pascal Urard (Ingénieur de recherche, STMicroelectronics, examinateur), Thomas Bollaert (Ingénieur de recherche, Mentor Graphics, invité).

Mention très honorable

Situation actuelle : Maître de conférences, Université de Bretagne-Sud

- Caaliph Andriamisaina 2005-2008 – Contrat

Flot de conception dédié aux architectures multi-modes pour les applications de traitement du signal et de l'image

Thèse de doctorat soutenue le 26 novembre 2008

En co-encadrement avec le Pr. Emmanuel Casseau Université Rennes 1 (50%)

Résumé : *Le marché des systèmes portatifs ne cessent de s'accroître et de s'ouvrir aux domaines des télécommunications et du multimédia impliquant, de ce fait, une demande de plus en plus forte en fonctionnalité, performance, durée d'autonomie, compacité et faible coût de conception. Ces besoins ont ainsi conduit à des recherches de dispositifs performants, flexibles, compacts et faible consommation. Comme technologie pour réaliser ces dispositifs, les concepteurs peuvent opter pour une solution « tout logiciel », « tout matériel » ou « mixte ». Les solutions « tout matériel » ou « mixte » reposent sur les architectures dédiées (ASIC) et les architectures reconfigurables (FPGAs) qui peuvent être partiellement et dynamiquement reconfigurées. Ces dernières, qui ont connu récemment une forte évolution, sont très flexibles mais ne fournissent actuellement pas encore, dans certains cas, une performance suffisante (débit, surface et/ou consommation, temps de*

reconfiguration). Les architectures dédiées (ASICs) sont quant à elles plus appropriées pour fournir une performance élevée (fort débit, compacte et faible consommation), mais n'offrent par contre aucune flexibilité.

Les travaux de cette thèse ont abouti à une méthodologie de conception automatisant la génération d'architectures multi-modes. Ce type d'architectures est spécialement conçu pour implémenter un ensemble d'applications mutuellement exclusives dans le temps. Le flot de conception prend en entrée plusieurs graphes flots de données (DFGs) représentant les différentes applications à implémenter. Ces DFGs sont ensuite ordonnancés sous contraintes de débit puis sont assignés. Dans le but de réduire la complexité des registres, des multiplexeurs et du contrôleur engendrée par le partage des ressources entre les différentes applications, un algorithme d'ordonnancement conjoint qui maximise la similarité entre les étapes de contrôle et des algorithmes d'assignation des opérations et des registres qui maximise la similarité entre les chemins de données a été proposé. Ces algorithmes sont intégrés dans un environnement de conception, dédié à la génération d'architecture multi-modes, qui s'articule autour de trois outils dont GAUT « Multi-modes », CDFG2UCOM et STARGene « Multi-modes ».

Composition du jury : Tanguy Risset (Professeur des Universités, ENS Lyon, Président), Frédéric Pétrot (Professeur des Universités, INP Grenoble, rapporteur), Patrice Quinton (Professeur des Universités, Université de Rennes 1, rapporteur), Eric Martin (Professeur des Universités, Université de Bretagne Sud, Examineur), Emmanuel Casseau (Professeur des Universités, Université de Rennes 1, directeur de thèse), Philippe Coussy (Maître de Conférences, Université de Bretagne Sud, examinateur), Franck Mamalet (Ingénieur de recherche, France Telecom, invité).

Mention très honorable

Situation actuelle : Ingénieur R&D, CEA LIST, Saclay

- Kods Trabelsi 2006-2009 – Bourse MESR

Méthodes d'optimisation pour la conception sous contraintes de systèmes et de circuits électroniques

Thèse de doctorat soutenue le 26 novembre 2008

En co-encadrement avec le Pr. Marc Sevaux (50%)

Résumé : Les circuits programmables ou FPGA (pour Field Programmable Gate Array) sont des circuits intégrés qui sont configurés après fabrication. Il s'agit d'une solution moins performante que les circuits construits spécifiquement pour une application (ou ASIC), mais à un coût de fabrication très inférieur. Pour configurer efficacement un FPGA, les industriels utilisent des outils de synthèse de haut niveau (ou HLS pour High Level Synthesis) qui convertissent une description comportementale d'un circuit électronique, exprimée dans un langage du type VHDL ou SystemC en une description logique sous-forme de portes. Les premiers outils commerciaux de HLS sont apparus dans les années quatre-vingt et ont connu depuis un essor considérable compte tenu de la concurrence importante entre les différents constructeurs et du développement industriel de ces systèmes. Le nombre de portes disponibles sur un FPGA est aujourd'hui de l'ordre du million. Dans ce contexte, la résolution des problèmes d'optimisation combinatoire requiert le développement de méthodes spécifiques pour obtenir le plus rapidement possible des solutions satisfaisantes.

Les travaux de cette thèse ont porté sur la proposition de plusieurs algorithmes pour résoudre un problème d'ordonnancement et d'allocation des ressources dans ce contexte. Différentes approches permettant l'exploration de l'espace de conception ont été proposées afin de minimiser la surface des circuits. Ces approches appartiennent à la famille des méthodes à base de voisinages. Elles s'appuient sur des techniques permettant de passer

d'une solution à une solution voisine par déplacements successifs pour choisir la solution de meilleure qualité. Les algorithmes développés ont été intégrés dans l'outil de synthèse de haut niveau GAUT.

Composition du jury : Alix Munier (Professeur des Universités, Université Paris 6, Président, rapporteur), Emmanuel Casseau (Professeur des Universités, Université de Rennes 1, rapporteur), Kenneth Sörensen (Professeur, Université d'Anvers, Belgique, examinateur), Marc Sevaux (Professeur des Universités, Université de Bretagne Sud, directeur de thèse), Philippe Coussy (Maître de Conférences, Université de Bretagne Sud, examinateur), André Rossi (Maître de Conférences, Université de Bretagne Sud, examinateur).

Mention très honorable

Situation actuelle : Ingénieur R&D, CEA LIST, Saclay

- Ghizlane Lebreton-Lhairech 2007-2011 – Contrat

Synthèse de haut niveau pour la conception faible consommation sur FPGA

Soutenance prévue en décembre 2011 (en cours de rédaction)

En co-encadrement avec le Pr. Eric Martin (50%)

- Awais Sani Hussein 2008-2011 – MESR

Méthodes de conception pour la génération d'entrelaceurs parallèles sans conflit pour les applications de codes correcteurs d'erreurs

Soutenance prévue en novembre 2011 (en cours de rédaction)

En co-encadrement avec le Pr. Eric Martin (50%)

- Aroua Briki 2009-2012 – MESR

Synthèse d'architecture d'unité mémoire complexe

Soutenance prévue en 2012

En co-encadrement avec le Pr. Eric Martin (30%) et Cyrille Chavet (40%)

- Paolo Burgio 2011-2013 – Contrat, cotutelle Université de Bologne

Utilisation de la mémoire partagée dans le contexte d'un processor multi-cœurs embarqué : exploration de la technologie et ses limites

Soutenance prévue en 2013

En co-direction avec le Pr. Luca Benini (50%)

La Table 1 ci-dessous résume l'ensemble des thèses réalisées en co-encadrement. Les thèses représentées par des blocs en pointillé démarreront après septembre 2011 et ne sont pas décrites dans ce document. Différentes sources de financement ont été utilisées afin de financer les doctorants (Contrat RNTL, ANR, FUI, bourse MESR, CIFRE).

	2002 / 2003	2003 / 2004	2004 / 2005	2005 / 2006	2006 / 2007	2007 / 2008	2008 / 2009	2009 / 2010	2010 / 2011	2011 / 2012	2012 / 2013	2013 / 2014
Farhat Thabet Contrat RNTL			Modélisation Comportementale Simulation & Synthèse			Ingénieur R&D CEA						
Cyrille Chavet CIFRE STMicroelectronics			Synthèse adaptateur spatio temporel /			Maitre de conférences						
Caaliph Andriamisaina CRE / Contrat ANR				Synthèse haut-niveau multi-mode			Ingénieur R&D CEA					
Kods Trabelsi Bourse MESR				Métaheuristique & synthèse haut niveau			Ingénieur R&D CEA					
Ghizlane Lebreton CRE / Contrat ANR						Synthèse haut niveau hiérarchique multi-horloge						
Awais Sani Hussein Bourse MESR						Conception entrela- ceurs parallèle						
Aroua Briki Bourse MESR						Synthèse architecture unité mémoire						
Paolo Burgio Contrat cotutelle Univ-Bologne						Hiérarchie mémoire Système many core						
Saeed Ur Rehman Allocation HEC, Pakistan						Co-direction M. Jezequel TelecomB		Flexible hw architecture Turbo-like decoders				
Thomas Peyret Cifre CEA LIST						Co-direction G. Corre CEA		Architecture FPGA tolérante aux fautes				
Mohamed Ben Hammouda Allocation MESR						Co-direction L. Lagadec UBO		Synthèse d'architecture et deverminage				

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

2.4.2. Encadrement de post-doctorants

Jorgiano Vidal – Contrat

Conception et réalisation d'un front-end et d'un modèle de représentation interne générique pour la synthèse de haut niveau

Juillet 2010-Juillet 2011

Caaliph Andriamisaina – Contrat

Génération de modèles de simulation par la synthèse de haut niveau pour le prototypage virtuel d'application du domaine du TDSI

Novembre 2008-Juillet 2009

2.4.3. Encadrement d'ingénieurs

LinFeng YE – Contrat

Réalisation d'un générateur d'entrelaceur parallèle

Janvier 2011-Juillet 2011

2.4.4. Encadrement de stagiaires de DEA et de Master recherche

J'ai encadré neuf étudiants en DEA ou en Master recherche sur des sujets relatifs à la modélisation comportementale, la synthèse de haut niveau et le prototypage virtuel.

Michelle Furtado Pinheiro Do Carmo

Prototypage virtuel d'une décodeur MP3 avec la plateforme SocLib

Master recherche I-MARS, année 2009/2010

Situation actuelle : sans emploi

Vianney Lapotre

Introduction de la prédiction de branchement dans la synthèse de haut niveau

Master recherche I-MARS, année 2009/2010

Situation actuelle : Doctorant à l'Université de Bretagne-Sud (Lorient, dep. 56)

Aroua Briki

Synthèse de haut niveau hiérarchique : étude de cas d'un encodeur de Reed Solomon

Master recherche I-MARS, année 2008/2009

Situation actuelle : Doctorante à l'Université de Bretagne-Sud (Lorient, dep. 56)

Hicham Lalaoui Hassani

Impact des étapes de synthèse de haut niveau dans l'environnement GAUT

Master recherche I-MARS, année 2007/2008

Situation actuelle : Enseignant à l'IGA, Casablanca Maroc

Moahamed Aabidi

Synthèse de haut-niveau : Etude de cas d'un algorithme Maximum A Posteriori pour la conception d'un Turbo décodeur

Master recherche I-MARS, année 2007/2008

Situation actuelle : inconnue

Ghizlane Lebreton

Prise en compte de la dynamique des données dans l'outil de synthèse haut niveau GAUT

Master recherche I-MARS, année 2006/2007

Situation actuelle : Doctorante à l'Université de Bretagne-Sud (Lorient, dep. 56)

Youcef Mekla

Utilisation de la synthèse comportementale sous contrainte d'entrées/sorties pour la conception faible consommation de composants matériels du TDSI

Master recherche I-MARS, année 2005/2006

Situation actuelle : Ingénieur à ACAMAS Electronique (Les Clayes sous Bois, dep. 78)

Sebastien Tregarot

Modélisation et simulation d'une application MIMO dans une plate-forme SoC décrite en langage de haut-niveau SystemC

Master recherche I-MARS, année 2004/2005

Situation actuelle : Ingénieur Docteur CAMKA System (Queven, dep. 56)

Jean-Baptiste Le Goff

Raffinement des communications en vue de la simulation et de la synthèse de composants virtuels algorithmiques

Master recherche I-MARS, année 2003/2004

Situation actuelle : Ingénieur à Atos Origin France (Rennes, dep. 35)

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

	2003	2004	2005	2006	2007	2008	2009	2010	2011
Jean-Baptiste Le Goff		Simulation / Modélisation	Ingénieur						
Sebastien Tregarot			Prototypage virtuel	Ingénieur docteur					
Youcef Mekla				Synthèse faible conso.	Ingénieur				
Ghizlane Lebreton					Synthèse bit près	Doctorante			
Moahamed Aabidi						Synthèse case study			
Hicham Lalaoui						Synthèse optimisation	Enseignant		
Aroua Briki						Synthèse hiérarchique	Doctorante		
Vianney Lapotre							Synthèse contrôle	Doctorant	
Michelle Furtado							Prototypage virtuel		

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

2.5. Responsabilités scientifiques

Cette partie concerne les responsabilités scientifiques, à savoir :

- Participation à des jurys de thèse
- Direction / édition d'ouvrages scientifiques
- Participation à des comités techniques internationaux
- Participation à des comités de programmes de conférences
- Participation à des comités de lecture de conférences
- Participation à des comités de lecture de journaux nationaux et internationaux
- Participation à des comités d'organisation
- Expertise scientifique

2.5.1. Participation à des jurys de thèse

J'ai participé à neuf jurys de thèse en tant qu'examinateur : cinq à l'Université de Bretagne Sud, deux à l'Université de Rennes 1, un à l'INSA de Toulouse et un à l'INP Grenoble.

Farhat Thabet

Modélisation Comportementale Unifiée pour la Simulation et la Synthèse Haut-Niveau de composants Virtuels Algorithmiques

Thèse de Doctorat soutenue le 4 mai 2007, Université de Bretagne-Sud - Lorient

Composition du jury : *Patrice Quinton (Professeur des Universités, Université de Rennes 1, président), Michel Auguin (Directeur de Recherche CNRS, I3S Nice – Sophia Antipolis, rapporteur), Jean-Paul Calvez (Professeur des Universités, Polytech'Nantes, rapporteur), Frédéric Blanc (Docteur Ingénieur, CEA-LIST, examinateur), Eric Martin (Professeur des Universités, Université de Bretagne Sud, directeur de thèse), Philippe Coussy (Maître de Conférences, Université de Bretagne Sud, examinateur).*

Mention très honorable

Fatma Abbes Ben Amor

Encapsulation des composants virtuels dans un système sur puce

Thèse de Doctorat soutenue le 14 mai 2007, Université de Bretagne-Sud - Lorient

Composition du jury : *Eric Martin (Professeur des Universités, Université de Bretagne Sud, président), Dominique Houzet (Professeur des Universités, INP Grenoble, rapporteur), Habib Youssef (Professeur des Universités, Faculté des Sciences de Monastir, rapporteur), Jean-Luc Philippe (Professeur des Universités, Université de Bretagne Sud, examinateur), Philippe Coussy (Maître de Conférences, Université de Bretagne Sud, examinateur), Emmanuel Casseau (Professeur des Universités, Université de Bretagne Sud, directeur de thèse), Mohamed Abid (Maître de Conférences, Faculté des Sciences de Monastir, examinateur).*

Mention très honorable

Cyrille Chavet

Synthèse automatique d'interfaces de communication matérielles pour la conception d'applications du domaine du traitement du signal

Thèse de Doctorat soutenue le 26 octobre 2007, Université de Bretagne-Sud - Lorient

Composition du jury : *Michel Paindavoine (Professeur des Universités, Université de Bourgogne, président), Daniel Etiemble (Professeur des Universités, Université de Paris Sud, rapporteur), Yves Mathieu (Professeur HDR, ENST Paris, rapporteur), Eric Martin (Professeur des Universités, Université de Bretagne Sud, directeur de thèse), Philippe Coussy (Maître de Conférences, Université de Bretagne Sud, examinateur), Pascal Urard (Ingénieur de recherche, STMicroelectronics, examinateur), Thomas Bollaert (Ingénieur de recherche, Mentor Graphics, invité).*

Mention très honorable

Caaliph Andriamisaina

Flot de conception dédié aux architectures multi-modes pour les applications de traitement du signal et de l'image

Thèse de Doctorat soutenue le 26 novembre 2008, Université de Bretagne-Sud - Lorient

Composition du jury : *Tanguy Risset (Professeur des Universités, ENS Lyon, président), Frédéric Pétrot (Professeur des Universités, INP Grenoble, rapporteur), Patrice Quinton (Professeur des Universités, Université de Rennes 1, rapporteur), Eric Martin (Professeur des Universités, Université de Bretagne Sud, examinateur), Emmanuel Casseau (Professeur des Universités, Université de Rennes 1, directeur de thèse), Philippe Coussy (Maître de Conférences, Université de Bretagne Sud, examinateur), Franck Mamalet (Ingénieur de recherche, France Telecom, invité).*

Mention très honorable

Kods Trabelsi

Méthodes d'optimisation pour la conception sous contraintes de systèmes et de circuits électroniques

Thèse de Doctorat soutenue le 26 novembre 2008, Université de Bretagne-Sud - Lorient

Composition du jury : *Alix Munier (Professeur des Universités, Université Paris 6, Président - rapporteur), Emmanuel Casseau (Professeur des Universités, Université de Rennes 1, rapporteur), Kenneth Sörensen (Professeur, Université d'Anvers, Belgique, examinateur), Marc Sevaux (Professeur des Universités, Université de Bretagne Sud, directeur de thèse), Philippe Coussy (Maître de Conférences, Université de Bretagne Sud, examinateur), André Rossi (Maître de Conférences, Université de Bretagne Sud, examinateur).*

Mention très honorable

Kevin Martin

Génération Automatique d'extensions de jeux d'instructions de processeurs

Thèse de Doctorat soutenue le 7 septembre 2010, Université de Rennes 1 - Rennes

Composition du jury : *Olivier Déforges (Professeur des Universités, INSA Rennes, président), Daniel Etiemble (Professeur des Universités, Université de Paris Sud, rapporteur), Frédéric Pétrot (Professeur des Universités, INP Grenoble, rapporteur), Philippe Coussy (Maître de Conférences, Université de Bretagne Sud, examinateur), Christophe Wolinski (Professeur des Universités, Université de Rennes 1, directeur de thèse), Francois Charot (Chargé de Recherche, INRIA, examinateur).*

Mention très honorable

Shafqat Khan

Développement d'architectures matérielles hautes performances pour les applications multimédia

Thèse de Doctorat soutenue le 29 septembre 2010, ENSSAT, Lannion

Composition du jury : *Patrice Quinton (Professeur des Universités, Université de Rennes 1, président), Michel Auguin (Directeur de Recherche CNRS, I3S Nice – Sophia Antipolis, rapporteur), Christophe Jego (Professeur des Universités, Université de Bordeaux, rapporteur), Philippe Coussy (Maître de Conférences, Université de Bretagne Sud, examinateur), Emmanuel Casseau (Professeur des Universités, Université de Rennes 1, directeur de thèse), Daniel Ménard (Maître de Conférences, Université de Rennes 1, examinateur).*

Mention très honorable

Mario Alberto Ibarra Manzano

Vision multi-caméras pour la détection d'obstacles sur un robot de service : des algorithmes à un système intégré

Thèse de Doctorat soutenue le 6 janvier 2011, LAAS, Toulouse

Composition du jury : *Michel Devy (Directeur de recherche, LAAS, président), Mohamed Akil (Professeur des Universités, Groupe ESIEE-Paris, rapporteur), Dominique Ginjac (Professeur des Universités, Université de Bourgogne, rapporteur), Philippe Coussy (Maître de Conférences, Université de Bretagne Sud, examinateur), Jean-Yves Fourniols, (Professeur des Universités, INSA Toulouse, directeur de thèse), Jean-Louis Boizard, (Maître de Conférences, Université Toulouse 2, examinateur).*

Mention très honorable

Luca Ferro

Vérification de propriétés logico-temporelles de spécifications SystemC TLM

Thèse de Doctorat soutenue le 11 juillet 2011, Institut National Polytechnique, Grenoble

Composition du jury : *Lionel Torres (Professeur des Universités, Université de Montpellier 2, rapporteur), El Mostapha Aboulhamid (Professeur des Universités, Université de Montréal Canada, rapporteur), Philippe Coussy (Maître de Conférences, Université de Bretagne Sud, examinateur), Laurent Maillet-Contoz (Docteur Ingénieur, STMicroelectronics, examinateur), Renaud Pacalet (Professeur, ParisTech, examinateur), Laurence Pierre (Professeur des Universités, INP Grenoble, directrice de thèse).*

2.5.2. Direction / édition d'ouvrages scientifiques

- Editeur invité du Journal of Electrical and Computer Engineering, Special Issue on Electronic System Level Design Methodology (ESL), Hindawi, January 2012
- Editeur invité du magazine IEEE Design and Test of Computer, Special Issue on High-Level Synthesis, July-August 2009
- Co-éditeur du livre "High-Level Synthesis: From Algorithm to Digital Circuit", P. Coussy and A. Morawiec, Springer, Berlin, Germany, 2008

2.5.3. Participation à des comités techniques internationaux

- Membre élu au comité technique de l'IEEE Signal Processing Society, Design and Implementation of Signal Processing Systems (DISPS) (2011 – 2014)
- Membre du comité de rédaction de la roadmap HiPEAC, 2011
- Membre du comité de rédaction de l'European Electronic Design Automation EDA Roadmap, Medea+/Catrene, 2009
- Membre du Synthesis Working Group (SWG) de de l'Open SystemC Initiative OSCI (Key Contributor 2009, Affiliate member 2010)

2.5.4. Participation à des comités de programmes de conférences

Internationales

Co-Président

- IEEE Design and Test in Europe DATE, Topic Architectural and High-Level Synthesis (2010, 2011, 2012)
- International ESL Synthesis Conference ESLsyn, in conjunction with DAC, 2011

Membre

- IEEE International Conference on Application-specific Systems, Architectures and Processors ASAP (2011)
- IEEE International Symposium on Circuits and Systems ISCAS (2005, 2006, 2007, 2008, 2009, 2010, 2011)
- IEEE International Great Lake Symposium on VLSI GLSVLSI (2007, 2008, 2009, 2010, 2011)
- IEEE Workshop on Signal Processing Systems SiPS (2011)
- ERDIAP workshop (Exploiting Regularity in the Design of IPs, Architectures and Platforms), in conjunction with ARCS (2011).

Nationales

- Membre du comité de programme du “Symposium en Architecture” SYMPA, Saint-Malo, France, 10-14 mai 2011
- Membre du comité de programme des Doctoriales de Bretagne 2002

2.5.5. Participation à des comités de lecture de conférences

Internationales

- IEEE Design Automation and Test in Europe DATE (2010, 2011, 2012)
- IEEE International Conference on Application-specific Systems, Architectures and Processors ASAP (2011)
- IEEE International Symposium on Circuits and Systems ISCAS (2005, 2006, 2007, 2008, 2009, 2010, 2011)
- IEEE International Great Lake Symposium on VLSI GLSVLSI (2007, 2008, 2009, 2010, 2011)
- IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP, (2011)
- IEEE Workshop on Signal Processing Systems Design and Implementation SIPS, (2011)
- IEEE International Conference on Electronics, Circuits, and Systems ICECS (2010)
- IEEE International Design Automation Conference DAC (2005)

Nationales

- Colloque sur le Traitement du Signal et de l'Image GRETSI (2011)
- Symposium en Architecture SYMPA (2011)
- MANifestation des Jeunes Chercheurs STIC MajecSTIC (2006, 2007)
- Journée Francophone sur l'Adéquation Algorithme/Architecture JFAAA (2002)

2.5.6. Participation à des comités de lecture de journaux internationaux

- IEEE Design and Test of Computers
- IEEE Transactions on Computer Aided-Design of Integrated Circuits and Systems
- IEEE Journal on Emerging and Selected Topics in Circuits and Systems (JETCAS)
- ACM Transactions on Design Automation of Electronics Systems (ACM TODAES)
- ACM Transactions on Embedded Computing System (ACM TECS)
- Integration, the VLSI Journal - Elsevier Science
- Journal of Embedded Computing (JEC)
- International Journal of Circuits, Systems, and Computers (JCSC)
- Journal on Embedded Systems, EURASIP (JES)
- Embedded Hardware Design (Microprocessors and Microsystems MICPRO) – Elsevier
- Microelectronics Journal – Elsevier

2.5.7. Participation à des comités d'organisation

Internationaux

- Co-organisateur du Tutorial “An Introduction to the SystemC Synthesis Subset Standard”, IEEE Embedded System Week (Codes+ISSS, CASES, EMSOFT), Scottsdale, USA, October 24-29 2010
- Co-organisateur du Tutorial on ESL Design and Virtual Prototyping of MPSOCs, IEEE Design Automation Conference DAC Anaheim, USA, June 13-18, 2009
- Co-organisateur du Tutorial on High-Level Synthesis, IEEE Design Automation Conference DAC San Francisco, USA, July 26-31, 2009
- Co-organisateur du workshop international “High-Level Synthesis: Next Step to Efficient ESL Design”, IEEE Asia and South Pacific Design Automation Conference ASP-DAC, Yokohama, Japan, January 19-22, 2009.
- Co-organisateur du workshop international “High-Level Synthesis: Back to the Future”, IEEE Design Automation Conference DAC Anaheim, USA, June 8-13, 2008.
- Co-organisateur du workshop international “The New Wave of the High-Level Synthesis”, IEEE Design and Test in Europe DATE Munich, Germany, March 10-14, 2008.
- Co-organisateur du workshop international ECSI-UBS « High-Level Synthesis », (in conjunction with the Forum on Design Language FDL), Darmstadt, Germany, 2006 September 18.

Nationaux

- Co-organisateur du Tutorial “Prototypage virtuel de plates-formes MPSoC”, Saint-Malo, France, 10 mai 2011
- Co-organisateur de la journée “Recontre de la Recherche en Electronique”, Lorient, France, 12 octobre 2010
- Co-organisateur de la journée “Systèmes Embarqués pour la Santé”, (journée inter GDR STIC-Santé thème D, SoC-SiP et ISIS thème C), Paris, France, 8 juillet 2010

2.5.8. Modération dans des conférences internationales

Session

- “Implementation techniques for Audio, Video and FFTs” (ICASSP 2011)
- “High-Level Synthesis” (DATE 2011)
- “High-Level and architectural Synthesis” (DATE 2010)
- “Logic & High-level Synthesis” (ISCAS 2010)
- “ASIP/ASIC” (GLSVLSI 2007)
- “Emerging Areas in CAD, Part II” (ISCAS 2005)

Tutoriel

- “ESL Design and Virtual Prototyping of MPSOCs” (DAC 2010)
- “High-Level Synthesis for ESL Design: Fundamentals and Case Studies” (DAC 2009)

Workshop

- “High-Level Synthesis: Back to the Future” (DAC 2008)
- “The New Wave of the High-Level Synthesis” (DATE 2008)
- “High-Level Synthesis” (FDL 2006)
- “High-Level Synthesis: Next Step to Efficient ESL Design” (ASP-DAC 2009)

2.5.9. Expertise scientifique

- Agence Nationale de la Recherche ANR, programme « Ingénierie Numérique & Sécurité » - INS 2011
- Ministère de l'Enseignement Supérieur et de la Recherche – Expert scientifique externe pour le Crédit Impôt Recherche CIR et les Jeunes Entreprises Innovantes JEI (de 2010 à aujourd'hui)

2.6. Diffusion des connaissances et publications scientifiques

Mes activités de recherche ont donné lieu à un certain nombre de publications scientifiques dont la liste est répertoriée dans les sous-sections suivantes.

2.6.1. Thèse de doctorat et rapport de DEA

[COU 2003a] P. Coussy, **Synthèse d'Interface de Communication pour les Composants Virtuels**, Thèse de doctorat, Université de Bretagne Sud, décembre 2003.

[COU 1999a] P. Coussy, **“DCT/Q and IQ/IDCT blocks for MPEG encoding”**, Rapport de DEA, Université Pierre et Marie Curie, Paris 6, Septembre 1999.

2.6.2. Brevets

[CHA 2009a] Cyrille Chavet, Philippe Coussy, Pascal Urard, Eric Martin, **“Apparatus for data interleaving algorithm”**, *CNRS – STMicroelectronics U.S. Patent application 20090031094*, January 29, 2009.

[CHA 2008a] Cyrille Chavet, Philippe Coussy, Pascal Urard, Eric Martin, **« Procédé et dispositif d'entrelacement de données »**, *CNRS – STMicroelectronics, Brevet Français n° 0754793 10*, 30 Avril 2007.

2.6.3. Articles dans des revues internationales avec comité de lecture

[AND 2010a] C. Andriamisaina, P. Coussy, E. Casseau, C. Chavet, **“High-Level Synthesis for Designing Multi-mode Architectures”**, *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems (TCAD)*, Vol. 29, Issue 11, pp. 1736-1749, November 2010.

[COU 2009a] P. Coussy, G. Gajski, A. Takach, M. Meredith, **“An Introduction to High-Level Synthesis”**, *Special issue on High-Level Synthesis, IEEE Design and Test of Computers*, Vol. 26, Issue 4, July/August, 2009.

[THA 2009a] F. Thabet, P. Coussy, D. Heller, E. Martin, **“Exploration and Rapid Prototyping of DSP Applications using SystemC Behavioral Simulation and High-Level Synthesis”**, *Journal of Signal Processing Systems, Springer*, Issue 56, 2-3, September 2009.

[COU 2008a] P. Coussy, G. Le Breton, D. Heller, **“Multiple Word-Length High-Level Synthesis”**, *EURASIP Journal on Embedded Systems*, July, 2008.

[COU 2007a] P. Coussy, E. Casseau, P. Bomel, A. Baganne, E. Martin, “**Constrained algorithmic IP design for system-on-chip**”, *Integration, the VLSI Journal, Elsevier Science*, Vol. 40, Issue 2, February 2007.

[COU 2006a] P. Coussy, E. Casseau, P. Bomel, A. Baganne, E. Martin, “**A Formal Method for Hardware IP Design and Integration under I/O and Timing Constraints**”, *ACM Transactions on Embedded Computing Systems*, Vol 5, No. 1, pp. 29-53, 2006.

2.6.4. Articles dans des revues internationales sans comité de lecture

[COU 2009b] P. Coussy, A. Takach, “**Raising the Abstraction Level of Hardware Design**”, *Special issue on High-Level Synthesis, IEEE Design and Test of Computers*, Vol. 26, Issue 4, July/August, 2009.

2.6.5. Chapitres de livre

[COU 2008b] P. Coussy, C. Chavet, P. Bomel, D. Heller, E. Senn, E. Martin, “**GAUT: A High-Level Synthesis Tool for DSP applications**”, *High-Level Synthesis: From Algorithm to Digital Circuits*, Springer, Berlin, Germany, 2008.

[TRA 2010a] K. Trabelsi, M. Sevaux, P. Coussy, A. Rossi, K. Sörensen, “**Advanced Metaheuristics for High-Level Synthesis**”, *In Metaheuristics. Springer*, 2010.

2.6.6. Communications internationales avec comité de lecture et actes

[SAN 2011a] A. Sani, P. Coussy, C. Chavet, E. Martin, “**A Methodology based on Transportation Problem Modeling for Designing Parallel Interleaver Architectures**”, *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011.

[SAN 2011b] A. Sani, P. Coussy, C. Chavet, E. Martin, “**An Approach Based on Edge Coloring of Tripartite Graph for Designing Parallel LDPC Interleaver Architecture**”, *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2011.

[SAN 2010a] A. Sani, P. Coussy, C. Chavet, E. Martin, “**Design of Parallel LDPC Interleaver Architecture: A Bipartite Edge Coloring Approach**”, *IEEE International Conference on Electronics, Circuits, and Systems (ICECS)*, 2010.

[LHA 2010a] G. Lhairech-Lebreton, P. Coussy, D. Heller, E. Martin, “**Bitwidth-Aware High-Level Synthesis for Designing Low-Power DSP Applications**”, *IEEE International Conference on Electronics, Circuits, and Systems (ICECS)*, 2010.

[LEF 2010a] Vincent Lefftz, Jean Bertrand, Hugues Cassé, Christophe Clienti, Philippe Coussy, Laurent Maillet-Contoz, Philippe Mercier, Pierre Moreau, Laurence Pierre, Emmanuel Vaumorin, “**A Design Flow for Critical Embedded Systems**”, *IEEE Symposium on Industrial Embedded Systems (SIES)*, 2010.

- [LHA 2010b] G. Lhairech-Lebreton, P. Coussy, E. Martin, “**Hierarchical and Multiple-Clock Domain High-Level Synthesis for Low-Power Design on FPGA**”, *IEEE International Conference on Field Programmable Logic and Applications (FPL)*, 2010.
- [CHA 2010a] C. Chavet, P. Coussy, P. Urard, E. Martin, “**Static Address Generation Easing: a Design Methodology for Parallel Interleaver Architectures**”, *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2010.
- [CHA 2010b] C. Chavet, P. Coussy, “**A Memory Mapping Approach for Parallel Interleaver Design with Multiples Read and Write Accesses**”, *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2010.
- [COU 2009c] P. Coussy, A. Rossi, M. Sevaux, K. Sørensen, and K. Trabelsi, “**VNS for High Level Synthesis**”, *In Proceedings of 8th Metaheuristics International Conference, MIC 2009*, July 2009.
- [CHA 2007a] C. Chavet, C. Andriamisaina, P. Coussy, E. Casseau, E. Juin, P. Urard, E. Martin, “**A Design Flow Dedicated to Multi-mode Architectures for DSP Applications**”, *IEEE International Conference on Computer Aided Design, (ICCAD)* 2007.
- [CHA 2007b] C. Chavet, P. Coussy, P. Urard, E. Martin, “**Application of a design space exploration tool to enhance interleaver generation**”, *European Signal Processing Conference (EUSIPCO)* 2007.
- [THA 2007a] F. Thabet, P. Coussy, D. Heller, E. Martin, “**Behavioral Description Model BDM for Design Space Exploration: a Case Study of His Algorithm for MC-CDMA System**”, *European Signal Processing Conference (EUSIPCO)* 2007.
- [CHA 2007c] C. Chavet, P. Coussy, P. Urard, E. Martin, “**A Methodology for Efficient Space-Time Adapter Design Space Exploration: A Case Study of an Ultra Wide Band Interleaver**”, *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2007.
- [AND 2007a] C. Andriamisaina, P. Coussy, E. Casseau, “**Synthesis of Multimode digital signal processing systems**”, *NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*, 2007.
- [CHA 2007d] C. Chavet, P. Coussy, P. Urard, E. Martin, “**A Design Methodology for Space-Time Adapter**”, *ACM Great Lakes Symposium on VLSI on VLSI (GLSVLSI)*, 2007.
- [THA 2006a] F. Thabet, P. Coussy, D. Heller, E. Martin, “**Design Space Exploration of DSP Applications Based on Behavioral Description Models**”, *IEEE Workshop on Signal Processing Systems Design and Implementation (SIPS)*, 2006.

- [KRI 2005a] L. Kriaa, S. Adriano, E. Vaumorin, R. Nouacer, F. Blanc, S. Pajaniardja, P. Coussy, E. Martin, D. Heller, F. Thabet et al, “**SystemC'mantic : A high level Modeling and Co-design Framework For Reconfigurable Real Time Systems**”, *Forum on Design Languages (FDL)*, 2005.
- [COU 2005a] P. Coussy, G. Corre, P. Bomel, E. Senn, E. Martin, “**A More Efficient and Flexible DSP Design Flow from MATLAB-SIMULINK**”, *IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, 2005.
- [COU 2005b] P. Coussy, G. Corre, P. Bomel, E. Senn, E. Martin, “**High-level synthesis under I/O Timing and Memory constraints**”, *IEEE International Symposium on Circuits And Systems (ISCAS)*, 2005.
- [COU 2004a] P. Coussy, D. Gnaëdig, A. Nafkha, A. Baganne, E. Boutillon, E. Martin, “**A Methodology for IP integration in DSP Soc: a case study of a MAP algorithm for turbo decoder**”, *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2004.
- [ABB 2004a] F. Abbes, E. Casseau, M. Abid, P. Coussy, J.-B. Legoff., “**IP integration methodology for SoC design**”, *International Conference on Microelectronics (ICM)*, 2004.
- [THA 2004a] F. Thabet, J.-B. Legoff, P. Coussy, E. Martin, “**A Methodology for Timing and Structural Communication Refinement in DSP Systems**”, *International Conference on Microelectronics (ICM)*, 2004.
- [COU 2003b] P. Coussy, A. Baganne, E. Martin, “**Communication and Timing Constraints Analysis for IP Design and Integration**”, *IFIP WG 10.5 Very Large Scale Integration of System-on-Chip Conference (VLSI-SOC)*, 2003.
- [COU 2002a] P. Coussy, A. Baganne, E. Martin, “**Virtual Component IP Re-use in Telecommunication Systems Design: A Case Study of MPEG-2 / JPEG2000 Encoder**”, *IEEE International Conference on Electronics, Circuits, and Systems (ICECS)*, 2002.
- [COU 2002b] P. Coussy, A. Baganne, E. Martin, “**IP Cores Integration in DSP System-On-Chip Designs**”, *European Signal Processing Conference (EUSIPCO)*, 2002.
- [COU 2002c] P. Coussy, A. Baganne, E. Martin, “**Platform-Based Design For Digital Signal Processing Systems: A Case Study of MPEG-2 / JPEG2000 Encoder**”, *IEEE International Conference on Communication Circuits and Systems (ICCCAS)*, 2002.
- [COU 2002d] P. Coussy, A. Baganne, E. Martin, “**A Design Methodology for IP Integration**”, *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2002.
- [COU 2002e] P. Coussy, A. Baganne, E. Martin, “**A Design Methodology for Integrating IP into SOC Systems**”, *IEEE International Custom Integrated Circuits Conference (CICC)*, 2002.

[SAV 2001a] G. Savaton, P. Coussy, E. Casseau, E. Martin, “**A Methodology for Behavioral Virtual Component Specification Targeting SoC Design with High-Level Synthesis Tools**”, *In Proc. of the Forum on Design Languages (FDL)*, 2001.

[PIL 2001a] S. Pillement, O. Sentieys, D. Chillet, E. Casseau, P. Coussy, E. Martin, G. Savaton, S. Roux, “**Design and synthesis of behavioral level virtual components**”, *IFIP WG 10.5 Very Large Scale Integration of System-on-Chip Conference (VLSI-SOC)*, 2001.

2.6.7. Communications internationales sans comité de lecture avec actes

[COU 2010a] P. Coussy, A. Takach, M. McNamara, M. Meredith, “**An Introduction to the SystemC Synthesis Subset Standard**”, *IEEE International Conference on Hardware-Software Codesign and System Synthesis (CODES+ISSS)*, 2010.

2.6.8. Communications internationales invitées sans actes

[COU 2011b] P. Coussy, panellist in “**Applying High-Level Synthesis (HLS) in an SoC Flow**”, *EE Times Virtual Event: System-on-Chip 2011*, May 12th, 2011.

[COU 2010b] P. Coussy “**An Introduction to High-Level Synthesis**”, *Embedded tutorial “An Introduction to the SystemC Synthesis Subset Standard”*, *IEEE International Conference on Hardware-Software Codesign and System Synthesis (CODES+ISSS)*, Scottsdale, USA, October 2010.

[COU 2009d] P. Coussy, “**High-Level Synthesis of dedicated coprocessors**”, *Tutorial on ESL Design and Virtual Prototyping of MPSOCs*, *IEEE Design Automation Conference DAC* Anaheim, USA, June 13th, 2009.

[COU 2008c] P. Coussy, “**Multi-Mode Architecture Design with High-Level Synthesis**”, *Workshop on High-level synthesis*, *IEEE Design Automation and Test in Europe DATE*, March 2008.

[COU 2006b] P. Coussy, “**GAUT: A High-Level Synthesis Tool**”, *Workshop on High-level synthesis*, *ECSI Institute & UBS Workshop*, September 18th, 2006.

2.6.9. Communications francophones avec comité de lecture et actes

[LHA 2011a] G. G. Lhairech-Lebreton, P. Coussy, E. Martin, “**Synthèse d’Architecture Multi-horloges pour la Conception Faible Consommation sur FPGA**”, *Colloque sur le Traitement du Signal et de l’Image (GRETSI)*, Septembre 2011.

[LAP 2011a] V. Lapotre, P. Coussy, C. Chavet, “**Prédiction de Branchement dans la Synthèse de Haut Niveau**”, *14^{ième} SYMPosium en Architecture (SYMPA)*, 10-13 Mai 2011.

[TRA 2008a] K. Trabelsi, P. Coussy, A. Rossi, M. Sevaux, “**Ordonnement et Assignation en Synthèse de Haut Niveau**”, *9ième congrès de la Société Française de Recherche Opérationnelle et d’Aide à la Décision (ROADEF)* Février, 2008.

- [LAU 2007a] J. Laurent, P. Coussy, **“Impact du type d'architecture sur la consommation d'une application”**, *Journées Faible Tension Faible consommation (FTFC)*, 21-23 mai, 2007.
- [THA 2005a] F. Thabet, P. Coussy, E. Martin, **“Approche Automatique pour le Raffinement des Communications”**, *Journées Francophones sur l'Adéquation Algorithme/Architecture (JFAAA)*, 2005.
- [COU 2005c] P. Coussy, G. Corre, P. Bomel, E. Senn, E. Martin, **“Synthèse Comportementale Sous Contraintes de Communication et de Placement Mémoire pour les composants du TDSI”**, *Colloque sur le Traitement du Signal et de l'Image (GRETSI)*, 2005.
- [CHA 2005a] C. Chavet, P. Coussy, P. Urard, E. Martin, **“Méthodologie de modélisation et d'implémentation d'adaptateurs spatio-temporels”**, *MANifestation des Jeunes Chercheurs STIC (MajecSTIC)*, 2005.
- [COU 2004b] P. Coussy, A. Baganne, E. Martin, E. Casseau, **“Intégration Optimisée de Composants Virtuels orientés TDSI par la Synthèse d'Architecture”**, *Colloque sur le Traitement du Signal et de l'Image (GRETSI)*, 2004.
- [COU 2002f] P. Coussy, A. Baganne, E. Martin, **“Réutilisation de Composant Virtuel dans les Systèmes de Télécommunication: Une Etude de Cas MPEG-2 / JPEG2000”**, *Journées Francophones sur l'Adéquation Algorithme Architecture (JFAAA)*, 2002.
- [COU 2002g] P. Coussy, A. Baganne, E. Martin, **“Analyse Fonctionnelle des Moyens de communication Proposés dans les Systèmes sur Silicium”**, *Journées Francophones sur l'Adéquation Algorithme/Architecture (JFAAA)*, 2002.

2.6.10. Communications nationales invitées sans actes

- [COU 2008d] P. Coussy, **“Conception d'architectures multi-mode pour les applications de traitement du signal”**, *GDR Soc-Sip, thème Architecture et logiciel embarqué*, 18 Mars 2008.
- [COU 2007b] P. Coussy, **“Panorama des techniques de synthèse de haut-niveau”**, *GDR Soc-Sip, thème Architecture et logiciel embarqué*, 19 février 2007

2.6.11. Participation à la rédaction de documents publics

- **“Simulation and Design Automation Tools”**, Section 5.4.3, Roadmap High Performance and Embedded Architecture and Compilation (HiPEAC), 2011, www.hipeac.net (à paraître)
- **“ESL Virtual System Design Modelling”**, chapter V.2, European Electronic Design Automation Roadmap 2009, Medea+/Catrene, www.catrene.org
- **“Esw and ESL to RTL design”**, chapter V.3, European Electronic Design Automation Roadmap 2009, Medea+/Catrene, www.catrene.org

2.6.12. Communications par affiches / Démonstrations (sélection)

- Computer-Aided Network DEsign Workshop, October, 2009
- Workshop on High-Level Synthesis, IEEE Asia South Pacific Design Automation Conference ASP-DAC, January 2009
- Workshop on High-Level Synthesis, IEEE Design Automation Conference DAC, June 2008
- ACM-Sigda University Booth, IEEE Design Automation Conference DAC, June 2008
- Workshop on High-Level Synthesis, IEEE Design Automation Conference DAC, June 2008
- ACM-Sigda University Booth, IEEE Design Automation Conference DAC, June 2008
- Workshop on High-Level Synthesis, IEEE Design Automation and Test in Europe DATE, March 2008
- EDACentrum University Booth, IEEE Design Automation and Test in Europe DATE, March 2008
- Workshop on High-level synthesis ECSI Institute & UBS Workshop, 2006
- ACM-Sigda University Booth IEEE IEEE Design Automation and Test in Europe DATE, 2005
- PhD Forum of the IEEE International Conference on Design Automation and Test in Europe DATE, 2004

2.6.13. Logiciels libres

- **GAUT** Outil de CAO (<http://lab-sticc.fr/www-gaut>), diffusé dans le domaine public sous la license Cecill-B (<http://www.cecill.info>) et téléchargé plus de 100 fois par an (depuis 2006) dans plus de 90 Universités et entreprises localisées dans plus d'une cinquantaine de pays : en Europe (Allemagne, Belgique, Danemark, Espagne, France, Grèce, Italie, UK, Ukraine...), Amérique du Nord (USA, Canada, Jamaïque, Brésil, Cuba...), Asie (Chine, Inde, Iran, Taiwan), Océanie (Nouvelle Zélande, Australie...) et Afrique (Afrique du Sud, Tunisie, Maroc, Réunion, Madagascar...). Cet outil est utilisé entre autre pour réaliser des travaux pratiques (France, Belgique, Italie, Espagne, Suède, USA...) et aussi des travaux de recherche doctoraux (France, USA, Canada, Italie, Espagne, Belgique, Grèce, Tunisie...).
- **DsxPlore** Exploration de l'espace de conception et prototypage rapide d'applications de traitement du signal. Simplification de la spécification comportementale en SystemC à différents niveaux d'abstraction pour la simulation et génération automatique des contraintes de synthèse de haut niveau pour l'outil.
- **STAR** Générateur automatique d'architecture d'adaptateurs spatiaux-temporels utilisés pour réaliser des entrelaceurs parallèles, des chemins de données dans la synthèse de haut niveau ou pour intégrer des composants virtuels dans des architectures point à point ou sur un média de communication partagé.

3. Activités d'enseignement

J'ai enseigné pendant quatre années (dont trois en tant que moniteur et une en tant que ½ ATER) au sein de l'Institut Universitaire Professionnalisé (IUP) de l'Université de Bretagne Sud sur le site de Lorient, principalement dans les filières GSI (Génie des Systèmes Industriels) et GEII (Génie Electrique et Informatique Industrielle). Durant cette période j'ai eu la chance d'intégrer des équipes enseignantes disponibles qui m'ont permis de découvrir l'enseignement et d'en appréhender la fonction au travers de nombreuses discussions. Le bilan très positif de ces années, sur le plan professionnel et personnel, m'a conduit à devenir Maître de conférences.

Mes enseignements ont couvert durant cette période des domaines variés allant de l'électronique numérique à l'informatique en passant par le traitement du signal, les automates programmables industriels et l'asservissement de systèmes. Certains enseignements m'ont permis de transmettre les connaissances acquises durant mon cursus universitaire informatique ou durant ma thèse, comme la programmation en langage C et VHDL, ou plus spécifiques comme la conception de circuits temps réel et la synthèse comportementale. D'autres enseignements comme l'asservissement ou les API m'ont demandé un investissement plus particulier me permettant d'approfondir mes connaissances dans ces domaines. De plus, j'ai enseigné à un public hétérogène de part le niveau (bac, Deug, licence, maîtrise), la spécialité (GMP, GSI, préparation CAPET, GEII) et la motivation. Ces premières expériences m'ont appris à m'adapter à différents domaines de formation et à modifier mon approche d'enseignement en fonction du cursus des étudiants.

Depuis 2004, date de ma nomination, j'effectue l'intégralité de mon service d'enseignement au sein du département Sciences et Techniques de l'UFR Sciences et Sciences pour l'Ingénieur de l'Université de Bretagne sud à Lorient. Le parcours EII de la Licence PCSI (Physique Chimie et Sciences pour l'Ingénieur), suivi du Master Electronique et Informatique Industrielle (EII) ont pour vocation de former des « ingénieurs » pluridisciplinaires (électronique, informatique industrielle et traitement du signal et de l'image) de façon à leur permettre d'appréhender facilement les nombreuses technologies mises en œuvre dans les entreprises. J'interviens ainsi principalement dans les enseignements relevant du domaine de l'électronique et de l'informatique industrielle.

Dans la suite je présente chacune de mes expériences d'enseignement depuis mon doctorat. Cette partie s'attache à décrire les activités d'enseignement par année et par filière. Y sont présentés pour chaque matière le type d'enseignement, le contenu, le volume horaire ainsi que le nombre d'étudiants par groupe.

3.1. De 2000 à 2003 en tant que Moniteur de l'Enseignement Supérieur

3.1.1. *Maîtrise GEII (2000/2001)*

Traitement d'Images Encadrement de 8 heures de TP (10 étudiants par groupe). Ces TP permettent aux étudiants d'appréhender les algorithmes utilisés dans le traitement des images (restauration, amélioration, codage). L'efficacité des filtres utilisés a été observée au travers d'images exemples à l'aide du logiciel MATLAB.

3.1.2. Licence GEII (2000/2002)

Création partielle de l'enseignement

Système d'Exploitation Encadrement de 16 heures de TP (14 étudiants par groupe) sous forme de projet. Afin qu'ils en comprennent le fonctionnement et l'intérêt, l'enseignant responsable de cette matière désireait que les étudiants conçoivent en langage C, une gestion de type pagination de la mémoire. J'ai rédigé un sujet définissant le cahier des charges et ai participé au jury de soutenances de ces projets.

Asservissement Linéaire Encadrement de 12 heures de TP (12 étudiants par groupe). Ces enseignements mettent en pratique les connaissances fondamentales de l'asservissement de systèmes linéaires. Pour cela les étudiants ont étudié différents types de correcteurs : proportionnel, proportionnel intégral, proportionnel intégral dérivé, avance et retard de phase... pour améliorer le fonctionnement d'un système bouclé. Les étudiants réalisent les expériences à l'aide du logiciel MATLAB.

Asservissement Non Linéaire / Numérique Encadrement de 2 groupes de 14 étudiants pour un total de 24 heures de TP. Ces enseignements mettent en pratique les connaissances fondamentales de l'asservissement de systèmes non linéaires. Afin de dimensionner les correcteurs, les notions de marge de phase, marge de gain et bande passante ont été, dans ce contexte, largement abordées. Les étudiants réalisent les expériences à l'aide du logiciel MATLAB.

Automates Programmables Industriels API Encadrement de 2 groupes de 14 étudiants pour un total de 16 heures de TP. Ces travaux pratiques ont pour objectif la prise en main du logiciel PL7 Pro pour la spécification de Grafcet. La validation se fait sur des maquettes électriques et pneumatiques.

3.1.3. DEUG GEII (2002/2003)

Logique Séquentielle Encadrement de 8 heures de TP (12 étudiants par groupe). Lors de ces séances de TP, il est proposé aux élèves d'expérimenter des circuits usuels de la logique séquentielle (Mémoire, registres). D'une manière générale, la démarche de travail est la suivante : recherche de l'équation logique et simplification éventuelle, déduction du schéma logique, réalisation (câblage), vérification du fonctionnement et conclusion.

3.1.4. Licence GSI (2000/2003)

Asservissement Linéaire Encadrement de 12 heures de TP (12 étudiants par groupe).

Echange de Données Informatiques Encadrement de 2 groupes de 14 étudiants pour un total de 24 heures de TP. Ces TPs permettent aux étudiants d'appréhender les protocoles et techniques basiques de l'échange de données entre logiciel. La mise en pratique a été réalisée en utilisant les standards OLE puis DDE entre les logiciels Word et Excel de Microsoft.

3.1.5. DEUG GSI (2001/2003)

Logique Combinatoire Encadrement de 10 heures de TD (2001/2002) et de 8 heures de TP (8 étudiants par groupe) (2002/2003). Ces enseignements visent à donner les connaissances théoriques et pratiques de la logique combinatoire : les types de représentation (décimal, octal, binaire), le codage de l'information, le calcul algébrique, les fonctions logiques, la

synthèse de systèmes logiques et la conception de circuits logiques en utilisant des composants basiques (Unité arithmétique et logique, multiplexeur, comparateur...).

3.1.6. Licence STI (2000/2001)

Microprocesseur Encadrement de 8 heures de TP (12 étudiants par groupe). Ce TP a pour but de faire une étude d'une architecture mixte matérielle / logicielle. Dans une première partie les élèves étudient une carte CAIM. La deuxième partie du TP consiste en l'étude de programmes fournis puis l'écriture d'une fonction permettant de visualiser à l'écran la tension aux bornes d'un potentiomètre.

3.1.7. DEUG STI (2002/2003)

Création partielle de l'enseignement

Famille Logique Encadrement de 2 groupes de 10 étudiants pour un total de 8 heures de TP. Ce cours, dont j'ai adapté et mis à jour le fond et la forme, a pour objectif d'initier les étudiants aux composants électroniques numériques des différentes familles logiques utilisées pour la conception de circuits VLSI. Pour cela sont présentés les méthodologies de conception d'ASIC, les caractéristiques des circuits intégrés et les aspects technologiques (CMOS, TTL, ECL). Les séances de TD sont dédiées à la conception de fonctions logiques basiques (ET, OU) ainsi qu'à leurs performances temporelles (temps de propagation) et électriques (consommation). Lors des séances de TP, les étudiants étudient et analysent la conception de portes logiques en technologie CMOS.

3.1.8. DEUG GMP (2000/2002)

Création partielle de l'enseignement

Langage C Encadrement de 2 heures de TD et 36 heures de TP (12 étudiants par groupe). Pour cet encadrement j'ai rédigé plusieurs sujets de travaux pratiques. Lors de ces séances de TP les étudiants abordent dans un premier temps la structure générale d'un programme. Sont ensuite abordées les notions de chaînes de caractères, les sous programmes, les variables structurées, les pointeurs et les entrées sorties standards.

3.1.9. Licence GMP (2001/2003)

Asservissement Linéaire : Encadrement de 12 heures de TP (12 étudiants par groupe).

Asservissement Linéaire : Encadrement de 2 groupes de 14 étudiants pour un total de 24 heures de TP. Ces TP ayant lieu avant les cours, chacune des séances commencent par un cours d'environ 1 heure.

3.1.10. Licence Pluridisciplinaire (préparation au CAPET) (2002/2003)

Création partielle de l'enseignement

Logique Combinatoire et Séquentielle Encadrement de 6 heures de TD et 8 heures de TP (5 étudiants). Cet enseignement a été particulier puisqu'il s'adressait à un groupe de cinq étudiants de la licence pluridisciplinaire de l'Université de Bretagne Sud préparant le concours CAPET. Mon choix s'est porté sur cette formation pour deux raisons : il s'agissait, d'une part, d'une formation toute nouvelle qui demandait à être mise en place et d'autre part, qui concernait un très faible nombre d'étudiants très motivés mais aussi et surtout très inexpérimentés dans la matière en question. Bien que l'enseignement de ce module soit

réparti sur deux ans, le volume horaire reste très limité pour aborder un ensemble de notions précises pour la préparation au concours : 3 séances de TD et 2 séances de TP par an. Les points abordés en TD et TP ont été discutés et choisis en accord avec le maître de conférences responsable des cours magistraux. La première année d'enseignement de la logique combinatoire dans cette filière étant consacrée aux notions de bases, j'ai rédigé deux sujets de TD adaptés au public visé mais aussi et surtout aux connaissances requises pour le concours. De plus, j'ai proposé un sujet de TP assez ludique puisqu'il consistait en un petit jeu de bataille entre deux joueurs. Ce type de sujet a l'avantage pour des élèves non-spécialistes du domaine, de faire un lien plus direct entre la théorie et la pratique mais surtout entre les fonctions de base de la logique combinatoire (tel que le multiplexage) et leurs applications concrètes.

3.2. De 2003 à 2004 en tant que ½ Attaché Temporaire d'Enseignement et de Recherche

3.2.1. Licence GEII

FPGA Encadrement de 8 heures de TP (14 étudiants par groupe) visant la compréhension de la synthèse logique sur FPGA en utilisant un logiciel de conception de circuits numériques (ISE Fondation de la société Xilinx).

VHDL Encadrement de 36 heures de TP (14 étudiants par groupe) dont 24 heures sous la forme d'un projet aboutissant à la conception d'un calculateur de distorsion entre images pour un estimateur de mouvement. J'ai défini, en collaboration avec l'équipe pédagogique, les points devant être traités par les étudiants et ai participé au jury de soutenance de ces projets.

Signaux et systèmes Encadrement de 8 heures de TP (14 étudiants par groupe). Ce module enseigne les bases et les outils théoriques du Traitement du Signal et de l'Image : Analyse fréquentielle (transformé de Fourier analogique et discrète), transformée en Z, filtrage numérique. Lors de la première séance de TP, les étudiants appréhendent les effets de l'échantillonnage et de la quantification sur un exemple de numérisation d'une fonction de transformation de Fourier FFT. Durant la deuxième séance, ils analysent un signal vocal et plus particulièrement sa décomposition en un ensemble d'harmoniques dans le domaine fréquentiel.

Automates Programmables Industriels API Encadrement de 2 groupes de 14 étudiants pour un total de 16 heures de TP.

3.2.2. DEUG GEII

Logique Combinatoire Encadrement de 10 heures TD (30 étudiants) et 16 heures de TP (2 groupes de 15 étudiants).

Logique Séquentielle Encadrement de 8 heures de TP (15 étudiants par groupe).

Famille Logique Encadrement de 8 heures de cours, 6 heures TD, 8 heures de TP.

3.3. Depuis 2004 en tant que Maître de Conférences

3.3.1. Master 2 Math-STIC, parcours SIAM (2006/2011)

Création de l'enseignement

System-On-Chip / ESL design Encadrement de 12 heures de cours, 4 heures TD et 4 heures de TP (15 étudiants par groupe). Ce cours est une introduction à la conception de systèmes numériques intégrés complexes. Dans un premier temps, le processus de fabrication des circuits intégrés est présenté. Suit un rappel détaillé de l'évolution des calculateurs, processeurs, mémoires, périphériques... pour aboutir à la notion de système sur puce multiprocesseur (MPSoC). L'approche de conception de niveau système (ESL design) est ensuite abordée. Dans ce contexte, l'évolution des méthodologies de conception est détaillée et les concepts tels que modélisation, prototypage virtuel, synthèse de haut-niveau, conception logiciel/matériel, partitionnement, intégration de coprocesseurs dédiés, composants virtuels... sont abordés.

3.3.2. Master 1 Math-STIC, parcours SIAM (2004/2011)

Création de l'enseignement

Méthodologie Encadrement de 10 heures de cours, 4 heures TD et 8 heures de TP (15 étudiants par groupe). L'objectif de cet enseignement est de permettre aux étudiants d'acquérir les compétences permettant de conduire un projet, de l'organiser, de le planifier et de le mener à bon terme. Les notions de gestion de projet, suivi des coûts, maîtriser des risques sont abordés. Une attention particulière est portée sur les phases de spécification, conception/analyse, et test. Durant les TPs, les étudiants sont amenés à réfléchir sur un cas concret de projet faisant appel à leurs connaissances scientifiques et leur demandant d'appliquer les concepts vus en cours pour aboutir à une spécification et une planification précise des travaux à partir d'un cahier des charges complexe et ambigu.

Création de l'enseignement

Réseaux locaux (2005/2011) Encadrement de 10 heures de cours, 4 heures TD et 8 heures de TP (15 étudiants par groupe). L'objectif de cet enseignement est de permettre aux étudiants de comprendre les fondamentaux des réseaux de terrain par une approche pragmatique. Les notions telles que topologie, adressage, type de transfert, synchronisation, gestion de flux... sont vues aux travers de l'étude des liens séries RS232, RS422, RS485, le protocole MODBUS, Bus CAN ou le Bus USB. Partant d'une base initiale constituées de deux joueurs, il est demandé aux étudiants de proposer une solution permettant de jouer à un nombre illimité de joueurs. Chaque joueur utilise une carte munie d'un microcontrôleur et de deux interfaces séries.

3.3.3. Licence 3 PCSI, parcours EII (2004/2011)

Création partielle de l'enseignement

Système d'exploitation encadrement de 10 heures de cours, 6 heures de TD et de 8 heures de TP (1 à 2 groupes de 15 étudiants). L'objectif de cet enseignement est de faire connaître les fonctions et les services offerts par un système d'exploitation : comprendre et faire le

lien entre l'architecture d'un microprocesseur, la programmation, la compilation... Cet enseignement sert d'introduction au module de « temps réels » du Master 1 GEII. Les notions de processus, gestion de la mémoire, gestion des entrées/sorties ainsi que la gestion des fichiers sont abordées. Les TP visent à la prise en main de Linux puis à l'étude des concepts étudiés en cours.

Création partielle de l'enseignement

Technologie des composants électroniques (2004) Encadrement de 8 heures de cours, 4 heures de TD et de 8 heures de TP. L'objectif de cet enseignement est de faire comprendre les principes de la logique CMOS et ses performances. A partir du transistor, la construction des blocs fonctionnels d'un système numérique est décrit (blocs logiques, bus, mémoires, pads d'entrée/sortie). Les principes de construction des circuits logiques programmables sont ensuite présentés. La caractérisation des systèmes numériques CMOS en termes de vitesse, de consommation et de surface est aussi abordée.

Création de l'enseignement

Projet Génie Informatique Encadrement de 36h de TP. L'objectif de cet enseignement est d'acquérir et consolider de bonnes connaissances en langage C, notamment concernant, les types structurés, les pointeurs et les listes chaînées. Au travers d'un projet complexe, les étudiants doivent mettre en œuvre les algorithmes et les approches de conception étudiées en cours d'algorithmique. Exemple de projet : modélisation du déplacement d'un robot dans un terrain, modélisation d'une colonie de fourmis pour trouver le plus court chemin entre des points géographiques, coloriage des sommets d'un graphe en utilisant un nombre minimum de couleurs, résolution automatique de grilles de SUDOKU...

Création de l'enseignement

Conception système Encadrement de 8 heures de cours, 6 heures de TD et de 8 heures de TP. L'objectif de cet enseignement est double : faire connaître les différents modes de communication (interruption versus attente active) entre composants et introduire une architecture de microcontrôleur. L'étude des composants d'E/S telle l'UART, des timers... est abordée. L'objectif est aussi de faire comprendre et de faire le lien avec les cours de microprocesseur, programmation et système d'exploitation.

3.3.4. Licence 1 Physique Chimie Sciences de l'Ingénieur (2008/2011)

Création de l'enseignement

Application des sciences aux métiers, compression d'image (2008/2011): Encadrement de 2 heures de cours, 2 heures de TD et de 4 heures de TP (2 groupes de 15 étudiants). Cet enseignement a comme objectif de faire percevoir à des étudiants de première année de licence l'intérêt de la Science au quotidien et dans les métiers. J'interviens donc dans ce contexte pour enseigner les rudiments de la représentation et de la compression d'image numérique qui sont utilisées dans les systèmes tels que les ordinateurs, les appareils photo numériques...

Création partielle de l'enseignement

Programmation 22 heures de TD/TP (2005/2006) Lors de ces séances de TP/TD les étudiants abordent dans un premier temps la structure générale d'un programme en langage C. Sont ensuite abordées les notions de chaînes de caractères, les sous-programmes, les variables structurées, les pointeurs et les entrées sorties standards

3.3.5. Master EII (Institut du Génie Appliqué, Maroc) (2008/2010)

J'ai été amené à enseigner à l'Institut du Génie Appliqué situé à Casablanca au Maroc dans le cadre de la délocalisation du diplôme de Master EII de l'Université de Bretagne Sud. Dans ce contexte, j'ai enseigné les deux matières nommées *méthodologie* et *System-on-Chip*, déjà présentées précédemment. Une adaptation du contenu du cours a été nécessaire pour prendre en compte la spécificité du parcours de l'IGA qui est plus orienté vers l'automatique que la formation EII de l'UBS.

3.3.6. Formation doctorale (Université de Mondragon, Espagne) (2010)

J'ai été invité en 2010 à aller enseigner à l'Université de Mondragon en Espagne un module complet sur les nouvelles approches de conception pour les systèmes sur puces. Durant vingt heures, j'ai fait un séminaire en anglais à l'ensemble des étudiants de Master 2 recherche, des doctorants et des enseignants de la filière « Signal Processing & Communication ». Dans ce contexte, les notions et approches telles que conception de niveau système, co-conception, synthèse de haut niveau... ont été introduites.

3.3.7. Ecole de printemps, ENS Lyon (2010)

J'ai été invité dans le cadre d'une école de printemps de l'Ecole Normale Supérieure (ENS) de Lyon, dont le thème était les architectures nouvelles, à faire un cours sur la synthèse de haut niveau qui est un des axes majeurs de mon domaine de recherche. Bien que cette école soit ouverte à tous, le public visé est principalement composé d'étudiants de master (1 ou 2) en informatique fondamentale. Le challenge est donc ici d'arriver à faire passer des notions de méthode de conception de composants électroniques (et les problèmes inhérents) à des étudiants qui n'ont que très peu de notion en architecture matérielle.

3.3.8. Master ENSIETA (2008/2011)

J'interviens depuis 2008 à l'ENSIETA (Ecole Nationale Supérieure des Ingénieurs des Etudes et Techniques d'Armement) pour enseigner la synthèse de haut niveau. Les étudiants qui assistent à ce module (8h) proviennent des filières IASE (Informatique Automatique des Systèmes Embarqués) ou ESSE (Electronique & Signal pour les Systèmes Embarqués). L'objectif de ce séminaire est surtout une ouverture à de nouvelles approches de conception dans le domaine de l'électronique.

3.4. Bilan de l'activité d'enseignement depuis 2000

3.4.1. De 2000 à 2004 (moniteur et ½ATER)

La Figure 1 présente la répartition des enseignements dont j'ai eu la responsabilité durant mes quatre années (dont trois en tant que moniteur et une en tant qu'ATER) au sein de l'Institut Universitaire Professionnalisé (IUP) de l'Université de Bretagne Sud.

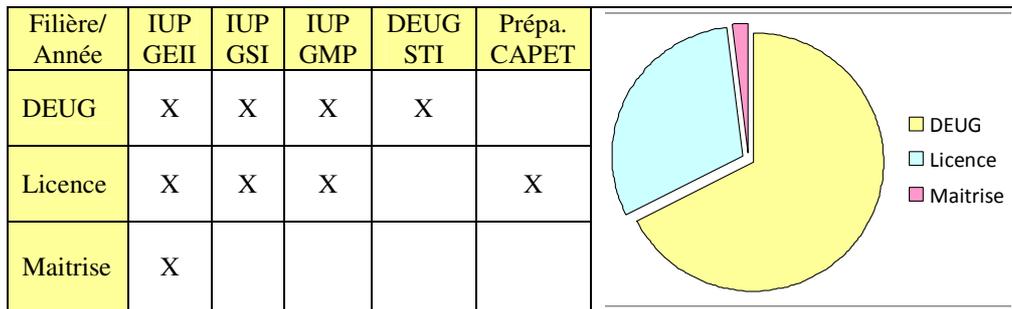


Figure 1: Répartition des enseignements entre DEUG, Licence et Maitrise

Le volume total représente 429 heures équivalent TP et s'est réparti comme suit quasi uniquement sous la forme de travaux pratiques.

Formations dispensées

- **Génie Informatique**
 - Langage C (72h TP) *
 - Système d'exploitation (16h projet) *
 - Echange de données informatiques (24h TP)
- **Informatique industrielle**
 - Automates Programmables Industriels (API) (32h TP)
 - Microprocesseur (8h TP)
- **Electronique numérique**
 - Logique combinatoire et séquentielle (26h TD, 48h TP) *
 - Technologie des familles logiques (8h Cours, 6h TD, 8h TP) *
 - Langage VHDL (8h TP, 24h projet)
 - FPGA (8h TP)
- **Signaux et systèmes**
 - Asservissement linéaire (60h TP)
 - Asservissement non linéaire et numérique (24h TP)
 - Signaux et Systèmes (8h TP)
 - Traitement d'Images (8h TP)

Le symbole * signifie que l'enseignement a été créé.

3.4.2. De 2004 à aujourd'hui

La Figure 2 ci-dessous présente la répartition par année des enseignements dont j'ai eu la responsabilité depuis 2004, année de mon recrutement en tant que Maître de conférences à l'Université de Bretagne Sud. Les enseignements réalisés sur les 4 années universitaires 2006-2010 (i.e. avant la réforme définissant l'équivalence entre TP et TD) au sein de l'Université de Bretagne Sud se répartissent de façon égale entre le niveau Licence (quasi en totalité en troisième année) et le niveau Master. Bien qu'elles représentent une part non négligeable du service d'un enseignant chercheur, les activités telles que les encadrements de projet, suivis de stage, tutorat... n'ont pas été intégrées dans la représentation d'activité.

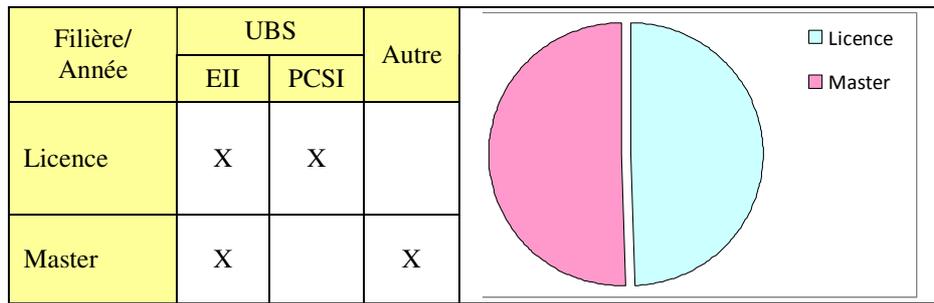


Figure 2: Répartition des enseignements entre niveau Licence et Master

Le volume total représente au minimum 192 heures équivalent TD plus au maximum 50 heures équivalent TD depuis 2008 suite à l'obtention d'une PEDR. La répartition s'équilibre entre les enseignements théoriques et les enseignements pratiques (Figure 3)

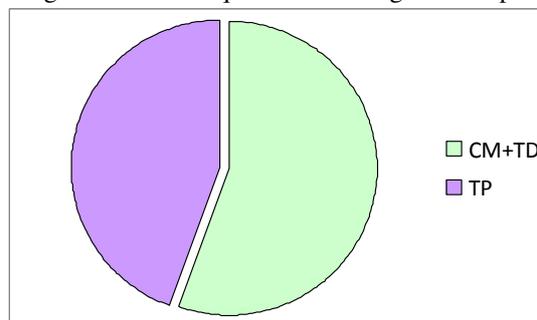


Figure 3: Répartition des enseignements théoriques et pratiques

Les formations dispensées en moyenne sur les quatre dernières années (150 heures équivalent TD en considérant CM = 1.5 TD et TD = 1.5 TP). Tous ces enseignements ont été créés en totalité ou en partie.

- **Génie informatique**
 - Programmation langage C (36h TP)
 - Système d'exploitation (10hCM, 4h TD, 8hTP)
- **Informatique industrielle**
 - Microcontrôleur (8hCM, 6h TD, 8hTP)
 - Réseaux locaux industriels (8hCM, 2h TD, 8hTP)
- **Electronique numérique**
 - Méthode de conception de systèmes numériques (20h CM, 4h TD, 4h TP)
- **Signaux et systèmes**
 - Traitement d'image (2h TD, 2h TD, 4h TP)
- **Enseignement général**
 - Méthodologie (8h TD, 4h TD, 8h TP)

Il est important de noter que la quasi totalité des étudiants qui intègrent le parcours EII en troisième année de licence, année durant laquelle j'effectue une grande partie de mon service d'enseignement, provient d'IUT ou de lycées (BTS) très variés. Ils ont de ce fait, en général, des profils hétérogènes qui ont tous, bien souvent malheureusement, un point commun : un très faible niveau en informatique au sens général (programmation, architecture, algorithmique...). C'est donc l'un des grands challenges de mes enseignements qui demandent de déployer des efforts pédagogiques importants afin d'amener les étudiants à un niveau acceptable et homogène dans cette discipline avant l'entrée en première année de Master.

4. Responsabilités collectives, animations et projets scientifiques

Dans cette dernière section, je présente les différentes responsabilités collectives que j'ai eues depuis que je suis à l'Université de Bretagne Sud. Je résume également mes activités d'animation scientifique au niveau national et international. Enfin, je liste les différents projets de recherche auxquels j'ai participé.

4.1. Au niveau de l'Université de Bretagne Sud

4.1.1. Au sein de l'UFR Sciences et Sciences de l'Ingénieur

Je suis **directeur des études** du parcours Electronique et Informatique Industrielle de la Licence Physique Chimie Sciences de l'Ingénieur depuis 2004, année de mon recrutement en tant que Maître de conférences à l'Université de Bretagne Sud. A ce titre, j'ai en charge la réalisation des emplois du temps, la gestion des étudiants au quotidien, l'animation de l'équipe pédagogique, la présidence des jurys, le suivi des étudiants partant en stage... Je participe également à l'élaboration des nouveaux supports et moyens de promotion (affiche, vidéo, plaquette...) afin de présenter la filière EII (Licence + Master) aux étudiants. Je représente aussi la filière dans différents salons, forum, IUT et lycées (BTS).

Je suis **membre du comité de pilotage** depuis 2006 et **membre de la commission de perfectionnement** depuis 2010 de la Licence Physique, Chimie, Sciences de l'Ingénieur. A ce titre je participe à l'élaboration (rédaction dossier AERES...), à l'amélioration de la formation des étudiants de cette filière, à l'amélioration de l'organisation des enseignements, du suivi des étudiants (tutorat...). Je suis également **membre de la commission formation** de l'UFR SSI.

Je suis par ailleurs **tuteur** en Licence 1 d'une quinzaine d'étudiant par an depuis 2005. Mon rôle est de conseiller les étudiants dans leur parcours universitaire : choix des UEs, méthode de travail...

Je suis **membre de la commission recherche et personnel** depuis 2011 au sein de l'UFR qui a pour rôle de réfléchir sur les profils d'enseignement et de recherche des futurs enseignants/chercheurs.

4.1.2. Au sein du laboratoire

Je suis **membre de comités de sélection** depuis 2008 et ai été **membre élu de la Commission de Spécialistes**, 61-63^{ème} section entre 2006 et 2008.

Je suis **membre élu au conseil du laboratoire Lab-STICC** depuis 2009 et ai été **membre élu au conseil du laboratoire LESTER** de 2006 à 2008.

Je suis **responsable/animateur scientifique du groupe** « synthèse d'architecture » depuis 2006. Ce groupe est composé d'une petite dizaine de personne (Maître de conférences, post-doc, ingénieur de recherche, doctorant, master recherche...). Je m'occupe aussi à ce titre de la gestion du site web de l'outil GAUT.

J'ai été, avec Guy Gogniat, à l'initiative de la mise en place des « réunions de permanents » du laboratoire en 2006. Le but de ces réunions était, à la différence des réunions du conseil (décision) ou de laboratoire (diffusion scientifique), d'aborder des sujets plus stratégiques et politiques (financements, recrutements...).

J'ai été responsable du montage de projet européen de type STREP, pour répondre au cinquième appel à projet de l'IST FP6 (septembre 2005). Bien qu'ayant été noté positivement, le projet nommé DESIRE pour "DESign Space exploration for Real Embedded systems" n'a pas été sélectionné. Le consortium était composé des partenaires suivants : STMicroelectronics (Fr), Infineon (Ger), Coware (Bel), Celoxica (Uk), Dune (It), LESTER (Fr), TU of Szczecin (Pol), ECSI (Fr).

J'ai été **tuteur** de trois moniteurs de l'enseignement supérieur entre 2005 et 2008 et participe depuis 2010 à deux comités de suivi de thèse.

4.1.3. Au sein des conseils de l'établissement

Je suis **membre élu du Conseil Scientifique** de l'université depuis 2009.

J'ai été **membre élu au Conseil de l'Ecole Doctorale** pluridisciplinaire entre 2001 et 2003.

4.2. Au niveau national

Je suis **membre des groupes de recherche GDR SoC-SiP** (System-on-Chip/System-in-Package) et **GDR ISIS** (Information, Signal, Images et ViSion) depuis 2005. J'assiste et participe régulièrement (en tant qu'auditeur ou présentateur) aux réunions d'animation qui permettent de rester en lien avec la communauté nationale, de découvrir de nouvelles problématiques et d'initier de nouvelles collaborations.

Je suis **animateur du thème C du GDR ISIS** en charge des interactions avec d'autres GDR depuis 2010. J'ai organisé une première journée inter-GDRs sur le thème des « systèmes embarqués pour la santé » qui a réuni plus de 50 personnes, venant des GDR SoC-SiP, ISIS, STIC-Santé, le 8 juillet 2010 sur le site de Jussieu. Je suis également membre du comité d'organisation du « barcamp » qui a eu lieu à Fréjus en avril 2011 sur la même thématique. Une deuxième journée est en cours d'organisation autour des méthodes d'optimisation pour la conception de systèmes embarqués (GDR RO, SoC-SiP, ISIS).

J'ai été **co-organisateur** avec Guy Gogniat d'une journée régionale intitulée « Rencontre de l'Electronique en Bretagne ». Cette journée, qui a eu lieu le mardi 12 octobre 2010 à Lorient dans les locaux de l'Université de Bretagne-Sud, a été organisée en collaboration avec l'Université Européenne de Bretagne (UEB), Université de Bretagne-Sud (UBS), la Maison de l'Electronique, de l'Informatique et de la Télématique de l'Ouest (MEITO) et le Conseil Général du Morbihan (CG56). Le thème principal était celui de la recherche en électronique en Bretagne : innovation, valorisation, formation. Cette journée, qui a réuni plus de cent personnes, a reçu un très bon accueil de la part des partenaires académiques et industriels de la région.

4.3. Au niveau international

Je suis **membre élu** au comité technique du Design and Implementation of Signal Processing Systems (DISPS) de l'**IEEE Signal Processing Society**.

Je suis **membre du réseau d'excellence** Européen HiPEAC depuis 2009 et participe à la rédaction de la roadmap 2011.

J'ai été membre du **groupe de travail SWG de l'OSCI** (Synthesis Working Group, Open SystemC Initiative) qui définit le sous ensemble synthétisable du langage SystemC pour la synthèse de haut niveau (Key Contributor 2009, Affiliate member 2010).

J'ai été **membre du comité de rédaction** de la Roadmap Européenne Catrene (ex MEDEA) sur l'EDA en 2009.

Je suis **membre des sociétés savantes** IEEE depuis 2004, ACM depuis 2009 et Eurasip de 2006 à 2008.

Je suis **co-organisateur** de nombreux tutoriaux et workshops autour des approches de conception de niveau système (HLS, ESL...). Ces évènements sont, à mon avis, un excellent moyen de promouvoir de nouvelles approches de conception notamment auprès des entreprises, de mettre en place des collaborations, de réunir la communauté, d'échanger sur des aspects scientifiques, mais aussi sur la place de la recherche et sa diffusion, de faire connaître le laboratoire, d'apprendre, de partager...

4.4. Participation à des collaborations scientifiques et à des contrats d'études

Depuis 2004, je me suis toujours impliqué dans des collaborations avec d'autres équipes universitaires et industrielles. Ces collaborations sont, à mon sens, essentielles pour ne pas dire « vitales » pour les activités de recherche.

4.4.1. Collaborations Académiques Internationales

Université de Californie, Los Angeles UCLA

Synthèse de haut niveau pour la conception faible consommation sur FPGA.

Type : fond propre

Durée : 4 mois en 2009

Partenaires : Lab-STICC, P. Coussy, G. Lhairech-Lebreton, UCLA J. Cong

Objet : *Développement d'un environnement de conception faible consommation multi horloges – multi-tensions sur FPGA.*

Contribution du Lab-STICC : *Définition d'un flot de synthèse de haut-niveau pour la génération d'architecture à multiple domaines d'horloge.*

Contribution du CADLab : *Définition d'une architecture FPGA multi Vdd/Vth et d'un flot de synthèse bas niveau faible consommation associé.*

Université de Bologne, Bologne, Italie

Flot de conception pour architecture MPSoC intégrant des accélérateurs matériels.

Type : fond propre

Durée : 2011-2013

Partenaires : Lab-STICC, P. Coussy, DEIS L. Benini, P. Burgio

Objet : *Développement d'un environnement de conception utilisant des directives de compilation OpenMP et la synthèse de haut niveau pour générer automatique des systèmes multiprocesseurs optimisés.*

Contribution du Lab-STICC : *Approche de synthèse de haut niveau et d'intégration d'accélérateur.*

Contribution de DEIS : *Architecture générique MPSoC et flot de conception partant d'une spécification système basée sur OpenMP.*

4.4.2. Contrats publics de recherche

GIGADEC

Etude des architectures de décodeur correcteur d'erreurs (FEC) très haut débit dédiés aux applications de télécommunication cellulaire de 4^e génération

Type : Projet région

Consortium: TurboConcept, Telecom-Bretagne/Lab-STICC, UBS-Lab-STICC

Budget : Coût 727k€, 203k€ d'aide demandée dont 104k€ pour l'UBS/Lab-STICC

Durée : 2011 – 2013

Rôle : responsable projet pour le Lab-STICC & montage

Projet P

P (conception d'une chaîne de génération de code multi-domaines sous licence libre)

Type : Projet FUI

Consortium: Aboard Engineering, AGC Solutions, AdaCore, Airbus, Altair, ASTRIUM, ATOS Origin, Continental, Ecole des Ponts ParisTech, INRIA/Aoste-Espresso-Metalau, IRIT/INPT/ENSEEIH, Lab-STICC, ONERA, Rockwell Collins, Sagem Défense Sécurité, Scilab, ST INFORMATIQUE SERVICES, Thales Alenia Space, Thales Avionics

Budget : Coût 9,8M€, 3.8M€ d'aide demandée dont 165k€ pour le Lab-STICC

Durée : 2011 – 2014

Rôle : responsable projet pour le Lab-STICC & montage

SoCKET

SoC toolKit for critical Embedded systems

Type : Projet Direction générale de la compétitivité, de l'industrie et des services DGCIS

Consortium: Airbus, Astrium, CNES, STMicroelectronics, Thales R&T, Schneider Electric Industries, PSI-S, CEA-LETI, Magilem, INPG-TIMA, UPS-IRIT, UBS-Lab-STICC

Budget : Coût 11,4M€, 3.5M€ d'aide demandée dont 177k€ pour le Lab-STICC

Durée : 2008 – 2011

Rôle : responsable projet pour le Lab-STICC & montage

SoCLib

open platform for virtual prototyping of multi-processors system on chip

Type : Projet plate-forme l'ANR/RNTL

Consortium : STMicroelectronics, Thales Communications, Thomson Silicon Components, Prosilog, TurboConcept, Silicomp, UPMC/LIP6, ENST, UPMC/LISIF, CEA LIST, INRIA Futurs, IRISA, LESTER, IETR INSA, TIMA, CEA LIST, CITI

Budget : Coût 4M€ aide demandée : 2.9 M€ dont 120k€ pour le LESTER

Durée : 2006-2009

Rôle : responsable projet pour le Lab-STICC & montage

SystemC'Mantic

A high level Modeling and Co-design Framework For Reconfigurable. Real Time Systems

Type : Projet RNTL

Consortium : Prosilog, THALES, TIMA, CEA LIST, TIMA, LESTER

Budget : Coût 1.9M€, aide demandée : 1.1M€ dont 215k€ pour le LESTER

Durée : 2003-2005

Rôle : responsable projet pour le Lab-STICC à partir de 2004

FLASH

Seed Optimisation and Indexing of Genomic Databases

Type : ARC INRIA

Consortium : IRISA, LIFL, LESTER, INSERM U694.

Durée : 2006-2007

Rôle : responsable projet pour le Lab-STICC

ANTELOP

A Novel approach for real-TimE and LOw-Power Embedded systems

Type : Région Bretagne pour l'aide au montage du projet Européen IST STREP DESIRE sur l'exploration de l'espace de conception et la synthèse de haut niveau

Aide : 22 k€

Date : 2005

Rôle : responsable projet

ALIPTA

Algorithmic Level IP for Telecom Applications

Type : Projet RNRT (projet précompétitif)

Consortium : TNI-Valiosys, THALES, SACET, Turbo-Concept, ENST Bretagne

Budget : 4.802 kF, dont 847 kF pour le LESTER

Durée : 2002-2004

Rôle : participation durant la thèse

4.4.3. Contrats privés de recherche

A-ST, Astrium Space Transportation

Evaluation de l'outil GAUT pour des prototypages en VHDL

Type : Contrat privée dans le cadre d'un projet R&T entre le CNES et la société ASTRIUM Space Transportation

Consortium : Astrium Space Transportation (Les Mureaux), Lab-STICC

Budget : 20 k€

Durée : 2011-2012

Rôle : responsable projet pour le Lab-STICC

ONAGRE

enviOnnement de coNception et de prototypAGed'applications Radio et mobile

Type : Contrat CRE (Contrat de Recherche Externalisée)

Consortium : France Telecom R&D (Issy les moulineaux), LESTER

Budget : 150 k€

Durée : 2005-2008

Rôle : responsable projet pour le Lab-STICC à partir de 2006

La Table 3 ci-dessous résume l'ensemble des projets auxquels j'ai participé. Les projets représentés par des blocs en pointillé ont ou démarreront en 2011 et ne sont pas décrits dans ce document. Différentes sources de financement ont été utilisées afin de financer les activités de recherche (RNRTL, RNRT, ANR, FUI, Région, privé).

	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014
ALIPTA		Projet RNRT Composant virtuel algorithmique												
SystemC/Mantic			RNRTL Modélisation comportementale											
Onagre					CRE Synthèse de haut niveau									
Antelop					PRIRE montage projet									
SocLib						Plateforme ANR Prototypage virtuel plateforme MPSoC								
Flash						ARC INRIA High-performance Computing								
SoCKeT								FUI-DGE Flot de conception systèmes critiques						
Astrium-ST											Contrat privé Synthèse de haut niveau			
Gigadec											Région Bretagne Archi. décodeur de correcteur d'erreurs			
Projet-P											FUI-DGE Chaîne générique de génération de code			

Table 3 • Résumé des projets de recherche

Partie 2 : Annexes, Sélection des publications significatives

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

5. Article concernant la synthèse d'architectures multi-modes

C. Andriamisaina, P. Coussy, E. Casseau, C. Chavet,

“High-Level Synthesis for Designing Multi-mode Architectures”,

IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems (TCAD),

Vol. 29, Issue 11, pp. 1736-1749, November 2010.

6. Article concernant la synthèse de haut niveau

P. Coussy, G. Gajski, A. Takach, M. Meredith,

“An Introduction to High-Level Synthesis”,

IEEE Design and Test of Computers, Special issue on High-Level Synthesis,

Vol. 26, Issue 4, July/August, 2009.

7. Article concernant la synthèse d'architecture multi-horloges

G. Lhairech-Lebreton, P. Coussy, E. Martin,

“Hierarchical and Multiple-Clock Domain High-Level Synthesis for Low-Power Design on FPGA”,

IEEE International Conference on Field Programmable Logic and Applications (FPL), 2010.

8. Article concernant la modélisation comportementale

F. Thabet, P. Coussy, D. Heller, E. Martin,

“Exploration and Rapid Prototyping of DSP Applications using SystemC Behavioral Simulation and High-Level Synthesis”,

Journal of Signal Processing Systems, Springer,

Issue 56, 2-3, September 2009.

9. Article concernant la synthèse d'architecture d'unité mémoire gros grain

C. Chavet, P. Coussy, P. Urard, E. Martin,

“Static Address Generation Easing: a Design Methodology for Parallel Interleaver Architectures”,

IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2010.

**10. Article concernant la synthèse d'architecture
d'unité mémoire grain fin**

C. Chavet, P. Coussy, P. Urard, E. Martin,

"A Design Methodology for Space-Time Adapter",

***ACM Great Lakes Symposium on VLSI on VLSI (GLSVLSI),
2007.***

Université de Bretagne Sud

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

Synthèse d'architecture pour les applications de traitement du signal et de l'image

Partie 3

Soutenue le 15 septembre 2011 devant la commission d'examen composée de :

F. Pétrot	Professeur des Universités	INP Grenoble	Rapporteur
M. Renovell	Directeur de recherche	Laboratoire LIRMM	Rapporteur
O. Sentieys	Professeur des Universités	Université de Rennes 1	Rapporteur
D. Etiemble	Professeur des Universités	Université de Paris Sud	Examineur
A. Greiner	Professeur des Universités	Université Pierre et Marie Curie	Examineur
L. Lavagno	Professeur des Universités	Ecole Polytechnique de Turin	Examineur
E. Martin	Professeur des Universités	Université de Bretagne Sud	Examineur
M. Paindavoine	Professeur des Universités	Université de Bourgogne	Examineur

Par

Philippe Coussy

Laboratoire Lab-STICC
Université de Bretagne Sud – CNRS UMR 3192
Centre de Recherche Christiaan Huygens
56321 Lorient Cedex

A mon père,

Table des matières

PARTIE 3 : TRAVAUX DE RECHERCHE DÉTAILLÉS ET PERSPECTIVES	- 9 -
1. Introduction	- 11 -
1.1. Contexte	- 11 -
1.2. Positionnement des travaux	- 15 -
2. Axe 1 : Synthèse d'architecture d'unité de calcul	- 17 -
2.1. Introduction	- 17 -
2.2. Présentation des travaux	- 18 -
2.2.1. Métaheuristiques pour l'optimisation	- 18 -
2.2.2. Synthèse d'architecture multi-modes	- 23 -
2.2.3. Synthèse hiérarchique d'architecture multi domaines d'horloge	- 26 -
2.3. Conclusion	- 30 -
2.4. Fiche de synthèse des travaux	- 30 -
3. Axe 2 : Modélisation et prototypage	- 35 -
3.1. Introduction	- 35 -
3.2. Présentation des travaux	- 36 -
3.2.1. Exploration algorithmique et prototypage FPGA	- 36 -
3.2.2. Modélisation comportementale	- 40 -
3.2.3. Conception d'architecture MPSoC	- 46 -
3.3. Conclusion	- 53 -
3.4. Fiche de synthèse des travaux	- 53 -
4. Axe 3 : Synthèse d'architecture d'unité mémoire	- 57 -
4.1. Introduction	- 57 -
4.2. Présentation des travaux	- 59 -
4.2.1. Unité mémoire à « grains fins »	- 59 -
4.2.2. Unité mémoire à « gros grains »	- 62 -
4.3. Conclusion	- 65 -
4.4. Fiche de synthèse des travaux	- 66 -
5. Conclusion et perspectives de recherche	- 69 -
6. Références	- 73 -

Table des figures

Figure 1 : Evolution des finesses de gravure et de la densité d'intégration	- 12 -
Figure 2 : Evolution des standards de communication.....	- 12 -
Figure 3 : Miniaturisation et diversification des systèmes sur puce	- 13 -
Figure 4 : Evolution du nombre de concepteurs.....	- 14 -
Figure 5 : Flot de conception type en synthèse de haut niveau	- 17 -
Figure 6 : Exemple d'architecture	- 19 -
Figure 7 : Estimateur de surface.....	- 19 -
Figure 8 : Illustration de la méthode de descente	- 20 -
Figure 9 : Illustration de la méthode de «descente multi-start».....	- 21 -
Figure 10 : Illustration de la méthode de recherche à voisinages variables VNS.....	- 21 -
Figure 11 : Comparaison des différentes méthodes avant synthèse logique	- 22 -
Figure 12 : Comparaison des différentes méthodes après synthèse logique.....	- 22 -
Figure 13: Vue générale du flot de conception proposé.....	- 24 -
Figure 14 : Combinaisons des applications et de leurs contraintes de cadence.....	- 25 -
Figure 15 : Comparaison en surface totale	- 26 -
Figure 16 : Flot de synthèse proposé.....	- 27 -
Figure 17 : Architecture à contrôle distribué et à horloges multiples.....	- 28 -
Figure 18 : Architecture (a) Flat, (b) mono ou multi-horloge avec contrôleur centralisé, (c) LIS ou GALS	- 28 -
Figure 19 : Comparaison en consommation	- 29 -
Figure 20 : Comparaison en surface.....	- 29 -
Figure 21 : Nombre de fils pour la FFT 32 points (600ns).....	- 29 -
Figure 22 : Flot de conception pour l'exploration algorithmique dans ONAGRE	- 37 -
Figure 23 : Flot de synthèse non bit près.....	- 38 -
Figure 24 : Flot de synthèse bit près.....	- 38 -
Figure 25 : Résultats de consommation puissance et de surface	- 40 -
Figure 26 : Flot de conception proposé par le projet SystemC'mantic.	- 41 -
Figure 27 : Comportement : (a) non pipeline sans recouvrement d'entrées/sorties, (b) non pipeline avec recouvrement d'entrées/sorties, (c) pipeline sans recouvrement d'entrées/sorties, et (d) pipeline avec recouvrement d'entrées/sorties.	- 42 -
Figure 28 : Problématique de raffinement manuel.	- 43 -
Figure 29 : (a) Approche proposée, (b) modèle BDM	- 44 -
Figure 30 : Temps de simulations des différents modèles BDM d'un ACS (pour 1 million d'itérations).	- 45 -
Figure 31 : Flot de conception général	- 46 -
Figure 32 : Application MJPEG pour une architecture (a) mono et (b) multiprocesseurs-	47
-	-
Figure 33 : Architecture mono ou multiprocesseurs : (a) sans coprocesseur et (b) avec coprocesseur(s).....	- 48 -
Figure 34 : Temps d'exécution de l'application (en #cycles).....	- 48 -
Figure 35: Temps de simulation (en secondes)	- 49 -
Figure 36 : Architecture cible.....	- 50 -
Figure 37 : Flot de conception.....	- 51 -
Figure 38 : Application JPEG	- 51 -
Figure 39 : Comparaisons des différentes approches pour l'application JPEG.....	- 52 -
Figure 40 : Effet du partage entre processeurs d'un HWPU	- 53 -
Figure 41 : Représentation simplifiée d'une chaîne de traitement du signal	- 57 -
Figure 42 : Architecture typique d'un entrelaceur parallèle	- 58 -
Figure 43 : Architecture STAR	- 60 -
Figure 44 : Flot de conception et outils associés	- 60 -
Figure 45 : Exemple de graphe de compatibilité des ressources	- 61 -

Figure 46 : Comparaison des résultats de synthèse entre l'architecture de référence et trois composants STAR.....	- 61 -
Figure 47 : Matrices références : Matrice d'ordre (a) naturel M_{Nat} et (b) entrelacé M_{Int} ..	- 63 -
Figure 48 : Matrices d'assignation : Matrice d'ordre (a) naturel MAP_{Nat} et (b) entrelacé MAP_{Int} (Etape 1).....	- 63 -
Figure 49 : Matrices d'assignation : Matrice d'ordre (a) naturel MAP_{Nat} et (b) entrelacé MAP_{Int} (Etape 2).....	- 64 -
Figure 50 : Matrices d'assignation : Matrice d'ordre (a) naturel MAP_{Nat} et (b) entrelacé MAP_{Int} (Résultat).....	- 64 -
Figure 51 : Résultats de synthèse (surface en porte NAND équivalente)	- 65 -

Partie 3 : Travaux de recherche détaillés et perspectives

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

1. Introduction

1.1. Contexte

Les Technologies de l'Information et de la Communication (TIC) au sens large du terme (télécommunications, multimédia, informatique, (micro)-électronique) sont depuis longtemps au carrefour des mutations de notre société. Ainsi, les 30 dernières années ont été aussi révolutionnaires dans ces domaines que les décennies qui ont connu l'invention du télégraphe, du phonographe ou du cinématographe ou celles plus récentes durant lesquelles le transistor, le circuit intégré ou l'ordinateur ont été inventés. Ainsi le grand public a vu l'apparition du premier téléphone portable et du disque compact (CD) durant les années 80, de l'internet et de la photo numérique dans les années 90, du GPS, des baladeurs MP3 et de la télévision numérique au début des années 2000 et finalement de l'alliance du tout au sein d'un même objet portable nommé « Smartphone » à partir du milieu des années 2000. Cette évolution n'est plus cantonnée au domaine dit « grand public » et l'industrie des systèmes embarqués concerne de nos jours tous les secteurs : santé, sécurité, énergie, transport, industrie... Dans ce contexte, le marché mondial des systèmes embarqués devrait représenter 230 milliards de dollars en 2015 (source Idate 2010). Dans l'ère du tout numérique dans laquelle nous évoluons depuis peu, 50% des habitants de la planète sont déjà équipés d'un téléphone portable. Les ventes de ces équipements dans le monde en 2009 s'élèvent à 1.16 milliards d'unité (source GfK). Le nombre de Smartphone vendus en 2009 puis en 2010 est de 173.5 et 302.6 millions (soit une croissance de 74.4 %, source IDC). L'internet, les ordinateurs et la télévision ne sont pas en reste puisque nous comptons en 2010, 366 millions d'ordinateurs et 218 millions de téléviseurs dont 171 millions de type LCD vendus mais aussi 1,6 milliards d'abonnés à internet (source Gartner). A cela s'ajoute des cycles de renouvellement toujours plus courts pour les équipements grands publics. Ainsi, un utilisateur change en moyenne tous les 6 mois de téléphone portable, tous les 1 à 2 ans de PC ou d'appareil photo numérique, tous les 2 à 3 ans de téléviseur et tous les 3 à 5 ans de voiture.

La « numérisation » de la société a, en partie, été possible grâce à la miniaturisation constante des composants électroniques, elle-même rendue possible par la création d'un cercle « vertueux ». En effet, la réduction des finesses de gravure s'est toujours accompagnée d'une augmentation significative de la capacité d'intégration, d'une amélioration des performances des circuits intégrés, d'un accroissement des capacités de stockage et d'un renforcement de la complexité des applications. L'existence, « garantie » par la loi de Moore, de technologies futures toujours plus fines associée à la diminution des coûts de production (un transistor coûtait 1\$ en 1968 et 1 nano\$ en 2004) et à la création de marchés toujours plus nombreux et rentables (le marché mondial des semi-conducteurs, en constante progression, a dépassé les 300 milliards de dollars en 2010 selon Gartner) ont permis des investissements massifs dans le domaine des semi-conducteurs. Ces investissements ont à leurs tours permis la miniaturisation de la technologie... Ainsi, au milieu des années 1990 la technologie CMOS 0.35 μ m était utilisée, aujourd'hui le 22nm voit le jour et bientôt les progrès technologiques semblent nous promettre des transistors d'une largeur inférieure à dix nanomètres grâce au graphène [ITRS 2010].

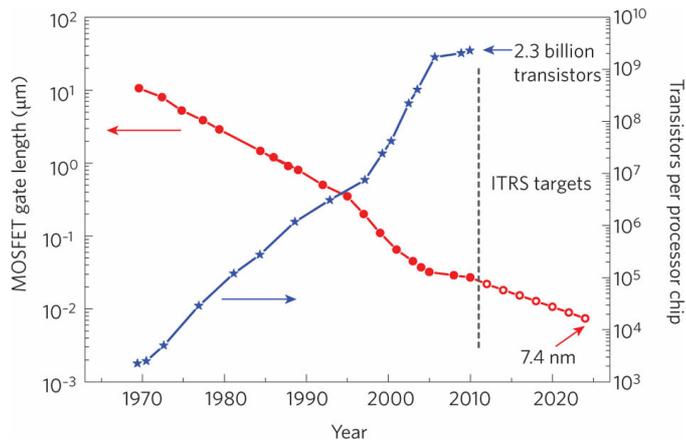


Figure 1 : Evolution des finesses de gravure et de la densité d'intégration

A l'inverse des finesses de gravures, la complexité des applications exécutées sur des systèmes numériques est en constante augmentation (voir Figure 1). Le domaine des télécommunications a par exemple vu la création de nombreux standards ces dernières années. Pour ne parler que des communications numériques sans fil, différentes générations se sont succédées : 2G (GSM, CDMA...), 2.5G (GPRS, EDGE...), 3G (UMTS, W-CDAM, CDMA2000...), 3.5-3.9G (HSPA, LTE...), 4G (LTE-advanced, HSPA+...), offrant chacune un accroissement significatif du débit de transmission allant pour le GSM de quelques dizaines de kilobits à plusieurs centaines de mégabits par seconde pour le LTE-advanced (voir Figure 2). Cette évolution se retrouve dans tous les domaines. Ainsi de nos jours, un détecteur de fumée de dernière génération ne se contente plus d'émettre une alarme sonore mais appelle également un centre de secours. Par ailleurs, la quantité de code embarqué sur le Concorde était de 4Ko alors que les calculateurs de vol des avions A310 (1982), A320 (1987), A340 (1990) utilisent respectivement 4, 10 et 20 Mo de code ce qui est toutefois sans commune mesure avec les avions actuels pour lesquels 1/3 du coût global est lié aux systèmes embarqués (dont 40% en développement de logiciels). Il en est de même dans le secteur automobile où environ 20% du coût de fabrication (40% prévu pour 2020 selon PSA Peugeot Citroën) d'une automobile vient de la conception et de la réalisation des systèmes embarqués (les dernières générations de berline embarquent plus de 10 millions de lignes de code et plus d'une centaine d'unités de contrôle ECU).

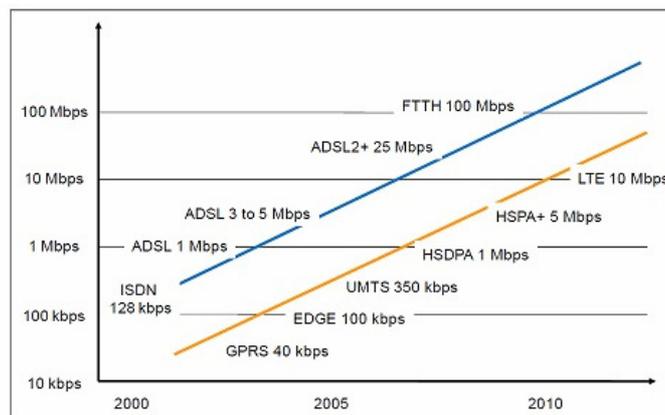


Figure 2 : Evolution des standards de communication

Les architectures sous-jacentes aux systèmes numériques intégrant des processeurs ont rapidement évolué au fil du temps pour s'adapter et répondre aux contraintes dictées par des utilisations sans cesse renouvelées : surface, performance, consommation. Ainsi, il est généralement admis que le premier système embarqué reposant sur des circuits intégrés était l'ordinateur de commande et de pilotage des vaisseaux spatiaux des missions Apollo, développé au MIT (Massachusetts Institute of Technology). La première version était composée de 4100 circuits contenant chacun une porte 3-NOR pour réaliser un processeur programmable 16 bits. Le premier microprocesseur (Intel 4004) a ensuite été réalisé en 1971 pour fabriquer une calculatrice. Le premier microcontrôleur TMS-1000 de Texas Instrument (premier SoC diront certains) a été mis sur le marché en 1974 et utilisé pour faire une calculatrice de poche (SR-16, TI). Le premier processeur de traitement du signal AT&T DSP1 de Bell Lab a été utilisé en 1979 dans la réalisation d'un commutateur pour réseau téléphonique commuté. Les ASICs sont apparus au début des années 80 sous la forme de « gate arrays » ou « standard cells » pour condenser l'ensemble des composants discrets d'une carte en un seul composant et ainsi réduire les coûts. Les FPGA ont ensuite rapidement fait leur apparition au milieu des années 80 pour palier le coût de production des ASICs. Initialement utilisés pour remplacer quelques dizaines de portes logiques, leur montée en puissance permet de réaliser des circuits toujours plus complexes. Les architectures mixtes logiciel/matériel hétérogènes composées d'un microprocesseur et d'un accélérateur (ASIC ou FPGA) permettant d'accélérer une partie du traitement se sont développées avec la diffusion des applications orientées multimédia et télécommunication. Ces évolutions technologiques nous permettent de nos jours de concevoir des systèmes multiprocesseurs sur puce nommés MPSoC. Ces systèmes submicroniques sont conçus autour d'un bus partagé ou d'un réseau enfoui (NoC), embarquent des mémoires organisées hiérarchiquement, des contrôleurs d'E/S, des accélérateurs, des systèmes d'exploitations... et seront à terme réalisés en 3D, deviendront reconfigurables dynamiquement, intégreront des capteurs (« More than Moore », voir Figure 3 de [ITRS 2010])...

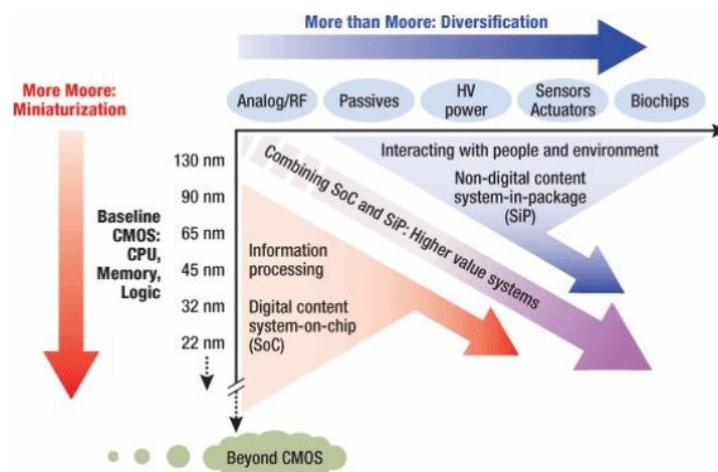


Figure 3 : Miniaturisation et diversification des systèmes sur puce

Selon la feuille de route publiée par l'ITRS (International Technology Roadmap for Semiconductors), le nombre de transistors pouvant être intégré sur une puce augmenterait de 58% par an, alors que la capacité de production n'augmenterait que de 21% par an. La réduction de ce fossé ne peut pas être adressée seulement par l'augmentation du nombre de concepteurs puisque cela implique des coûts de conception faramineux. Le tableau de la Figure 4, tiré de [URA 2007], montrait en 2006 que pour une technologie de 65nm et un

taux de productivité annuel de 200000 portes logiques par concepteur, il fallait compter 75 Homme.An pour concevoir un circuit intégrant 15 millions de portes logiques. Sans évolution des approches de conception et donc de productivité, ces chiffres seraient portés à 300 Homme.An pour concevoir un circuit intégrant 60 millions de portes logiques pour une technologie de 32nm en 2010.

1992	1994	1996	1998	2000	2002	2004	2006	2008	2010
Finesse de gravure									
0,6	0,5	0,35	0,25	0,18	0,13	90	65	45	32
# de portes / mm²									
1k	5k	15k	30k	45k	80k	150k	300k	600k	1,2M
# de portes / die(50mm²)									
50k	250k	750k	1,5M	2,25M	4M	7,5M	15M	30M	60M
# de portes / concepteur / an									
4k	6k	9k	40k	56k	91k	125k	200k	200k	200k
Homme.Année / die(50mm²)									
~10	~40	~80	~40	~40	~43	~60	~75	~150	~300

Figure 4 : Evolution du nombre de concepteurs

Ainsi, la demande toujours croissante de nouveautés, la multitude et la diversité des produits, la complexité des systèmes à concevoir, la complexité des architectures sous-jacentes, l'accroissement des performances, la réduction de la consommation énergétique, la diminution des finesses de gravures, la différenciation de l'offre, les délais et fenêtres de mise sur le marché, les cycles de renouvellement, la maîtrise des coûts... demandent une évolution constante des approches de conception des systèmes et des circuits qui les composent.

L'abstraction couplée à l'automatisation des processus de vérification et de synthèse ainsi que la réutilisation de blocs préconçus ont toujours été des facteurs clés dans l'évolution des méthodes de conception qui permettent une exploration plus rapide et plus efficace de l'espace de conception.

Dans le domaine du logiciel, le langage machine (séquence binaire) était à l'origine l'unique moyen pour programmer un ordinateur. Dans les années 50, le concept d'assembleur a été introduit avant que ne soient proposés, pour améliorer la productivité, des langages de haut niveau et des techniques de compilation. Les langages de programmation haut niveau sont indépendants de la plateforme et suivent les règles du langage humain avec une grammaire, une syntaxe et une sémantique. Ils fournissent donc de la flexibilité et de la portabilité en masquant les détails de l'architecture sous-jacente sur laquelle les programmes sont exécutés. De nos jours, l'utilisation de l'assembleur est assez rare et se cantonne à l'optimisation de parties critiques d'un programme lorsqu'il y a un besoin absolu de performance, de compaction ou des deux. Aux vues de la complexité des applications et des architectures mais aussi des temps de développements imposés, l'utilisation de langages de haut niveau permet d'obtenir globalement de bien meilleurs résultats que l'utilisation de l'assembleur avec lequel plus personne ne songerait programmer une application complexe dans sa totalité.

Dans le domaine du matériel, les langages de spécifications et les méthodes de conception ont évolué de façon similaire. Ainsi, jusqu'à la fin des années 60 les circuits étaient conçus, optimisés et dessinés entièrement à la main. La simulation au niveau porte et la simulation au niveau cycle sont respectivement apparues au début et à la fin des années 70. Les nouvelles techniques telles que placement-routage, capture schématique de circuit, vérification formelle et analyse temporelle statique ont été introduites durant les

années 80. Les langages de description de matériel tel que Verilog (1986) ou VHDL (1987) ont permis une large adoption des outils de simulation. Ces langages ont ensuite servi d'entrée aux outils de synthèse logique amenant la définition de leurs sous-ensembles synthétisables. Durant les années 90, les outils de synthèse de haut niveau de première génération ont été commercialisés. A la même époque, les travaux de recherche autour de la conception mixte matérielle-logicielle (estimation, exploration, partitionnement, interfaçage, communication, synthèse, co-simulation...) se sont accentués. La conception par réutilisation et la conception basée sur des plateformes ont ensuite émergé. Dans les années 2000, des approches de conception dites « de niveau système » (System Level Design SLD ou Electronic System Level Design ESLD) ont vu le jour pour permettre d'explorer, de synthétiser et de vérifier plus efficacement des architectures complexes. Ce saut en abstraction s'est accompagné de la création de nouveaux langages tels que SystemC ou SystemVerilog et de la modélisation de niveau transactionnelle (TLM). Toutes ces évolutions ont permis d'augmenter la productivité c'est-à-dire de réduire le temps de conception, le temps de vérification, le temps d'exploration de l'espace de conception (performance, surface, consommation)...

1.2. Positionnement des travaux

Les flots de conception ont évolué pour permettre aux concepteurs d'exploiter les modèles de haut niveau, qui étaient jusqu'alors utilisés pour valider des principes algorithmiques, directement en entrée des outils de synthèse. L'objectif est de pouvoir mettre au point les systèmes (architecture + application) plus rapidement à l'aide de spécifications, de modélisations et de langages de haut niveau. Une fois la spécification fonctionnelle de l'application validée, le concepteur a alors besoin de méthodes et d'outils lui permettant de raffiner cette description abstraite pour aller vers l'implémentation. Les flots de conception modernes permettent de spécifier une application, d'explorer l'espace des solutions architecturales, de réutiliser des blocs de plus en plus complexes et doivent également permettre la synthèse des parties logicielles et matérielles et la validation formelle des architectures obtenues. La suite de cette section présente uniquement les étapes d'un flot de conception « idéal » qui ont un lien avec les travaux exposés ultérieurement dans ce document. De ce fait, la description du flot de conception est volontairement incomplète.

Spécification et raffinement de l'application

L'application fait en premier lieu l'objet d'une modélisation à haut niveau (MatLab, C, C++, SystemC, Esterel...), appelée à ce stade spécification fonctionnelle, avant d'être parallélisée si besoin puis raffinée. Dans le domaine du traitement du signal, des environnements de spécification et de simulation tels que Cocentric System Studio de Synopsys (ex COSSAP) ou SPW [SYN 2011] présentent l'avantage d'une spécification algorithmique en langage de haut niveau, en l'occurrence C (facile d'exploitation par les « algorithmiciens ») et une spécification système sous forme graphique (appréciée des « algorithmiciens » et des ingénieurs application), reposant sur le modèle formel SDF (Synchronous Data Flow) [LEE 1987]. Grâce aux vitesses de simulation importantes qu'ils offrent, ces modèles parallèles de l'application sont utilisés pour déterminer rapidement un découpage fonctionnel du système et pour tester les algorithmes retenus pour chaque acteur (aussi nommée tâche).

Exploration de l'espace de conception

Une fois le modèle de haut niveau validé, la spécification de l'application peut être implantée sous une forme logicielle, matérielle ou mixte. Pour cela, une étude de précision des calculs est en générale réalisée et la spécification doit être raffinée pour être plus proche de l'implémentation finale (code embarqué, spécification plus proche d'une réalisation

matérielle, parallélisation...). Pour explorer l'espace de conception (analyse de performance, partitionnement...), le prototypage virtuel est de plus en plus utilisé [BAI 2007]. Ce type d'approche permet de proposer une modélisation des parties matérielles du système très en avance de phase, pour pouvoir démarrer le plus tôt possible le développement des parties logicielles embarquées dans le système. Les architectures ainsi modélisées permettent une simulation rapide de l'application autorisant à son tour une exploration de l'espace de conception efficace.

Synthèse d'architecture

La complexité croissante des algorithmes implémentés, et l'augmentation continue des volumes de données et des débits applicatifs ainsi que les contraintes de consommation requièrent encore la conception d'accélérateurs matériels dédiés. Typiquement l'architecture d'un composant complexe du TDSI utilise des *éléments ou unité de calculs* de plus en plus complexes, des *unités mémoires* (mémoires et/ou modules de brassage de données tels les entrelaceurs/désentrelaceurs pour les Turbo-Codes/LDPC ou les blocs de redondance spatio-temporelle dans les systèmes OFDM/MIMO...), privilégie des connexions point à point pour la communication inter éléments de calcul et demande d'intégrer dans une même architecture plusieurs configurations et/ou algorithmes (systèmes (re)configurables pour supporter de multiple standards/applications). Ces accélérateurs matériels peuvent être conçus à la main ou par des outils de synthèse d'architecture par l'équipe utilisant ce composant ou par une autre équipe (de la même entreprise, division ou non).

Les thèmes de recherche que j'ai développés ces dernières années se sont concentrés autour du thème principal de la conception des systèmes numériques avec une forte dominante pour les méthodes et en particulier pour la synthèse automatique d'architecture. La synthèse d'architecture est devenue de nos jours un des moyens nécessaires pour une conception efficace des systèmes numériques et des circuits qui les composent. L'utilisation de la synthèse de haut niveau peut par exemple :

- aider au raffinement et au paramétrage de la spécification algorithmique dans une démarche de prototypage rapide,
- permettre la conception optimisée d'une architecture pour une spécification algorithmique donnée et un ensemble de contraintes et objectifs,
- aider à explorer l'espace de conception d'un système sur puce en générant des architectures matérielles à partir de modèles de simulation comportementaux,
- aider à explorer l'espace de conception d'un système sur puce en générant des modèles de simulation à partir d'une spécification algorithmique non temporisée,
- ...

Trois axes de recherche peuvent ainsi être dégagés : *la synthèse d'architecture d'unité de traitement, la modélisation et le prototypage* ainsi que *la synthèse d'architecture d'unité mémoire*. Ces travaux ont été validés sur différents types d'applications du domaine du traitement du signal ou de l'image dans différents projets de recherche collaboratifs. Ainsi, huit doctorants, neuf stagiaires de DEA ou Master recherche, deux post-doctorants, un ingénieur ont travaillé ou travaillent actuellement sur ces différents thèmes de recherche. Après un bref résumé présenté dans cette section, ces différents thèmes de recherche seront développés dans les sections 2, 3 et 4.

La suite du document est organisée en quatre parties. Les trois premières présentent ma contribution dans le domaine de la conception des systèmes numériques selon les trois axes mentionnés précédemment. Enfin la quatrième conclut ce document et propose un ensemble de perspectives et de pistes de recherche à explorer dans de futurs travaux.

2. Axe 1 : Synthèse d'architecture d'unité de calcul

2.1. Introduction

Cet axe de recherche adresse la génération automatique d'accélérateurs matériels ou coprocesseurs dédiés. A partir de descriptions abstraites, des architectures de niveau transfert de registres (RTL) sont générées sous contraintes (temps, ressources...) en utilisant des techniques de synthèse de haut niveau et visent à optimiser un ou plusieurs objectifs (consommation, temps, surface...).

La première étape d'un flot de synthèse de haut niveau est l'étape de compilation (voir Figure 5). Cette étape réalise la vérification syntaxique et sémantique de la description algorithmique et la traduit en un format intermédiaire propre à l'environnement de synthèse. De plus, la phase de compilation réalise des opérations telles que : l'élimination du code mort, la propagation des expressions constantes, le déroulage de boucles, la mise en ligne des fonctions... Deux types de représentations intermédiaires sont couramment employés dans les outils de synthèse pour la représentation intermédiaire : les graphes flot de données (Data Flow Graph DFG) et les graphes flot de données et de contrôle (Control and Data Flow Graph CDFG) [GAJ 1992].

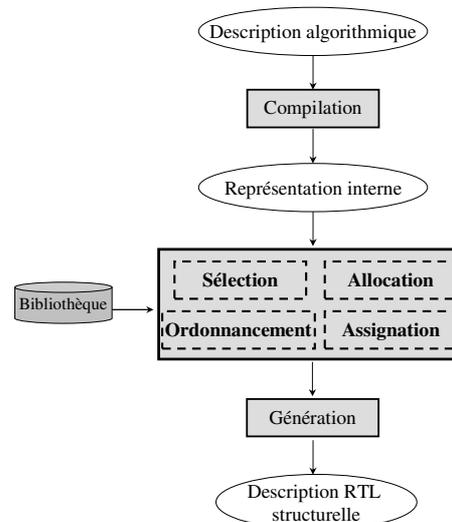


Figure 5 : Flot de conception type en synthèse de haut niveau

Les différentes étapes, effectuées par la suite, opèrent sur le modèle formel de représentation interne :

- L'étape de **sélection** consiste à choisir la nature des ressources matérielles (opérateurs) qui réaliseront les opérations présentes dans l'application. Le choix des composants se fait sur des critères tels que leur surface, leur vitesse ou leur consommation.
- L'étape **d'allocation** détermine, pour chaque type d'opérateur ou d'opération, le nombre de ressources à utiliser dans l'architecture finale. Il est à noter que, souvent, dans la littérature, l'étape d'allocation englobe l'activité de sélection.

- L'étape *d'ordonnement* a pour rôle d'affecter une date d'exécution à chacune des opérations en tenant compte d'une part des dépendances de données et d'autre part des contraintes imposées par le concepteur. Ainsi, l'ordonnement peut chercher à : minimiser le nombre d'étapes de contrôle en fonction d'une quantité de ressources, minimiser le nombre de ressources en fonction d'un nombre de cycles d'horloge...
- L'étape *d'assignation* associe à chaque opération un opérateur matériel dans l'architecture et à chaque variable un élément de mémorisation.
- L'étape de *génération* qui permet de fournir des descriptions utilisables pour la synthèse logique (VHDL, Verilog...) ou pour la simulation (VHDL, Verilog, SystemC...)

Les travaux de recherche de cet axe ont été développés dans le cadre de différents projets. Ainsi,

- la conception d'architecture multi-modes par la synthèse de haut niveau a été abordée dans le cadre des travaux de thèse de Caaliph Andriamisaina (2005-2008),
- l'optimisation de la surface à l'aide de méta-heuristiques a été explorée dans le cadre de la synthèse de haut niveau durant la thèse de Kods Trabelsi (2006-2009),
- la génération par la synthèse de haut niveau hiérarchique d'architecture à multiples domaines d'horloge pour la conception faible consommation sur FPGA est étudiée dans les travaux de thèse de Ghizlane Lhairech-Lebreton (2007-2011),
- l'optimisation des architectures en surface et consommation par la synthèse de haut niveau de spécifications bit-près dans le cadre du projet de recherche privée ONAGRE en partenariat avec France Telecom (2005-2008),
- l'introduction de la prédiction de branchement dans la synthèse comportementale a été évaluée durant le stage de master recherche de Vianney Lapotre (2010).

Seuls les trois premiers ensembles de travaux seront présentés dans cette section. La prise en compte de spécification bit-près est détaillée dans la section 3 relativement à l'utilisation du langage SystemC. De plus, afin de rester concis, les travaux sur la prédiction de branchement récemment initiés ne seront pas abordés dans ce document (voir [LAP 2011a]).

2.2. Présentation des travaux

2.2.1. Métaheuristiques pour l'optimisation

Dans ces travaux de recherche, nous avons pour objectif la minimisation de la surface des architectures réalisant des applications de traitement de signal et de l'image implémentées sur FPGA. Le choix de cibler les FPGAs s'appuie essentiellement sur le fait que le marché de cette technologie est en pleine croissance grâce à ses multiples avantages (performances, temps réduit de mise sur le marché, coût relativement faible, solution fiable...). L'optimisation des architectures a été abordée en utilisant des méthodes de Recherche Opérationnelle (RO) : les métaheuristiques. Ces travaux ont été réalisés dans le cadre de la thèse de Kods Trabelsi [TRA 2009] en co-encadrement avec Marc Sevaux et en collaboration avec André Rossi, tous deux, membres de l'équipe RO du laboratoire Lab-STICC.

L'un des objectifs des outils de synthèse de haut niveau dédiés aux applications, dont la contrainte principale est le temps, est de trouver une architecture qui a un coût minimal. Trouver les techniques d'ordonnement, d'allocation, d'assignation des variables et des opérations qui mènent systématiquement à la solution optimale en surface est quasiment impossible. Dans la plupart des contributions de l'état de l'art qui abordent la génération

automatique de la description structurelle, un ordre d'exécution des étapes (allocation, ordonnancement, assignation) est prédéfini afin de résoudre le problème étape par étape. Dans cette optique, une étape est entièrement résolue avant de passer aux suivantes. Cette approche sacrifie la bonne résolution d'une étape en faveur des précédentes et ne conduisent pas, par conséquent, à une solution optimale au problème global. Ceci est dû à la complexité et l'interdépendance entre ces sous-problèmes. En effet, l'ordre de résolution des étapes de la synthèse de haut niveau a un impact fort sur la qualité de l'architecture obtenue. Parmi les stratégies utilisées dans les outils de synthèse, on trouve par exemple le choix de résolution de l'allocation avant ou après l'ordonnancement [COU 2008]. Dans le cas où l'allocation est réalisée avant l'ordonnancement, le nombre d'opérateurs défini à la fin de l'étape d'allocation servira de guide pour l'ordonnancement. Dans certains cas, ce nombre n'est pas définitif et pourra être mis en cause par le résultat de l'ordonnancement. Ainsi, l'ordonnancement se fait sous contrainte de temps (le débit) et de ressources. La contrainte de ressources est dans ce cas flexible et peut être revue si son respect s'oppose à celui de la contrainte de temps. Dans le cas où l'allocation se fait après l'ordonnancement le nombre de ressources allouées sera supérieur ou égal au nombre maximum de ressources utilisées en même temps. Ainsi, le choix de l'ordre d'exécution de ces deux phases aura un impact sur la solution. Une forte interdépendance existe aussi entre les étapes d'ordonnancement et d'assignation. Ces deux problèmes sont NP-complets [GAR 1990] [PAN 1991]. En outre, la résolution optimale de chacun de ces sous-problèmes ne mène pas forcément à la solution optimale du problème global [CON 2008]. Ainsi, le choix de l'ordre d'exécution des étapes de la synthèse définira l'espace explorable des solutions. Par exemple, si l'on décide de résoudre les problèmes d'ordonnancement et d'assignation des opérations dans une première phase, le résultat obtenu limitera les possibilités d'assignations des variables.

Ainsi, comme les méthodes exactes se retrouvent incapables de résoudre le problème de la minimisation de la surface du circuit en un temps raisonnable et que les heuristiques proposées sont dédiées aux sous-problèmes de la HLS ne résolvent que des sous-parties du problème [GAJ 1992] [COU 2008], nous avons travaillé sur des méthodes permettant d'explorer un espace de solution plus large, afin de retenir la meilleure solution rencontrée dans l'espace exploré. Le principe retenu repose sur la génération d'une première architecture appelée solution initiale qui est ensuite raffinée par modifications successives. L'espace de solutions pouvant être colossal pour certaines applications, nous avons eu recours aux métaheuristiques pour l'obtention du plus grand nombre de solutions possible. Pour l'évaluation de la qualité de chacune des solutions traduite par la surface de celle-ci, l'utilisation d'un estimateur a été nécessaire. Celui-ci, faisant l'estimation au plus haut niveau, doit être bien étudié afin de s'approcher au mieux du résultat obtenu suite à l'implémentation réelle de l'application considérée.

Pour dénombrer les composants de l'architecture on procède de la manière suivante. Pour les opérateurs, le nombre est fixé de manière explicite à la fin des phases d'ordonnancement et d'assignation des opérations. Quant au nombre de registres, il est également défini de façon explicite à la fin de la phase d'assignation des variables. Reste à trouver le type et le nombre de multiplexeurs. Celui-ci peut être aisément déduit à partir de la grille, symbolisant les connexions entre les registres et les opérateurs, que nous proposons.

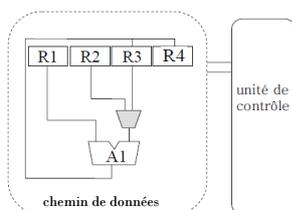


Figure 6 : Exemple d'architecture

Dest./Souc.	A1	R1	R2	R3	R4	Type du Mux
A1	A11	0	1	0	0	
	A12	0	0	1	1	(2 :1)
	R1	0	0	0	0	
	R2	0	0	0	0	
	R3	1	0	0	0	
	R4	1	0	0	0	
Composants	1add	1reg	1reg	1reg	1reg	1 mux (2 :1)

Figure 7 : Estimateur de surface

Le tableau de la Figure 7 est une fidèle représentation du chemin de donnée de la figure Figure 6. Les lignes du tableau correspondent aux entrées des opérateurs (A11 et A12 sont les entrées de l'additionneur A1) et des registres, alors que les colonnes correspondent à leurs sorties. Une case du tableau contient le nombre de variables qui transitent entre la sortie et l'entrée correspondante. Une connexion sera créée entre une source et une destination si la case correspondante contient un entier non nul. Si une destination est desservie par plus d'une seule source, un multiplexeur devra être introduit. La deuxième ligne du tableau indique que l'entrée A12 de l'additionneur A1 nécessite un multiplexeur de deux entrées et une sortie noté (2:1). Celui-ci se chargera d'aiguiller les variables arrivant des sorties des registres R2 et R3.

Ainsi, ce tableau nous permet de compter tous les composants de l'architecture: opérateurs, registres et multiplexeurs. Si l'on connaît la surface de chacun de ces composants, on peut estimer la surface totale occupée par le chemin de données, une fois l'application implémentée sur le FPGA cible. Nous avons pour cela caractérisé en surface une bibliothèque d'opérateurs (registres, multiplexeurs et opérateurs) en utilisant la synthèse logique. Les unités retenues sont les *gates* pour une synthèse logique non hiérarchique et *slices* pour une synthèse logique hiérarchique.

Grâce à l'utilisation d'un estimateur, l'évaluation de la qualité d'une solution se fait rapidement, sans passer par la longue phase de synthèse logique. Ceci permet d'avoir recours à des approches qui optimisent une solution initiale par raffinements successifs. Différentes approches plus ou moins complexes ont été étudiées : méthodes de simple descente, plus grande descente, Multi-start, Recherche à voisinage variable et Greedy Randomized Adaptative Search Procedure (GRASP). À partir d'une solution initiale trouvée, par une heuristique par exemple, on peut facilement appliquer des méthodes de descente. Les méthodes de descente s'articulent toutes autour d'un principe simple : partir d'une solution existante, puis chercher une solution dans le voisinage et accepter cette solution si elle améliore la solution courante (voir figure Figure 8).

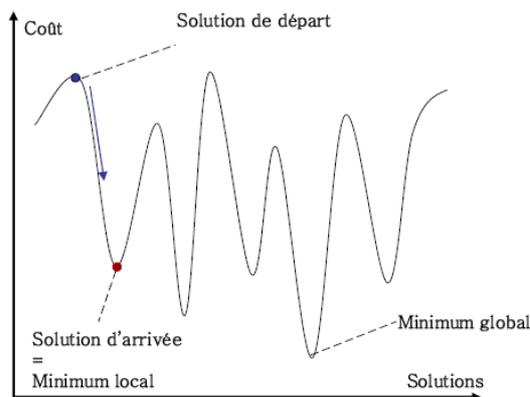


Figure 8 : Illustration de la méthode de descente

Pour une méthode de descente simple, à partir d'une solution initiale x , on choisit une solution x' dans le voisinage $N(x)$ de x . Si cette solution est meilleure que x , ($f(x') < f(x)$) alors on accepte cette solution comme nouvelle solution x et on répète le processus jusqu'à ce qu'il n'y ait plus aucune solution améliorante dans le voisinage de x . Cette solution peut être: (1) un minimum global i.e. solution qui correspond à la solution optimale du problème ou (2) un minimum local i.e. une solution qui correspond à une solution optimale d'un sous ensemble de l'espace des solutions.

Définir un voisinage revient à définir une technique simple et rapide qui transforme une solution admissible x en une solution différente x' toujours admissible. Par exemple, le

voisinage peut être la réassignation des variables, la réassignation des opérations, la réallocation d'opérateurs ou de registres, de ré-ordonnement... Ainsi, dans le cas de la réassignation des variables, pour obtenir la nouvelle solution x' , nous choisissons une variable d_i assignée à un registre R_i et nous l'assignons à un registre existant R_j si toutes les variables déjà assignées à R_j sont compatibles avec la variable d_i . Deux variables sont compatibles si leurs durées de vie ne se chevauchent pas. Une fois l'assignation de la variable modifiée, si la surface totale diminue (par suppression d'un registre et/ou d'un multiplexeur) alors nouvelle solution est acceptée, sinon elle est rejetée.

Une autre version de la méthode de descente est la méthode de plus grande descente. Au lieu de choisir une solution x' dans le voisinage de x , on choisit toujours la meilleure solution x' du voisinage de x . Cette méthode nécessite, à chaque étape, d'explorer l'ensemble du voisinage et sera donc plus coûteuse en temps de calcul. Dans le cas de la réassignation des variables, une variable d_i assignée à un registre R_i sera « assignée » à tous les registres existants et seule la meilleure solution de réassignation sera conservée.

Les méthodes de descente simple et celle de plus grande descente décrites se basent sur une amélioration progressive de la solution et donc restent bloquées dans un minimum local dès qu'elles en rencontrent un. L'absence de technique de diversification est à l'origine de ce comportement. L'équilibre nécessaire entre intensification et diversification n'existe pas. Un moyen simple de diversifier la recherche peut consister à re-exécuter un des algorithmes de descente en prenant un autre point de départ. Comme l'exécution de ces méthodes est souvent très rapide; on peut inclure cette répétition au sein d'une boucle générale. On obtient alors un algorithme de type « descente Multi-start » [GLO 2002]. Il s'agit donc de générer différents points de départ, comme le montre la Figure 9 et d'exécuter une méthode de descente (descente simple, plus grande descente, recherche à voisinages variables...) à partir de ces différentes solutions initiales pour obtenir différentes solutions d'arrivée. La dernière étape consiste à retenir la meilleure solution. Pour trouver plusieurs solutions de départ, il suffit de jouer sur le résultat d'une des sous étapes du flot de synthèse qui génère la solution initiale.

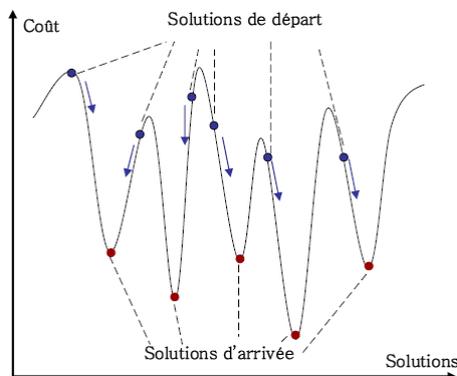


Figure 9 : Illustration de la méthode de « descente multi-start »

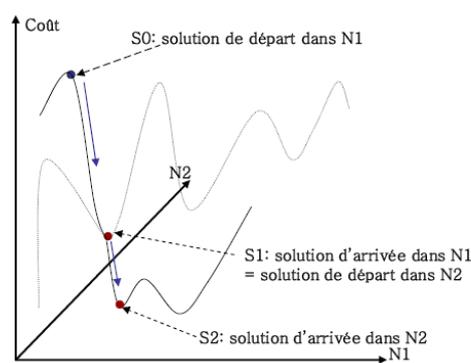


Figure 10 : Illustration de la méthode de recherche à voisinages variables VNS

La recherche à voisinages variables VNS, est une méthode relativement récente, basée sur la performance des méthodes de descente [GLO 2002]. Cette méthode suggère simplement d'utiliser plusieurs voisinages successivement dès qu'on se retrouve bloqué dans un minimum local. Ainsi, si l'on considère un ensemble de trois voisinages $N1$, $N2$ et $N3$. En partant d'une solution initiale, on applique une méthode de recherche locale (méthode de descente par exemple) jusqu'à arriver dans un minimum local. Ce minimum local pour le premier voisinage $N1$ est considéré comme solution initiale pour appliquer la méthode de recherche locale sur le voisinage $N2$ jusqu'à arriver dans un minimum (voir Figure 10). Ce minimum local pour $N2$ est considéré comme solution initiale pour

appliquer la méthode de recherche locale sur N3 jusqu'à arriver dans un minimum. Dans ce cas, le minimum local pour N3 est utilisé comme solution initiale pour N1 et ainsi de suite.

La méthode du GRASP a été introduite en 1989 par Feo et Resende puis présentée dans sa forme définitive en 1995 [GLO 2002]. Cette méthode d'exploration combine une heuristique gloutonne et une recherche locale. A chaque itération, on construit une solution en utilisant une heuristique gloutonne dont une partie est modifiée aléatoirement. Ensuite, cette solution est raffinée par l'intermédiaire d'une méthode de descente. Nous utilisons dans nos expériences l'algorithme du Left-Edge (LEA) pour optimiser le nombre de registres de la solution initiale. L'introduction d'une modification de façon aléatoire à l'algorithme LEA d'assignation des variables, consiste à tirer au sort un entier dans [0..100], si l'entier est inférieur à un certain seuil, l'assignation se fait comme le veut le LEA. Sinon, un nouveau registre doit être créé pour sauvegarder la variable.

Un grand nombre d'expériences a été réalisé pour comparer les différentes approches et les différents voisinages [TRA 2009]. La Figure 11 présente un résultat typique obtenu en appliquant aucune optimisation à la Solution Initiale (SI), une optimisation du nombre de registres avec l'algorithme du Left-Edge (LEA), une Descente Simple sur un voisinage de registre, une recherche multi-start combinée avec une descente simple sur voisinage de registre (MDS), une recherche multi-start combinée avec un VNS avec réassignation de registre et de variable comme méthode de descente (MS-VNS) et un GRASP. La Figure 12 présente les résultats après synthèse logique de la solution initiale, LEA et GRASP.

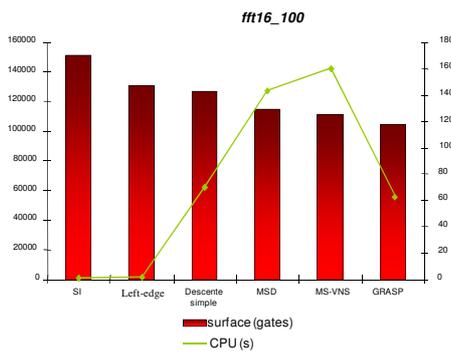


Figure 11 : Comparaison des différentes méthodes avant synthèse logique

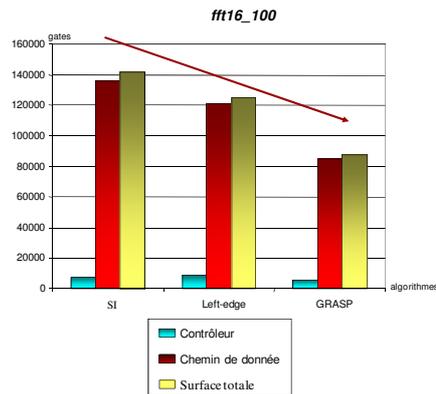


Figure 12 : Comparaison des différentes méthodes après synthèse logique

2.2.2. Synthèse d'architecture multi-modes

Afin de répondre aux besoins de performance, de flexibilité et de consommation des applications multi-fonctions et multi-standards modernes, nous avons proposé une approche de conception basée sur la synthèse de haut niveau automatisant la génération d'architectures multi-modes. Ces travaux ont été réalisés dans le cadre des thèses de Caaliph Andriamisaina [AND 2008] en co-encadrement avec Emmanuel Casseau (ENSSAT, Lannion) et de Cyrille Chavet [CHA 2007] en co-encadrement avec Eric Martin (Lab-STICC, Lorient) et Pascal Urard (STMicroelectronics, Crolles).

Une architecture multi-modes est conçue pour pouvoir exécuter plusieurs fonctions mutuellement exclusives dans le temps, c'est-à-dire qui s'exécutent à des instants différents. Les fonctions à réaliser peuvent différer sur le plan algorithmique, sur le format de données (précisions des calculs), sur le débit... Afin d'optimiser une architecture multi-mode, il faut favoriser la ressemblance des chemins de données des différents modes (opérateurs, registres et multiplexeurs) et des contrôleurs : il faut donc favoriser la création de comportements spatiaux temporels « semblables ». Pour cela, il est possible pour un graphe non encore traité (1) de suivre le plus possible les traces (i.e. les résultats d'allocation, d'ordonnancement...) des graphes préalablement synthétisés ou (2) de créer préalablement une trace commune représentant les ressources nécessaires et le comportement pour l'ensemble des graphes. La première méthode force ainsi le graphe en cours d'ordonnancement et d'assignation à suivre la trace des graphes déjà ordonnancés et assignés ce qui a pour effet de favoriser la ressemblance des étapes de contrôle de l'ensemble des graphes. Dans ce cas, le traitement du graphe courant ne prend toutefois pas en compte le traitement des graphes à venir. La seconde méthode permet, quant à elle, de créer une trace initiale globale pour l'ensemble des graphes à traiter. Cette trace globale permet ainsi d'augmenter la ressemblance entre les étapes de contrôle des graphes puisqu'elle prend en compte lors de l'ordonnancement et l'assignation du DFG courant les traces des graphes déjà ordonnancés et assignés ainsi que celles à priori des graphes non encore ordonnancés et non encore assignés (mais présente dans la trace commune puisque tous les graphes ont été prétraités pour créer la trace commune). Notons que quelque soit la méthode utilisée, il est nécessaire de trier les graphes afin de les traiter séquentiellement.

Dans notre approche, nous proposons un flot de conception issu du flot de synthèse de haut niveau conventionnel. La Figure 13 présente une vue générale de notre flot de conception. Le point d'entrée est une spécification fonctionnelle multi-modes dans laquelle un ensemble de fonctions mutuellement exclusives dans le temps est représenté. Cette description est, par la suite, compilée pour fournir un ensemble de DFGs.

L'étape suivante du flot consiste à trier les DFGs pour identifier le DFG qui servira de trace à suivre par les autres DFGs. Ce DFG est nommé DFG *principal* (*mDFG*), et les autres DFGs sont nommés DFGs *secondaires* (*sDFGs*). Le DFG principal est ordonnancé et assigné, en premier lieu, et les DFGs secondaires sont ordonnancés et assignés par la suite sur les traces créées par le *mDFG*. Les DFGs sont triés en fonction de leur parallélisme moyen par type d'opérations et de leur capacité à servir de trace (partage des ressources) pour les autres DFGs.

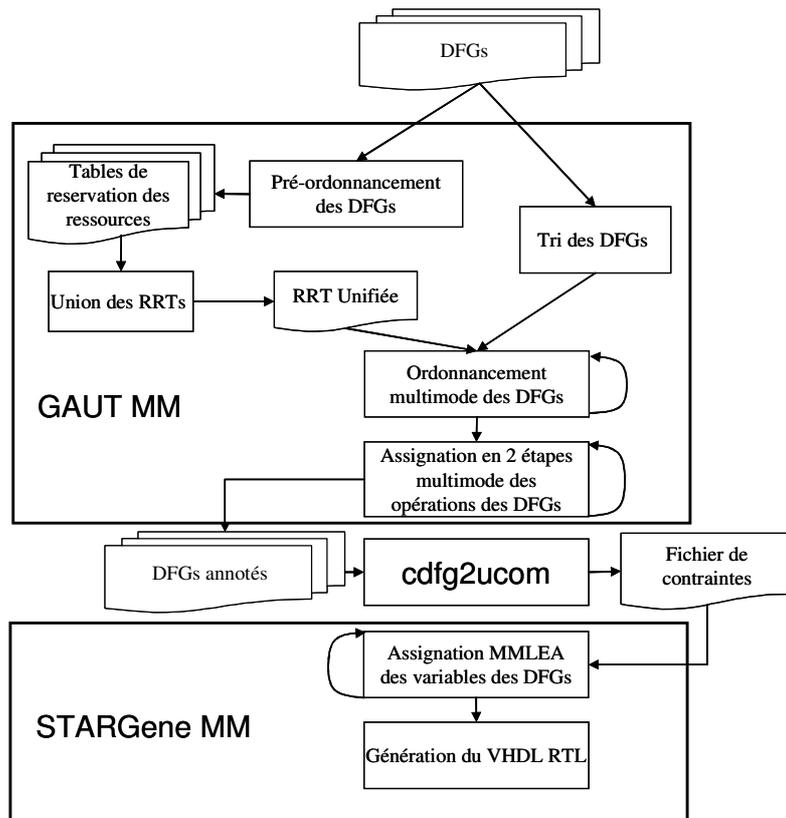


Figure 13: Vue générale du flot de conception proposé

L'étape d'allocation détermine initialement un nombre minimum de ressources (une borne inférieure), pour chaque type d'opérations, permettant a priori de respecter une contrainte de débit. Deux types d'allocation ont été considérés : (1) allocation incrémentale et (2) allocation unifiée.

L'allocation incrémentale remplit initialement une table de réservation de ressources RRT à l'aide de l'ordonnement du *mDFG*. Une table de réservation de ressources RRT est une table qui contient, pour chaque étape de contrôle, le nombre de ressources allouées et utilisables pour chaque type d'opérations. Cette RRT est ensuite mise à jour si nécessaire au fur et à mesure de l'ordonnement des DFGs suivants (*sDFGs*). L'allocation unifiée, quant à elle, détermine au préalable un nombre de ressources nécessaires pour ordonner l'ensemble des DFGs. Ce nombre de ressources nécessaires est obtenu par l'union des allocations, c'est-à-dire des ressources, de chacun des DFGs.

L'étape d'ordonnement débute par l'ordonnement du *mDFG* sous une contrainte de cadence et avec la prise en compte du résultat de l'allocation. Une fois le *mDFG* ordonné, les *sDFGs* sont ordonnés sous contraintes de cadence et tentent de respecter le plus possible la RRT. Cette dernière est mise à jour, si nécessaire, après ordonnement d'un *sDFG*.

L'étape d'assignation commence, selon le même principe, par l'assignation des ressources du *mDFG* à l'aide d'un graphe bipartite et d'un algorithme « Maximum Weighted Bipartite Matching (MWBM) » [PAP 1998]. Ensuite, les opérations des *sDFGs* sont assignées sur les ressources utilisées par les DFGs précédemment assignés. Après l'assignation des ressources de calcul, l'assignation et le partage des registres et la génération du circuit d'aiguillages (multiplexeurs) sont effectués.

Enfin, la dernière étape consiste en la génération du VHDL au niveau transfert de registres (RTL) de l'architecture multi-modes.

L'ensemble du flot présenté précédemment est fonctionnel et a été implémenté dans l'outil GAUT et dans l'outil STARGene détaillé ultérieurement dans ce document. Afin de montrer la pertinence de l'approche, un ensemble d'expériences a été réalisé sur des applications du domaine du traitement du signal. Un grand nombre de comparaisons a été réalisé sur la quantité d'opérateurs, de multiplexeurs, de registres mais aussi sur la complexité des contrôleurs, la consommation, la fréquence d'horloge et la surface totale en faisant varier les politiques d'allocation, d'ordonnancement et d'assignation (registre et opérations). La surface totale sera le seul résultat présenté ici (pour plus de détails sur les surfaces, performances et consommation voir [AND 2008] [AND 2010a]). Les résultats ont été comparés avec des méthodes de l'état de l'art : approche cumulative où chaque mode est synthétisé séparément (aucun partage) et approche multi-contrôleur [CHI 2005] où seul le chemin de données est mis en commun entre les différents modes. La Figure 14 présente les différentes combinaisons utilisées pour les expériences. Ces combinaisons dépendent des contraintes de débit et des applications de traitement du signal et de l'image : un filtre à réponse impulsionnelle finie (*FIR*), une transformée de Fourier rapide (*FFT*), une transformée en cosinus discrète (*DCT*), un produit de matrice (*PRODMAT*), une somme de la valeur absolue des différences et une somme des différences quadratiques (*SSD*).

	Combinaisons	Applications	Contraintes de cadence (ns)
Même algorithme, paramètres différents	combi1	SAD 4x4	400
		SAD 8x4	350
		SAD 8x8	380
	combi2	FIR7	110
		FIR11	150
		FIR15	190
		FIR19	230
	combi3	FIR16	200
		FIR32	360
		FIR64	690
		FIR128	1320
		FIR256	2600
Algorithmes différents	combi4	DCT-II (4x4)	1000
		FFT16	1500
		FIR128	1400
	combi5	DCT-II (4x4)	550
		PRODMAT (8x8)	1760
	combi6	FIR16	190
		FDCT-I (8)	220
	combi7	FFT16	580
		IFFT16	620
	combi8	FFT16	580
SSD 8x8		580	

Figure 14 : Combinaisons des applications et de leurs contraintes de cadence

Les résultats en termes de surface totale sont présentés dans la Figure 15. Comparé à l'approche cumulative, nous atteignons un gain en surface totale variant de 12% (*FFT16-IFFT16*) à 60% (*FIR19-15-11-7*). En analysant les différentes combinaisons, ce gain varie en fonction de la ressemblance entre les algorithmes constituant ces combinaisons. Pour celles formées d'algorithmes de structures quasi-identiques, tel qu'un même algorithme avec des paramètres différents, nous obtenons un gain variant de 23% à 60%. Par contre, lorsque les algorithmes formant une combinaison sont distincts comme la combinaison *FFT16-SSD8x8*, nous avons un gain allant de 12% à 19%.

Comparé à une approche multi-contrôleurs, le gain obtenu varie de 4% à 15% (*combi2*) et en moyenne, ce gain est de 8%. Par ailleurs, dans l'approche multi-contrôleurs [CHI 2005], les auteurs mettent en avant un gain en surface de 44% avec l'expérience sur le *combi2* par rapport à l'approche cumulative, alors qu'avec l'approche multi-contrôleurs que

nous avons implémentée, nous atteignons un gain en surface de 53%. Nous pouvons donc envisager que les gains en surface obtenus avec notre approche soient, en moyenne, de l'ordre de 15% comparée à la méthode proposée dans [CHI 2005].

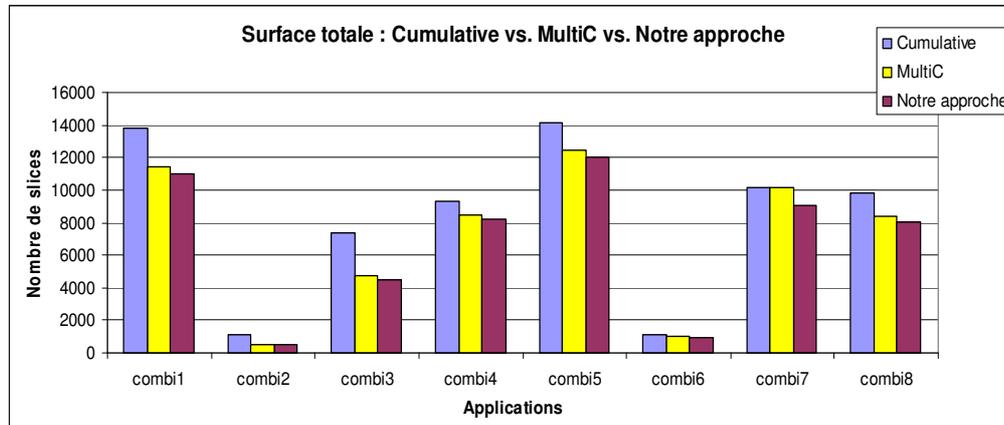


Figure 15 : Comparaison en surface totale

2.2.3. Synthèse hiérarchique d'architecture multi domaines d'horloge

La minimisation de la consommation des accélérateurs matériels sur FPGA est actuellement ciblée dans le cadre de la thèse de Ghizlane Lebreton-Lhairech en co-encadrement avec Eric Martin (Lab-STICC, Lorient). Une approche de synthèse de haut niveau, permettant de générer automatiquement des architectures ayant plusieurs domaines d'horloge et intégrant des mécanismes de gel d'horloge, est proposée.

Généralement, les architectures que l'on retrouve dans les FPGAs pour réaliser des chaînes de traitements intensifs ne contiennent qu'un seul domaine d'horloge. L'horloge doit donc être répartie sur la totalité du circuit afin d'alimenter l'ensemble des composants nécessitant le signal d'horloge (registres, opérateurs « pipelinés »...). Ceci aboutit à l'utilisation d'un réseau complexe de distribution d'horloge qui s'étend sur une large surface. De plus, cet arbre de distribution de l'horloge est, bien souvent, toujours actif bien que toutes les parties (ressources) du circuit ne soient pas utilisées simultanément. Malheureusement, la consommation dans les FPGAs est classiquement liée à la tension d'alimentation, à la fréquence d'horloge et au taux d'activité [PIG 2005]. Cette consommation est dominée par les interconnexions (45% de la puissance totale) et par le réseau de distribution d'horloge (40%) alors que les blocs logiques ne représentent, quant à eux, que 15% de la consommation totale [DEG 2005]. Ainsi, les interconnexions et le réseau de distribution d'horloge sont les principaux contributeurs à la consommation.

Nous visons la conception faible consommation des applications de traitement numérique du signal (TNS) sur FPGA en utilisant la synthèse de haut niveau hiérarchique afin de générer automatiquement des architectures constituées de plusieurs « blocs » ayant chacun leur propre horloge. De plus, grâce aux composants de gestion d'horloges (Digital Clock Manager DCM chez Xilinx [XIL 20011] ou Phase-Locked Loop PLL chez Altera [ALT 2011]) présents dans les FPGAs actuels, la communication entre les blocs peut être rendue synchrone. L'approche que nous proposons permet (1) de réduire la fréquence d'horloge dans certaines parties du circuit, (2) de mettre en œuvre le gel d'horloge, (3) de réduire le nombre de longs fils dans le circuit final au travers de la hiérarchie, (4) de réduire la complexité du réseau d'horloge grâce à l'utilisation de plusieurs domaines d'horloge et (5) de réduire le surcoût en surface par l'introduction d'une nouvelle architecture synchrone

multi-horloge au lieu d'utiliser des architectures Globalement Asynchrones Localement Synchrones GALS [MUT 2000].

Notre flot de synthèse de haut niveau, présenté dans la Figure 16, utilise en entrée une spécification décrite en langage C ainsi qu'une contrainte de débit exprimée sous la forme d'une cadence (i.e. un intervalle d'initiation en nombre de cycle) et d'une période d'horloge nommée *horloge système*. La spécification d'entrée peut contenir des *opérations élémentaires* spécifiées en utilisant les symboles du langage (+,-,...) et des *opérations composites* exprimées par des appels de fonctions. La bibliothèque d'opérateur contient des opérateurs simples et complexes. Les opérateurs simples sont utilisés pour implémenter des opérations simples (registres, multiplexeurs, additionneurs...). Les opérateurs complexes sont utilisés pour implémenter des opérations composites i.e. les fonctions. Un composant complexe est un composant séquentiel composé d'un contrôleur et d'un chemin de données.

Nous traduisons dans un premier temps la spécification initiale en un graphe flot de données hiérarchiques (H-DFG) en fonction des directives de compilation : le concepteur peut (1) mettre en ligne les fonctions en utilisant dans le code source le mot clé *inline* ou (2) conserver les appels de fonctions. Lorsqu'il existe au moins une fonction non mise en ligne, nous effectuons une synthèse hiérarchique. Le H-DFG qui est généré contient dans ce cas des nœuds hiérarchiques représentant les appels de fonction. L'étape d'allocation définit ensuite le nombre d'opérateurs a priori nécessaires (borne inférieure) à mettre en œuvre pour respecter la contrainte de débit. L'ordonnancement repose sur un algorithme de liste où la mobilité des opérations est utilisée comme critère de priorité [COU 2008b]. L'assignation des opérations sur les opérateurs et des données dans les registres est réalisée à l'aide d'un algorithme d'appariement maximal de graphe bipartite pondéré « Maximum Weighted Bipartite Matching (MWBM) » [PAP 1998] avec comme objectif la minimisation du coût des interconnexions (multiplexeurs).

Afin d'optimiser la consommation, notre approche synthétise dans un premier temps chaque fonction sous la forme d'un opérateur complexe si celui-ci n'est pas déjà présent dans la bibliothèque. Si cet opérateur complexe contient des opérateurs simples multi-cycles, alors son horloge sera ralentie. En effet, les opérateurs multi-cycle requièrent par définition plus d'un cycle pour délivrer leurs résultats et la réduction de la fréquence de l'horloge pour ces opérateurs n'impacte pas leur latence. De ce fait, les opérateurs complexes sont synthétisés en amont de l'application et sont cadencés avec des fréquences d'horloge plus faibles que celle de l'horloge système quand cela est possible. Nous obtenons ainsi, une synthèse automatique d'architectures hiérarchiques à horloges multiples.

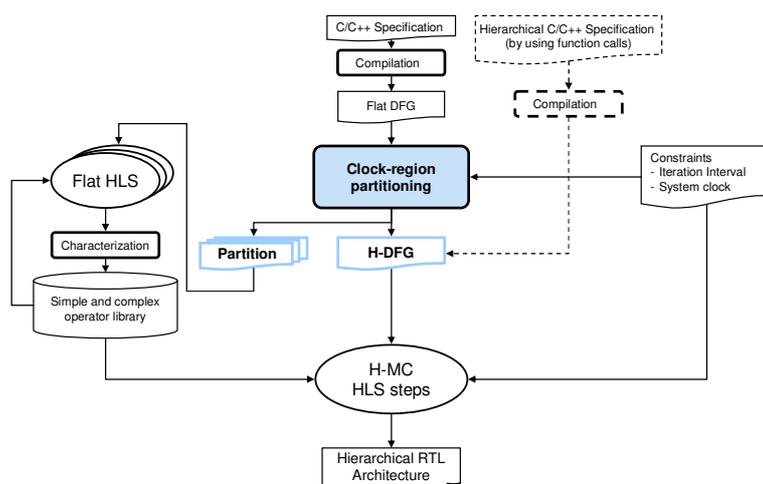


Figure 16 : Flot de synthèse proposé

Un opérateur complexe a son propre contrôleur et son chemin de données. Il peut donc fonctionner indépendamment de son environnement. Par conséquent, lors de la génération de l'architecture RTL, nous générons un signal de validation de la commande afin d'activer l'opérateur complexe lorsque ce dernier doit fonctionner. Ce même signal de validation est utilisé pour geler l'horloge du composant lorsqu'il est inactif. La communication synchrone entre un composant complexe et son environnement se fait grâce à des registres générés lors de l'étape d'assignation des variables. Afin d'assurer la synchronisation entre différents domaines d'horloge, nous utilisons des ressources dédiées fournies par les FPGA actuels [XIL 2001][ALT 2011]. Ce dispositif spécifique est représenté par la « boîte » en pointillé en Figure 17 appelé DCM (Digital Clock Manager). L'architecture ainsi générée est nommée architecture à contrôle distribué et à horloges multiples (DC-MuC). Elle est composée de plusieurs niveaux hiérarchiques. Le niveau hiérarchique le plus haut fonctionne à la fréquence d'horloge spécifiée par le concepteur (*horloge système*). Chaque niveau hiérarchique est cadencé à une fréquence d'horloge sous-multiple de la fréquence du niveau hiérarchique qui lui est supérieur.

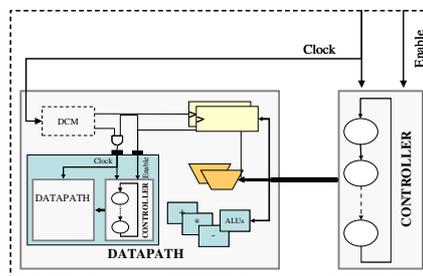


Figure 17 : Architecture à contrôle distribué et à horloges multiples

Nous avons validé notre approche en utilisant plusieurs applications du domaine de traitement numérique de signal. Nous avons synthétisé pour cela plusieurs architectures pour chaque application, afin de comparer notre approche à celles proposées dans l'état de l'art. Nous avons ainsi utilisé pour nos comparaisons (Figure 18) : (1) des architectures mono-horloge non hiérarchiques (*Flat*) ; (2) des architectures à mono-horloge dans lesquelles les fonctions sont implémentées par des chemins de données pilotés par un contrôleur centralisé (*CC-MoC*) ; (3) des architectures à multi-horloge dans lesquelles les fonctions sont implémentées par des chemins de données ayant leur propre horloge mais pilotés par un contrôleur centralisé (*CC-MuC*) ; (4) des architectures insensibles à la latence (*LIS*) ; (5) des architectures globalement asynchrone et localement synchrone (*GALS*) ; (6) des architectures à mono-horloge dans lesquelles les fonctions sont implémentées par des chemins de données pilotés par leur propre contrôleur (*DC-MoC*) , (7) des architectures à multi-horloge dans lesquelles les fonctions sont implémentées par des chemins de données ayant leur propre horloge et leur propre contrôleur (*DC-MuC*).

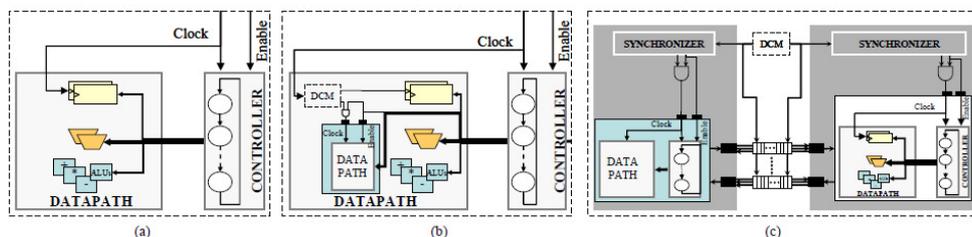


Figure 18 : Architecture (a) Flat, (b) mono ou multi-horloge avec contrôleur centralisé, (c) LIS ou GALS

Afin de générer les signaux d'horloge de chaque bloc, nous avons utilisé les gestionnaires de l'horloge DCM disponibles sur notre cible Xilinx XC5VLX110. Un DCM consomme 44mw à 250Mhz±0.2mW/Mhz. La consommation a été mesurée en utilisant un analyseur de puissance DC qui fournit une tension/courant programmable ainsi que des dispositifs de mesure.

Les applications de traitement du signal que nous avons, entre autres, utilisées pour nos premières expériences sont les suivantes: une FFT (Fast Fourier Transform) et une DCT (Discrete Cosine Transform) avec plusieurs complexités de calcul. Nous avons fait varier les tailles de la DCT de 16*16 à 32*32 et de la FFT de 16 à 32 points. La contrainte de cadence a été fixée à 60 cycles et la période d'horloge système à 10ns. La Figure 19 et la Figure 20 présentent respectivement les mesures de consommation et les surfaces des différentes architectures. Les résultats que nous avons obtenus montrent que la distribution du contrôle dans l'architecture DC-MuC, que nous proposons de générer automatiquement, réduit la consommation et la surface du circuit final (5% de réduction de la surface par rapport à l'architecture « Flat » et jusqu'à 11% de réduction de puissance). L'architecture GALS réduit la consommation (jusqu'à 7%), mais présente 50% de surcoût en surface suite à l'utilisation de FIFOs et d'un processeur de synchronisation [MUT 2000].

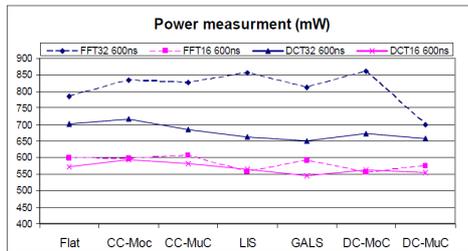


Figure 19 : Comparaison en consommation

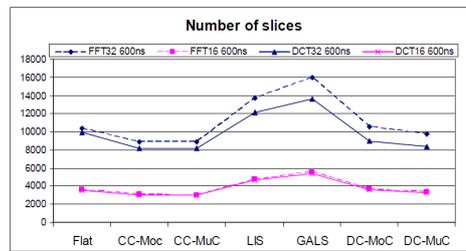


Figure 20 : Comparaison en surface

Les connexions dans les FPGA Xilinx se répartissent en deux catégories nommées « longues » et « courtes ». Les longs fils regroupent les fils de type VLONG, HLONG et PENT alors que les fils courts regroupent les fils de type GENERIC, DOUGLE et GLOBAL. Dans l'architecture DC-MuC que nous proposons, les longs fils sont réduits jusqu'à 29% par rapport à une architecture plate pour la FFT 32 points (voir Figure 21).

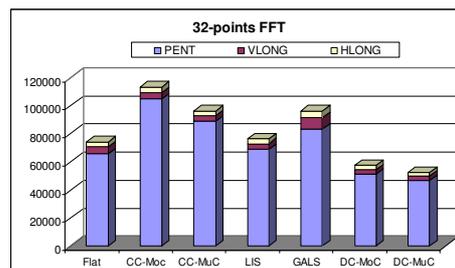


Figure 21 : Nombre de fils pour la FFT 32 points (600ns)

2.3. Conclusion

La synthèse automatique d'architecture, aussi nommée synthèse de haut niveau, est un processus extrêmement complexe qui requiert de nombreuses compétences afin d'aboutir à la définition de solutions efficaces. A travers les travaux menés dans cet axe de recherche, nous avons tenté de répondre à la problématique, largement traitée depuis plus d'une vingtaine d'année, de façon originale : la conception d'architectures multi-modes, la génération automatique d'architectures mutli-horloges et l'optimisation des architectures par l'utilisation de métaheuristiques, l'introduction automatique de la prédiction de branchement dans les architectures générées... L'ensemble des travaux a été intégré dans l'outil GAUT qui est diffusé sous licence CECILL-B (open source). Cet outil a été téléchargé respectivement 120, 145 et 180 fois en 2008, 2009 et 2010 dans plus de 50 pays par des académiques (majoritaires) et des industriels. L'outil est évalué et utilisé dans différents pays à des fins pédagogiques (Cours, TP), à des fins de recherche (thèses, projet) ou à des fins de conception et de prototypage (industriels).

Afin de mener à bien ces travaux, trois doctorants, un post-doctorant, deux ingénieurs et quatre stagiaires de DEA ou Master Recherche ont participé ou participent actuellement au projet.

Les travaux menés au sein de cet axe de recherche (hors publications réalisées durant la thèse) ont conduit à quinze communications scientifiques réparties comme suit : trois revues internationales [AND 2010a] [COU 2009a] [COU 2009b], deux chapitres de livre [TRA 2010a] [COU 2008b], six conférences internationales [LHA 2010b] [COU 2009c] [AND 2007a] [CHA 2007a] [COU 2005b] [COU 2004a], quatre conférences nationales [TRA 2008a] [LAP 2011a] [LHA 2011a] [LAU 2007a]. Les références sont détaillées à la fin de la section 2.4.

2.4. Fiche de synthèse des travaux

▪ **Co-encadrement de doctorants**

Ghizlane LHAIRECH-LEBRETON, 2007-2011 (Co-encadrement Eric Martin, 50%)
Synthèse de haut niveau pour la conception d'architecture faible consommation

Kods TRABELSI, 2006-2009 (Co-encadrement Marc Sevaux, 50%)
Méthodes d'optimisation pour la conception sous contraintes de systèmes et de circuits électroniques

Caaliph ANDRIAMISAINA, 2004-2008 (Co-encadrement Emmanuel Casseau, 50%)
Flot de conception dédié aux architectures multi-modes pour les applications de traitement du signal et de l'image

▪ **Encadrement de stagiaires de DEA et de Master**

Vianney Lapotre 2009/2010
Introduction de la prédiction de branchement dans la synthèse de haut niveau
Master recherche I-MARS

Aroua Briki 2008/2009

Synthèse de haut niveau hiérarchique : étude de cas d'un encodeur-décodeur Reed Solomon

Master recherche I-MARS

Hicham Lalaoui Hassani, 2007/2008

Impact des étapes de synthèse de haut niveau dans l'environnement GAUT

Master recherche I-MARS

Moahamed Aabidi, 2007/2008

Synthèse de haut-niveau : Etude de cas d'un algorithme Maximum A Posteriori pour la conception d'un Turbo décodeur

Master recherche I-MARS

▪ **Collaborations scientifiques**

SoCKET : SoC toolKit for critical Embedded sysTems

Type : Projet direction générale de la compétitivité, de l'industrie et des services DGCIS

Consortium : Airbus, Astrium, CNES, STMicroelectronics, Thales R&T, Schneider Electric Industries, PSI-S, CEA-LETI, Magilem Design Services, INPG-TIMA, UPS-IRIT, UBS-Lab-STICC

Durée : 2008 – 2011

SoCLib : open platform for virtual prototyping of multi-processors system on chip

Type : Projet plate-forme l'ANR/RNTL

Consortium : STMicroelectronics, Thales Communications, Thomson Silicon Components, Prosilog, TurboConcept, Silicomp, UPMC/LIP6, ENST, UPMC/LISIF, CEA LIST, INRIA Futurs, IRISA, LESTER, IETR INSA, TIMA, CEA LIST, CITI

Durée : 2006-2009

ONAGRE : envirOnnement de coNception et de prototypAGE d'applications Radio et mobile

Type : Contrat CRE (Contrat de Recherche Externalisée)

Consortium : France Telecom R&D (Issy les moulineaux), LESTER

Durée : 2005-2008

FLASH : Seed Optimisation and Indexing of Genomic Databases

Type : ARC INRIA

Consortium : IRISA, LIFL, LESTER, INSERM U694.

Durée : 2006-2007

Synthèse de haut niveau pour la conception faible consommation sur FPGA

Université de Californie, Los Angeles UCLA

Type : fond propre

Consortium : Cad-Lab UCLA, Lab-STICC

Durée : 4 mois en 2009

▪ **Publications scientifiques**

Revues internationales

- [AND 2010a] C. Andriamisaina, P. Coussy, E. Casseau, C. Chavet, “**High-Level Synthesis for Designing Multi-mode Architectures**”, *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems (TCAD)*, Vol. 29, Issue 11, pp. 1736-1749n, November 2010.
- [COU 2009a] P. Coussy, G. Gajski, A. Takach, M. Meredith, “**An Introduction to High-Level Synthesis**”, *Special issue on High-Level Synthesis, IEEE Design and Test of Computers*, Vol. 26, Issue 4, July/August, 2009.
- [COU 2009b] P. Coussy, A. Takach, “**Raising the Abstraction Level of Hardware Design**”, *Special issue on High-Level Synthesis, IEEE Design and Test of Computers*, Vol. 26, Issue 4, July/August, 2009.

Chapitres de livre

- [COU 2008b] P. Coussy, C. Chavet, P. Bomel, D. Heller, E. Senn, E. Martin, “**GAUT: A High-Level Synthesis Tool for DSP applications**”, *High-Level Synthesis: From Algorithm to Digital Circuits*, Springer, Berlin, Germany, 2008.
- [TRA 2010a] K. Trabelsi, M. Sevaux, P. Coussy, A. Rossi, K. Sörensen, “**Advanced Metaheuristics for High-Level Synthesis**”, *In Metaheuristics. Springer*, 2010.

Conférences internationales

- [LHA 2010b] G. Lhairech-Lebreton, P. Coussy, E. Martin, “**Hierarchical and Multiple-Clock Domain High-Level Synthesis for Low-Power Design on FPGA**”, *IEEE International Conference on Field Programmable Logic and Applications (FPL)*, 2010.
- [COU 2009c] P. Coussy, A. Rossi, M. Sevaux, K. Sörensen, and K. Trabelsi, “**VNS for High Level Synthesis**”, *In Proceedings of 8th Metaheuristics International Conference, MIC 2009*, July 2009.
- [CHA 2007a] C. Chavet, C. Andriamisaina, P. Coussy, E. Casseau, E. Juin, P. Urard, E. Martin, “**A Design Flow Dedicated to Multi-mode Architectures for DSP Applications**”, *IEEE International Conference on Computer Aided Design (ICCAD)* 2007.
- [Andria07a] C. Andriamisaina, P. Coussy, E. Casseau, “**Synthesis of Multimode digital signal processing systems**”, *NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*, 2007.
- [COU 2005b] P. Coussy, G. Corre, P. Bomel, E. Senn, E. Martin, “**High-level synthesis under I/O Timing and Memory constraints**”, *IEEE International Symposium on Circuits And Systems (ISCAS)*, 2005.

[COU 2004a] P. Coussy, D. Gnaëdig, A. Nafkha, A. Baganne, E. Boutillon, E. Martin, “**A Methodoly for IP integration in DSP Soc: a case study of a MAP algorithm for turbo decoder**”, *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2004.

Conférences nationales

[LHA 2011a] G. G. Lhairech-Lebreton, P. Coussy, E. Martin, “**Synthèse d’Architecture Multi-horloges pour la Conception Faible Consommation sur FPGA**”, *Colloque sur le Traitement du Signal et de l’Image (GRETSI)*, Septembre 2011.

[LAP 2011a] V. Lapotre, P. Coussy, C. Chavet, “**Prédiction de Branchement dans la Synthèse de Haut Niveau**”, *14^{ième} SYMPosium en Architecture (SYMPA)*, 10-13 Mai 2011.

[TRA 2008a] K. Trabelsi, P. Coussy, A. Rossi, M. Sevaux, “**Ordonnement et Assignment en Synthèse de Haut Niveau**”, *9ieme congrès de la Société Française de Recherche Opérationnelle et d’Aide à la Décision (ROADEF)* Février, 2008.

[LAU 2007a] J. Laurent, P. Coussy, “**Impact du type d’architecture sur la consommation d’une application**”, *Journées Faible Tension Faible consommation (FTFC)*, 21-23 mai, 2007.

3. Axe 2 : Modélisation et prototypage

3.1. Introduction

Cet axe de recherche concerne la modélisation et l'intégration de la synthèse d'architecture dans les approches de conception de niveau « système ». Les travaux réalisés portent sur le raffinement algorithmique, le prototypage virtuel, l'intégration d'accélérateur dans des architectures MPSoC, l'exploration de l'espace de conception...

La complexité des architectures augmente avec la densité d'intégration tout comme la complexité des systèmes s'accroît avec notre capacité à créer et intégrer des applications toujours plus complexes. A cela s'ajoute la difficulté de conception et de réalisation liée à la nécessité de maîtriser des objectifs tels que la consommation, les performances, la robustesse, la surface... De ce fait, l'automatisation est requise pour aider les concepteurs à gérer et à maîtriser la complexité des systèmes actuels et leur permettre de focaliser leur attention sur les décisions qui auront le plus d'impact sur la qualité des résultats. Le rehaussement des niveaux d'abstraction des parties logicielles et matérielles est nécessaire pour autoriser une exploration efficace de l'espace de conception, faisant de ce fait de la simulation et de la synthèse, les clés de voutes des approches de conception nommées de nos jours ESL (« Electronic System Level »). Cette constante évolution de l'abstraction s'est accompagnée de la création de nouveaux langages tels que SystemC ou SystemVerilog. Dès la fin des années 90 et le début des années 2000, les équipes de recherche ont défini des langages de spécification dérivés du C pour décrire des systèmes (i.e. les parties logicielles bien sûr mais aussi les parties matérielles) avec pour objectif la simulation [LIA 1997] [PET 1997] [LAV 1999] [COC 2000] et la synthèse [AUB 1995] [GAJ 2000]... Un consensus a en partie été trouvé sous la forme du SystemC. Ce langage, devenu standard IEEE 1666 en 2005 a évolué depuis sa création et propose maintenant une bibliothèque de modélisation de niveau transactionnel (TLM 2.0) ainsi qu'un ensemble d'extension pour modéliser des parties analogiques (SystemC AMS).

Dans ce contexte, la synthèse d'architecture a dû évoluer pour considérer ces nouveaux langages et niveau d'abstraction mais aussi de nouvelles approches de conception. Les outils ont vu leurs langages d'entrée évoluer pour passer de langages propriétaires ou de description de matériel (VHDL, Verilog) à des langages principalement dérivés du C [MAR 2009]. De plus, la génération de modèles VHDL/Verilog RTL visant la synthèse logique s'accompagne désormais de la fourniture de modèles de simulation (généralement en SystemC) plus ou moins abstraits (précis au cycle ou à la transaction) pour permettre une exploration architecturale de l'ensemble du SoC via du prototypage virtuel. Les modèles de simulation peuvent aussi être utilisés dans certains cas comme spécification fonctionnelle pour la synthèse d'architecture. Les spécifications algorithmiques sont réalisées dans ces langages à l'aide de type de données dits « matériels » offrant la possibilité d'utiliser des variables, signées ou non, « entières bit près » ou en « virgule fixe »...

Les travaux de recherche de cet axe ont été développés dans le cadre de différents projets. Ainsi :

- la modélisation formelle des contraintes temporelles aux entrées / sorties (comportement aux interfaces) pour la synthèse de composants virtuels algorithmiques

- a tout d'abord été abordée durant ma thèse. Les prises en compte simultanées des contraintes mémoire et d'E/S ont été étudiées durant mon année d'ATER (2000-2004),
- la modélisation, à l'aide du langage SystemC, en vue d'une simulation avant synthèse des composants a été explorée durant le projet RNTL SystemC'Mantic ainsi que dans le cadre des travaux de thèse de Farhat Thabet (2004-2007),
 - la modélisation d'application de traitement du signal et la synthèse d'architecture prenant en compte des spécifications bit-près ont été abordées dans le cadre du contrat de recherche externe (CRE) France Telecom ONAGRE (2005-2008),
 - la modélisation et la génération de modèles de simulation cycle-près ou transactionnel par des outils de synthèse de haut niveau a été adressée plus récemment dans le cadre du projet plateforme ANR SocLib (2006-2009) et dans le cadre du post-doctorat de Caaliph Andriamisaina (2008-2009),
 - l'utilisation conjointe de la compilation et de la synthèse de haut niveau dans un flot de conception automatisé visant les architectures MPSoC est étudiée dans la thèse en co-tutelle de Paolo Burgio (2010-2013).

Les travaux de recherche effectués dans le cadre de ma thèse, poursuivis et étendus durant mon année d'ATER ne seront pas décrits dans ce document.

3.2. Présentation des travaux

3.2.1. Exploration algorithmique et prototypage FPGA

3.2.1.1. Flot de conception

L'exploration algorithmique a été abordée dans le cadre du projet privé ONAGRE en collaboration avec France Telecom FT- division R&D (projet initié en 2005 par E. Casseau et E. Martin et dont j'ai eu la responsabilité en 2006). Le pôle "Communications Numériques" de FT R&D développe des techniques radio, de codage et d'accès novatrices. Ces travaux de recherche nécessitent non seulement des simulations algorithmiques, mais également la réalisation de prototypes permettant un fonctionnement temps réel des applications. Ces prototypes sont généralement réalisés à l'aide de solutions technologiques configurables de type FPGA.

Le projet avait pour but de proposer un flot complet de conception et de prototypage sur plateforme reconfigurable. Il s'agissait à terme de disposer d'un flot de conception complet partant de la spécification algorithmique parallèle d'une chaîne de traitement du signal précise au bit près et aboutissant à la chaîne de traitement matérielle toute ou en partie implantée dans un FPGA. Le flot devait reposer sur un environnement de spécification et de simulation de « type COSSAP » [SYN 2011] et sur la synthèse de haut niveau. Le modèle formel sous jacent était de type « flot de données synchrones » (Synchronous Data Flow SDF) [LEE 1987] et chaque acteur devait pouvoir être implémenté sous la forme d'un composant matériel. Ce modèle formel couplé à une approche de spécification graphique ont été retenus par FT R&D pour leur adéquation avec le domaine cible et leur utilisation très répandue par les algorithmiciens et les ingénieurs application. Ce flot complet présente l'avantage d'utiliser une spécification algorithmique en langage C (facile d'exploitation par les algorithmiciens), une spécification schématique de l'application et offre de plus la possibilité d'obtenir des estimations matérielles rapides même pour un non expert en électronique numérique.

Le flot repose sur les deux outils logiciels suivants : l'environnement Galacsy, développé par un consultant extérieur, et utilisé pour la spécification algorithmique par FT R&D et

l'outil de synthèse de haut niveau GAUT utilisé pour la génération des parties matérielles synthétisables. L'utilisateur décrit son application au niveau fonctionnel avec l'environnement Galacsy et valide cette application par simulation. Les parties synthétisables sont automatiquement extraites des descriptions SystemC générées par Galacsy et l'outil GAUT se charge de générer l'architecture matérielle correspondante. Cette automatisation évite une étape fastidieuse de spécification des architectures matérielles réalisées manuellement auparavant et donc très sujette aux erreurs. La validation de ces architectures est ensuite réalisée sur une plate-forme de prototypage à base de FPGA.

La spécification simulable obtenue à partir de l'environnement de spécification Galacsy comporte un ensemble de modules et de processus SystemC dont une partie seulement correspond à l'algorithme à synthétiser (typiquement la partie calculatoire et la définition des entrées/sorties associées). C'est cette partie que l'outil GAUT traite afin de fournir la description matérielle correspondante. Une approche d'identification et d'extraction automatique des parties pertinentes a donc été proposée. Le langage de spécification SystemC a été dans un premier temps retenu pour sa facilité à exploiter les types matériels de données et à présenter une entrée potentielle pour les flots de conception matérielle. Toutefois, les types de données « Algorithmic C » de la société Mentor Graphics [MEN 2011] ont finalement été retenus par FT R&D aux vues de leurs meilleures performances en simulation pour spécifier les algorithmes des parties calculatoires. Sans entrer dans les détails, cette librairie permet au concepteur de définir des variables, signées ou non, « entières bit près » et « virgule fixe » en utilisant les types *ac_int* et *ac_fixed*. Tout comme SystemC, cette librairie fournit aussi toutes les opérations arithmétiques ainsi que différentes fonctions de quantification (arrondi, troncature...) et de gestion de débordement (saturation, wrap-around...). Par exemple, une variable de type *ac_fixed* <5,2,true,AC_RND,AC_SAT> est un nombre signé en virgule fixe de la forme bb.bbb (5 bits de largeur total dont 2 bits avant la virgule) avec un mode de quantification de type arrondi et un mode de gestion de débordement de type saturation.

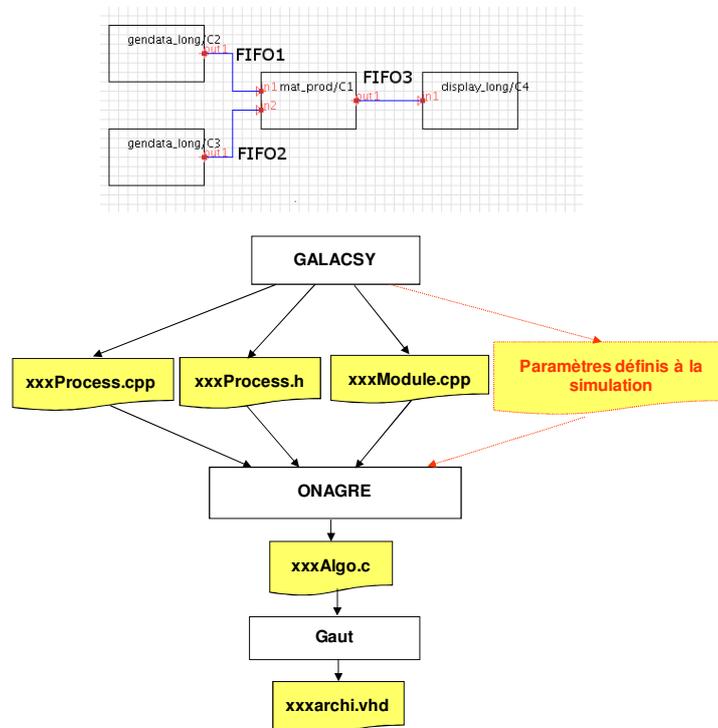


Figure 22 : Flot de conception pour l'exploration algorithmique dans ONAGRE

L'ensemble du flot partant de la spécification d'un acteur SDF jusqu'à sa réalisation matérielle a été validé au travers, entre autres, d'un algorithme Levinson-Durbin [HAY 1993] et d'une carte FPGA APEX 20ke de chez Altera. Malheureusement, pour des raisons de confidentialité, les résultats de synthèse, toutefois comparables en surface et en performances à ceux obtenus à titre comparatif dans le projet par un outil de synthèse de haut niveau commercial, ne seront pas présentés dans ce document.

3.2.1.2. Prise en compte d'une spécification bit près pour la synthèse

Afin de prendre en compte la spécification bit près réalisée à l'aide des types de données « Algorithmic C », le front-end ainsi que la quasi-totalité des étapes de synthèse implémentées dans l'outil GAUT ont dû être modifiées (caractérisation de la bibliothèque, ordonnancement, assignation, génération de l'architecture) et des étapes ont dû être ajoutées (analyse/raffinement, « clustering », dimensionnement des opérateurs). La Figure 23 et la Figure 24 présentent le flot supportant ou non une spécification bit près.

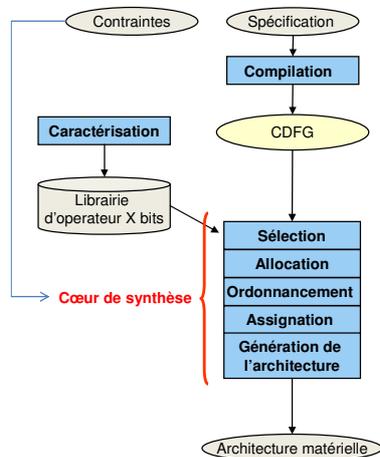


Figure 23 : Flot de synthèse non bit près

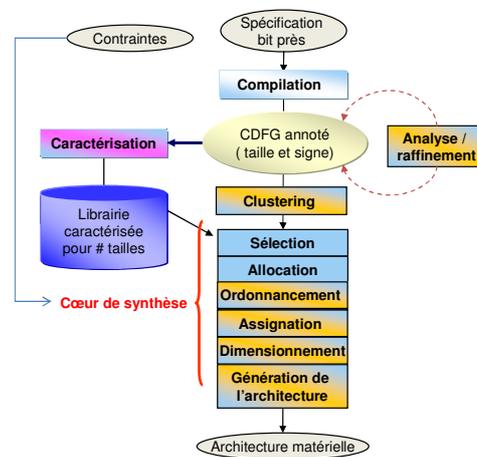


Figure 24 : Flot de synthèse bit près

Le point d'entrée du flot modifié est une spécification déjà raffinée utilisant les types « Algorithmic C » de Mentor Graphics. Bien qu'il puisse aider le concepteur durant la tâche de raffinement en fournissant des estimations de surface par exemple, l'outil ne réalise pas la transformation flottant vers fixe. La description générée par l'étape de compilation est un DFG dans lequel chaque nœud est annoté avec des informations telles que la taille et le type des variables pour les nœuds de données.

La phase d'analyse et de raffinement (voir Figure 24) permet de dimensionner les variables créées dans la représentation intermédiaire (DFG) par la phase de compilation (par exemple l'expression $a = b+c+d$ sera traduite en $tmp = b+c$ suivi de $a=tmp+d$). Cette phase permet également de définir le nombre de bits utilisés pour représenter les constantes qui ont toutes par défaut la taille d'un entier (taille fixée à la compilation par gcc, outil sur lequel est basé le front-end de GAUT). Elle réalise enfin, d'une part la propagation des besoins de définition par une passe avant partant des entrées et d'autre part, la réinjection des besoins d'utilisation afin d'éliminer les bits inutiles par une passe arrière en partant des sorties. Les données d'entrée/sortie étant définies par le concepteur ne sont pas modifiées par les algorithmes de propagation, néanmoins, des alertes sont générées par l'outil d'analyse en cas de sous ou surdimensionnement.

Le groupement d'opérations nommé « clustering » consiste à définir des groupes d'opérations pouvant partager un opérateur. A des fins expérimentales, plusieurs types de groupement ont été proposés : par fonction mathématique (addition...), par taille des opérands ou par temps de propagation (en nombre de cycle).

Les modifications apportées à la phase d'ordonnancement concernent la fonction de priorité de l'algorithme d'ordonnancement par liste utilisé dans GAUT. Les deux premières solutions trient les opérations « ordonnancables » par leurs mobilités puis pour une même mobilité par leurs tailles (1) croissantes ou (2) décroissantes. La troisième solution est *mixte* et considère la mobilité et la taille des opérands pour le calcul de la priorité. Un paramètre permet de modifier le poids de la mobilité ou de la taille dans ce calcul.

L'étape d'assignation définit l'opérateur qui exécutera l'opération et a, de ce fait, un grand impact sur la surface finale de l'architecture. Dans l'ancien flot, l'assignation d'une opération sur un opérateur n'était pas contrainte en taille et seule la disponibilité de l'opérateur importait. Plusieurs politiques d'assignation ont été définies : (1) simultanément à l'ordonnancement, l'opération est assignée à l'opérateur dont la taille courante est la plus proche de celle de l'opération ; (2) suite à l'ordonnancement, les opérations exécutées dans un même cycle sont assignées par ordre croissant ou (3) par ordre décroissant de largeur; (4) après l'ordonnancement, les opérations exécutées dans un même cycle sont assignées simultanément à l'aide d'un algorithme « Maximum Weighted Bipartite Matching (MWBM) » [PAP 1998] appliqué sur un graphe bipartite.

La fusion des registres peut être réalisée en utilisant un algorithme du Left-Edge (LEA) ou un algorithme MWBM prenant en compte la taille des données et la complexité des multiplexeurs engendrés par l'assignation des variables aux registres.

Durant la phase de dimensionnement, la taille d'un opérateur est définie en fonction des opérations qui lui sont assignées. Pour qu'un opérateur exécute correctement une opération, la taille de ses signaux d'entrées doit être au minimum égale à la taille de celles de l'opération. Toutefois, les tailles des opérands d'une même opération peuvent être différentes. Ainsi, nous avons défini trois calculs différents. Le premier est utile comme point de comparaison avec les méthodes de la littérature qui considèrent que les opérands d'une opération ont une même taille. En revanche, le deuxième calcule la largeur maximale pour chaque entrée en respectant l'ordre des opérands (sans commutativité). Enfin, le dernier permet de trouver la taille optimale de l'opérateur en calculant la largeur maximale des entrées (avec commutativité).

La Figure 25 présente un exemple typique des résultats de synthèse d'une architecture d'une transformée en cosinus discret DCT 2D 32*32 par un flot de synthèse de haut niveau prenant en compte l'information sur les tailles des opérands durant : (1) le dimensionnement uniquement sans optimiser les registres, (2) le dimensionnement uniquement avec optimisation des registres par le LEA ; (3) le dimensionnement seul avec optimisation des registres et des multiplexeurs par un algorithme MWBM et (4) un « clustering » des opérations par temps, un ordonnancement mixte, une optimisation de registre et des multiplexeurs par un algorithme MWBM prenant en compte la taille des opérands et le dimensionnement. Nous avons fait varier les tailles des opérands comme suit 8, 12, 16, 20, 24, 28 et 32 bits pour les données d'entrées et avons déduit les tailles des données de sorties. Les expériences ont été réalisées sur un FPGA Xilinx Virtex5 xc5vlx110 en utilisant la suite logicielle ISE 10.1. La consommation en puissance du cœur du FPGA a été mesurée avec un analyseur de puissance DC qui contient une alimentation tension/courant programmable ainsi que des appareils de mesure (tels que voltmètre et ampèremètre). La précision du voltmètre est de 0.016% + 1.5 mV et celle de l'ampèremètre est de 0.03% + 15 μ A.

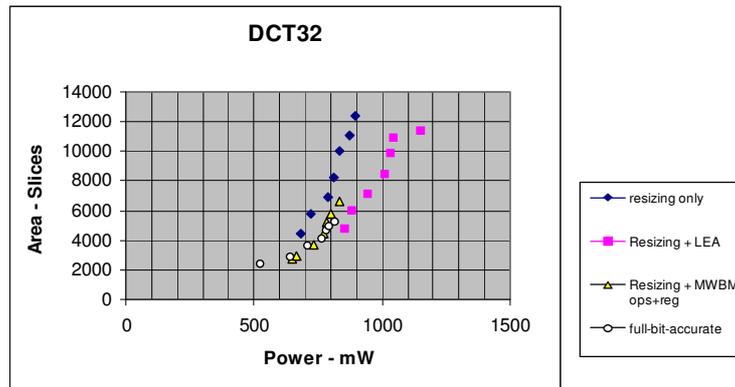


Figure 25 : Résultats de consommation puissance et de surface

Nous obtenons pour des entrées sur 8 bits une consommation de puissance de 685 mW et une surface de 4477 slices en appliquant uniquement l'étape de dimensionnement (1). En appliquant le flot "bit près" complet (4) pour des entrées sur 16 bits nous obtenons une consommation de puissance de 712 mW et une surface de 3569 slices. A consommation comparable et surface plus faible, la précision des calculs est grandement améliorée puisque le nombre de bit a doublé. Comparé à la combinaison (2) qui est l'approche la plus concurrente, nous n'améliorons pas significativement les résultats sur 8 et 16 bits. Toutefois pour 24 bits, une consommation de 786 mW et une surface de 4925 slices sont obtenues en appliquant la combinaison (2) alors que nous obtenons de meilleurs résultats sur 32 bits en utilisant l'approche proposée : 815 mW et 5172 slices. En effet, notre approche permet systématiquement, pour une taille de donnée, d'obtenir les meilleurs résultats (les points sont tous les plus en bas et les plus à gauche) ce qui permet d'augmenter, parfois grandement, la précision des calculs.

3.2.2. Modélisation comportementale

3.2.2.1. Flot de conception

La modélisation comportementale pour la simulation et la synthèse a été abordée dans le cadre du projet collaboratif RNTL SystemC'Mantic (projet, initié en 2003 par E. Martin, dont j'ai eu la responsabilité en 2004). Ce projet visait à définir un flot de conception de systèmes sur puce (voir Figure 26), allant des spécifications jusqu'à l'implémentation à l'aide du langage SystemC. Dans ce contexte, nos travaux, réalisés dans le cadre de la thèse de Farhat Thabet en co-encadrement avec Eric Martin, portaient sur l'utilisation du langage SystemC à différents niveaux d'abstraction pour la modélisation comportementale en amont de la synthèse de haut niveau. Pour rappel, la modélisation transactionnelle et la bibliothèque TLM SystemC en étaient, à l'époque, à leurs débuts. TLM 1.0, « standardisé » en juin 2005, ciblait la modélisation de systèmes à mémoire partagée à base de processeurs.

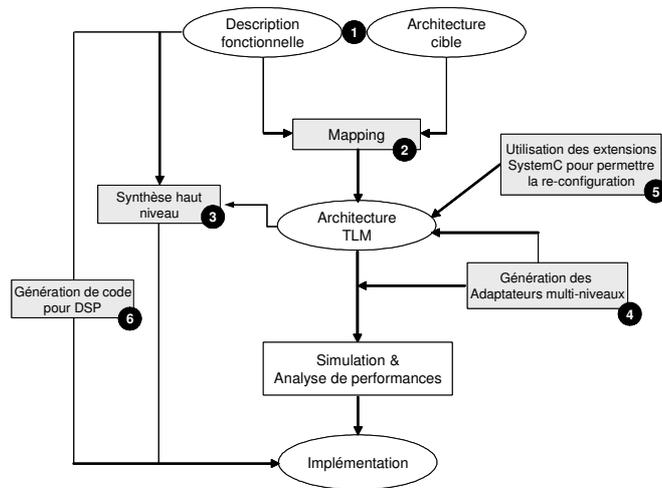


Figure 26 : Flot de conception proposé par le projet SystemC'mantic.

Dans le flot de conception général retenu, l'application est dans un premier temps décrite par un modèle fonctionnel en SystemC (1) et l'architecture cible du système est modélisée à l'aide des composants de la bibliothèque : processeurs, accélérateurs matériels... Le concepteur réalise ensuite un partitionnement HW/SW et l'assistant de « mapping » (2) génère une plate-forme transactionnelle en SystemC. Une fois validée, cette architecture est considérée comme la référence pour l'implémentation. Afin de descendre en abstraction, le comportement aux entrées/sorties des accélérateurs matériels doit être raffiné et exploré. Les contraintes temporelles sont ensuite extraites lors des simulations et utilisées pour piloter la synthèse de haut niveau (3) afin d'obtenir des accélérateurs matériels RTL synthétisables. Le problème d'hétérogénéité de niveau d'abstraction entre les modèles est résolu grâce à la génération automatique des « transacteurs » (4). Des extensions au langage SystemC (5) ont aussi été proposées pour permettre une modélisation des parties reconfigurables du système. De plus, (6) le code correspondant aux fonctions implémentées logiquement (Embedded C) est généré à partir des spécifications SystemC.

3.2.2.2. Approche pour la simulation et la synthèse

Afin de répondre à des contraintes de conception (temps, consommation...), un accélérateur matériel peut être conçu avec différents interfaces (parallélisme...) et comportements temporels. Comme présentés dans la **Figure 27**, les comportements aux entrées/sorties peuvent en effet être (1) non pipeline et sans recouvrement entre les entrées et les sorties ; (2) non pipeline mais avec recouvrement entre les entrées et les sorties ; (3) pipeline et sans recouvrement entre les entrées et les sorties, et (4) pipeline avec recouvrement entre les entrées et les sorties.

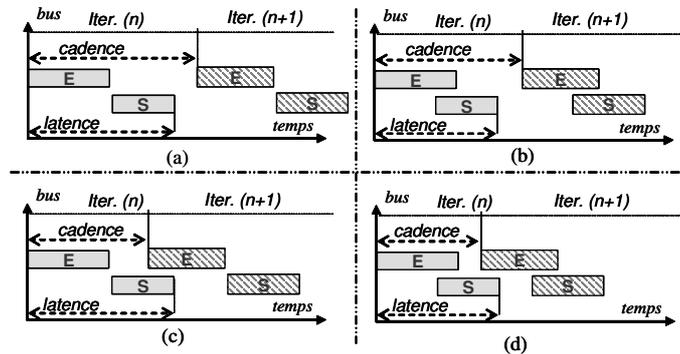
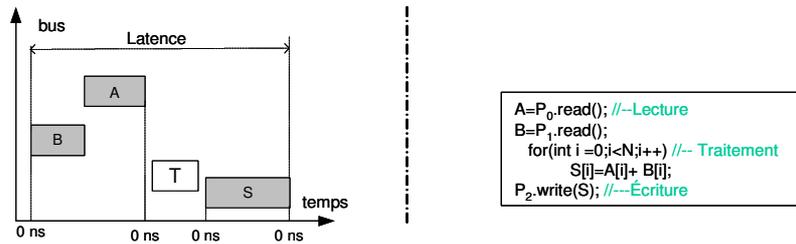


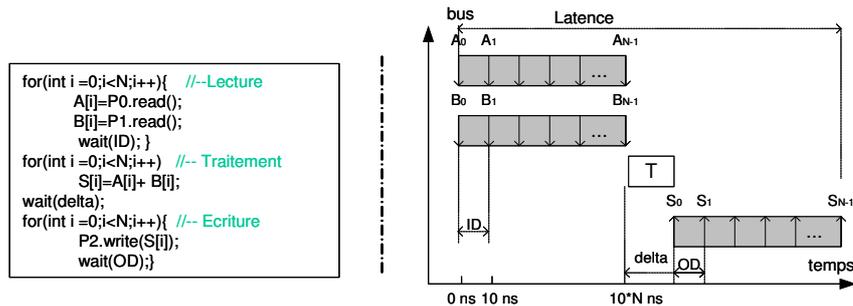
Figure 27 : Comportement : (a) non pipeline sans recouvrement d'entrées/sorties, (b) non pipeline avec recouvrement d'entrées/sorties, (c) pipeline sans recouvrement d'entrées/sorties, et (d) pipeline avec recouvrement d'entrées/sorties.

Pour modéliser de tels comportements, le concepteur a besoin de réécrire la spécification, initialement non temporelle, du modèle pour entrelacer les opérations de lecture, de traitement, et d'écriture (**Figure 28**). La **Figure 28.c** présente une spécification décrivant un comportement d'exécution avec recouvrement entre les entrées et les sorties du composant *addv*. Cette transformation de code, abordable dans le cas d'un exemple simple (ici l'addition de deux vecteurs), devient rapidement impossible, dans le cas de la modélisation d'un système plus complexe d'un point de vue algorithmique et comportemental. La nécessité de recodage pour passer d'un comportement d'exécution à un autre, à un même niveau d'abstraction, rend la tâche d'exploration comportementale complexe. Un tel processus de raffinement, s'il est réalisé manuellement, ne permet pas un passage rapide et fiable d'un comportement d'exécution à un autre, et ne permet donc pas au concepteur de faire des analyses sur les performances de son système dans les premières étapes de processus de conception. Un modèle de spécification et de raffinement rapide et fiable d'un composant virtuel algorithmique est donc nécessaire.

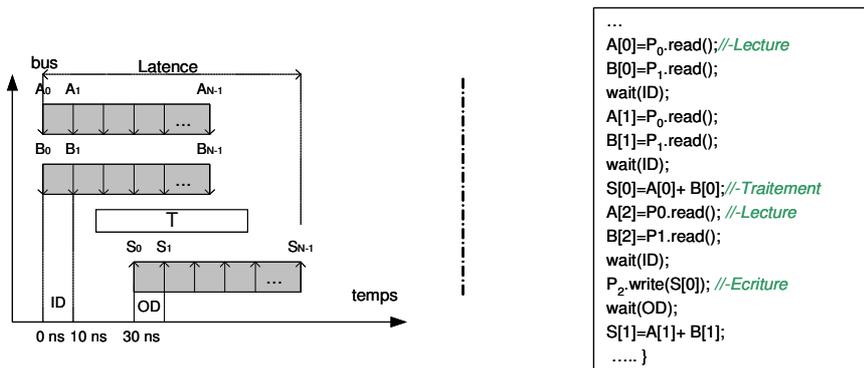
L'approche proposée pour la modélisation comportementale en vue de la synthèse repose à la fois sur (1) un modèle de spécification comportementale, nommé « Behavioral Description Model » (BDM), et (2) sur la synthèse de haut niveau HLS. Le modèle BDM, utilisé pour la simulation, permet de modéliser l'interface d'un composant (nombre de port d'E/S, type de données i.e. scalaires, vecteurs, tableaux 2D...) et ainsi d'explorer pour chaque type d'interface différents comportements temporels. Il permet finalement de générer automatiquement un ensemble de contraintes temporelles utilisées lors de la synthèse haut-niveau de la fonction de traitement (**Figure 29.a**). Le principe consiste à encapsuler une fonction de traitement séquentielle (une fonction C traditionnelle, ici *addv*) dans un module SystemC (**Figure 29.b**). Un processus est associé à chaque port d'entrée (IP_0 et IP_1) et port de sortie (OP_0). Le module inclut aussi deux processus de contrôle : un pour contrôler les processus d'entrée (ICP) et un pour contrôler les processus de sortie (OCP). L'architecture de communication est définie par le nombre de port et leurs caractéristiques (format, entrée/sortie...). Le comportement temporel aux interfaces est décrit de façon générique à l'intérieur des processus d'entrées/sorties. Avec une modélisation basée sur la séparation de la communication et du traitement, les paramètres de simulation relatifs à la communication peuvent être modifiés pour modéliser différents comportements, sans toutefois modifier le code initial de la fonction. Le concepteur peut ainsi aisément spécifier les comportements précédemment décrits et présentés dans la **Figure 28**.



(a) Comportement d'exécution séquentiel au niveau gros grain non temporisé



(b) Comportement d'exécution séquentiel au niveau grain fin temporisé



(c) Comportement d'exécution concurrent avec recouvrement d'entrée/sortie au niveau grain fin temporisé

Figure 28 : Problématique de raffinement manuel.

Afin de faciliter la description comportementale, nous avons défini un ensemble de méthodes génériques de lecture/écriture qui fournissent les paramètres suivants : (1) le nombre des ports ; (2) les granularités des données et des ports ; (3) les délais et les ordres d'acquisition et de production des données ; et (4) le délai de traitement. Une fois l'architecture et le comportement aux E/S définis, les traces temporelles, illustrant le déroulement de la communication sous forme de dates, sont automatiquement générées lors de la simulation du modèle BDM. Ces traces sont ensuite utilisées comme contraintes lors de la synthèse haut-niveau de la fonction de traitement. Les composants ciblés à l'époque devaient avoir un comportement statique i.e. la durée du traitement ne devait pas dépendre des valeurs des opérandes.

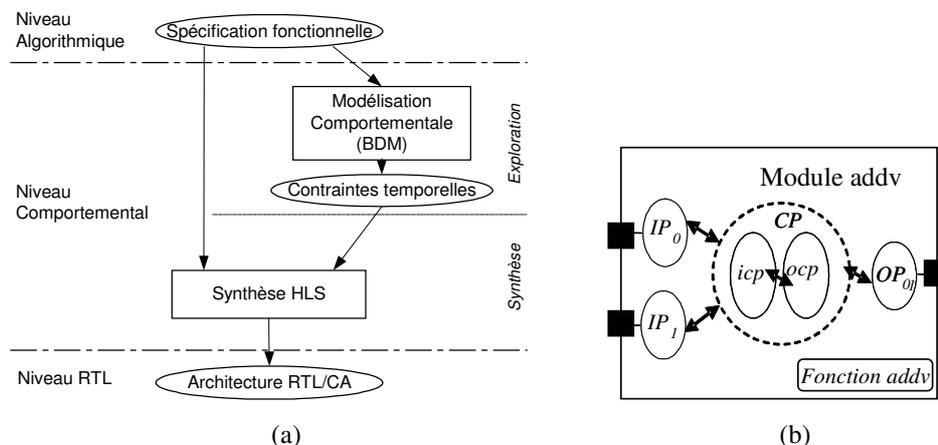


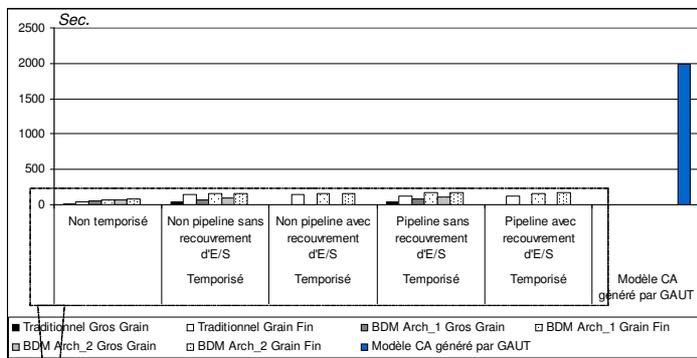
Figure 29 : (a) Approche proposée, (b) modèle BDM

Un environnement d'exploration et de conception basé sur le langage C/C++/SystemC allant de la spécification fonctionnelle jusqu'au niveau d'abstraction RTL/CA a été développé sous la forme d'un outil, nommé DsXplore (plug-in Eclipse). Cet outil permet de spécifier et de configurer graphiquement l'architecture d'un système (une chaîne de traitement du signal) et de générer les modèles BDM correspondants. Le concepteur spécifie son système graphiquement en instanciant et reliant des blocs algorithmiques génériques préconçus et configurables. Pour chaque bloc instancié, le concepteur peut visualiser le code C/C++ décrivant le traitement effectué par ce bloc. Le concepteur dispose d'une bibliothèque de fonction de traitement du signal (FFT, FIR...). Le concepteur définit les paramètres algorithmiques (les tailles de données d'entrée/sortie, la configuration de certains opérateurs ou sous-blocs d'un algorithme...), les paramètres architecturaux (les granularités des données et des ports d'entrée/sortie...) et les paramètres temporels (les délais de communication et du traitement) permettant d'obtenir l'architecture et les comportements du modèle composant à simuler puis à synthétiser. Ces travaux ont aussi requis une modification de l'outil de synthèse haut niveau GAUT pour prendre en compte en entrée des descriptions en langage C. En effet, avant la réalisation de ce projet, l'outil GAUT acceptait uniquement en entrée une spécification algorithmique décrite à l'aide du langage VHDL. Pour cela, le front end de GAUT est basé sur l'outil de compilation gcc. Un prototype de moteur de génération de code SystemC simulable au niveau CA (Cycle Accurate) décrivant la microarchitecture de l'unité de traitement générée a de plus été développé et intégré dans l'outil GAUT afin de faire des simulations post synthèse en SystemC.

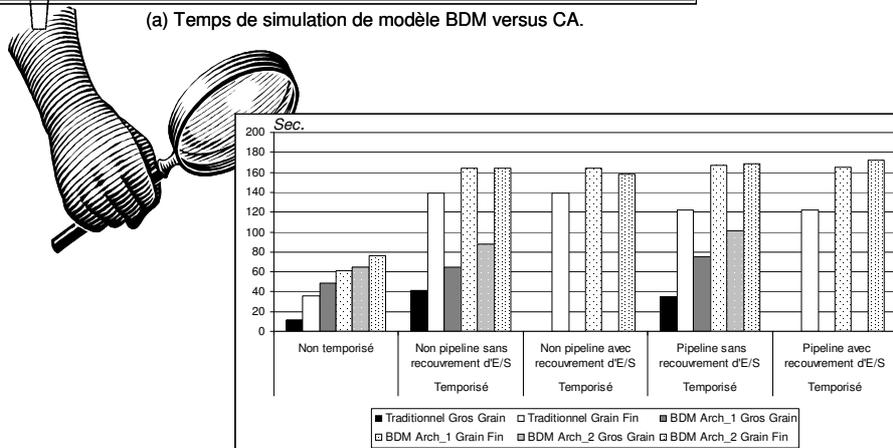
Ce flot de conception a été appliqué à des algorithmes du domaine du TDSI et des Télécommunications. Ainsi, un décodeur multiutilisateurs HISD (pour *Hyper-plane Intersection and Selection Detector*), développé au LESTER dans le cadre de la thèse de A. Nafkha [NAF 2005], a été utilisé pour évaluer l'impact de différentes contraintes d'intégration modélisées au niveau système sur le modèle d'implémentation micro-architectural [THA 2007a]. Une deuxième expérience, réalisée dans le cadre d'un projet RNTL SystemC'mantic, a montré l'intérêt de notre méthodologie pour la conception et l'intégration d'un composant virtuel de niveau algorithmique (Transformée de Fourier Rapide FFT) dans le contexte de conception d'un modem WiMax, la norme 802.16a [KRI 2005a].

A titre d'exemple, la Figure 30 présente les temps de simulation au niveau comportemental de différents comportements temporels du modèle de spécification traditionnel, du modèle BDM. L'approche traditionnelle représente la description utilisant un seul processus pour spécifier les opérations de lecture, de traitement et d'écriture des

données. Le composant utilisé est un ACS (Add, Compare and Select) utilisant 4 entrées. Les composants ACS sont des éléments de base des décodeurs MAP utilisés dans les codes convolutifs type turbo-codes. Les architectures *arch_1* dites « série » ont 1 port d'entrée et 1 port de sortie. Les architectures *arch_2* dites « mixte » ont 2 ports d'entrée et 2 ports de sortie. Les niveaux de description grain fin font références à des entrées de types scalaires et les niveaux de description gros grain font références à des entrées de types tableaux. Dans la Figure 30.a, nous comparons les temps de simulations du niveau comportemental avec les temps de simulation de niveau CA/RTL. Le modèle CA décrivant la microarchitecture est automatiquement généré par l'outil GAUT après synthèse. Ces simulations ont été effectuées sur un ordinateur portable (CPU : 1.1 Ghz, RAM : 1.24 Go) pour un million d'itérations de l'algorithme ACS. La modélisation BDM a un impact non négligeable sur les temps de simulation par rapport à une approche traditionnelle. Toutefois, elle évite au concepteur de devoir modifier le code pour explorer différents comportements temporels à différents niveau de granularité ce qui représente un temps non négligeable de développement dans le cas d'algorithmes complexes.



(a) Temps de simulation de modèle BDM versus CA.



(b) Temps de simulation de modèle BDM versus modèle traditionnel.

Figure 30 : Temps de simulations des différents modèles BDM d'un ACS (pour 1 million d'itérations).

3.2.3. Conception d'architecture MPSoC

3.2.3.1. Prototypage virtuel

L'utilisation de la synthèse d'architecture dans le contexte du prototypage virtuel a été abordée dans le cadre de la plateforme ANR SocLib. L'objectif de ce projet était de proposer un environnement de conception systèmes multiprocesseurs intégrés sur puce (MP-SoC). La plate-forme SoCLib est constituée (1) d'une bibliothèque de modèles de simulation de composants virtuels et (2) d'un ensemble d'outils logiciels qui permettent une exploration de l'espace de conception efficace. Les composants virtuels sont matériels sous la forme de CPU, DSP, contrôleurs de périphérique ou de mémoire, coprocesseurs spécialisés, réseaux d'interconnexion sur puce... et logiciels sous la forme d'OS, intergiciels de communication, logiciels applicatifs, etc. Les outils logiciels sont des moteurs de simulation, outils de configuration, outils de déverminage, outils de génération de documentation et des outils de génération automatique de modèles de simulation. L'ensemble des modèles de simulation ont été réalisés à l'aide du langage SystemC. Afin de faire des explorations plus ou moins fines et rapides, les modèles de simulation peuvent être précis au niveau du bit et du cycle (Cycle Accurate / Bit Accurate CABA) ou bien précis au niveau de la transaction (Transaction Level Modeling with Distributed Time TLM-DT). Finalement, afin de garantir une interopérabilité entre les composants le protocole d'interface VCI/OCP est utilisé.

Le flot de conception défini dans le cadre du projet SoCLib est présenté dans la Figure 31. Le point d'entrée est une spécification parallèle d'une application, décrite sous la forme d'un graphe de tâches communiquant par des canaux FIFOs (Kanh Process Network KPN). L'architecture cible est un système sur puce multiprocesseurs ayant un espace mémoire partagé décrit en utilisant des composants dans la bibliothèque de modèles de simulation. Après l'étape de partitionnement, les tâches sont implantées sous la forme de tâches logicielles ou de coprocesseurs spécialisés. Le simulateur est généré et la simulation peut permettre un déverminage de l'application ou une analyse des performances à l'issue de laquelle l'architecture ou l'assignation des tâches peut être revue.

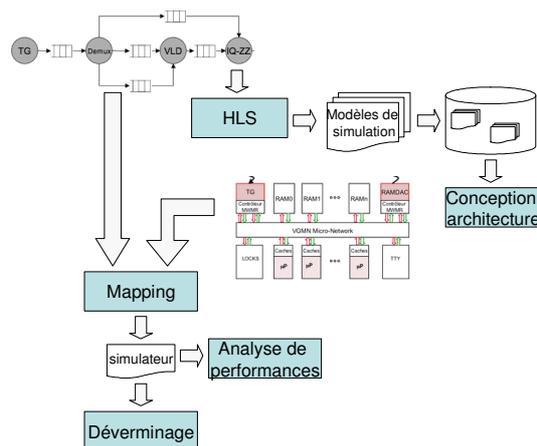


Figure 31 : Flot de conception général

Dans ce contexte, la synthèse de haut niveau (notamment à travers l'outil GAUT) doit, à partir d'une description algorithmique non temporisée (donc une des tâches de la spécification parallèle de l'application) et d'un ensemble de contraintes de synthèse,

généraliser automatiquement des modèles de simulation de niveau CABA ou TLM-DT d'un coprocesseur spécialisé. L'interface de communication retenue pour les coprocesseurs spécialisés est basée sur un protocole simple de type FIFO sans adresse. Chaque FIFO offre des primitives d'accès simple : écrire une valeur dans une FIFO et lire une valeur dans la FIFO. Ainsi, en suivant le principe d'« orthogonalisation » des calculs et du traitement, l'interface du coprocesseur est « décorrélée » du protocole utilisé par le média de communication (crossbar, NoC, bus...). Elle peut aussi de ce fait supporter plusieurs types de contrôleur de bus/interface. Ainsi, en fonction du « wrapper » utilisé pour adapter le protocole d'interface FIFO et le protocole du média de communication, le composant peut être maître ou esclave, utiliser des « transferts DMA » ou non... Différents modèles de programmation tels que KPN ou MWMR (Multi Writer, Multi Reader) défini dans le cadre du projet peuvent aussi être utilisés [FAU 2007].

Les modèles de simulation générés par l'outil GAUT ont été validés dans des architectures mono et multiprocesseur de niveau CABA ou TLM-DT. Nous avons dans les expériences que nous présentons ici, utilisé une application de décompression de vidéo Motion JPEG (MJPEG). L'application a été modélisée dans une première version sous la forme d'un graphe de tâche présenté dans la **Figure 33.a**. Cette description a été parallélisée comme indiqué dans la **Figure 33.b** pour obtenir une deuxième version capable de décoder quatre images en parallèle.

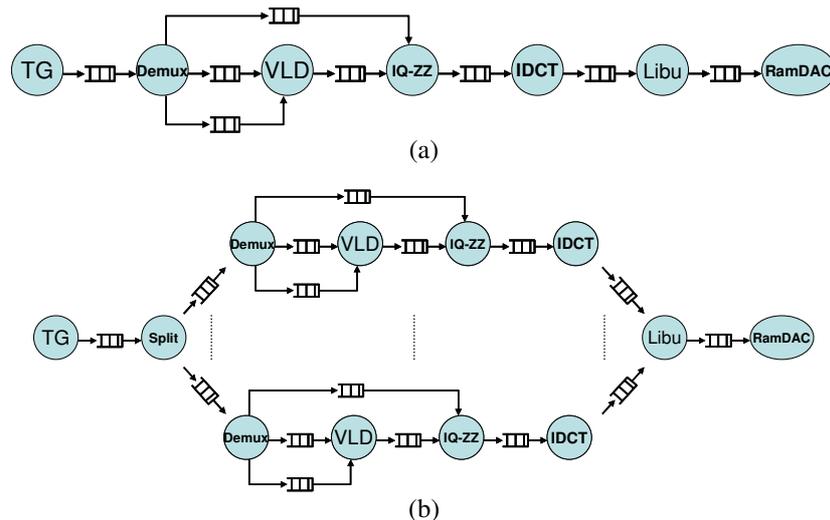


Figure 32 : Application MJPEG pour une architecture (a) mono et (b) multiprocesseurs

Les composants de la bibliothèque SoCLib qui ont été utilisés ont été développés au laboratoire Lip6 : MIPS-R3000, Cache, RAM, Crossbar générique... Quatre architectures ont été modélisées pour ces expériences : monoprocesseur et quadri-processeurs (**Figure 33.a**) avec ou sans coprocesseur(s) (**Figure 33.b**).

Les deux premières versions des systèmes étaient « purement logiciel » i.e. aucun coprocesseur n'était employé pour accélérer les calculs. L'outil DSX [SOC 2011], développé au Lip6, a été utilisé pour assigner la tâche *TG* au composant « File access », la tâche *RamDAC* au composant « Frame Buffer » et la chaîne de tâches *Demux*, *VLD*, *IQ-ZZ*, *IDCT* à chaque processeur MIPS-R3000 présent dans l'architecture.

Les deux versions suivantes du système étaient « mixtes » i.e. un ou quatre coprocesseurs ont été instanciés pour accélérer les applications. L'outil DSX a été utilisé

pour assigner la tâche *TG* au composant « File access », la tâche *RamDAC* au composant « Frame Buffer », la chaîne de tâches *Demux*, *VLD*, *IQ-ZZ*, à chaque processeur MIPS R3000 et la tâche IDCT à chaque coprocesseur présent dans l'architecture.

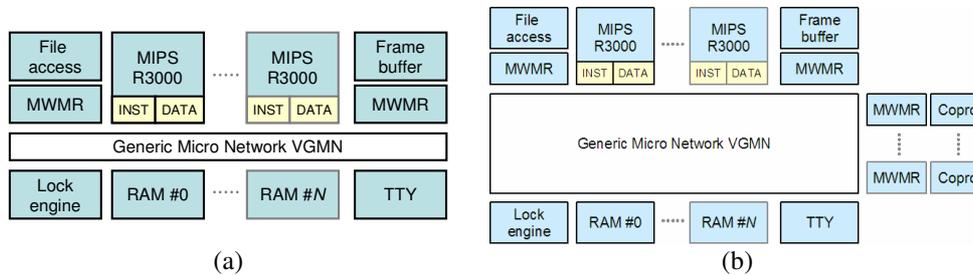


Figure 33 : Architecture mono ou multiprocesseurs : (a) sans coprocesseur et (b) avec coprocesseur(s)

Les simulations, dont les résultats pour décoder une séquence de 50 images de 48*48 pixels, sont présentés dans la **Figure 34** et la Figure 35, ont été réalisées à l'aide du simulateur SystemCASS [SOC 2011]. Dans l'architecture monoprocesseur, l'introduction d'un coprocesseur IDCT permet de réduire la latence de l'application de 14,4%. La parallélisation de l'application sur 4 processeurs offre quant à elle une réduction de la latence de 21%, ce qui est faible au regard du surcoût matériel (3 CPUs + caches et mémoires associés). Finalement, l'introduction de 4 IDCT matérielles dans l'architecture multiprocesseur permet quant à elle de réduire de 10% supplémentaire la latence totale. Concernant les temps de simulation, l'introduction d'un coprocesseur IDCT dans l'architecture monoprocesseur et de quatre coprocesseurs IDCT dans l'architecture multiprocesseurs augmente respectivement de 10% et 38% les temps des simulations. Ceci est lié aux modèles SystemC générés par GAUT qui n'ont pas été réellement optimisés durant le projet. La parallélisation de l'application sur 4 processeurs sans accélérateur aboutit quant à elle à une dégradation de 65% du temps de simulation par rapport à une architecture monoprocesseur sans coprocesseur et de 50% par rapport à une architecture monoprocesseur avec coprocesseur.

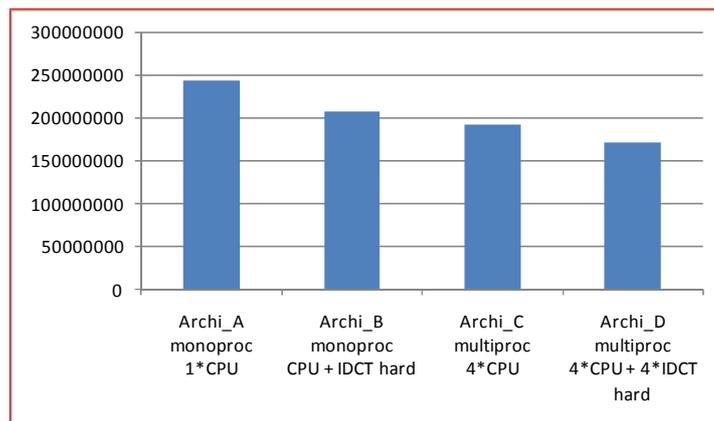


Figure 34 : Temps d'exécution de l'application (en #cycles)

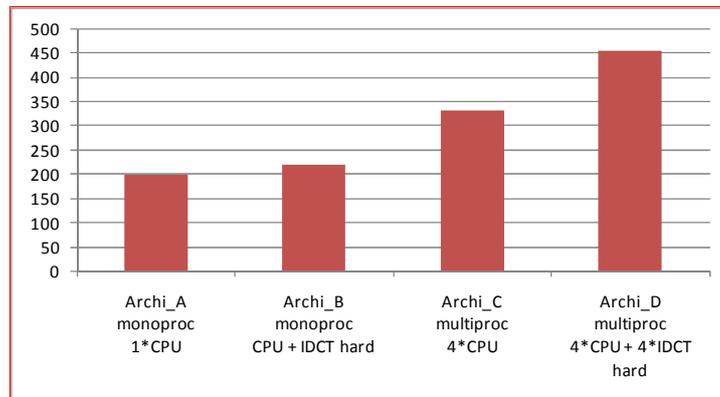


Figure 35: Temps de simulation (en secondes)

3.2.3.2. Compilation et synthèse d'architecture

L'utilisation de la synthèse d'architecture dans le contexte d'une utilisation conjointe avec un compilateur est abordée dans le cadre de la thèse de Paolo Burgio, en co-direction avec Luca Benini (Université de Bologne, Italie), qui a débuté en janvier 2010. Le but de ces travaux est de proposer une approche reposant sur l'utilisation d'une API OpenMP et sur un flot de conception intégré. L'approche doit permettre la parallélisation et l'accélération d'une application sur une architecture multiprocesseur régulière intégrant des accélérateurs matériels (HWPU) tout en masquant les détails d'implémentation. Bien que le flot ne soit pas encore totalement intégré, nous avons obtenu des premiers résultats intéressants.

Le patron architectural cible que nous avons défini dans une première approche est présenté dans la Figure 36. Il est constitué d'un nombre paramétrable de processeurs ayant chacun un cache privé d'instructions, d'un nombre paramétrable d'accélérateurs matériels, d'un réseau d'interconnexion combinatoire de type Mesh-of-Tree (MoT), d'un ensemble de bancs mémoire et de périphériques de synchronisation. Le modèle mémoire considéré est de type espace d'adressage global partitionné (Partitioned Global Address Space PGAS) utilisant donc une mémoire distribuée partagée. La communication entre les CPUs et les HWPU se fait au travers d'une mémoire partagée de niveau L1 (multi-ports multi-bancs avec un nombre de port égal au nombre de bancs). L'utilisation d'une mémoire partagée permet de limiter les transferts de données (zero-copy) entre les CPUs et les HWPU. De plus, chaque processeur a accès à une mémoire partagée sur puce de niveau L2 et à une mémoire partagée hors puce de niveau L3 plus importante. Chaque HWPU a deux ensembles de registres de configuration offrant ainsi deux canaux de programmation. Ces registres sont « memory-mapped » et appartiennent à l'espace d'adresses visibles des CPUs. La duplication des registres de configuration permet un recouvrement de la configuration et du traitement via une utilisation classique dite en « ping-pong » de ces registres : un CPU peut programmer un HWPU alors que ce dernier est en cours d'exécution. Lorsqu'un CPU veut accéder aux registres de configuration d'un HWPU, le réseau d'interconnexion intercepte les requêtes et les route vers le port esclave du HWPU. Lorsque le HWPU veut accéder à ses opérandes ou fournir ses résultats, il le fait au travers de son interface « maître ».

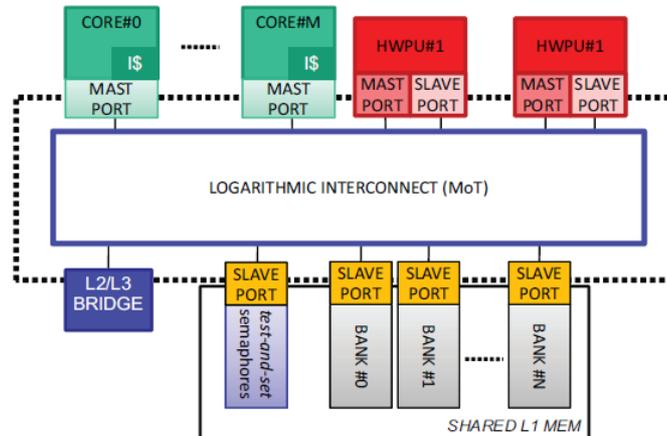


Figure 36 : Architecture cible

Lorsqu'un processeur veut utiliser un HWPU, il initie une séquence de programmation qui est exécutée automatiquement. Une fois la programmation effectuée, le CPU peut réaliser un traitement indépendant sans attendre la fin des calculs du HWPU. La séquence de programmation est la suivante :

- Acquisition d'un verrou pour vérifier que l'accélérateur est programmable,
- Notification du nombre d'entrées et du nombre de sorties,
- Notification de l'adresse de chaque entrée et de chaque sortie,
- Activation du HWPU,
- Attente de la fin du traitement après exécution d'une instruction de synchronisation.

Le flot de conception défini dans le cadre de ces travaux est présenté dans la Figure 37. Le point d'entrée est une spécification de l'application distribuée sur différents CPU et/ou HWPU via des directives de compilation spécifiées sous la forme de pragma dans le code source. Pour cela, un nombre réduit de directives a été créé pour permettre au concepteur d'indiquer, aussi facilement qu'une parallélisation via un pragma OpenMP, les parties du code doivent être accélérées sur un HWPU. Le compilateur modifié GCC OpenMp (GOMP) remplace ainsi de façon transparente les parties du code à accélérer par une séquence d'instructions qui permet de décharger les calculs sur un accélérateur. Il extrait aussi de façon transparente les parties du code à accélérer sous la forme d'un fichier utilisé ensuite par un outil de synthèse de haut niveau (ici GAUT) pour générer les modèles de simulation et de synthèse du HWPU correspondant. L'utilisation d'une mémoire partagée facilite l'intégration des accélérateurs matériels dans le modèle d'exécution et le modèle mémoire « OpenMP ». Un simulateur SystemC est ensuite généré pour permettre une analyse des performances à l'issue de laquelle l'architecture ou l'assignation des tâches peuvent être revues.

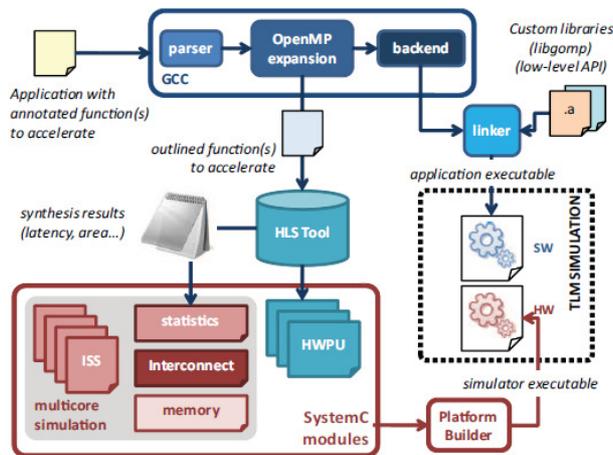


Figure 37 : Flot de conception

L'intérêt de l'approche proposée a été étudié, entre autre, sur différentes applications dont une décompression d'image au format JPEG. Il a été décidé, dans la série d'expérience présentée ici, d'accélérer l'application (1) en parallélisant la fonction de quantification inverse (IQTZ) et en l'exécutant sur un ou plusieurs processeurs et (2) en utilisant un ou plusieurs accélérateurs matériels pour la DCT inverse (IDCT) (Figure 38).

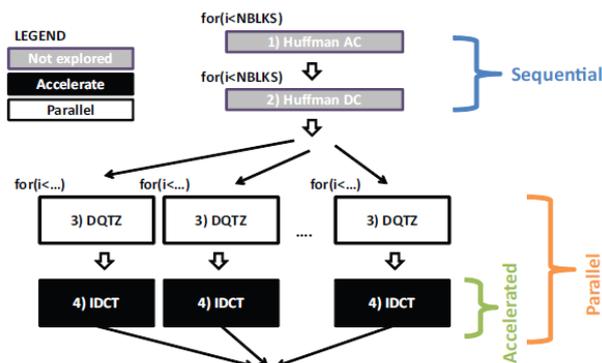


Figure 38 : Application JPEG

A l'aide de la plateforme de simulation MPARM [MPA 2011] développée à l'université de Bologne, nous avons comparé notre modèle architectural basé sur des accélérateurs matériels sans copie (ZC-HWPU) avec des approches typiques d'accélération d'application : ASIP et accélérateurs avec copies des données. Les ASIP peuvent être considérés comme une technique d'accélération dans laquelle une ALU et la logique d'accélération d'instruction partagent une mémoire de niveau L0 (i.e. la file de registres). Bien que ce type d'approche soit intéressant, le nombre restreint de ports de lecture et d'écriture des files de registres réduit malheureusement en général le parallélisme d'E/S de l'accélérateur (typiquement 2 port d'entrée et 1 port de sortie) et limite de ce fait les gains en performance. Après analyse de l'application, des clusters d'opérations ont été modélisés sous la forme d'instructions dédiées pouvant s'exécuter en un cycle CPU. Une autre approche traditionnelle consiste à découpler l'accélérateur des files de registres du CPU, à connecter l'accélérateur sur un bus système ou un NoC, et à transférer ses données explicitement depuis ou vers une mémoire privée de niveau L1 (généralement une mémoire

scratchpad), ou une mémoire principale de niveau L2 (cas du processeur CELL) ou L3 (typique des GPUs). Cette famille d'approche sera nommée par la suite C-ACC.

Nous avons étudié trois variantes d'accélérateurs C-ACC en considérant des transferts depuis des mémoires de niveau L1, L2 et L3. Pour chaque variante nous avons considéré deux schémas d'interfaçage dans lesquels la communication et les traitements matériels pouvaient être pipelinés (*double buffering C-ACC(DB)*) ou non (*single buffering C-ACC*).

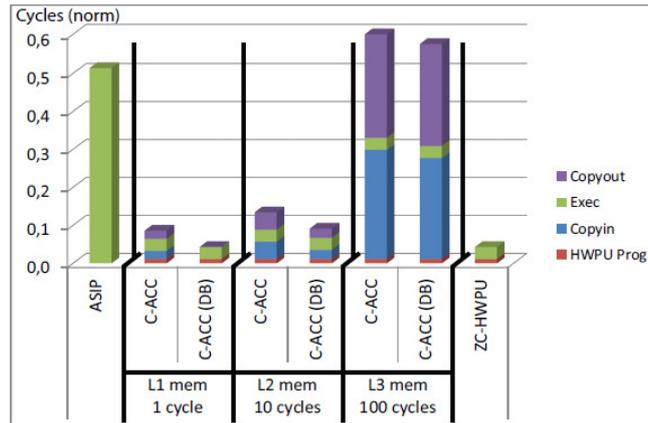


Figure 39 : Comparaisons des différentes approches pour l'application JPEG

Le schéma ZC-HWPU que nous proposons égale ou surpasse les autres modèles architecturaux. Ses performances sont comparables aux C-ACC(DB) lorsqu'une mémoire de niveau L1 est considérée, ce qui n'est toutefois pas une architecture classique. Pour l'application JPEG, notre approche permet une accélération d'un facteur supérieur à 3 comparée au C-ACC et d'un facteur supérieur à 2 par rapport au C-ACC(DB) lorsque les échanges se font à travers une mémoire de niveau L2. Comparée aux échanges à travers une mémoire de niveau L2, notre approche permet une accélération d'un facteur supérieur à 14 comparée au C-ACC et d'un facteur supérieur à 13 par rapport au C-ACC(DB). La solution ASIP est 12 fois plus lente.

Afin d'étudier l'effet du partage d'un ZC-HWPU, nous avons fait varier le nombre de processeurs. Dans l'architecture multiprocesseurs ciblée, les macro-blocs sont quantifiés inversement (DQ-ZZ) en parallèle par les processeurs et l'IDCT est réalisée matériellement pour chaque macro-bloc par un unique accélérateur. Dans ce contexte, les processeurs requièrent continuellement et concurremment le coprocesseur, impliquant potentiellement des contentions. Les résultats présentés dans la Figure 40.a montrent que le partage d'un unique accélérateur offrant un unique canal de programmation évolue bien jusqu'à 4 processeurs. A partir de 8 CPUs, les contentions sont trop importantes et les performances du système se dégradent. L'utilisation d'un deuxième canal de programmation (double buffering) permet de supprimer cette limitation (Figure 40.b).

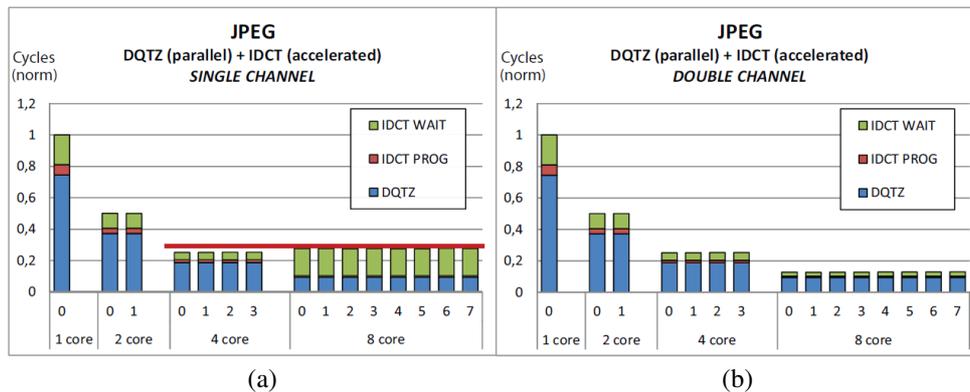


Figure 40 : Effet du partage entre processeurs d'un HWPU

3.3. Conclusion

Depuis ces dernières années, les travaux dans le domaine des approches de conception de niveau « système » (langage, prototypage virtuel, synthèse d'architecture, vérification formelle...) ont abouti à des évolutions majeures dans la façon de concevoir les systèmes. Afin d'accompagner ces évolutions, nous avons mené plusieurs études dans cet axe principalement appuyées sur la synthèse d'architecture et en particulier sur l'outil GAUT. Ainsi, des travaux sur la prise en compte du langage C et du langage SystemC en entrée de la HLS ont été menés pour permettre une meilleure intégration de cette technologie dans les flots de conception. Des travaux autour de la modélisation comportementale et de la génération automatique de modèles de simulation plus ou moins précis ont été réalisés afin de faciliter la conception de systèmes mixtes complexes.

Afin de mener à bien ces travaux, deux doctorants, deux post-doctorants et quatre stagiaires de DEA ou Master Recherche ont participé ou participent actuellement au projet.

Les travaux menés au sein de cet axe de recherche (hors publications réalisées durant ma thèse) ont conduit à treize communications scientifiques réparties comme suit : trois revues internationales [THA 2009a] [COU 2008a] [COU 2010a], huit conférences internationales [LEF 2010a] [LHA 2010a] [THA 2007a] [THA 2006a] [KRI 2005a] [COU 2005a] [ABB 2004a] [THA 2004a], deux conférences nationales [THA 2005a] [COU 2005c]. Les références sont détaillées à la fin de la section 3.4.

3.4. Fiche de synthèse des travaux

▪ Co-encadrement de doctorants

Farhat THABET, 2003-2007 (Co-encadrement Eric Martin, 50%)

Modélisation Comportementale Unifiée pour la Simulation et la Synthèse Haut-Niveau de composants Virtuels Algorithmiques

Paolo BURGIO, 2010-2013 (Co-direction Luca Benini, Univ. Bologne, 50%)

Utilisation de la mémoire partagée dans le contexte d'un processeur multi-cœurs embarqué : exploration de la technologie et ses limites.

Co-tutelle : Université de Bologne

▪ **Encadrement de post-doctorants**

Jorgiano Vidal, 2010-2011

Conception et réalisation d'un front-end et d'un modèle de représentation interne générique pour la synthèse de haut niveau

Caaliph ANDRIAMISAINA, 2008-2009

Génération de modèles de simulation par la synthèse de haut niveau pour le prototypage virtuel d'application du domaine du TDSI

▪ **Encadrement de stagiaires de DEA et de Master**

Michelle Furtado Pinheiro Do Carmo, 2009/2010

Prototypage virtuel d'un décodeur MP3 avec la plateforme SocLib
Master recherche I-MARS

Ghizlane Lebreton, 2006/2007

Prise en compte de la dynamique des données dans l'outil de synthèse de haut niveau GAUT

Sebastien Tregarot, 2004/2005

Modélisation et simulation d'une application MIMO dans une plate-forme SoC décrite en langage de haut-niveau SystemC
Master recherche I-MARS

Jean-Baptiste Le Goff, 2003/2004

Raffinement des communications en vue de la simulation et de la synthèse de composants virtuels algorithmiques
Master recherche I-MARS

▪ **Collaborations scientifiques**

SoCLib

open platform for virtual prototyping of multi-processors system on chip

Type : Projet plate-forme l'ANR/RNTL

Consortium : STMicroelectronics, Thales Communications, Thomson Silicon Components, Prosilog, TurboConcept, Silicomp, UPMC/LIP6, ENST, UPMC/LISIF, CEA LIST, INRIA Futurs, IRISA, LESTER, IETR INSA, TIMA, CEA LIST, CITI

Durée : 2006-2009

SystemC'Mantic

A high level Modeling and Co-design Framework For Reconfigurable. Real Time Systems

Type : Projet RNTL

Consortium : Prosilog, THALES, TIMA, CEA LIST, TIMA, LESTER

Durée du projet : 2003-2005

ONAGRE

environnement de conception et de prototypage d'applications Radio et mobile

Type : Contrat CRE (Contrat de Recherche Externalisée)

Consortium : France Telecom R&D (Issy les moulineaux), LESTER

Durée du projet : 2005-2008

▪ **Publications scientifiques**

Revue internationale

[THA 2009a] F. Thabet, P. Coussy, D. Heller, E. Martin, **“Exploration and Rapid Prototyping of DSP Applications using SystemC Behavioral Simulation and High-Level Synthesis”**, *Journal of Signal Processing Systems, Springer*, Issue 56, 2-3, September 2009.

[COU 2008a] P. Coussy, G. Le Breton, D. Heller, **“Multiple Word-Length High-Level Synthesis”**, *EURASIP Journal on Embedded Systems*, July, 2008.

[COU 2010a] P. Coussy, A. Takach, M. McNamara, M. Meredith, **“An Introduction to the SystemC Synthesis Subset Standard”**, *IEEE International Conference on Hardware-Software Codesign and System Synthesis (CODES+ISSS)*, 2010.

Conférences internationales

[LEF 2010a] V. Lefftz, J. Bertrand, H. Cassé, C. Clienti, P. Coussy, L. Maillet-Contoz, P. Mercier, P. Moreau, L. Pierre, E. Vaumorin, **“A Design Flow for Critical Embedded Systems”**, *IEEE Symposium on Industrial Embedded Systems (SIES)*, 2010.

[LHA 2010a] G. Lhairech-Lebreton, P. Coussy, D. Heller, E. Martin, **“Bitwidth-Aware High-Level Synthesis for Designing Low-Power DSP Applications”**, *IEEE International Conference on Electronics, Circuits, and Systems (ICECS)*, 2010.

[THA 2007a] F. Thabet, P. Coussy, D. Heller, E. Martin, **“Behavioral Description Model BDM for Design Space Exploration: a Case Study of His Algorithm for MC-CDMA System”**, *European Signal Processing Conference (EUSIPCO)* 2007.

[THA 2006a] F. Thabet, P. Coussy, D. Heller, E. Martin, **“Design Space Exploration of DSP Applications Based on Behavioral Description Models”**, *IEEE Workshop on Signal Processing Systems Design and Implementation (SIPS)*, 2006.

[KRI 2005a] L. Kriaa, S. Adriano, E. Vaumorin, R. Nouacer, F. Blanc, S. Pajaniardja, P. Coussy, E. Martin, D. Heller, F. Thabet et al , **“SystemC'mantic : A high level Modeling and Co-design Framework For Reconfigurable Real Time Systems”**, *Forum on Design Languages (FDL)*, 2005.

[COU 2005a] P. Coussy, G. Corre, P. Bomel, E. Senn, E. Martin, “**A More Efficient and Flexible DSP Design Flow from MATLAB-SIMULINK**”, *IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, 2005.

[ABB 2004a] F. Abbes, E. Casseau, M. Abid, P. Coussy, J.-B. Legoff., “**IP integration methodology for SoC design**”, *International Conference on Microelectronics (ICM)*, 2004.

[THA 2004a] F. Thabet, J.-B. Legoff, P. Coussy, E. Martin, “**A Methodoly for Timing and Structural Communication Refinement in DSP Systems**”, *International Conference on Microelectronics (ICM)*, 2004.

Conférences nationales

[THA 2005a] F. Thabet, P. Coussy, E. Martin, “**Approche Automatique pour le Raffinement des Communications**”, *Journées Francophones sur l'Adéquation Algorithme/Architecture (JFAAA)*, 2005.

[COU 2005c] P. Coussy, G. Corre, P. Bomel, E. Senn, E. Martin, “**Synthèse Comportementale Sous Contraintes de Communication et de Placement Mémoire pour les composants du TDSI**”, *Colloque sur le Traitement du Signal et de l'Image (GRETSI)*, 2005.

4. Axe 3 : Synthèse d'architecture d'unité mémoire

4.1. Introduction

Ce troisième axe de recherche se concentre sur la génération automatique d'unité mémoire. Il s'agit de trouver un placement des données dans des éléments mémoire permettant des accès parallèles sans conflit. Ce problème complexe est fréquemment rencontré durant la conception des entrelaceurs utilisés dans les circuits correcteurs d'erreur type LDPC ou turbo codes utilisés dans des applications telles que Wifi, Wimax, DVB-S, DVB-T, 3GPP-LTE.

Les algorithmes de décodage sont conçus pour corriger des erreurs dans une trame de données [BER 1993]. Toutefois, pour que ces corrections soient efficaces, il faut disperser ces erreurs sur toute la trame. Ainsi, dans le cas de chaînes de traitement du signal (cf. **Figure 41**), l'un des éléments les plus importants pour la qualité du décodage est le bloc d'entrelacement/dé-entrelacement. Sa fonction est de briser les relations de voisinage entre les différents éléments d'une trame de données avant son émission sur le canal. Ainsi, si une perturbation modifie plusieurs bits consécutifs de la trame entrelacée, une fois les données dé-entrelacées ces erreurs se retrouvent réparties sur toute la trame et sont de ce fait « plus faciles » à localiser et à corriger (i.e. c'est une décorrélation des erreurs susceptibles d'être véhiculées). La fonction d'un entrelaceur consiste donc à mélanger des données : les données entrent dans un ordre OA, et ressortent de l'entrelaceur dans un ordre OB.

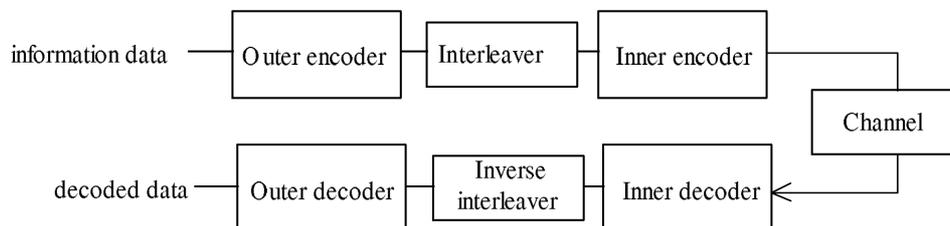


Figure 41 : Représentation simplifiée d'une chaîne de traitement du signal

Afin d'offrir des débits de transmission toujours plus élevés (voir Figure 2), les codes de type turbo-codes ou LDPC doivent de nos jours être parallélisés (jusqu'à 16 ou 32 éléments de calcul s'exécutant en parallèle). La **Figure 42** représente un exemple d'architecture destinée à la mise en œuvre d'un entrelaceur parallèle matériel. Elle comporte un ensemble de blocs ou modules de traitement de données PE1, PE2, ...PE_n générant ou consommant en parallèles des données, un ensemble de bancs mémoire M1, M2, ..., M_n disposés en parallèle et reliés aux éléments de calculs PE1, PE2, ...PE_n via des modules d'interconnexion I1 et I2.

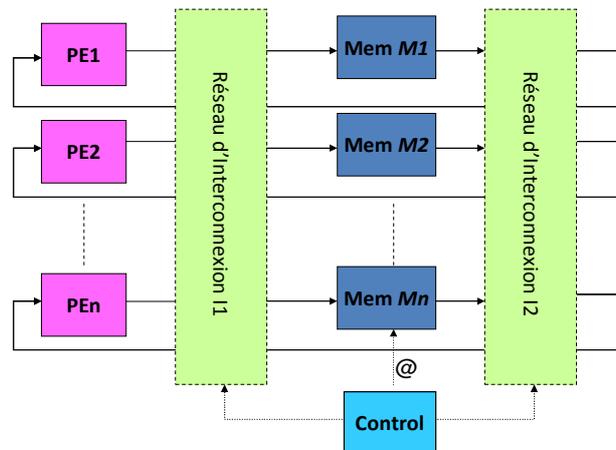


Figure 42 : Architecture typique d'un entrelaceur parallèle

L'un des problèmes majeurs liés à la mise en œuvre de ce type d'architecture d'entrelacement concerne les conflits d'accès en mémoire. Il est en effet possible que plusieurs blocs de traitement tentent d'écrire ou, au contraire, de lire en même temps différentes données dans ou à partir d'un même banc mémoire. Résoudre le problème consiste à générer une architecture de mémorisation/temporisation des données composée d'éléments mémoire, d'un contrôleur et d'un ensemble de composant d'aiguillage qui minimisent le plus les conflits ou qui les suppriment ce qui est toujours possible dans le cas des entrelaceurs actuels [TAR 2004]. Nous sommes donc en présence d'un problème d'optimisation d'une architecture dans laquelle les accès aux données ont la particularité d'être statiques i.e. ils ne dépendent pas des stimuli.

Il est important de noter que ce problème de ré-ordonnancement, c'est-à-dire d'aiguillage et de temporisation, des données apparaît aussi parfois durant la conception d'architectures de types chaîne de traitement (macro-pipeline d'accélérateurs connectés en point à point), durant l'intégration d'accélérateur dans une architecture MPSoC ou dans la synthèse de haut niveau lorsque les données sont placées dans des bancs mémoires et que l'application doit être parallélisée pour respecter les contraintes de temps/consommation ou durant l'optimisation du chemin de données (ensemble d'opérateur, d'élément de mémorisation et de composants d'aiguillage) et de son contrôleur.

Les travaux de recherche de cet axe ont été développés dans le cadre de différents projets. Ainsi :

- le placement des données dans des éléments mémoire à grains fins à forte sémantique (FIFO, LIFO, registre...) a été abordé dans le cadre de la thèse CIFRE STMicroelectronics de Cyrille Chavet (2004-2007);
- le placement des données dans des éléments mémoire gros grains (bancs mémoires) a été :
 - o initié dans le cadre de la thèse CIFRE STMicroelectronics de Cyrille Chavet qui a donné lieu à un dépôt de brevet en Europe et aux USA (2004-2007),
 - o poursuivi sur des aspects méthodologiques par les travaux de thèse d'Awais Sani Hussain (2008-2011) ;
- le placement des données dans des éléments mémoire à gros grains et à grains fins par relaxation de contraintes dans le cadre des travaux de thèse d'Aroua Briki (2009-2012).

4.2. Présentation des travaux

4.2.1. Unité mémoire à « grains fins »

Dans une première approche, nous avons abordée la problématique de conception d'entrelaceur parallèle en utilisant des éléments mémoire à forte sémantique en lieu et place des bancs mémoire. L'architecture cible nommée STAR pour Space-Time AdapteR (voir la Figure 43) se compose de chemins de données (utilisant des FIFO, des LIFO et/ou des registres) et de contrôleurs permettant de piloter le système. L'adaptation spatiale (une donnée en peut être transmise de n'importe quel port d'entrée vers un ou plusieurs ports de sortie) est effectuée par un réseau d'interconnexion adapté. L'adaptation temporelle est réalisée par les éléments de mémorisation. Le flot de conception proposé peut générer des architectures STAR pouvant intégrer plusieurs modes de fonctionnement (par exemple, plusieurs longueurs de trames pour un entrelaceur). Afin de rester synthétique, l'approche de conception d'architecture multi-modes ne sera pas détaillée ici [CHA 2007] [AND 2010a].

La méthodologie de conception permettant de générer automatiquement un adaptateur STAR. Notre flot de conception prend en entrée (1) des diagrammes temporels (fichier de contraintes) ou (2) une description en langage C de l'ordonnancement des données d'entrée-sortie (par exemple une règle d'entrelacement) et des contraintes utilisateur (débit, latence...). Ce flot formalise ensuite ces contraintes de communication sous la forme d'un Graphe de Compatibilité des Ressources (GCR), dont les propriétés permettent une exploration efficace de l'espace des solutions architecturales afin de générer un composant STAR.

Le flot de conception (voir la Figure 44) est basé sur trois outils :

- **StarTor** prend en entrée la description en langage C de l'algorithme d'entrelacement, et les contraintes de l'utilisateur (latence, débit, interface de communication, parallélisme d'entrée-sortie...). Il en extrait l'ordre des données d'entrée-sortie en produisant d'une trace à partir de la description fonctionnelle. Ensuite, l'outil génère le fichier de contraintes de communication qui sera utilisé par l'outil *STARGene*.

- **STARGene**, basé sur un flot à cinq étapes, génère l'architecture STAR : (1) construction des graphes de compatibilité des ressources, à partir du fichier de contraintes, correspondant à chacun des modes de fonctionnement du design, (2) fusion des graphes de compatibilités pour chaque mode en un seul graphe multi-modes, (3) assignation des structures de mémorisation (FIFO, LIFO ou Registre) sur le GCR (4) optimisation de l'architecture et (5) génération du VHDL niveau transfert de registre (RTL) intégrant les différents modes de communication.

- **StarBench** génère un banc de test basé sur les contraintes de communication et permet de valider les architectures générées en comparant les résultats de simulation avec le comportement théoriquement attendu.

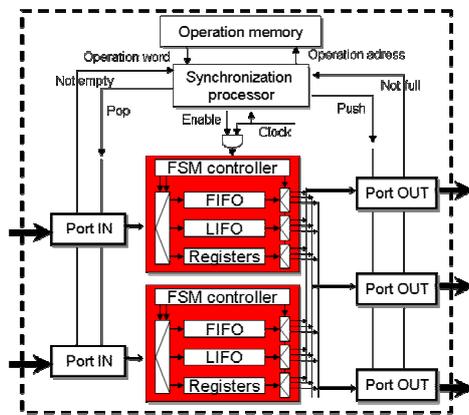


Figure 43 : Architecture STAR

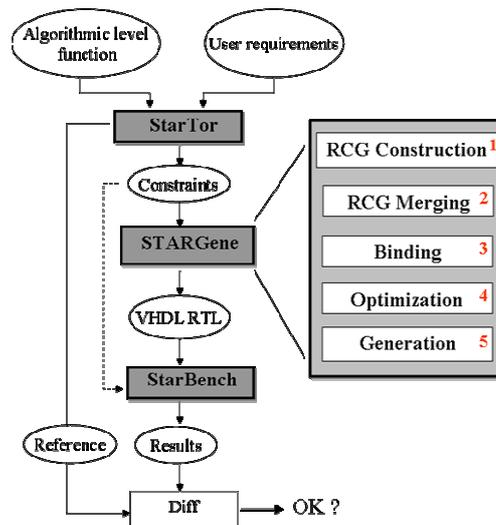


Figure 44 : Flot de conception et outils associés

Flot de conception (*STARGene*) détaillé :

1 - Construction du graphe de compatibilité des ressources : Les nœuds du RCG représentent les données qui doivent transiter au sein de l'adaptateur; les arcs, étiquetés et orientés, représentent les types de compatibilité entre ces données. L'étiquette est le type d'élément de stockage, $T = \{\text{Registre, FIFO, LIFO}\}$. Afin d'assigner des étiquettes aux arcs, nous analysons les relations entre les durées de vie des données. Nous avons défini à cette fin un ensemble de règles basées sur les propriétés temporelles de chaque type d'éléments mémoire. Deux données sont compatibles Registre si les intervalles de durée de vie de ces données sont "non recouvrant". Deux données sont compatibles FIFO si leurs intervalles de durée de vie sont "partiellement recouvrant" et que ces données respectent entre elles une sémantique de type "première entrée, première sortie". Deux données sont compatibles LIFO si leurs intervalles de durée de vie sont "recouvrant-incluant" i.e. ces deux données respectent entre elles une sémantique de type "dernière entrée, première sortie".

2 - Fusion des graphes : Dans le cas d'architecture devant gérer plusieurs modes de fonctionnement au sein d'un même adaptateur, l'approche que nous avons retenue consiste à fusionner, à moindre coût architectural, les différents graphes au sein d'un GCT unifié. Cette étape est basée sur un algorithme de fusion des graphes s'inspirant de celui proposé dans [DUT 1993] et basé sur une approche de type Kernighan-Lin.

3 - Assignment des éléments mémorisant :

- *Identification des ressources* : L'identification des éléments de mémorisation est facilitée par les propriétés intrinsèques de nos arcs étiquetés et par l'utilisation de la notion de chemin dans un graphe. En effet, parcourir un chemin d'un type donné (FIFO ou LIFO) dans notre RCG est équivalent, par construction, à la recherche d'une clique de compatibilité. Ainsi, notre modèle nous permet de nous passer d'un algorithme de recherche de clique maximum (NP-complet).

- *Dimensionnement des ressources* : La taille d'une structure FIFO ou d'une LIFO n'est pas égale au nombre de sommets sur le chemin correspondant. De fait, pour dimensionner correctement ces éléments, il faut identifier le recouvrement maximal de données dans ce même chemin. Notre modèle de représentation, basé sur un graphe GCR, facilite cette exploration.

- *Assignation des ressources* : Notre algorithme glouton est basé sur un ensemble de métriques, paramétrables par l'utilisateur, pour assigner autant de structures FIFO ou LIFO possibles sur le GCR. Nous utilisons alors un algorithme en deux étapes : (1) identification d'un élément mémoire, (2) fusion des nœuds mis en jeu au sein d'un nœud hiérarchique. Naturellement, le choix des nœuds à fusionner influence considérablement l'architecture résultante, et les choix de l'utilisateur concernant le réglage des différentes métriques est de première importance.

4 - Optimisation de l'architecture : Enfin, l'étape d'optimisation vise à maximiser l'utilisation des éléments mémorisant, afin de minimiser leur nombre et par là même, réduire le nombre de structures à contrôler.

Le graphe de compatibilité des ressources est construit en traitant les données dans l'ordre chronologique de leur arrivée dans l'adaptateur de communication (voir **Figure 45**). Pour chaque donnée, on construit alors les arcs de compatibilité la reliant aux autres données, selon les règles de constructions énoncées précédemment. Ainsi, si l'on considère l'exemple présenté dans la **Figure 45**, on constate :

- un arc de compatibilité Registre entre les données *a* et *d*,
- un arc de compatibilité LIFO entre les données *a* et *c*,
- un arc de compatibilité FIFO entre les données *a* et *b*,
- ...

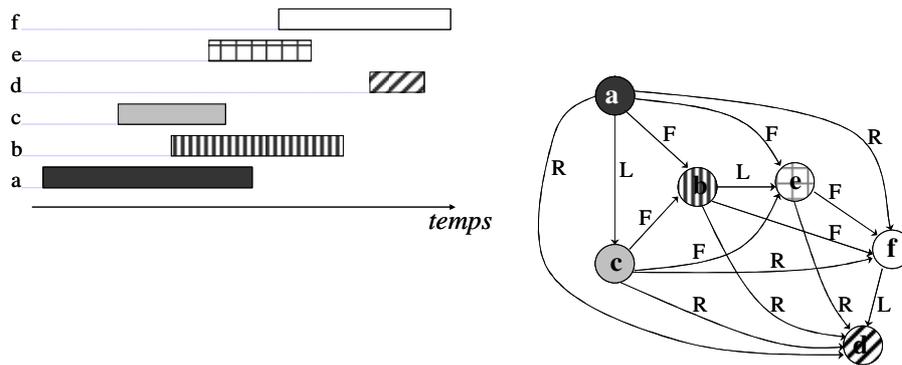


Figure 45 : Exemple de graphe de compatibilité des ressources

A titre d'exemple pour les expériences, nous avons considéré un entrelaceur pour l'Ultra Wide Band [IEEE 2007] pour lequel nous disposons d'architectures de référence, issues de l'équipe FTM-HLS de STMicroelectronics, pour comparer nos résultats. L'entrelaceur que nous ciblons doit pouvoir être utilisé dans trois modes de fonctionnement différents en fonction de la longueur de la trame à entrelacer (300, 600 ou 1200 données). L'architecture générée doit respecter des contraintes de débit et de latence pour pouvoir s'intégrer dans le système cible. Pour des raisons de confidentialité, ces informations ne peuvent toutefois pas être divulguées. Les architectures de références ont été réalisées à l'aide d'un outil de synthèse de haut niveau du commerce. Suite à des accords de non divulgation, nous ne pouvons révéler dans ce document ni le nom de cet outil, ni les informations concernant les bibliothèques technologiques utilisées pour la synthèse, ni les surfaces précises obtenues.

Mode	Référence		F/L (Min 7, Tx 95%)		F/L (Min 15, Tx 90%)		Registres		Débit
	Gain mémoire	Ctrl	Gain mémoire	Ctrl	Gain mémoire	Ctrl	Gain mémoire	Ctrl	
300	0	300	56	77	60	240	60	240	434,8
600	0	600	83	101	130	470	130	470	438
1200	0	1200	96	117	120	609	168	1032	412,4

Figure 46 : Comparaison des résultats de synthèse entre l'architecture de référence et trois composants STAR

La **Figure 46** synthétise les résultats que nous avons obtenus au cours de cette expérience. L'architecture de référence n'est pas optimisée en termes de points mémoire (aucun point mémoire économisé), et est composée d'un ensemble de registres. Cette architecture de référence est comparée à trois architectures STARs. Les deux premières sont des architectures STARs classiques exploitant des composants FIFOs, LIFOs et registres, générés avec des métriques d'exploration différentes (Longueur minimum d'un chemin typé, Taux d'utilisation moyen d'une structure) : dans le premier cas, la longueur minimum d'un chemin typé est fixé à 7 données et le taux d'utilisation moyen est fixé à 95% ; pour le second cas, la longueur minimum d'un chemin typé est fixé à 15 données et le taux d'utilisation moyen est fixé à 90% . La dernière architecture STAR générée n'utilise que des registres. Enfin, les débits obtenus (Exprimés en Mb./s.) pour les architectures STAR ont été indiqués. Nous pouvons remarquer que pour un mode donné, quelle que soit la solution architecturale du composant STAR, le débit reste inchangé. Notre approche permet donc d'explorer efficacement l'espace des solutions tout en garantissant le respect de la contrainte de débit.

L'analyse de ces résultats montre que notre approche permet de réduire le nombre de points mémoire et le nombre d'éléments à piloter dans tous les cas, par rapport à l'architecture de référence. Ainsi, pour la première architecture STAR avec FIFOs/LIFOs, le gain moyen en termes de signaux de contrôle est de 77% et en termes de points mémoire, le gain moyen est de 86%. Dans le cas de la seconde architecture STAR avec FIFOs/LIFOs, le gain moyen en termes de signaux de contrôle est de 30% et en termes de points mémoire, le gain moyen est de 82%. Enfin, dans le cas du composant ne comportant que des registres, le gain moyen en termes de signaux de contrôle est de 19% et en termes de points mémoire, le gain moyen est de 81%. De même, si l'on considère les modes 300 et 600 données, nous pouvons constater que les architectures générées pour le second STAR à base de FIFOs/LIFOs, et le STAR n'utilisant que des registres, ont les mêmes caractéristiques. De fait, ces architectures sont identiques car les contraintes utilisées ne permettent pas d'assigner d'éléments FIFOs ou LIFOs ; Seul le mode 1200 données offre des chemins de compatibilités permettant d'assigner ce type de structures. Cela signifie que les éléments FIFOs et LIFOs assignés dans le cas de la première architecture STAR avec FIFOs/LIFOs pour les modes 300 et 600 données sont des éléments de petites tailles. Toutefois, si l'on compare la surface de l'architecture de référence intégrant les trois modes de fonctionnement, et la surface cumulée des composants STAR à base de registres générés pour chacun des modes, alors la surface totale de l'architecture de référence est de 14% supérieure à la surface cumulée des architectures.

4.2.2. Unité mémoire à « gros grains »

Dans les approches de conception d'architecture d'entrelacement dites « gros grain », les éléments mémoire ciblés ne sont pas des registres mais des bancs mémoire. Dans ce contexte, nous avons proposé différentes approches permettant de supprimer les conflits d'accès aux données : des approches basées sur des modèles matriciels, des approches basées sur des graphes bipartite ou tripartite, des approches utilisant sur la relaxation de contraintes... Afin de rester synthétique, seule la famille d'approches basées sur la représentation matricielle sera présentée dans ce document. Toutefois, de plus amples détails sur les autres méthodes sont fournies dans [SAN 2011a] [SAN 2011b] [SAN 2010a] [CHA 2010d].

Les approches que nous proposons utilisent un modèle basé sur des matrices pour représenter le problème à résoudre. Deux matrices sont ainsi créées : la première M_{nat} représente l'ordre naturel et la deuxième M_{int} l'ordre entrelacé. L'ordre naturel et l'ordre entrelacé considérés dans l'exemple illustré dans la **Figure 47** sont $Nat = 1, 2, 3, 4, 5...$ et $Int = 6, 7, 8, 3, 11...$ Supposons que la contrainte applicative impose d'utiliser trois éléments de calcul en parallèle pour respecter le débit de transmission. Dans ces deux

matrices, les lignes représentent les données accédées par chaque élément de calcul et les colonnes correspondent aux « instants » de traitement. Une colonne représente donc les accès simultanés aux données en lecture puis en écriture.

P0	1	2	3	4
P1	5	6	7	8
P2	9	10	11	12

(a)

P0	6	7	8	3
P1	11	1	5	2
P2	4	9	12	10

(b)

Figure 47 : Matrices références : Matrice d'ordre (a) naturel M_{Nat} et (b) entrelacé M_{Int}

Ainsi, à l'instant t_0 de l'ordre naturel noté t_n0 , les données 1, 5 et 9 doivent être lues, traitées puis mises à jour en mémoire respectivement par les processeurs P0, P1 et P2. A l'instant t_n1 , les données 2, 6 et 10 doivent être lues, traitées puis mises à jour en mémoire respectivement par les processeurs P0, P1 et P2... Une fois l'ensemble des traitements réalisés dans l'ordre naturel (**Figure 47.a**), les données doivent être traitées dans l'ordre entrelacé (**Figure 47.b**). La solution classique pour concevoir les entrelaceurs parallèles consiste généralement à assigner les données de sorte qu'il n'y ait pas de conflit dans l'ordre naturel [TUL 2002] [MUL 2006] [MOU 2008]. Dans notre exemple, les données de la ligne 0 (i.e. les données utilisées par P0) de M_{nat} seraient assignées au banc mémoire B0, les données de la ligne 1 au banc mémoire B1...Malheureusement ce type d'approche, pourtant répandu, ne supprime pas les conflits d'accès. Les conflits peuvent apparaître lors des accès effectués dans l'ordre entrelacé. Ils sont dans ce cas résolus par l'ajout de tampon mémoire (registre/FIFO) dans le réseau d'interconnexion pour permettre de temporiser certains accès et sérialiser de ce fait les requêtes mémoire. Ainsi pour notre exemple, à l'instant t_3 de l'ordre entrelacé (3^{ème} colonne de M_{int} dans la **Figure 47.b**), les données 8, 5 et 12 sont accédées en parallèles par les processeurs P0, P1 et P2 ce qui crée un conflit d'accès au banc B2 dans lequel sont assignées les données 5 et 8.

A la différence des méthodes existantes, l'approche que nous proposons consiste à placer les données dans les bancs mémoire de façon à éviter l'ensemble des conflits d'accès (i.e. dans l'ordre naturel mais aussi dans l'ordre entrelacé). Contrairement à [DIN 2005] qui est basé sur un algorithme de recuit simulé, notre approche, en plus d'offrir une complexité algorithmique réduite, permet de respecter un modèle de réseau d'interconnexion (par exemple un barillet/« barrel shifter »). Pour cela, nous utilisons deux matrices MAP_{nat} et MAP_{int} reflétant respectivement l'assignation mémoire des données pour les ordres naturel et entrelacé. L'assignation doit être identique dans MAP_{nat} et MAP_{int} . De plus, un banc mémoire ne peut être utilisé plus d'une seule fois dans chaque colonne des matrices afin de ne pas créer de conflit d'accès. L'assignation des bancs doit respecter les possibilités offertes par le modèle de réseau d'interconnexion imposé : pour un barillet, l'assignation dans une colonne d'une matrice MAP_x doit être une permutation circulaire de n'importe qu'elle autre colonne de MAP_x . Ainsi, à l'instant initial, l'algorithme assigne la première colonne de la matrice MAP_{nat} et reporte cette assignation dans MAP_{int} (voir **Figure 48**).

B0			
B1			
B2			

(a)

		B1	
	B0		
			B2

(b)

Figure 48 : Matrices d'assignation : Matrice d'ordre (a) naturel MAP_{Nat} et (b) entrelacé MAP_{Int} (Etape 1)

La colonne la plus contrainte (i.e. la colonne contenant le plus grand nombre de données assignées dans MAP_{nat} ou MAP_{int}) est ensuite traitée et le résultat est reporté dans la deuxième matrice. Dans notre exemple, la deuxième colonne de MAP_{int} est traitée et la matrice MAP_{nat} est mise à jour en conséquence (**Figure 49**).

B0			
B1		B1	B2
B2			

(a)

	B2	B1	
	B0		
	B1		B2

(b)

Figure 49 : Matrices d'assignation : Matrice d'ordre (a) naturel MAP_{Nat} et (b) entrelacé MAP_{Int} (Etape 2)

L'algorithme se poursuit jusqu'à ce que l'ensemble des données aient été assignées en respectant l'ensemble des contraintes ou s'arrête sinon. La **Figure 50** présente le résultat de l'assignation pour notre exemple. Les bancs mémoires $B0$, $B1$ et $B2$ contiendront respectivement les données $\{1, 6, 3, 12\}$, $\{5, 10, 7, 4\}$ et $\{9, 2, 11, 8\}$. Cette solution permet d'utiliser un barillet comme réseau d'interconnexion puisque chaque colonne est une permutation circulaire des autres.

B0	B2	B0	B1
B1	B0	B1	B2
B2	B1	B2	B0

(a)

B1	B2	B1	B0
B2	B0	B2	B1
B0	B1	B0	B2

(b)

Figure 50 : Matrices d'assignation : Matrice d'ordre (a) naturel MAP_{Nat} et (b) entrelacé MAP_{Int} (Résultat)

Cette approche (nommée ci après *Meth_1*) [CHA 2010a] est limitée aux règles d'entrelacement n'utilisant qu'une et une seule fois chaque données comme par exemple les turbo codes en bloc [BER 1993]. De ce fait, seuls des placements mémoire dit « in-place », dans lesquels chaque donnée est assignée à un unique banc mémoire, sont possibles. Malheureusement, cette limitation interdit la conception d'entrelaceur dans lesquels les données sont utilisées plusieurs fois tels les LDPC [MAC 1996] [GUT 2010]. Cette restriction a été supprimée dans une deuxième approche [CHA 2010b] (nommée ci après *Meth_2*) qui en contrepartie n'accepte pas de contrainte sur le réseau d'interconnexion. Finalement, les travaux de recherche actuellement menés dans le cadre de la thèse d'Aroua Briki visent à proposer une méthode (nommée ci après *Meth_3*) permettant de traiter des règles d'entrelacement dans lesquelles les données sont utilisées plus d'une fois tout en supportant une contrainte sur le réseau d'interconnexion. L'approche repose sur une technique de relaxation de contrainte. Dans ce contexte, lorsque qu'il n'existe aucune solution d'assignation dans un banc mémoire permettant de respecter la contrainte architecturale imposée par le réseau d'interconnexion, un sous ensemble des données conflictuelles est stocké dans des registres annexes. La **Figure 51** présente les résultats obtenus en appliquant les différentes approches pour concevoir deux types entrelaceurs. L'Ultra-WideBand (UWB) [IEEE 2007] ne traite qu'une seule fois chaque donnée par itération ce qui n'est pas le cas des LDPC non binaires [GUT 2010]. L'entrelaceur UWB brasse 1200 données 4 par 4. Le LDPC brasse quant à lui 192 données 6 par 6 et accède à chaque donnée deux fois par itération. Un barillet a été utilisé lorsque qu'une contrainte sur le réseau d'interconnexion a été fixée.

Surface	UWB			LDPC Non binaire		
	[DIN 2005]	Meth_1	Meth_3	Meth_1	Meth_2	Meth_3
Mémoire	262800	262800	262800	n.a	42048	41391
Réseau d'interconnexion	1680	840	840	n.a	1960	3920
Contrôleur	11970	5358	5358	n.a	738	1693
Total	276450	268998	268998	n.a	44746	47004

Figure 51 : Résultats de synthèse (surface en porte NAND équivalente)

La règle d'entrelacement de l'UWB autorise intrinsèquement l'utilisation d'un barillet comme réseau d'interconnexion. Ceci permet à la méthode *Meth_1* de limiter la complexité architecturale et d'obtenir une meilleure solution que la méthode proposée dans [DIN 2005]. Pour rappel, cette dernière est basée sur un recuit simulé qui, bien qu'elle trouve une solution d'assignation sans conflit, ne prend pas en compte le coût du réseau d'interconnexion. La méthode *Meth_3* permet d'obtenir une architecture identique à celle fournie par *Meth_1*. La règle d'entrelacement utilisée dans les LDPC non binaire traite les données deux fois par itération. De ce fait la méthode *Meth_1* ne peut être utilisée. Les architectures générées en utilisant les méthodes *Meth_2* et *Meth_3* ont des surfaces équivalentes (à 3.5% près). Le réseau d'interconnexion fourni par *Meth_3* intègre un barillet. Toutefois, la règle d'entrelacement n'étant pas intrinsèquement basé sur ce type de permutation, une partie importante du réseau est composé de registres et multiplexeurs additionnels.

4.3. Conclusion

La conception optimisée d'entrelaceur parallèle est un des grands défis pour les futurs systèmes de communication. Bien que les standards évoluent pour prendre en compte le problème des contentions mémoire dans la définition même des règles d'entrelacement, le problème reste entier pour bien des parties de la chaîne de communication et bien des applications. Afin de résoudre cette famille de problème, nous avons mené plusieurs études. Ainsi, la génération d'entrelaceur à grains fins (registres, FIFO...), gros grains (bancs mémoire) et mixte a été adressée. Ces travaux ont et font l'objet de collaborations étroites avec les équipes de recherche en communications numériques du laboratoire mais aussi avec des industriels (STMicroelectronics, Turboconcept). L'ensemble des approches, qui ont été formellement définies, a été validé au travers de démonstrateurs et les résultats ont été et sont utilisés dans différents projets (IP FP7 DaVinci [GUT 2010], Thèse Cifre Rubisoft de S. Tregarot...).

Afin de mener à bien ces travaux, trois doctorants et un ingénieur ont participé ou participent actuellement au projet.

Les travaux menés au sein de cet axe de recherche ont conduit à dix communications scientifiques réparties comme suit : un brevet [CHA 2009a] [CHA 2008a], huit conférences internationales [SAN 2011a] [SAN 2011b] [SAN 2010a] [CHA 2010a] [CHA 2010b] [CHA 2007b] [CHA 2007c] [CHA 2007d], une conférence nationale [CHA 2005a]. A noter qu'un deuxième brevet est en cours de dépôt dans ce domaine. Les références sont détaillées à la fin de la section 4.4.

4.4. Fiche de synthèse des travaux

- **Co-encadrement de doctorants**

Awais Hussain SANI, 2008-2011 (Co-encadrement Eric Martin, 50%)
Approches de conception d'entrelaceurs parallèles pour les applications de codes correcteurs d'erreurs

Aroua BRIKI, 2009-2012 (Co-encadrement Cyrille Chavet 50%)
Conception d'unités mémoires complexes pour les applications de traitement du signal

Cyrille CHAVET, 2004-2007 (Co-encadrement Eric Martin, 50%)
Synthèse automatique d'interfaces de communication matérielles pour la conception d'applications du domaine du traitement du signal

- **Encadrement d'ingénieur**

LinFeng YE, 2011
Réalisation d'un générateur d'entrelaceur parallèle

- **Collaborations scientifiques**

Cifre STMicroelectronics
Turboconcept
Equipe Interaction Architecture Silicium (IAS) du Lab-STICC

- **Publications scientifiques**

- Brevets**

[CHA 2009a] Cyrille Chavet, Philippe Coussy, Pascal Urard, Eric Martin, "**Method and device for interleaving data**", *CNRS – STMicroelectronics U.S. Patent application 20090031094*, January 29, 2009.

[CHA 2008a] Cyrille Chavet, Philippe Coussy, Pascal Urard, Eric Martin, « **Procédé et dispositif d'entrelacement de données** », *CNRS – STMicroelectronics, Brevet Français n° 0754793 10*, 30 Avril 2007.

- Conférences internationales**

[SAN 2011a] A. Sani, P. Coussy, C. Chavet, E. Martin, "**A Methodology based on Transportation Problem Modeling for Designing Parallel Interleaver Architectures**", *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011.

[SAN 2011b] A. Sani, P. Coussy, C. Chavet, E. Martin, "**An Approach Based on Edge Coloring of Tripartite Graph for Designing Parallel LDPC Interleaver**"

Architecture”, IEEE International Symposium on Circuits and Systems (ISCAS), 2011.

[SAN 2010a] A. Sani, P. Coussy, C. Chavet, E. Martin, **“Design of Parallel LDPC Interleaver Architecture: A Bipartite Edge Coloring Approach”, IEEE International Conference on Electronics, Circuits, and Systems (ICECS), 2010.**

[CHA 2010a] C. Chavet, P. Coussy, P. Urard, E. Martin, **“Static Address Generation Easing: a Design Methodology for Parallel Interleaver Architectures”, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2010.**

[CHA 2010b] C. Chavet, P. Coussy, **“A Memory Mapping Approach for Parallel Interleaver Design with Multiples Read and Write Accesses”, IEEE International Symposium on Circuits and Systems (ISCAS), 2010.**

[CHA 2007b] C. Chavet, P. Coussy, P. Urard, E. Martin, **“Application of a design space exploration tool to enhance interleaver generation”, European Signal Processing Conference (EUSIPCO) 2007.**

[CHA 2007c] C. Chavet, P. Coussy, P. Urard, E. Martin, **“A Methodology for Efficient Space-Time Adapter Design Space Exploration: A Case Study of an Ultra Wide Band Interleaver”, IEEE International Symposium on Circuits and Systems (ISCAS), 2007.**

[CHA 2007d] C. Chavet, P. Coussy, P. Urard, E. Martin, **“A Design Methodology for Space-Time Adapter”, ACM Great Lakes Symposium on VLSI on VLSI (GLSVLSI), 2007.**

Conférence nationales

[CHA 2005a] C. Chavet, P. Coussy, P. Urard, E. Martin, **“Méthodologie de modélisation et d’implémentation d’adaptateurs spatio-temporels”, MANifestation des Jeunes Chercheurs STIC (MajecSTIC), 2005.**

5. Conclusion et perspectives de recherche

Ce mémoire a présenté une synthèse de mes activités de recherche réalisées depuis septembre 2004, date de ma nomination en tant que Maître de Conférences à l'UFR Sciences et Sciences de l'Ingénieur de l'Université de Bretagne-Sud. Elles ont été effectuées au laboratoire LESTER jusqu'en 2008 puis au laboratoire Lab-STICC au sein de l'équipe Méthodes et Outils pour les Circuits et Systèmes (MOCS) jusqu'à ce jour. Mes travaux de recherche se situent dans le domaine des sciences de l'ingénieur qui est un domaine pluridisciplinaire combinant l'informatique, l'électronique, l'automatique, le traitement du signal et les télécommunications. Plus précisément, ils se positionnent dans le domaine des méthodes de conception des systèmes numériques avec une forte dominante pour la synthèse automatique d'architecture et les approches dites de niveau système. Cette synthèse a été organisée dans ce mémoire autour de trois principaux axes. Un premier axe correspond à la synthèse d'architecture d'unité de calcul générant automatiquement des architectures optimisées d'accélérateurs matériels au niveau RTL. Un deuxième axe de recherche a pour thème la modélisation et le prototypage qui reposent sur des langages de description dits « systèmes ». Un troisième axe de recherche regroupe des travaux autour de la synthèse d'architecture d'unité mémoire générant automatiquement des architectures parallèles d'entrelaceur permettant ainsi de concevoir des turbo-décodeurs très haut-débit. Il est à noter que j'ai toujours insisté pour que les méthodes proposées soient formalisées, puis mises en œuvre au travers d'outils de CAO pour enfin être validées sur des applications réalistes au travers de prototypage sur des plateformes matérielles.

Dans les années à venir, je compte m'attacher à poursuivre et à diversifier mes efforts dans le domaine des méthodes et des outils visant la conception optimisée de systèmes embarqués. Ainsi, à court terme, je désire poursuivre les efforts initiés pour l'optimisation des architectures par la synthèse de haut niveau. A moyen terme, je souhaite développer mon activité vers la conception (semi-)automatisée d'architecture MPSoC, la génération automatique d'architectures robustes, la synthèse d'architectures de coprocesseurs offrant des possibilités de déverminage et la conception automatique de décodeurs itératifs très hauts débits. Enfin, à plus long terme, il me paraît intéressant de travailler sur la définition de méthodes et d'outils de conception d'architectures utilisant le codage neuronale.

Ainsi, je souhaite approfondir les travaux initiés dans le cadre de la thèse de Ghizlane Lhairech-Lebreton qui visent actuellement la réduction de la consommation par la génération automatique d'architectures ayant plusieurs domaines d'horloge et pouvant geler partiellement le signal d'horloge. Une piste intéressante consisterait à utiliser plusieurs niveaux de tension (approche multi-V_{th} multi-V_{dd}) en plus d'utiliser plusieurs domaines d'horloge. Une étude préliminaire a été menée lors du séjour de Ghizlane Lhairech-Lebreton dans le laboratoire CADLab de l'Université de Californie à Los Angeles (UCLA), en collaboration avec le Pr. Jason Cong.

Je souhaite aussi à court terme poursuivre les travaux initiés dans le cadre du stage de master recherche de Vianney Lapôtre qui consistaient à étudier l'intérêt du couplage de la prédiction de branchement et de la synthèse de haut niveau. La prise en compte et l'optimisation des applications orientées contrôle seront abordées dans le cadre du projet collaboratif national P qui débutera en octobre 2011.

A moyen terme, l'utilisation plus ou moins massive de circuit FPGA comme technologie cible pour réaliser les systèmes sur puces multiprocesseurs hétérogènes semble incontournable. Les approches ESL (incluant le prototypage virtuel, la synthèse

d'architecture, la conception conjointe HW/SW...) sont maintenant assez matures et permettent d'envisager une automatisation complète du flot de conception à moyen terme. La démocratisation et le couplage de ce type de technologies programmables et des approches de conception de haut niveau permettront à terme de concevoir et de réaliser rapidement des produits innovants visant des marchés de petites ou moyennes tailles par des PME ou des grands groupes. Dans ce contexte, il me semble intéressant de développer des approches visant à améliorer l'exploration de l'espace de conception pour les systèmes complets. Ces travaux à la frontière entre le domaine de la compilation et de la synthèse d'architecture permettraient d'estimer rapidement le coût et les performances d'une architecture mais aussi de transformer le code à haut niveau afin d'explorer efficacement et automatiquement l'espace des solutions. Plusieurs propositions de projets coopératifs (nationaux et européens) visant cet objectif ont été déposées ces dernières années mais n'ont malheureusement pas été retenues pour financement. Les efforts seront poursuivis dans ce sens. Les travaux actuellement réalisés dans le cadre de la thèse de Paolo Burgio (voir Partie 1 du document), en co-tutelle avec l'université de Bologne, qui se terminera fin 2013 s'inscrivent dans cet axe bien qu'ils ne ciblent pas de technologie particulière.

Les systèmes utilisés entre autre dans les domaines du spatial, de l'avionique, du médical ou du nucléaire fonctionnent dans des environnements pouvant provoquer des erreurs dans leurs fonctionnements. L'évolution actuelle des technologies CMOS (diminution des tensions d'alimentation, réduction des finesses de gravure, élévation des fréquences d'horloge...) utilisées dans les produits « grand public », n'est pas sans avoir de conséquences sur la fiabilité des circuits intégrés. Il devient alors nécessaire d'utiliser, parfois en plus des technologies, des architectures tolérantes aux fautes. Ceci doit se faire à tous les niveaux : système, algorithmique et logique pour maîtriser les mécanismes de défaillances physiques et leurs effets au niveau des technologies commerciales (dans le sens où elles ne sont pas durcies aux Single Event Upset - SEU ou Single Event Latch-up - SEL). Les flots de conception doivent donc intégrer les contraintes de fiabilité vis-à-vis des erreurs en prenant en compte les contraintes liées à l'environnement, aux applications et aux architectures afin de permettre la génération systématique d'architectures fiables pour les systèmes d'instrumentation. Cette problématique sera en partie abordée dans le cadre de la thèse de Thomas Peyret en partenariat le Laboratoire Capteurs et Architectures Electroniques (LCAE) du CEA LIST en octobre 2011.

L'émergence de la synthèse de haut niveau, permettant de générer de façon automatique ou semi-automatique un composant électronique conforme à une spécification logicielle permet de réduire l'effort de vérification. Toutefois, bien qu'en utilisant des outils de synthèse d'architecture, la spécification puisse être validée par simulation de la description algorithmique, les erreurs liées à des aspects temporels ou aux données restent difficiles à déverminer. La validation des circuits à base de FPGAs est une activité complexe, identifiée comme critique, qui requiert des simulations RTL précises temporellement, des outils d'introspection, l'instrumentation du bitStream ou encore l'usage d'oscilloscopes. Dans le cas de la synthèse d'architecture, malgré les efforts pour conserver un lien entre la spécification RTL (générée automatiquement) et la description abstraite initiale, la complexité de déverminage par simulation reste forte. Il est donc nécessaire d'offrir une capacité de déverminage logiciel aux circuits produits automatiquement à partir de spécifications de haut niveau. L'objectif est de pouvoir enrichir, à la demande et avec un surcoût matériel maîtrisé, les architectures de capacités de déverminage (contrôlabilité, introspection, abstraction, agilité) telles qu'on les rencontre dans les environnements logiciels. Cette problématique sera traitée dans le cadre de la thèse MESR de Mohamed Ben Hammouda en co-direction avec Loic Lagadec (Université de Bretagne Occidentale UBO) qui démarrera en octobre 2011.

L'omniprésence programmée de l'informatique et de l'électronique dans notre environnement va provoquer l'explosion du nombre d'objets communicant et donc des systèmes embarqués dans un futur assez proche. Cette évolution s'accompagnera d'un accroissement de la quantité d'information à transmettre et d'une demande de fiabilité importante. De ce fait, afin d'offrir des transmissions très haut débit efficaces, les architectures des décodeurs devront être massivement parallélisées. La conception d'entrelaceur matériel sans conflit sera donc, en plus de l'évolution des standards de communication, une des clés du succès. Les travaux initiés dans ce domaine seront poursuivis et renforcés notamment dans le cadre du projet collaboratif régional GIGADEC (TurboConcept, Telecom Bretagne/Lab-STICC, UBS/Lab-STICC) débutant en juillet 2011. Les travaux de recherche effectués dans le cadre de la thèse de Saeed Ur Rehman (Financement HEC, Pakistan) qui débutera en septembre 2011 (en co-direction M. Jezequel, Telecom Bretagne), viseront à concevoir et réaliser une architecture générique d'entrelaceur parallèle auto-configurable qui fait actuellement l'objet d'un dépôt de brevet.

A plus long terme, je compte initier des activités de recherche dans le cadre du projet « codage neural », piloté par Claude Berrou (Professeur à Télécom Bretagne, co-inventeur des turbo-codes et membre de l'Académie des Sciences), qui est l'un des défis scientifiques du Labex COMIN-Labs (Bretagne-Pays de Loire). Ce travail s'inscrit dans une démarche fortement réductionniste. Il s'agit de dégager de l'ensemble des connaissances acquises ces dernières années par les neurobiologistes un terreau minimum qui offre à la théorie de l'information (codage, communication, graphes, etc.) l'opportunité d'une contribution forte à la compréhension et à l'imitation du comportement cérébral. Les travaux menés principalement au niveau théorique et algorithmique ont déjà abouti à des résultats en augmentant d'un ordre de magnitude la capacité de mémorisation des réseaux de neurones artificiels (mémoire associative neurale). L'objectif de ce grand projet est d'identifier, d'approfondir et d'exploiter les fortes analogies que l'on peut observer entre les structures et les propriétés du cortex cérébral et celles des décodeurs correcteurs d'erreurs modernes, de type turbo ou propagation de croyance. Dans ce contexte, je souhaite proposer des architectures matérielles innovantes mais aussi des méthodes et des outils de conception associés pour pouvoir réaliser à moyen terme des systèmes numériques et pourquoi pas, à long terme, un « modèle numérique du cerveau ».

6. Références

- [ABB 2004a] F. Abbes, E. Casseau, M. Abid, P. Coussy, J.-B. Legoff., “*IP integration methodology for SoC design*”, International Conference on Microelectronics (ICM), 2004.
- [ALT 2011] Altera, “*PLL Clock Management Features in Altera FPGA*”, www.altera.com/support/devices/pll_clock/pll-overview.html
- [AND 2010a] C. Andriamisaina, P. Coussy, E. Casseau, C. Chavet, “*High-Level Synthesis for Designing Multi-mode Architectures*”, IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems (TCAD), Vol. 29, Issue 11, pp. 1736-1749, November 2010.
- [AND 2008] C. Andriamisaina, “*Flot de conception dédié aux architectures multi-modes pour les applications de traitement du signal et de l'image*”, Thèse de doctorat, Université de Bretagne Sud, 2008.
- [AND 2007a] C. Andriamisaina, P. Coussy, E. Casseau, “*Synthesis of Multimode digital signal processing systems*”, NASA/ESA Conference on Adaptive Hardware and Systems (AHS), 2007.
- [AUB 1995] Aubury, M. and Watts, R. “*Handel-C Compiler, Release 1 Documentation and User's Manual*”, Oxford University Computing Laboratory, Oxford, 1995.
- [BAI 2007] B. Bailey, G. Martin, and A. Piziali, “*ESL Design and Verification: A Prescription for Electronic System Level Methodology*”, Morgan Kaufman Publishers, 2007.
- [BER 1993] C. Berrou, A. Glavieux, P. Thitimajshima, “*Near Shannon Limit Error-Correcting Coding and Decoding: Turbo Code*”. Proc. ICC'93, Geneva, Switzerland, p.1064-1070, Mai 1993.
- [CHA 2010a] C. Chavet, P. Coussy, P. Urard, E. Martin, “*Static Address Generation Easing: a Design Methodology for Parallel Interleaver Architectures*”, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2010.
- [CHA 2010b] C. Chavet, P. Coussy, “*A Memory Mapping Approach for Parallel Interleaver Design with Multiples Read and Write Accesses*”, IEEE International Symposium on Circuits and Systems (ISCAS), 2010.
- [CHA 2009a] Cyrille Chavet, Philippe Coussy, Pascal Urard, Eric Martin, “*Apparatus for data interleaving algorithm*”, CNRS – STMicroelectronics U.S. Patent application 20090031094, January 29, 2009.
- [CHA 2008a] Cyrille Chavet, Philippe Coussy, Pascal Urard, Eric Martin, « *Procédé et dispositif d'entrelacement de données* », CNRS – STMicroelectronics, Brevet Français n° 0754793 10, 30 Avril 2007.
- [CHA 2007] C. Chavet, “*Synthèse automatique d'interfaces de communication matérielles pour la conception d'applications du domaine du traitement du signal*”, Thèse de doctorat, Université de Bretagne Sud, 2007.

- [CHA 2007a] C. Chavet, C. Andriamisaina, P. Coussy, E. Casseau, E. Juin, P. Urard, E. Martin, “*A Design Flow Dedicated to Multi-mode Architectures for DSP Applications*”, IEEE International Conference on Computer Aided Design, (ICCAD) 2007.
- [CHA 2007b] C. Chavet, P. Coussy, P. Urard, E. Martin, “*Application of a design space exploration tool to enhance interleaver generation*”, European Signal Processing Conference (EUSIPCO) 2007.
- [CHA 2007c] C. Chavet, P. Coussy, P. Urard, E. Martin, “*A Methodology for Efficient Space-Time Adapter Design Space Exploration: A Case Study of an Ultra Wide Band Interleaver*”, IEEE International Symposium on Circuits and Systems (ISCAS), 2007.
- [CHA 2007d] C. Chavet, P. Coussy, P. Urard, E. Martin, “*A Design Methodology for Space-Time Adapter*”, ACM Great Lakes Symposium on VLSI on VLSI (GLSVLSI), 2007.
- [CHA 2005a] C. Chavet, P. Coussy, P. Urard, E. Martin, “*Méthodologie de modélisation et d’implémentation d’adaptateurs spatio-temporels*”, MANifestation des Jeunes Chercheurs STIC (MajecSTIC), 2005.
- [CHI 2005] L. Chiou, S. Bhunia and K. Roy, “*Synthesis of Application-Specific Highly Efficient Multi-mode Cores for Embedded Systems*”, ACM Transactions on Embedded Computing Systems, 2005.
- [COC 2000] J. C. Cockx, “*Efficient modeling of preemption in a virtual prototype*,” in Proceeding of International Symposium on Rapid System Prototyping RSP, 2000.
- [CON 2008] J. Cong and J. Xu, “*Simultaneous FU and register binding based on network flow method*”, In Proceedings of the conference on Design, automation and test in Europe (DATE '08), 2008.
- [COU 2010a] P. Coussy, A. Takach, M. McNamara, M. Meredith, “*An Introduction to the SystemC Synthesis Subset Standard*”, IEEE International Conference on Hardware-Software Codesign and System Synthesis (CODES+ISSS), 2010.
- [COU 2009a] P. Coussy, G. Gajski, A. Takach, M. Meredith, “*An Introduction to High-Level Synthesis*”, Special issue on High-Level Synthesis, IEEE Design and Test of Computers, Vol. 26, Issue 4, July/August, 2009.
- [COU 2009b] P. Coussy, A. Takach, “*Raising the Abstraction Level of Hardware Design*”, Special issue on High-Level Synthesis, IEEE Design and Test of Computers, Vol. 26, Issue 4, July/August, 2009.
- [COU 2009c] P. Coussy, A. Rossi, M. Sevaux, K. Sörensen, and K. Trabelsi, “*VNS for High Level Synthesis*”, In Proceedings of 8th Metaheuristics International Conference, MIC 2009, July 2009.
- [COU 2008] P. Coussy, A. Morawiec (EDS), “*High-Level Synthesis: From Algorithm to Digital Circuit*”, Springer, 2008
- [COU 2008a] P. Coussy, G. Le Breton, D. Heller, “*Multiple Word-Length High-Level Synthesis*”, EURASIP Journal on Embedded Systems, July, 2008.
- [COU 2008b] P. Coussy, C. Chavet, P. Bomel, D. Heller, E. Senn, E. Martin, “*GAUT: A High-Level Synthesis Tool for DSP applications*”, “High-Level Synthesis: From Algorithm to Digital Circuits”, Springer, Berlin, Germany, 2008.

- [COU 2007a] P. Coussy, E. Casseau, P. Bomel, A. Baganne, E. Martin, “*Constrained algorithmic IP design for system-on-chip*”, Integration, the VLSI Journal, Elsevier Science, Vol. 40, Issue 2, February 2007.
- [COU 2006a] P. Coussy, E. Casseau, P. Bomel, A. Baganne, E. Martin, “*A Formal Method for Hardware IP Design and Integration under I/O and Timing Constraints*”, ACM Transactions on Embedded Computing Systems, Vol 5, No. 1, pp. 29-53, 2006.
- [COU 2005a] P. Coussy, G. Corre, P. Bomel, E. Senn, E. Martin, “*A More Efficient and Flexible DSP Design Flow from MATLAB-SIMULINK*”, IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP), 2005.
- [COU 2005b] P. Coussy, G. Corre, P. Bomel, E. Senn, E. Martin, “*High-level synthesis under I/O Timing and Memory constraints*”, IEEE International Symposium on Circuits And Systems (ISCAS), 2005.
- [COU 2005c] P. Coussy, G. Corre, P. Bomel, E. Senn, E. Martin, “*Synthèse Comportementale Sous Contraintes de Communication et de Placement Mémoire pour les composants du TDSI*”, Colloque sur le Traitement du Signal et de l'Image (GRETSI), 2005.
- [COU 2004a] P. Coussy, D. Gnaëdig, A. Nafkha, A. Baganne, E. Boutillon, E. Martin, “*A Methodology for IP integration in DSP Soc: a case study of a MAP algorithm for turbo decoder*”, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2004.
- [COU 2004b] P. Coussy, A. Baganne, E. Martin, E. Casseau, “*Intégration Optimisée de Composants Virtuels orientés TDSI par la Synthèse d'Architecture*”, Colloque sur le Traitement du Signal et de l'Image (GRETSI), 2004.
- [COU 2003b] P. Coussy, A. Baganne, E. Martin, “*Communication and Timing Constraints Analysis for IP Design and Integration*”, IFIP WG 10.5 Very Large Scale Integration of System-on-Chip Conference (VLSI-SOC), 2003.
- [COU 2002a] P. Coussy, A. Baganne, E. Martin, “*Virtual Component IP Re-use in Telecommunication Systems Design: A Case Study of MPEG-2 / JPEG2000 Encoder*”, IEEE International Conference on Electronics, Circuits, and Systems (ICECS), 2002.
- [COU 2002b] P. Coussy, A. Baganne, E. Martin, “*IP Cores Integration in DSP System-On-Chip Designs*”, European Signal Processing Conference (EUSIPCO), 2002.
- [COU 2002c] P. Coussy, A. Baganne, E. Martin, “*Platform-Based Design For Digital Signal Processing Systems: A Case Study of MPEG-2 / JPEG2000 Encoder*”, IEEE International Conference on Communication Circuits and Systems (ICCCAS), 2002.
- [COU 2002d] P. Coussy, A. Baganne, E. Martin, “*A Design Methodology for IP Integration*”, IEEE International Symposium on Circuits and Systems (ISCAS), 2002.
- [COU 2002e] P. Coussy, A. Baganne, E. Martin, “*A Design Methodology for Integrating IP into SOC Systems*”, IEEE International Custom Integrated Circuits Conference (CICC), 2002.
- [COU 2002f] P. Coussy, A. Baganne, E. Martin, “*Réutilisation de Composant Virtuel dans les Systèmes de Télécommunication: Une Etude de Cas MPEG-2 /*

- JPEG2000*”, Journées Francophones sur l'Adéquation Algorithme Architecture (JFAAA), 2002.
- [COU 2002g] P. Coussy, A. Baganne, E. Martin, “*Analyse Fonctionnelle des Moyens de communication Proposés dans les Systèmes sur Silicium*”, Journées Francophones sur l'Adéquation Algorithme/Architecture (JFAAA), 2002.
- [DEG 2005] V. Degalahal, V. and T. Tuan, “*Methodology for high level estimation of FPGA power consumption*”, In Proceedings of the Asia and South Pacific Design Automation Conference ASP-DAC, 2005.
- [DIN 2005] L.Dinoi, S.Benedetto, “*Variable-size interleaver design for parallel turbo decoder architecture*”, IEEE Trans.On communications, Vol.53, No11, Nov. 2005.
- [DUT 1993] S. DUTT, “*New Faster Kernighan-Lin-Type Graph-Partitioning Algorithms*”, IEEE International Conference on Computer Aided Design (ICCAD), 1993.
- [FAU 2007] E. Faure, “*Communications matériel/logiciel dans les systèmes sur puce multi-processeurs orientés télécommunications*”, Thèse de doctorat, Université Paris 6, 2007
- [GAJ 1992] D. Gajski et al. “*High-Level Synthesis : Introduction to Chip and System Design*”, Kluwer Academic Publishers, 1992
- [GAJ 2000] D. Gajski, Z. Jianwen, R. Dömer, A. Gerstlauer, Z. Shuqing, “*SpecC: Specification Language and Methodology*”, Kluwer Academic Publisher, 2000.
- [GAR 1990] Michael R. Garey and David S. Johnson. “*Computers and Intractability; a Guide to the Theory of Np-Completeness*”, W. H. Freeman & Co., New York, NY, USA, 1990.
- [GLO 2002] Glover, F., & Kochenberger, G. (Eds.). “*Handbook of Metaheuristics*”. Norwell, MA, Kluwer Academic Publishers, 2002.
- [GUE 2010] X. Guerin, “*Approche Efficace de Développement de Logiciel Embarqué pour des Systèmes Multiprocesseurs sur Puce*”, Thèse de doctorat, Université de Grenoble, 2010.
- [GUT 2010] I. Gutierrez, A. Mourad, J. Bas, S. Pfletschinger, G. Bacci, A. Bourdoux, H. Gierszal, “*DAVINCI Non-Binary LDPC codes: Performance and Complexity Assessment*”, Published at proceedings of Future Network & Mobile Summit, Italy, June 2010.
- [HAY 1993] S. Haykin, “*Adaptive Filter Theory*”, Prentice-Hall Publishers, N.J., USA, 1993.
- [IEEE 2007] IEEE 802.15.3a, WPAN High Rate Alternative.
- [ITRS 2010] International Technology Roadmap for Semiconductors, “*Process Integration, Devices, and Structures*”, [en ligne] <http://www.itrs.net/home.html>
- [KRI 2005a] L. Kriaa, S. Adriano, E. Vaumorin, R. Nouacer, F. Blanc, S. Pajaniardja, P. Coussy, E. Martin, D. Heller, F. Thabet et al , “*SystemC'mantic : A high level Modeling and Co-design Framework For Reconfigurable Real Time Systems*”, Forum on Design Languages (FDL), 2005.

- [LAP 2011a] V. Lapotre, P. Coussy, C. Chavet, “*Prédiction de Branchement dans la Synthèse de Haut Niveau*”, 14ieme SYMPosium en Architecture (SYMPA), 10-13 Mai 2011.
- [LAU 2007a] J. Laurent, P. Coussy, “*Impact du type d'architecture sur la consommation d'une application*”, Journées Faible Tension Faible consommation (FTFC), 21-23 mai, 2007.
- [LAV 1999] L. Lavagno and E. Sentovich, “*ECL: A specification environment for system-level design*,” in Proc. Design Automation Conf. DAC'99, 1999.
- [LEE 1987] Edward A. Lee and David G. Messerschmitt, “*Synchronous Data Flow*”, in Proceedings of the IEEE, vol. 75, no. 9, p 1235-1245, September, 1987.
- [LEF 2010a] V. Lefftz, J. Bertrand, H. Cassé, C. Clienti, P. Coussy, L. Maillat-Contoz, P. Mercier, P. Moreau, L. Pierre, E. Vaumorin, “*A Design Flow for Critical Embedded Systems*”, IEEE Symposium on Industrial Embedded Systems (SIES), 2010.
- [LHA 2011a] G. Lhairech-Lebreton, P. Coussy, E. Martin, “*Synthèse d'Architecture Multi-horloges pour la Conception Faible Consommation sur FPGA*”, Colloque sur le Traitement du Signal et de l'Image (GRETSI), Septembre 2011.
- [LHA 2010a] G. Lhairech-Lebreton, P. Coussy, D. Heller, E. Martin, “*Bitwidth-Aware High-Level Synthesis for Designing Low-Power DSP Applications*”, IEEE International Conference on Electronics, Circuits, and Systems (ICECS), 2010.
- [LHA 2010b] G. Lhairech-Lebreton, P. Coussy, E. Martin, “*Hierarchical and Multiple-Clock Domain High-Level Synthesis for Low-Power Design on FPGA*”, IEEE International Conference on Field Programmable Logic and Applications (FPL), 2010.
- [LIA 1997] S. Liao, S. Tjiang, and R. Gupta. “*An efficient implementation of reactivity for modeling hardware in the Scenic design environment*”. In Proceedings of the Design Automation Conference, pages 70-75, June 1997.
- [MAC 1996] J.C.MacKay David and R.M.Neal, “*Near Shannon limit performance of low density parity check codes*”, Electronics letters, July 1996.
- [MAR 2009] G. Martin, G. Smith, “*High-Level Synthesis: Past, Present, and Future*”, *Special issue on High-Level Synthesis, IEEE Design and Test of Computers*, Vol. 26, Issue 4, July/August, 2009.
- [MEN 2011] MentorGraphics, “*Algorithmic C datatypes*”, <http://www.mentor.com/esl/catapult/algorithmic>
- [MPA 2011] MPARM: *Multi-processor cycle-accurate architectural simulator*, <http://www-micrel.deis.unibo.it/sitonew/research/mparm.html>
- [MOU 2008] H.Moussa, A.Baghdadi, M.Jezequel, “*Binary de Bruijn on-chip network for a flexible multiprocessor LDPC decoder*”. 45th ACM/IEEE Design Automation Conference (DAC), 2008.
- [MUL 2006] O.Muller, A.Baghdadi, M.Jezequel, “*ASIP-based multiprocessor SoC design for simple and double binary turbo decoding*”, in Design, Automation, and Test in Europe conference (DATE), 2006.

- [MUT 2000] J.Muttersbach, T.Villiger, W.Fichtner. “*Practical design of globally asynchronous locally synchronous systems*”. In Proc. Int. Symp. On Advanced Research in Asynchronous Circuits and Systems, 2000.
- [NAF 2005] A. Nafkha, C. Roland, E. Boutillon, “*A Near-Optimal Multiuser Detector for MC-CDMA systems Using Geometrical Approach*”, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2005.
- [PAN 1991] B. Pangrle, “*On the complexity of connectivity binding*”, Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, Vol 10, iss. 11, 1991.
- [PAP 1998] C. H. Papadimitriou, K. Steiglitz, “*Combinatorial Optimization: Algorithms and Complexity*”, Prentice-Hall Publishers, N.J., USA, 1998.
- [PET 1997] F. Pétrot, D. Hommais, and A. Greiner, “*Cycle precise core based hardware/software system simulation with predictable event propagation*,” in Proc. 23rd Euromicro Conf.,1997.
- [PIG 2005] C. Piguet, “*Low Power Electronics Design*”, CRC Press, 2005.
- [PIL 2001a] S. Pillement, O. Sentieys, D. Chillet, E. Casseau, P. Coussy, E.Martin, G. Savaton, S. Roux, “*Design and synthesis of behavioral level virtual components*”, IFIP WG 10.5 Very Large Scale Integration of System-on-Chip Conference (VLSI-SOC), 2001.
- [SAN 2011a] A. Sani, P. Coussy, C. Chavet, E. Martin, “*A Methodology based on Transportation Problem Modeling for Designing Parallel Interleaver Architectures*”, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2011.
- [SAN 2011b] A. Sani, P. Coussy, C. Chavet, E. Martin, “*An Approach Based on Edge Coloring of Tripartite Graph for Designing Parallel LDPC Interleaver Architecture*”, IEEE International Symposium on Circuits and Systems (ISCAS), 2011.
- [SAN 2010a] A. Sani, P. Coussy, C. Chavet, E. Martin, “*Design of Parallel LDPC Interleaver Architecture: A Bipartite Edge Coloring Approach*”, IEEE International Conference on Electronics, Circuits, and Systems (ICECS), 2010.
- [SAV 2001a] G. Savaton, P. Coussy, E. Casseau, E. Martin, “*A Methodology for Behavioral Virtual Component Specification Targeting SoC Design with High-Level Synthesis Tools*”, In Proc. of the Forum on Design Languages (FDL), 2001.
- [SOC 2011] SocLib, “*SoCLib an open platform for virtual prototyping of multi-processors system on chip (MP-SoC)*” [en ligne], <http://www.soclib.fr>
- [SYN 2011] Synopsys, <http://www.synopsys.com>
- [TAR 2004] A. Tarable, S. Benedetto, G. Montorsi, “*Mapping interleaving laws to parallel turbo and LDPC decoder architectures*”, IEEE Transactions on Information Theory, vol.50, p.2002 - 2009, septembre 2004.
- [THA 2009a] F. Thabet, P. Coussy, D. Heller, E. Martin, “*Exploration and Rapid Prototyping of DSP Applications using SystemC Behavioral Simulation and High-Level Synthesis*”, Journal of Signal Processing Systems, Springer, Issue 56, 2-3, September 2009.

- [THA 2007] F. Thabet, *Modélisation Comportementale Unifiée pour la Simulation et la Synthèse Haut-Niveau de composants Virtuels Algorithmiques*, Thèse de doctorat, Université de Bretagne Sud, 2007.
- [THA 2007a] F. Thabet, P. Coussy, D. Heller, E. Martin, “*Behavioral Description Model BDM for Design Space Exploration: a Case Study of His Algorithm for MC-CDMA System*”, European Signal Processing Conference (EUSIPCO) 2007.
- [THA 2006a] F. Thabet, P. Coussy, D. Heller, E. Martin, “*Design Space Exploration of DSP Applications Based on Behavioral Description Models*”, IEEE Workshop on Signal Processing Systems Design and Implementation (SIPS), 2006.
- [THA 2005a] F. Thabet, P. Coussy, E. Martin, “*Approche Automatique pour le Raffinement des Communications*”, Journées Francophones sur l'Adéquation Algorithme/Architecture (JFAAA), 2005.
- [THA 2004a] F. Thabet, J.-B. Legoff, P. Coussy, E. Martin, “*A Methodology for Timing and Structural Communication Refinement in DSP Systems*”, International Conference on Microelectronics (ICM), 2004.
- [TRA 2010a] K. Trabelsi, M. Sevaux, P. Coussy, A. Rossi, K. Sørensen, “*Advanced Metaheuristics for High-Level Synthesis*”, In *Metaheuristics*. Springer, 2010.
- [TRA 2009] K. Trabelsi, *Méthodes d'optimisation pour la conception sous contraintes de systèmes et de circuits électroniques*, Thèse de doctorat, Université de Bretagne Sud, 2009
- [TRA 2008a] K. Trabelsi, P. Coussy, A. Rossi, M. Sevaux, “*Ordonnancement et Assignation en Synthèse de Haut Niveau*”, 9ieme congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF) Février, 2008.
- [TUL 2002] M.J. Thul, F. Gilbert, and N. Wehn, “*Optimized concurrent interleaving architecture for high-throughput turbo-decoding,*” in Proc. 9th Int. Conf. Electron., Circuits, Syst., vol.3, pp.1099–1102, 2002.
- [URA 2007] P. Urard, J. Yi, Hyukmin Kwon, and A. Gouraud, “User needs”, “*High-Level Synthesis: From Algorithm to Digital Circuits*”, Springer, Berlin, Germany, 2008.
- [XIL 2011] Xilinx, “*Digital Clock Manager (DCM) Module*”, xilinx.com/support/documentation/ip_documentation/dcm_module.pdf