Conception des systèmes embarqués : applications, algorithmes et architectures

Yannick Le Moullec

HABILITATION A DIRIGER DES RECHERCHES Présentée par Yannick Le Moullec

UNIVERSITE DE BRETAGNE-SUD
sous le sceau de l'Université Bretagne Loire

Préparée à l'Unité Mixte de recherche n° 6285 - Laboratoire des sciences et techniques de l'information de la communication et de la connaissance (Lab-STICC), Université de Bretagne-Sud

Ecole doctorale : Santé, Information, Communications, Mathématiques, Matière (SICMA)

Mention : Sciences et Technologies de l'Information et de la Communications

Conception des systèmes embarqués : applications, algorithmes et architectures

Habilitation à Diriger des Recherches soutenue le 14/11/2016

Devant le jury composé de :

Christophe Jégo
Professeur des universités, Laboratoire IMS-CNRS UMR 5218, ENSEIRB-MATMECA, rapporteur

Daniel Chillet
Professeur des universités, UMR 6074 IRISA/INRIA, ENSSAT, Université de Rennes 1, rapporteur

Daniel Menard
Professeur des universités, IETR UMR CNRS 6164, INSA de Rennes, rapporteur

Frank Singhoff
Professeur des universités, Lab-STICC UMR 6285, Université Bretagne Occidentale, Examinateur

Jean-Philippe Diguet
Directeur de recherche (HDR), Lab-STICC UMR 6285, Université Bretagne Sud, Examinateur

Jean-Luc Philippe
Professeur des universités, Lab-STICC UMR 6285 Université, Bretagne Sud, Examinateur

Conception des systèmes embarqués : applications, algorithmes et architectures

Habilitation à diriger des recherches

Yannick Le Moullec

15 septembre 2016

Table des matières

Liste des figures

Liste des tableaux

Préface

Remerciements

Je remercie les membres du jury de cette d'habilitation à diriger des recherches : Messieurs Christophe Jégo, Daniel Chillet et Daniel Menard qui m'ont fait l'honneur d'être rapporteurs, ainsi que Messieurs Jean-Philippe Diguet, Jean-Luc Philippe et Frank Singhoff pour avoir chaleureusement acceptés d'être examinateurs.

Je suis reconnaissant envers Marc Sevaux, directeur du site lorientais du Lab-STICC, qui m'a donné l'opportunité de soutenir cette habilitation à diriger des recherches dans de bonnes conditions. Je remercie également les membres du Lab-STICC avec qui j'ai eu, et ai encore, la chance de travailler depuis mes années de doctorat ; je m'y sens le bienvenu à chacune de mes visites grâce aux échanges scientifiques et pédagogiques que j'ai avec ses membres, notamment Jean-Philippe Diguet, Johann Laurent et Guy Gogniat. Je remercie aussi Florence Palin et Virginie Guillet pour leur accueil toujours souriant.

Mes remerciements vont aussi à Monsieur Peter Koch pour m'avoir accueilli et soutenu pendant mes dix années à Aalborg University au Danemark. Grâce à lui j'ai eu l'opportunité de travailler sur des projets scientifiques divers et variés et de co-encadrer mes premiers doctorants ; une bonne partie de ce document n'existerait pas sans eux. C'est aussi grâce à lui que j'ai pu développer mes compétences en apprentissage par problèmes *problem-based learning (PBL)* via l'encadrement de nombreux projets et thèses de master.

Je remercie Monsieur Toomas Rang pour son accueil amical au sein de Tallinn University of Technology en Estonie où j'ai actuellement le plaisir de poursuivre ma carrière d'enseignant-chercheur, notamment de piloter et contribuer à divers projets de recherche et encadrer ou co-encadrer des doctorants.

Enfin, toute ma gratitude va à l'ensemble des étudiants de master et de doctorat que j'ai eu l'honneur de (co)encadrer ces dernières 12 années ; les résultats présentés dans ce document sont aussi les leurs.

Objectifs du document et "avertissement"

Ce document synthétise mes activités de recherche, d'enseignement et d'animation scientifique effectuées depuis l'obtention de mon doctorat en avril 2003. Ce doctorat m'a été conféré par l'Université de Bretagne Sud (UBS) pour mes travaux de recherche effectués au sein du Laboratoire d'Électronique des Systèmes Temps-Réels (LESTER, maintenant Lab-STICC) sous la direction de messieurs Jean-Luc Philippe et Jean-Philippe Diguet.

J'ai ensuite passé dix ans à Aalborg University (AAU) au Danemark, où j'ai été successivement post-doctorant, *assistant professor* (équivalent maître de conférences 'junior'), *associate professor* (équivalent maître de conférences 'confirmé'). Durant mes années danoises, j'ai été rattaché à Center for Embedded Software Systems (CISS), Center for Software Defined Radio (CSDR) et Department of Electronics.
Depuis août 2013 je suis *Senior Research Scientist* (équivalent chargé de recherche de 1ère classe) au sein de Thomas Johann Seebeck Department of Electronics à Tallinn University of Technology (TUT) en Estonie.

Je souhaite dès à présent attirer l'attention du lecteur sur le fait que j'ai été, et suis encore, employé sur contrats. Ces contrats sont généralement le résultat d'une combinaison de diverses sources financières, principalement projets de recherche et obligations d'enseignement. De plus, et notamment à Aalborg University, l'obtention des financements pour ces projets de recherche était conditionnée par la participation d'entreprises (PME).
Je souhaite également attirer l'attention du lecteur sur mon implication forte en enseignement via notamment l'encadrement de 65 étudiants ou groupes, dont 34 thèses de master (certaines en coopération avec des entreprises).

Bien que ces deux types de coopérations m'aient donné l'avantage de pouvoir travailler sur des problèmes concrets, elles ont aussi abouti à une peut-être trop grande diversité de mes travaux de recherche et à la coloration ingénierie de beaucoup d'entre eux. En conséquence, ce document est lui-même organisé par projet plutôt que par thématique, et certains lecteurs regretteront peut-être l'absence d'un fil conducteur plus net. J'ai fait ce choix car il correspond au parcours qui fût le miens, ce cheminement m'a permis d'acquérir l'expérience qui m'amène à présenter aujourd'hui cette HDR.

Organisation du document

Ce document est organisé en trois parties. La première a pour objectif de donner une vue d'ensemble de mon profil d'enseignant-chercheur et de résumer ma carrière jusqu'à aujourd'hui. Elle est constituée de mon CV détaillé qui couvre mon parcours professionnel et ma formation (chapitre 1), les animations et services scientifiques auxquels j'ai contribué (chapitre 2), mes responsabilités administratives (chapitre 3), le résumé de mes activités de recherche (chapitre 4), la liste de mes publications (chapitre 5) et mes activités d'enseignement (chapitre 6).

La deuxième partie présente une sélection de mes travaux de recherche. J'ai choisi cinq projets de recherche couvrant les aspects suivants : conception conjointe logicielle/matérielle (chapitre 7), systèmes reconfigurables (chapitre 8), systèmes embarqués pour aide à la personne (chapitre 9), calcul stochastique (chapitre 10) et architecture embarquée pour nano-satellite (chapitre 11). Pour chacun de ces projets je présente de manière relativement concise le contexte dans lequel les travaux ont été réalisés, la ou les problématiques abordées et les contributions apportées. Chacune de ces descriptions est suivie d'une ou plusieurs reproductions d'articles qui offrent au lecteur la possibilité de parcourir ces travaux de recherche de manière plus détaillée.

Enfin, la troisième partie donne une vue d'ensemble de mes projets en cours (chapitre 12), de mes perspectives de recherche (chapitre 13) ainsi que quelques réflexions sur l'enseignement et ses liens avec la recherche (chapitre 14).

Partie I

CV détaillé

Chapitre 1

Parcours professionnel et formation

Né le 29 juin 1975, Ploërmel, France
Nationalité française
Adresse professionnelle : Ehitajate tee 5, U02B-209, EE-19086 Tallinn, Estonie
Adresse électronique : yannick.lemoullec@ttu.ee
Téléphone : +372 5844 6540 (portable)

1.1 Parcours professionnel

- 2013/ - : *senior research scientist* (équivalent chargé de recherche de 1ère classe)[1], Johann Seebeck Department of Electronics, Tallinn University of Technology, Tallinn, Estonie.
  - Juin/août 2015 (3 mois) : chercheur invité, National Taipei University of Technology, Taipei, Taiwan.
  - Mi-juin/Mi-juillet 2014 (1 mois) : professeur invité, Lab-STICC, Université de Bretagne Sud, Lorient, France.
- 2009/2013 : *associate professor* (équivalent maître de conférences 'confirmé')[2], Technology Platforms Section (TPS), Department of Electronic Systems, Aalborg University, Danemark.
  - Mi-septembre/mi-décembre 2012 (3 mois) : chercheur invité, Shanghai Institute of Microsystem And Information Technology (SIMIT), Chinese Academy of Sciences (CAS), Shanghai, Chine.
  - 2009/2010 : également associé au Center for Software Defined Radio (CSDR), Department of Electronic Systems, Aalborg University, Danemark.

[1] En Estonie, les corps de chercheurs sont research scientist, research scientist (PhD)(équivalent post-doc), senior research scientist (équivalent CR2/CR1 selon ancienneté) et lead research scientist (équivalent DR2/DR1 selon ancienneté) ; ceux des enseignant-chercheurs sont assistant/lecturer (équivalent vacataire/chargé de cours), assistant (PhD)/lecturer (PhD)(équivalent maître de conférences 'junior'), associate professor (équivalent maître de conférences 'confirmé') et professor (équivalent professeur des universités).

[2] Au danemark, les corps d'enseignant-chercheurs sont external lecturer (chargé de cours), post-doc, assistant professor (équivalent maître de conférences 'junior'), associate professor (équivalent maître de conférences 'confirmé') et professor (équivalent professeur des universités).

- 2005/2009 : *assistant professor*[3], TPS & CSDR, Department of Electronic Systems, Aalborg University, Danemark.
  - Juillet 2006 (1 mois) et septembre 2007 (1 mois) : professeur invité, LESTER/Lab-STICC, Université de Bretagne Sud, Lorient, France.
- 2003/2005 : post-doc, Center for Embedded Software Systems (CISS), Aalborg University, Danemark.
- 2002/2003 : ATER, Université de Bretagne Sud, Lorient, France.
- 1999/2002 : vacataire, Université de Bretagne Sud, Lorient, France.

Les postes que j'ai successivement occupés (doctorat inclus) sont également résumés dans la figure 1.1.

1.2 Formation

- 2003 : doctorat sciences pour l'ingénieur, Laboratoire d'Électronique des Systèmes Temps-Réels (LESTER, maintenant Lab-STICC), Université de Bretagne Sud, Lorient, France.

  Titre de la thèse : Aide à la conception de systèmes sur puce hétérogènes par l'exploration paramétrable des solutions au niveau système.

  Jury : Michel Auguin, Jean-Philippe Diguet, Christian Gamrat, Jean-Luc Philippe, Olivier Sentieys et Lionel Torres.
- 1999 : DEA Électronique, Université de Rennes I (co-habilité INSA Rennes et Supélec Rennes), France.
- 1998 : maîtrise EEA, Université de Bretagne Sud, Lorient, France.
- 1997 : licence EEA, Université de Bretagne Sud, Lorient, France.
- 1996 : formation post-BTS réseaux, Lycée la Croix-Rouge, Brest, France.
- 1995 : BTS informatique industrielle, Lycée St-Joseph, Lorient, France.
- 1993 : baccalauréat F3, Lycée St-Ivy, Pontivy, France.

1.3 Développement professionnel et personnel

- Qualification MCF section 61 du CNU (31/01/2013 — 31/12/2017).
- "Giving Presentations and Lectures in English", Tallinn University of Technology, 2016.
- "Estonian for Beginners (short introductory course)", Tallinn University of Technology, 2014.

[3] Voir note 2 en page précédente.

- "PhD Supervisor Course", Aalborg University, 2011.
- "Introductory Course in First Aid", Aalborg University, 2010.
- "How to Write a Professional and Successful EU Proposal", Aalborg University, 2008.
- "Academic Writing and Presentation in English", Aalborg University, 2008.
- "University Course in Pedagogy for Assistant Professors", Aalborg University, 2005-2007.
- "Teaching in English", Aalborg University, 2007.
- "IEEE Region 8 workshop on Development of Leadership Skills", Aalborg University, 2007.
- "Writing Competitive Research Project Proposals for the First Time", Aalborg University, 2007.
- "Introduction to Problem Based Learning - the AAU way", Aalborg University, 2003.

1.4 Bourses obtenues à titre personnel

- 2015, Récipiendaire, Taiwan Fellowship, Ministry of Foreign Affairs, République de Chine (Taiwan). Chercheur invité, National Taipei University of Technology, Taipei, Taiwan.
- 2012, Récipiendaire, Grant for Research Stays Abroad, Otto Mønsted Foundation, Danemark. Chercheur invité, Shanghai Institute of Microsystem and Information Technology (SIMIT), Chinese Academy of Sciences (CAS), Shanghai, Chine.

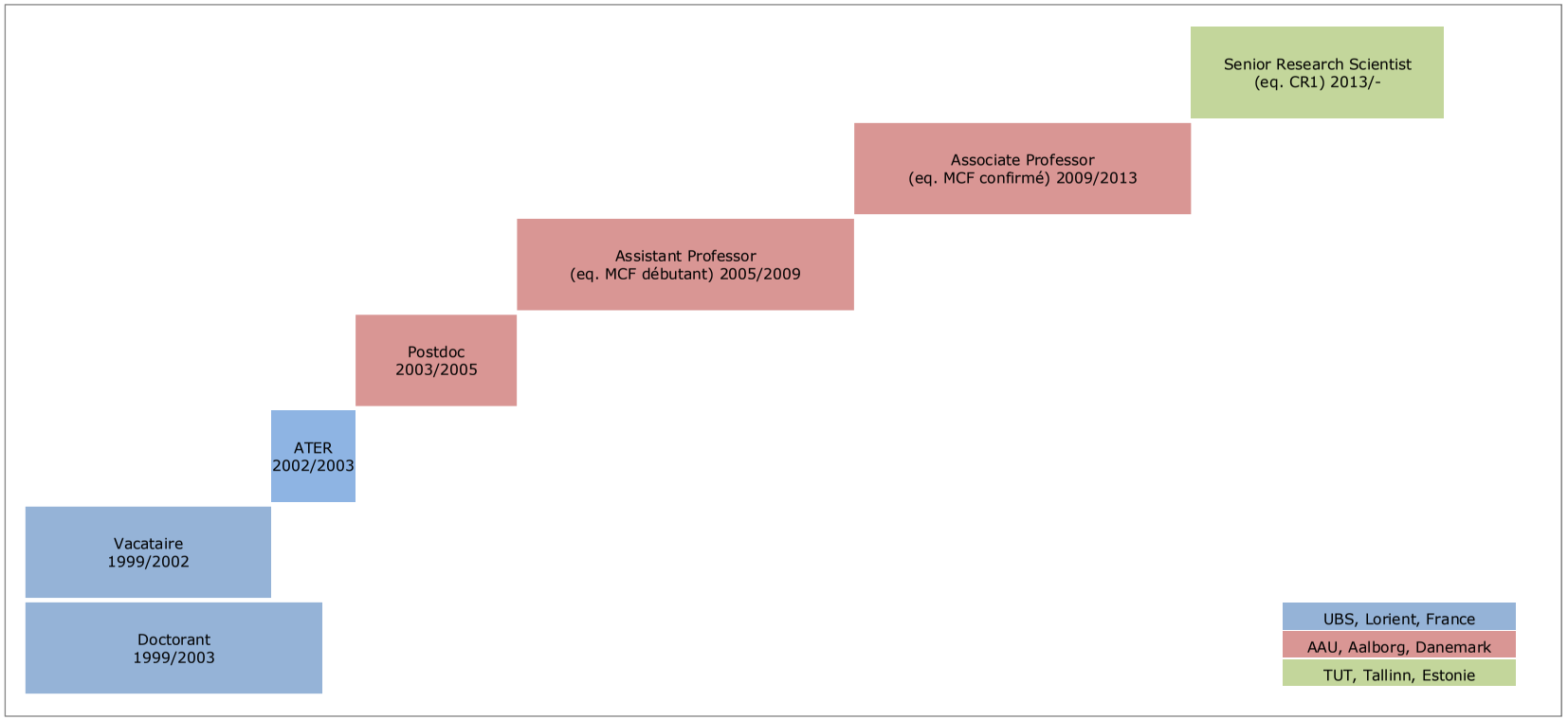


Figure 1.1: Postes occupés depuis octobre 1999.

Chapitre 2

Animations et services scientifiques

2.1 Séminaires et assimilés

- Aôut 2015 : "Information and Communication Technologies for Societal Challenges: Personalized Healthcare", Discours programme, ICBIS 2015, Taipei, Taiwan.
- Juillet 2014 : "HW/SW Codesign: Selected Research Activities", Shanghai Institute of Microsystem and Information Technology (SIMIT), Chinese Academy of Sciences (CAS), Shanghai, Chine.
- Juillet 2014 : "HW/SW Codesign: Selected Research Activities", Northeast Microelectronic Institute, Shenyang, Chine.
- Février 2016, mars 2015, mars 2014, janvier 2012, octobre 2011, mars 2010, mars 2009, mars 2008, décembre 2006 : bourse Erasmus enseignant, séminaires sur l'implantation du traitement numérique en bande de base (2015-2016) et la conception conjointe logicielle/matérielle (2006-2014) ; suivi échanges étudiants UBS-TUT/UBS-AAU, Université de Bretagne Sud, Lorient, France.
- Octobre 2009 : cours invité sur la conception conjointe logicielle/matérielle et la technologie FPGA, Tallinn University of Technology, Tallinn, Estonie.
- Février 2009 : atelier, "SDR in Terrestrial Applications", Copenhague, Danemark.
- Septembre 2008 : salon, "Demonstration of FPGA Technology", Elektronikmesse i Odense, Danemark.
- Août 2006 : séminaire, "HW/SW Codesign : Motivation, History, Trends, and Design-Trotter", Copenhagen University College of Engineering, Ballerup, Danemark.
- Mai 2006 : séminaire, "HW/SW Codesign : Motivation, History, Trends, and Design-Trotter", Aalborg University, Danemark.
- Janvier 2005 : atelier, "Demonstration of Design-Trotter", SummIT05, Aarhus, Danemark.

2.2 Services scientifiques

- Membre, panel d'experts évaluateurs, demandes de financements post-doc, Estonian Research Council, Estonie, 2016.
- Membre, jury de thèse de doctorat, E. Moorits, TUT, Tallinn, Estonie, 2016.
- Membre, comité de sélection pour un poste *associate Professor*, School Of Engineering, Aarhus University, Danemark, 2012.
- Évaluateur, dossier de projet IWT, Agency for Innovation by Science and Technology for Flemish Companies and Research Centers, Belgique, 2010.
- Membre, jury de thèse de doctorat, A. Vander Biest, ULB, Bruxelles, Belgique, 2009.
- Membre, comité de sélection pour un poste *assistant Professor*, Engineering College of Aarhus, Danemark, 2009.
- Relecteur pour les journaux suivants :
  - IEEE Transactions on Circuits and Systems II: Express Briefs (IEEE)
  - Journal of Real-Time Image Processing (Springer)
  - EURASIP Journal on Wireless Communications and Networking (EURASIP)
  - Journal of Embedded Computing (IOS PRESS)
  - International Journal of Reconfigurable Computing (Hindawi)
- Membre du comité de pilotage, BEC'16.
- Co-président de programme, DASIP'16 ; Co-président, ICBIS'15.
- Organisateur et président de séance, session spéciale 'Implementation Methodologies for Communication and Biomedical Systems', ISABEL'08, Aalborg, Denmark.
- Membre, comités scientifiques/techniques ou relecteur, IEEE NORCAS'15-16, DASIP'07-16, IAIT'13 ; '15, RUC'12-15, IEEE NORCHIP'07-12, IEEE BEC'12 ; '14, ISABEL'11, COSIT'11, AUC'10, SPL'09-10, ICSPCS'09, MOWIN'09, AISPC'08, IEEE ICICS'07, AISPC'07, PIMRC'07, SEC'07, EUC'05, etc.

Chapitre 3

Responsabilités administratives

- Membre adjoint, comité d'évaluation des doctorants de deuxième année, Faculty of Information Technology, Tallinn University of Technology, Estonie, 2016.
- Membre, comité d'évaluation des doctorants de deuxième année, Faculty of Information Technology, Tallinn University of Technology, Estonie, 2015.
- Facilitateur Erasmus (Université de Bretagne Sud, France et University of Trento, Italie), Tallinn University of Technology, Estonie, 2014/-.
- Coordinateur Erasmus pour Université de Bretagne Sud (France), École Nationale Supérieure des Sciences Appliquées et de Technologie (France), Polytech Nice-Sophia (France), Telecom-Bretagne (France) et Tallinn University of Technology (Estonie), Aalborg University, Danemark, 2004/2012.
- Coordinateur semestriel pour le master international Applied Signal Processing and Implementation (ASPI), Aalborg University, 2008/2011, Danemark.

Chapitre 4

Résumé de mes activités de recherche

Mes activités de recherche concernent, pour l'essentiel, les méthodes et outils de conception pour les systèmes embarqués visant une vaste gamme d'applications. Pour reprendre la nomenclature de la section 61 du CNU, les mots-clefs associés à mes travaux sont : ordonnancement, adéquation algo-architecture, objets communicants, systèmes de telecom, systèmes embarqués.
J'ai participé, à divers degrés, à 19 projets (dont 16 après mon doctorat) au travers desquels j'ai pu confirmer mon intérêt pour la recherche et développer les qualités qu'elle requière (créativité, curiosité scientifique, rigueur, etc.) Un récapitulatif de ces projets est donné dans la section 4.1.
J'ai également eu plaisir à (co)encadrer des doctorants (voir section 4.2) et à participer à la vie de la communauté scientifique au travers de divers séminaires, participation à l'organisation de conférences, relecture pour journaux et conférences etc. (voir chapitre 2 plus haut). Enfin, la liste de mes publications est disponible dans le chapitre 5.

4.1 Récapitulatif des mes projets de recherche

Comme indiqué en préface, l'ensemble de ma carrière après mon doctorat repose sur des contrats financés en partie par des projets de recherche. De plus, lorsque j'étais à Aalborg University, ces projets de recherche nécessitaient l'implication d'entreprises (PME) locales ou succursales d'entreprises étrangères.
Le tableau 4.1 présente un récapitulatif des projets auxquels j'ai participé après mon doctorat (pour information, ceux auxquels j'ai participé pendant mon doctorat sont donnés dans le tableau 4.2).
Pour chaque projet sont indiqués la période (triée par année de fin décroissante), l'intitulé, mon rôle (participant, leader d'activité, chercheur principal, etc.), des mots-clefs, la ou les sources de financement, le budget lorsque l'information est publique, et le ou les partenaires.

4.2 Encadrement doctoral

La durée nominale du doctorat est de quatre ans en Estonie et de trois ans au Danemark. Pour information, les thèses de master que j'ai (co)encadrées sont indiquées en section 6.2.

Période (par année de fin)	Intitulé	Rôle	Mots-clefs	Financement	Budget	Partenaire(s)
2015/2019	ERA-Chair COEL (COgnitive ELectronics)	Co-demandeur principal et responsable scientifique	Cognitive-, smart-, micro-, nano-, Embedded Electronics, IoT, Monitoring, Integration, Synergy, Cohesion, spreading excellence	UE/EC H2020 Widespread/ERA-Chair H2020	2500000 EUR	TUT (EE)
2016/2017	ESS/Ethercat	Participant	Ethercat, FPGA, European Spallation Source	UE/ESS	–	TUT (EE), ESS (UE)
2014/2017	B38-Hardware and Software Solutions for Cognitive Embedded Networks Systems - Application to Personalized Health-Monitoring	Chercheur principal	Personalized Health-Monitoring, Wireless Sensor Networks, Reasoning Solutions	TUT Baseline	72450 EUR	TUT (EE)
2014/2014	A Software-Defined Radio Platform for Teaching Cognitive Communication Systems	Chercheur principal	SDR, enseignement	Information Technology Foundation for Education - Tiger University	4199 EUR	TUT (EE)
2013/2014	Black Box Project - Wearable Signal Processing and Wireless Communication Solutions for WBASN Targeting Health-Monitoring Applications	Participant	WSN, 6LowPAN, DSP	CEBE	–	TUT (EE), CEBE (EE), ELIKO (EE)
2010/2013	FPGAs in Space - Low-Power Signal Processing Capacity for Nano-Satellites	Participant, co-encadrement d'un doctorant	'Probabilistic Computing', FPGA	EU Structural Funds/AAU	204174 EUR	AAU (DK), Gomspace ApS (DK)
2010/2012	Global Air Traffic Awareness and Optimization through Space Based Surveillance (GATOSS)	Co-chercheur principal, leader de l'activité d'implantation sur FPGA	Nano-satellite, FPGA	Danish National Advanced Technology Foundation	495825 EUR	AAU (DK), Gompsace ApS (DK)
2009/2010	Adaptive Tongue-Controlled Interface	Leader de l'activité d'implantation interfaces Bluetooth et Zigbee	Paralysie, interface homme-machine, Systèmes sans fils	Danish Ministry of Higher Education and Science	165915 EUR	AAU (DK), TKS AS (DK)
2009/2010	Reconfigurable MMSE Equalizer for a MIMO Receiver	Participation à la rédaction de la demande de financement et co-encadrement d'un doctorant	Systèmes sans fils, systèmes reconfigurables, FPGA	CSDR, Embassade France au DK	–	AAU (DK), Telecom-Bretagne (F)
2007/2010	Methodologies for Mapping Multiple Functionalities to Reconfigurable Heterogeneous Architectures	Participant, co-encadrement d'un doctorant	Systèmes reconfigurables, FPGA, partitionnement, ordonnancement	Danish Ministry of Higher Education and Science	228154 EUR	AAU (F), Rohde & Schwarz Technology Center A/S (DK), OFFIS (DE)
2005/2010	Methods for Accelerated Design to FPGA Technology	Participant, co-encadrement d'un doctorant	Métriques pour exploration de l'espace de conception, estimation de l'effort d'implantation sur FPGA	MHES: Danish Ministry of Higher Education and Science/ETI A/S	200000 EUR	AAU (DK), ETI A/S (DK), UBS (F)
2009/2009	Switchable MIMO Receiver	Participant	Systèmes sans fils, Systèmes reconfigurables, FPGA	–	–	AAU (DK), Agilent (B)
2009/2009	SDR Implementation of a Multipoint to Point Channel Emulator	Participant	Systèmes sans fils, systèmes reconfigurables, FPGA	–	–	AAU (DK), Agilent (B)
2003/2009	Design Trotter, partie 2	Extension des travaux de doctorat	Métriques et ordonnancement pour exploration de l'espace de conception	–	–	UBS(F)/AAU (DK)
2004/2006	MAGNET	Participant, co-encadrement de deux ingénieurs de recherche	Évaluation d'outils de conception conjointe logicielle/matérielle	EU FP6-IST	1666667 EUR	AAU (DK), 37 partenaires UE
2005/2005	RaaR	Participant	Veille technologique et partage de connaissances dans le domaine des systèmes reconfigurables	–	–	AAU (DK), UBS (F)

Tableau 4.1: Récapitulatif des projets de recherche auxquels j'ai participé après mon doctorat.

Période (par année de fin)	Intitulé	Rôle	Mots-clefs	Financement	Budget	Partenaire(s)
1999/2009	Design Trotter, partie 1	Participant, principaux travaux de doctorat	Métriques et ordonnancement pour exploration de l'espace de conception	LESTER	–	UBS (F)
2000/2003	EPICURE	Participant, travaux connexes à ceux de doctorat	Métriques et ordonnancement pour exploration de l'espace de conception	RNTL	–	UBS (F), Université Nice (F), CEA (F), Esterel Technologies (F), Thales (F)
1999/2000	MACGTT	Participant, travaux connexes à ceux de doctorat	Métriques et ordonnancement pour exploration de l'espace de conception	CNRS	–	UBS (F), Université Nice (F), ENSSAT (F)

Tableau 4.2: Récapitulatif des projets de recherche auxquels j'ai participé pendant mon doctorat.

4.2.1 Thèses de doctorat en cours

- Encadrant (100%). Tauseef Ahmed, "Advanced Radio Resource Management in Wireless Networks with Emphasis on Cognitive Radio Networks and Wireless Sensor Networks" (titre de travail), 2014-2018. Tallinn University of Technology.

- Co-encadrant (approx. 70%). Faisal Ahmed, "Energy Harvesting and Qos Solution for the Reliability of the Communication Channel in the context of WSNs - Application to Wireless Body Area Networks" (titre de travail), 2014-2018 (prévision). Co-encadrant : Paul Annus (approx. 30%). Tallinn University of Technology.

- Co-encadrant (approx. 30%). Anindya Gupta, "Automatic Detection of Lung Nodules from CT Images" (titre de travail), 2014-2018 (prévision) : Co-encadrants : Olev Märtens (approx. 30%), Tõnis Säär (approx. 40%). Tallinn University of Technology.

- Co-encadrant (approx. 70%). Tariq Meeran, "The Impact of Wireless Mesh Networks on Voice over Internet Protocol" (titre de travail), 2013-2017 (prévision). Co-encadrant : Paul Annus. Coopération Kabul University/Tallinn University/Tallinn University of Technology.

4.2.2 Thèses de doctorat finalisées

- Co-encadrant (approx. 40%). Yar Muhammad Mughal, "Parametric Framework for Modelling of Bioelectrical Signals". 2011 - 2015 (pris en cours, 2013). Co-encadrants : Toomas Rang (30%), Paul Annus (30%). Situation actuelle du diplômé : Chargé de cours, Tartu University, Estonie.

- Co-encadrant (approx. 50%). Alex Birklykke, "Modeling and Predicting the Behavior of Computers Operating without Guard-Bands - An Experimental Approach Based on Voltage-Scaled FPGAs". 2010 - 2013 (soutenance Février 2015 cause emploi industrie). Co-encadrants : Peter Koch (30%), Rakesh Kumar (10%), Ramjee Prasad (5%), Lars Alminde (5%). Situation actuelle du diplômé : Ingénieur applications FPGA, Rohde & Schwarz Technology Center A/S, Danemark.

- Co-encadrant (approx. 70%). Andreas Popp, "Mapping Framework for Heterogeneous Reconfigurable Architectures – Combining Temporal Partitioning and Multiprocessor Scheduling", 2007 - 2010 (soutenance 2010 cause emploi industrie). Co-encadrants : Peter Koch (20%), Kim Gruettner (10%). Situation actuelle du diplômé : Ingénieur logiciel, Baader Logistix A/S, Danemark.

- Co-encadrant (approx. 60%). Rasmus Abildgren, "Implementation Effort and Parallelism – Metrics for Guiding Hardware/Software Partitioning in Embedded System Design", 2006 - 2010. Co-encadrants : Peter Koch (30%), Jean-Philippe Diguet (10%). Situation actuelle du diplômé : Ingénieur logiciel sénior, Samsung Denmark Research Center, Danemark.

Chapitre 5

Publications

Revues avec comité de lecture (19)

[J1] Y. M. Mughal, P. Annus, Y. Le Moullec, and T. Rang, "A parametric framework for the development of bio-electrical applications - application to a bio-impedance signal simulator," *Proceedings of the Estonian Academy of Sciences*, vol. To appear, no. To appear, To appear, 2016.

[J2] F. Ahmed, Y. Le Moullec, and P. Annus, "Fypsim: Evaluation tool for solar-based energy harvesting for wsns," *International Journal of Bioelectromagnetism*, vol. 17, no. 2, pp. 75–86, 2015.

[J3] A. F. Cattoni, Y. Le Moullec, and C. Sacchi, "Efficient fpga implementation of a stbc-ofdm combiner for an ieee 802.16 software radio receiver," *Telecommunication Systems*, vol. 56, no. 2, pp. 245–255, 2014.

[J4] A. Ulbinaite, M. Kucinskiene, and Y. Le Moullec, "The complexity of the insurance purchase decision making process," *Transformations in Business & Economics*, vol. 13, no. 3, 2014.

[J5] B. Paul, S. Marcombes, A. David, L. N. Andreasen Struijk, and Y. Le Moullec, "A context-aware user interface for wireless personal-area network assistive environments," *Wireless Personal Communications*, vol. 69, no. 1, pp. 427–447, 2013.

[J6] A. Ulbinaite, K. Marija, and Y. Le Moullec, "Determinants of insurance purchase decision making in lithuania," *Engineering Economics*, vol. 24, no. 2, pp. 144–159, 2013.

[J7] M. U. R. Awan, Y. Le Moullec, P. Koch, and F. Harris, "Hardware architecture of polyphase filter banks performing embedded resampling for software-defined radio front-ends," *Special issue on digital front-end and RF processing for ZTE Communications: An International Journal*, vol. 10, no. 1, pp. 54–62, 70, 2012.

[J8] ——, "Polyphase filter banks for embedded sample rate changes in digital radio front-ends," *Special issue on digital front-end and RF processing for ZTE Communications: An International Journal*, vol. 9, no. 4, pp. 3–9, 2012.

[J9] A. Ulbinaite, M. Kucinskiene, and Y. Le Moullec, "Conceptualising and simulating insurance consumer behaviour: An agent-based-model approach," *International Journal of Modeling and Optimization*, vol. 1, no. 3, pp. 250–257, 2011.

[J10] S. Chitti, G. Kulkarni, A. Popp, and Y. Le Moullec, "Flexible and reconfigurable implementation of link adaptation algorithms," *Wireless Personal Communications*, vol. 54, no. 1, pp. 83–93, 2010.

[J11] A. Ulbinaite and Y. Le Moullec, "Towards an abm-based framework for investigating consumer behaviour in the insurance industry," *Economics*, vol. 89, no. 2, pp. 95–110, 2010.

[J12] R. Abildgren, J.-P. Diguet, P. Bomel, G. Gogniat, P. Koch, and Y. Le Moullec, "A priori implementation effort estimation for hardware design based on independent path analysis," *EURASIP J. Embedded Syst.*, vol. 2008, 2:1–2:12, Jan. 2008.

[J13] D. Idris, Y. Le Moullec, and P. Eggers, "Design and implementation of self-calibration for digital predistortion of power amplifiers," *Trans. Cir. and Sys.*, vol. 7, no. 2, pp. 75–84, Feb. 2008.

[J14] J. P. Diguet, G. Gogniat, J. L. P. Philippe, Y. Le Moullec, S. Bilavarn, C. Gamrat, K. Ben Chehida, M. Auguin, X. Fornari, and P. Kajfasz, "Epicure: A partitioning and co-design framework for reconfigurable computing," *Microprocessors and Microsystems, Special Issue on FPGA's*, vol. 30, no. 6, pp. 367–387, 2006.

[J15] Y. Le Moullec, J.-P. Diguet, N. Amor, T. Gourdeaux, and J.-L. Philippe, "Algorithmic-level specification and characterization of embedded multimedia applications with design trotter," *Journal of VLSI signal processing systems for signal, image and video technology*, vol. 42, no. 2, pp. 185–208, 2006.

[J16] S. Munagala, R. Muchanthula, Y. Le Moullec, P. Koch, and L. Kristensen, "Hsdpa-adaptive modulation and coding: Simulation and implementation," *Trans. Comp. Res.*, vol. 1, no. 2, pp. 133–139, 2006.

[J17] N. B. Amor, Y. Le Moullec, J. P. Diguet, J. L. Philippe, and M. Abid, "Design of a multimedia processor based on metrics computation," *Adv. Eng. Softw.*, vol. 36, no. 7, pp. 448–458, Jul. 2005.

[J18] Y. Le Moullec, J.-P. Diguet, T. Gourdeaux, and J.-L. Philippe, "Design-trotter: System-level dynamic estimation task a first step towards platform architecture selection," *J. Embedded Comput.*, vol. 1, no. 4, pp. 565–586, Dec. 2005.

[J19] Y. Le Moullec, J.-P. Diguet, D. Heller, and J. L. Philippe, "Estimation du parallélisme au niveau système pour l'exploration de l'espace de conception de systèmes enfouis," *Technique et Science Informatiques*, vol. 22, no. 3, pp. 315–349, 2003.

Conférences avec comité de lecture (63)

[C1] T. Ahmed, F. Ahmed, and Y. Le Moullec, "Optimization of channel allocation in wireless body area networks by means of reinforcement learning," in *IEEE Asia Pacific Conference on Wireless and Mobile 2016*, 2016.

[C2] T. M. Meeran, P. Annus, and Y. Le Moullec, "The current state of voice over internet protocol in wireless mesh networks," in *International Conference on Advances in Computing, Communications and Informatics 2016*, 2016.

[C3] F. Ahmed, Y. Le Moullec, and P. Annus, "Analytical evaluation of indoor energy harvesting technologies for wsns with fypsim framework," in *IEEE International Conference on Industrial Informatics and Computer Systems 2016*, 2016.

[C4] Y. Le Moullec and J.-P. Diguet, "Transient computing pour les réseaux de capteurs sans fil à récolte d'énergie : Méthodes d'hibernation et aspects architecturaux," in *Conférence d'informatique en Parallélisme, Architecture et Système*, 2016.

[C5] F. Ahmed, Y. Le Moullec, and P. Annus, "Fypsim: An estimation framework for energy harvesting and energy prediction for wsns," in *IEEE International Conference on Consumer Electronics - Taiwan, 2016*, 2016.

[C6] A. Gupta, O. Märtens, Y. Le Moullec, and T. Saar, "Methods for increased sensitivity and scope in automatic segmentation and detection of lung nodules in ct images," in *15th IEEE International Symposium on Signal Processing and Information Technology 2015*, 2015.

[C7] T. Ahmed and Y. Le Moullec, "Frequency and power allocation schemes for heterogeneous networks including femto cells," in *23rd Telecommunications Forum TELFOR 2015*, 2015.

[C8] M. El-Sayed, P. Koch, and Y. Le Moullec, "Architectural design space exploration of an fpga-based compressed sampling engine: Application to wireless heart-rate monitoring," in *IEEE Nordic Circuits and Systems Conference (NORCAS) 2015*, 2015.

[C9] T. Ahmed and Y. Le Moullec, "Power-efficient frequency allocation algorithms for self-organized networks," in *RTUWO 2015 - Advances in Wireless and Optical Communications*, 2015.

[C10] Y. M. Mughal, Y. Le Moullec, P. Annus, and K. A., "A bio-impedance signal simulator (biss) for research and training purposes," in *The 26th Irish Signals and Systems Conference 2015 (ISSC2015)*, 2015.

[C11] A. Gupta, O. Märtens, Y. Le Moullec, and T. Saar, "A tool for lung nodules analysis based on segmentation and morphological operation," in *IEEE International Symposium on Intelligent Signal Processing 2015 (WISP2015)*, 2015.

[C12] F. Ahmed, P. Annus, and Y. Le Moullec, "Energy harvesting technologies: Potential application to wearable health-monitoring," in *10th International Conference in Bioelectromagnetism; 01/2015*, 2015.

[C13] T. Ahmed and Y. Le Moullec, "Power optimization in non-coordinated secondary infrastructure in a heterogeneous cognitive radio network," in *19th International Conference ELECTRONICS 2015, Elektronika ir Elektrotechnika*, 2015.

[C14] Y. M. Mughal, Y. Le Moullec, P. Annus, and M. Min, "Development of a bio-impedance signal simulator on the basis of the regression based model of the cardiac and respiratory impedance signals," in *16th Nordic-Baltic Conference on Biomedical Engineering & Medical Physics and the 10th MedTech Days 2014 (IFMBE Proceedings)*, vol. 48, 2015, pp. 92–95.

[C15] A. Gupta, O. Märtens, and Y. Le Moullec, "A preliminary computer-aided technique for ct based lung segmentation," in *Annual Conference of the Estonian National Doctoral School in Information and Communication Technologies, Tallinn*, 2014.

[C16] B. Knudsen, M. Jensen, A. Birklykke, P. Koch, J. Christiansen, K. Laursen, L. Alminde, and Y. Le Moullec, "Ads-b in space: Decoder implementation and first results from the gatoss mission," in *16th biennial Baltic Conference on Electronics 2014*, 2014.

[C17] A. Gupta, O. Märtens, and Y. Le Moullec, "A survey on open-access reference image databases for lung cancer," in *Annual Conference of the Estonian National Doctoral School in Information and Communication Technologies, Tallinn*, 2014.

[C18] Y. Le Moullec, Y. Lecat, P. Annus, R. Land, A. Kuusik, M. Reidla, T. Hollstein, U. Reinsalu, K. Tammemäe, and P. Ruberg, "A modular 6lowpan-based wireless sensor body area network for health-monitoring applications," in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference 2014 (APSIPA ASC 2014)*, 2014.

[C19] A. Cattoni, Y. Le Moullec, and C. Sacchi, "Zero-forcing pre-coding for mimo wimax transceivers: Performance analysis and implementation issues," in *Aerospace Conference, 2013 IEEE*, 2013, pp. 1–7.

[C20] A. Birklykke, P. Koch, R. Prasad, L. Alminde, and Y. Le Moullec, "Empirical verification of fault models for fpgas operating in the subcritical voltage region," in *International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS), 2013 IEEE*, 2013.

[C21] A. Birklykke, Y. Le Moullec, L. Alminde, and R. Prasad, "An automated test framework for experimenting with stochastic behavior in reconfigurable logic," in *Reconfigurable Computing and FPGAs (ReConFig), 2012 International Conference on*, 2012, pp. 1–6.

[C22] L. Alminde, J. Christiansen, K. Laursen, A. Midtgaard, M. Bisgard, M. Jensen, B. Gosvig, A. Birklykke, P. Koch, and Y. Le Moullec, "Gomx-1: A nano-satellite mission to demonstrate improved situational awareness for air traffic control," in *26th Annual AIAA/USU Conference on Small Satellites*, 2012.

[C23] J. Buthler, M. Buhl, Y. Le Moullec, G. Berardinelli, and A. Cattoni, "Implementation of a fft module on the fpga of usrp2 boards," in *3rd International Workshop of the COST Action IC0902*, 2012.

[C24] C. Sacchi, O. Tonelli, A. Cattoni, and Y. Le Moullec, "Implementation aspects of a flexible frequency spectrum usage algorithm for cognitive ofdm systems," in *Aerospace Conference, 2011 IEEE*, 2011, pp. 1–9.

[C25] Y. Le Moullec, "A first step towards high-level cost models for the implementation of sdrs on multiprocessing reconfigurable systems," in *Wireless Personal Multimedia Communications (WPMC), 2011 14th International Symposium on*, 2011, pp. 1–5.

[C26] A. Birklykke, Y. Le Moullec, R. Prasad, and L. Alminde, "Statistical re-acquisition method for gps receivers on satellites in low earth orbit," in *Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology (Wireless VITAE), 2011 2nd International Conference on*, 2011, pp. 1–5.

[C27] J. Johansen, S. Enevoldsen, V. Pucci, O. Tonelli, A. Cattoni, and L. M. Y., "Analysis of the usrp2 firmware: System architecture overview," in *2nd International Workshop of the COST Action IC0902*, 2011.

[C28] A. Ulbinaite, M. Kucinskiene, and Y. Le Moullec, "Integration of the decoy effect in an agent-based-model simulation of insurance consumer behavior," in *International Conference on Software and Computer Applications*, 2011, pp. 152–157.

[C29] A. Popp, A. Herrholz, K. Gruttner, Y. Le Moullec, P. Koch, and W. Nebel, "Systemc-ams sdf model synthesis for exploration of heterogeneous architectures," in *Design and Diagnostics of Electronic Circuits and Systems (DDECS), 2010 IEEE 13th International Symposium on*, 2010, pp. 133–138.

[C30] A. Popp, Y. Le Moullec, and P. Koch, "Fast feasibility estimation of reconfigurable architectures," in *Industrial Electronics and Applications, 2009. ICIEA 2009. 4th IEEE Conference on*, 2009, pp. 117–122.

[C31] U. Cerasani, Y. Le Moullec, and T. Tong, "A practical fpga-based lut-predistortion technology for switch-mode power amplifier linearization," in *NORCHIP, 2009*, 2009, pp. 1–5.

[C32] A. Corneliussen, E. Poulsen, P. Silpakar, T. steraa, and Y. Le Moullec, "Hardware-accelerated nios-ii implementation of a turbo decoder," in *Computer and Electrical Engineering, 2009. ICCEE '09. Second International Conference on*, vol. 1, 2009, pp. 367–371.

[C33] A. Popp, Y. Le Moullec, and P. Koch, "Scheduling temporal partitions in a multi-processing paradigm for reconfigurable architectures," in *Adaptive Hardware and Systems, 2009. AHS 2009. NASA/ESA Conference on*, 2009, pp. 230–235.

[C34] J. Kristensen, P. Simonsen, A. Popp, P. Koch, and Y. Le Moullec, "Ds-cdma descrambling and despreading with the cell broadband engine," in *Signal Acquisition and Processing, 2009. ICSAP 2009. International Conference on*, 2009, pp. 128–133.

[C35] A. Popp, Y. Le Moullec, and B. Olech, "Designing heterogeneous reconfigurable systems: Feasibility analysis, temporal partitioning and multi-processor scheduling," in *Elektronika - Konstrukcje, Technologie, Zastosowania (Electronics -Constructions, Technologies, Applications)*, 2009, pp. 130–133.

[C36] R. Abildgren, J.-P. Diguet, P. Bomel, G. Gogniat, P. Koch, and Y. Le Moullec, "A method for a priori implementation effort estimation for hardware design," in *Electronic Design, 2008. ICED 2008. International Conference on*, 2008, pp. 1–6.

[C37] B. Can, M. Portalski, and Y. Le Moullec, "Hardware aspects of fixed relay station design for ofdm(a) based wireless relay networks," in *Electrical and Computer Engineering, 2008. CCECE 2008. Canadian Conference on*, 2008, pp. 000 355–000 360.

[C38] A. Jensen, N. Jorgensen, K. Laugesen, and Y. Le Moullec, "Non-data aided carrier offset compensation for sdr implementation," in *NORCHIP, 2008.*, 2008, pp. 158–161.

[C39] S. Lal, S. Kaur Warar, A. Popp, and Y. Le Moullec, "Flexible m-qam modulator and scalable fft/ifft: Design and implementation for a sdr multi-carrier transmitter with link adaptation," in *Proceedings of the 5th Karlsruhe Workshop on Software Radios*, 2008, pp. 27–34.

[C40] G. Kulkarni, S. Chitti, A. Popp, and Y. Le Moullec, "Attack - a methodology for realizing partially reconfigurable fpga systems," in *Norchip, 2007*, 2007, pp. 1–5.

[C41] A. Saramentovas, P. Ruzgys, R. Abildgren, and Y. Le Moullec, "Hsdpa design space exploration and implementation guidance with design-trotter," in *Information, Communications Signal Processing, 2007 6th International Conference on*, 2007, pp. 1–5.

[C42] D. Idris, Y. Le Moullec, and P. Eggers, "Design and implementation of self-calibration for digital predistortion of power amplifiers," in *Proceedings of the 9th International Conference on Data Networks, Communications, Computers*, ser. DNCOCO'07, 2007, pp. 345–350.

[C43] R. Abildgren, A. Saramentovas, P. Ruzgys, P. Koch, and Y. Le Moullec, "Algorithm-architecture affinity - parallelism changes the picture," in *Conference on Design and Architectures for Signal and Image Processing (DASIP) 2007*, 2007.

[C44] T. Singh, Y. Le Moullec, D. Uppudi, and P. Kyritsi, "Demonstration of time reversal communications," in *Wireless Personal Multimedia Communications Symposium (WPMC) 2007*, 2007.

[C45] S. Chitti, G. Kulkarni, A. Popp, and Y. Le Moullec, "Flexible reconfigurable implementation of link adaptation," in *Wireless Personal Multimedia Communications Symposium (WPMC) 2007*, 2007.

[C46] A. Rashid, F. Fitzek, O. Olsen, Y. Le Moullec, and M. Gade, "A low complexity, high speed, regular and flexible reed solomon decoder for wireless communication," in *Design and Diagnostics of Electronic Circuits and systems, 2006 IEEE*, 2006, pp. 31–36.

[C47] M. Kristensen, S. Sorensen, Y. Le Moullec, and P. Koch, "Efficient algorithm and system architecture for the suppression of mpeg artifacts," in *Norchip Conference, 2006. 24th*, 2006, pp. 293–296.

[C48] S. Munagala, R. Muchanthula, Y. Le Moullec, P. Koch, and L. Kristensen, "Hsdpa-adaptive modulation and coding: Simulation and implementation," in *Proceedings of the 6th International Conference on Applied Computer Science*, ser. ACS'06, 2006, pp. 242–248.

[C49] Y. Le Moullec, S. Christensen, W. Chenpeng, P. Koch, and S. Bilavarn, "Design space exploration for rapid development of dsp applications," in *Information, Communications and Signal Processing, 2005 Fifth International Conference on*, 2005, pp. 1407–1410.

[C50] A. Veiverys, V. Prasad Goluguri, Y. Le Moullec, C. Rom, O. Olsen, and P. Koch, "A generic hardware-accelerated ofdm system simulator," in *NORCHIP Conference, 2005. 23rd*, 2005, pp. 62–65.

[C51] Y. Le Moullec, S. Christensen, W. Chenpeng, P. Koch, and S. Bilavarn, "Fast system-level design of wireless applications," in *Wireless Personal Multimedia Communications Symposium (WPMC) 2005*, 2005.

[C52] Y. Le Moullec, N. Ben Amor, J.-P. Diguet, and P. Koch, "Adaptive wireless systems optimization based on follow-up modeling," in *Global Signal Processing Expo and Conference (GSPx) 2004*, 2004.

[C53] Y. Le Moullec, C. Leroux, E. Baud, and P. Koch, "Power consumption estimation of the multi-threaded xinc processor," in *Norchip Conference, 2004. Proceedings*, 2004, pp. 210–213.

[C54] Y. Le Moullec, N. Ben Amor, J.-P. Diguet, and P. Koch, "Follow-up modelling for wireless personal communication systems," in *Wireless Personal Multimedia Communications Symposium (WPMC) 2004*, 2004, pp. 255–259.

[C55] Y. Le Moullec, N. Amor, J.-P. Diguet, M. Abid, and J.-L. Philippe, "Multi-granularity metrics for the era of strongly personalized socs," in *Design, Automation and Test in Europe Conference and Exhibition, 2003*, 2003, pp. 674–679.

[C56] Y. Le Moullec, P. Koch, J.-P. Diguet, and J.-L. Philippe, "Design trotter: Building and selecting architectures for embedded multimedia applications," in *IEEE International Symposium on Consumer Electronics (ISCE)*, 2003.

[C57] M. Auguin, K. Ben Chehida, J.-P. Diguet, X. Fornari, A.-M. Fouilliart, C. Gamrat, G. Gogniat, P. Kajfasz, and Y. Le Moullec, "Partitioning and codesign tools and methodology for reconfigurable computing: The epicure philosophy," in *SAMOS Workshop*, 2003.

[C58] Y. Le Moullec, J.-P. Diguet, and J.-L. Philippe, "Design-trotter: A multimedia embedded systems design space exploration tool," in *Multimedia Signal Processing, 2002 IEEE Workshop on*, 2002, pp. 448–451.

[C59] Y. Le Moullec, P. Koch, and J.-P. Diguet, "A power aware system-level design space exploration framework," in *Design & Diagnostics of Electronic Circuits & Systems (DDECS)*, 2002.

[C60] Y. Le Moullec, J.-P. Diguet, and J.-L. Philippe, "Design-trotter: Recombinaisons hiérarchiques dans l'étape d'estimation intra-fonction," in *Journées francophones Adéquation Algorithmes Architectures (JFAAA)*, 2002.

[C61] Y. Le Moullec, J.-P. Diguet, D. Heller, and J.-L. Philippe, "Fast and adaptive data-flow and data-transfer scheduling for large design space exploration," in *Great Lakes Symposium on VLSI (GLSVLSI)*, 2002.

[C62] Y. Le Moullec, J. Diguet, and J.-L. Philippe, "A scheduling framework for system-level estimation," in *Electronics, Circuits and Systems, 2000. ICECS 2000. The 7th IEEE International Conference on*, vol. 1, 2000, 277–280 vol.1.

[C63] S. Bilavarn, J.-P. Diguet, G. Gogniat, Y. Le Moullec, and J.-L. Philippe, "Méthode de conception d'architectures hétérogènes pour les applications de traitement numérique du signal," in *Journées Nationales du Réseau Doctoral en Micro–nanoélectronique (JNRDM)*, 2000.

Rapport technique (1)

[R1] R. Abildgren, J.-P. Diguet, G. Gogniat, P. Koch, and Y. Le Moullec, "Technical report: Real-time aware hardware implementation effort estimation," Department of Electronic Systems, Aalborg University, Tech. Rep., 2010.

Poster avec comité de relecture (1)

[P1] A. Popp, Y. Le Moullec, and P. Koch, *Temporal partitioning and multi-processor scheduling for reconfigurable architectures*, Poster session presented at HiPEAC Advanced Computer Architectures and Compilation for Embedded Systems Summer School, 2008.

Chapitre 6

Résumé de mes activités d'enseignement

J'ai très vite pris goût à l'enseignement, particulièrement pour l'interaction avec les étudiants lors des TP qui constituaient l'essentiel de mes premières interventions. J'ai ensuite découvert à la fois les CM/TD et l'encadrement de projets suivant la pédagogie par problèmes *problem-based learning' (PBL)* telle que pratiquée à Aalborg University. J'ai également, pendant mes emplois de post-doc et d'*assistant professor*, bénéficié de deux formations pédagogiques : "Introduction to Problem Based Learning - the AAU way" (portant essentiellement sur l'encadrement de projets) et "University Course in Pedagogy for Assistant Professors" (incitant à la réflexion sur le métier d'enseignant, le processus d'apprentissage, les méthodes et outils, etc.).

Je suis enthousiaste à l'idée de partager mes connaissances, à interagir avec les étudiants et à participer au développement d'activités pédagogiques (nouveau programmes, internationalisation et échanges étudiants, etc.). J'apprécie également le fait de pouvoir travailler avec des étudiants d'horizons divers et variés (une dizaine de nationalités jusqu'à présent). Enfin, j'essaye aussi de joindre recherche et enseignement, notamment en M1 et M2 où il est possible de transférer une partie des retombées des avancées scientifiques, et surtout en M2 où j'ai pu encadrer des projets au travers desquels les étudiants ont contribué soit aux travaux de recherches de notre groupe soit aux activités de recherches de partenaires industriels.

Les tableaux 6.1 et 6.2 donnent une vue d'ensemble de mes activités d'enseignement, respectivement cours et encadrement, en licence et master). Ces activités sont détaillées dans les deux sections qui suivent.

Période	Programme	Intitulé	Format	Niveau	Établissement
2014/-	CE	Cognitive Communication	CM, TP	M1	TUT
2014/-	CE	C Programming for Embedded Microcontrollers	CM, TP	M1	TUT
2010/2012	ASPI, SDR, SPC	Reconfigurable Computing	CM, TD/TP	M2 puis M1	AAU
2005/2011	ASPI, SDR	Hardware/Software Co-design	CM, TD/TP	M2	AAU
2007/2011	COMSYS	Programmable Digital Platforms	CM, TP	L3	AAU
2008/2011	COMSYS	Microcomputer Hardware	CM, TP	L2	AAU
2005/2007	ASPI	Hardware Platform Analysis	CM/TP	M1	AAU
2004/2005	ASPI	DSP Design Methodology	CM/TD	M1	AAU
1999/2003	EEA	Électronique numérique de base	TP	L2	UBS
1999/2003	EEA	Électronique numérique avancée	TP	L3	UBS
1999/2003	EEA	Systèmes d'exploitation	TP	L2	UBS
1999/2003	EEA	Bases de données	TP	L2	UBS
1999/2003	EEA	Automates Programmables Industriels	TP	L2	UBS

Tableau 6.1: Tableau récapitulatif de mes enseignements. CM : cours magistraux, TD : travaux dirigés, TP : travaux pratiques. CE : Communicative Electronics, ASPI : Applied Signal Processing and Implementation, SDR : Software Defined Radio, SPC : Signal Processing and Computation, COMSYS : Communication Systems, EEA : Électronique, Électrotechnique, Automatique

Période	Programme	Intitulé	Format	Niveau	Établissement
2014/-	CE	4 thèses de Master	Encadrement	M2	TUT
2014/-	CE	6 projets semestriels	Encadrement	M2	TUT
2004/2012	ASPI, SDR	30 thèses de Master	(Co)encadrement	M2	AAU
2004/2012	ASPI	25 projets semestriels	(Co)encadrement	M1/M2	AAU
1999/2003	EEA	6 projets	Encadrement	L3	UBS

Tableau 6.2: Tableau récapitulatif de mes encadrements en licence et master. CE : Communicative Electronics, ASPI : Applied Signal Processing and Implementation, SDR : Software Defined Radio, SPC : Signal Processing and Computation, COMSYS : Communication Systems, EEA : Électronique, Électrotechnique, Automatique

6.1 Résumé des cours donnés après mon doctorat

Les cours sont classés par année de fin décroissante.

6.1.1 Cognitive Communication (2014/-)

L'objectif de ce cours est de donner aux étudiants les notions fondamentales sous-jacentes l'implantation de systèmes de télécommunication cognitifs. Le cours prend comme point de départ une plateforme de radio logicielle et se concentre sur l'implantation des blocs de traitement numérique en bande de base. Le cours est une combinaison de cours magistraux et de travaux pratiques (carte Nuand BladeRF x115, langage VHDL, outil Altera Quartus II).
Le cours est organisé de manière à donner une vue d'ensemble de tels systèmes, d'introduire la plateforme de radio logicielle, de présenter les blocs de traitement numérique en bande de base les plus courants, de présenter l'architecture de divers éléments de traitement de signal (GPP, DSP, FPGA, etc), de présenter les méthodes, techniques et outils pour passer des algorithmes aux architectures (compilation, synthèse). Tout ceci est appliqué en travaux pratiques (tâches, problèmes, et mini-projet) afin que les étudiants puissent acquérir et développer leur expérience pratique.
J'ai développé ce cours par moi-même. Il est actuellement ouvert aux étudiants de M1 (notamment pour ceux inscrits au programme 'Communicative Electronics' à TUT) et passera prochainement en M2. Volume horaire : 24 heures de cours et 48 heures de travaux pratiques réparties sur 16 semaines.

6.1.2 C Programming for Embedded Microcontrollers (2014/-)

Ce cours est destiné aux étudiants de master qui n'ont pas ou très peu d'expérience en programmation C pour les microcontrôleurs. L'objectif du cours est donc de fournir aux étudiants les connaissances, méthodes et techniques essentielles à ce type de programmation. Le cours combine cours magistraux et travaux pratiques (carte Texas Instrument MSP-EXP430G2, langage C, outil Texas Instrument Code Composer Studio).
Ce cours fournit aux étudiants les principes de fonctionnement des microprocesseurs et l'architecture des microcontrôleurs, une vue d'ensemble du langage C et du processus de développement. De plus, le cours présente de manière plus détaillée l'architecture du microcontrôleur sélectionné et les techniques pour le programmer et l'interfacer (p. ex. communication série). Tout ceci est exploité lors de travaux pratiques (tâches, problèmes, et mini-projet) afin que les étudiants puissent acquérir et développer leur expérience pratique, en complément au cours ci-dessus.
J'ai développé ce cours par moi-même. Il est actuellement ouvert aux étudiants de M1 (notamment pour ceux inscrits au programme 'Communicative Electronics' à TUT). Bien que pouvant être vu comme un cours de rattrapage/mise à niveau, il attire (un peu surprenamment) plus d'étudiants qu'il n'y a de places disponibles. Volume horaire : 24 heures de cours et 24 heures de travaux pratiques réparties sur 16 semaines.

6.1.3 Reconfigurable Computing (2010/2012)

L'objectif de ce cours était de fournir aux étudiants les connaissances et méthodes nécessaires à l'implantation optimisée d'algorithmes de traitement du signal sur des plateformes

logicielles/matérielles reconfigurables. Le cours combinait cours magistraux et travaux pratiques (carte Xilinx ML506, langage VHDL, outils Xilinx ISE et Planahead)
Le cours couvrait les méthodes analytiques par simulation pour l'évaluation des fonctions de coût associées à de tels systèmes, notamment les contraintes de performances (p. ex. temps d'exécution) et d'utilisation des ressources logicielles et/ou matérielles (p. ex. surface, empreinte mémoire, consommation énergétique), les aspects théoriques et pratiques des systèmes et architectures reconfigurables (p. ex. partitionnement spatial et temporel, reconfiguration statique/dynamique, complète/partielle sur FPGA). Des travaux pratiques permettaient d'appliquer et de développer ces connaissances et méthodes.
J'avais co-développé ce cours avec le doctorant Andreas Popp pour la partie théorique et l'ingénieur de recherche Pierre Bomel (Lab-STICC/UBS) pour la partie pratique. Il était ouvert aux étudiants de M1 (notamment pour ceux inscrits au programme 'Signal Processing and Computing' à AAU). Volume horaire : 10 heures de cours et 10 heures de travaux pratiques réparties sur 5 semaines .

6.1.4 Hardware/Software Co-design (2005/2011)

Ce cours avait pour objectif de former les étudiants à la conception conjointe logicielle/matérielle. Il prenait la forme de cours magistraux, travaux dirigés (notamment études d'articles scientifiques) et travaux pratiques (carte Altera DE2, outils Altera Quartus).
Les thèmes abordés comprenaient l'exploration de l'espace des solutions, les modèles de calcul, modélisation et estimation pour la décision de conception à plusieurs niveaux d'abstraction, sélection d'architectures hétérogènes, synthèse des calculs et des communications, principes d'optimisation et méthodes de prototypage rapide.
J'avais développé ce cours par moi-même avec les conseils de Messieurs Ole Olsen et Peter Koch (AAU) ainsi que ceux de Jean-Philippe Diguet (UBS). Il était ouvert aux étudiants de M1 (notamment pour ceux inscrits aux programmes 'Applied Signal Processing and Communication' et 'Software Defined Radio' à AAU). Volume horaire : 20 heures de cours et 20 heures de travaux pratiques réparties sur 10 semaines.

6.1.5 Programmable Digital Platforms (2007/2011)

Je m'occupais de la partie FPGA de ce cours (les deux autres étant DSP et test logiciel). L'objectif de cette partie était de fournir les bases de la conception et programmation sur FPGA ; cette partie était constituée de cours magistraux et travaux pratiques (carte Digilent Spartan-3, langage VHDL, outil Xilinx ISE).
Cette partie du cours couvrait les aspects architecturaux des FPGA, les étapes du flot de conception (spécification, synthèse, placement/routage, simulation, etc.) et une introduction au langage VHDL. Plusieurs tâches et petits problèmes permettaient de mettre ces connaissances en pratique.
J'avais développé cette partie du cours par moi-même avec les conseils de Lilian bossuet (UBS). Cette partie du cours était ouverte aux étudiants de L3 (notamment pour ceux inscrits au programme 'Communication Systems' à AAU). Volume horaire pour cette partie du cours : 10 heures de cours et 10 heures de travaux pratiques réparties sur 5 semaines.

6.1.6 Microcomputer Hardware (2008/2011)

Ce cours était fait en liaison avec le cours Microcomputer Software (donnée par Messieurs Flemming Christensen et Sofus Nielsen). Il s'agissait de fournir aux étudiants les notions essentielles à la conception et l'implantation d'un système à microprocesseur, du point de vue matériel. Le cours consistait de cours magistraux et travaux pratiques (Motorola 68000, pas de carte dédiée mais wrapping).
Le cours couvrait des éléments tels que architecture minimale, power-on-reset, cartographie mémoire, adressage, périphériques, communications série et parallèle, etc.
J'étais parti d'un support de cours existant que j'avais ensuite re-développé. Il était ouvert aux étudiants de L2 (notamment pour ceux inscrits au programme 'Communication Systems' à AAU). Volume horaire pour cette partie du cours : 20 heures de cours et 20 heures de travaux pratiques réparties sur 10 semaines.

6.1.7 Hardware Platform Analysis (2005/2007)

Cours relativement semblable à Programmable Digital Platforms (voir plus haut), mais avec langage langage Handel-C et cartes Celoxica RC203 (outil Celoxica DK Design Suite) et Altera DE2 (processeur NIOS2, outils Altera Quartus II et NIOS2 EDS).
J'avais développé ce cours par moi-même. Il était ouvert aux étudiants de M1 (notamment pour ceux inscrits aux programmes 'Applied Signal Processing and Communication' et 'Software Defined Radio' à AAU). Volume horaire : 10 heures de cours et 10 heures de travaux pratiques réparties sur 5 semaines.

6.1.8 DSP Design Methodology (2004/2005)

Ce cours visait à présenter aux étudiants les méthodes et techniques nécessaires au passage des algorithmes de traitement du signal sur architecture matérielle. Le cours consistait de cours magistraux et travaux dirigés.
Le contenu du cours couvrait les modèles tels que Y-Chart, Rugby Meta-Model, graphes de flot de données et FSM/D ainsi que des méthodes d'ordonnancement, partage de ressource, etc. Les travaux dirigés permettaient d'appliquer et vérifier ces notions sur des exemples relativement simples ('papier et crayon').
J'étais parti d'un support de cours existant que je n'avais pas beaucoup retouché par la suite. Il était ouvert aux étudiants de M1 (notamment pour ceux inscrits au programme 'Appplied Signal Processing and Implementation' à AAU). Volume horaire : 10 heures de cours et 10 heures de travaux dirigés réparties sur 5 semaines.

6.2 Encadrement de thèses de master et de projets semestriels après mon doctorat

Abréviations :
EU : encadrant unique
EP : encadrant principal (au moins 50%)
ES : encadrant secondaire
PC : projet court (1 semestre)
PL : projet long (2 semestres)
L2 : deuxième année de licence

M1 : première année de master
M2 : seconde année de master
Seuls les sujets de projets en licence sont récurrents et éventuellement choisis par plusieurs groupes un même semestre ; ceux de master sont définis en fonction des projets de recherche des encadrants ou proposés par des entreprises.

6.2.1 Encadrement ou co-encadrement au niveau master depuis 2004 (65 étudiants ou groupes)

Encadrement ou co-encadrement de thèses de Master depuis 2004

Total : 34 étudiants ou groupes (4 à TUT, 30 à AAU), 1 (PC) ou 2 (PL) semestres M2. Quelques exemples de ces thèses de Master sont disponibles ici : https://goo.gl/4Gbgfq.

34. N. A. Jalali, "MPLS-VPN Impact on VoIP-QoS", 2016, thèse de master, coopération Tallinn University of Technology/Tallinn University/Kabul University, ES, PL, M2. Situation actuelle du diplômé : chargé de cours, Kabul University, Afghanistan.

33. A. A. O. Jademi, "Modeling of Carrier Sense Multiple Access with Collision Avoidance in the Context of Wireless Body Area Networks", 2016, thèse de master, Tallinn University of Technology, EP, PC, M2. Situation actuelle du diplômé : directeur, Dynamism Global Enterprises, Lituanie.

32. I. Kesküla, "Electrical Energy Metering Solution for a Three Phase Load in a Domestic Climate Control System", 2015, thèse de master, Tallinn University of Technology, EU, PC, M2. En coopération avec Liewenthal Electronics pour leur client Innovative Solutions OÜ, Estonie. Situation actuelle du diplômé : responsable ingénierie, Genosity et doctorant, T.J. Seebeck Department of Electronics, Tallinn University of Technology, Estonie.

31. S. Leima, "IJTAG Controlled Test for Parallel FPGA Board Communication Links", 2014-2015, thèse de master, Tallinn University of Technology, EU, PL, M2. En coopération avec Ericsson Estonia. Situation actuelle du diplômé : Test Developer, Ericsson Estonia, Estonie.

30. Y. Li, "Low Cost 3D Scanner, Part 2", 2012, thèse de master, Aalborg University, ES, PC, M2. Situation actuelle du diplômé : doctorant, School of EECS, Queen Mary University of London, Royaume-Uni.

29. E. Dhiver, "Speed Optimization of a 3D Printer by Means of a FPGA", 2012, thèse de master, Aalborg University, EP, PC, M2. En coopération avec CreateIT Real ApS, Danemark. Situation actuelle du diplômé : ingénieur conception et développement logiciels bancs de test, MBDA, France.

28. M. Buhl and J. L. Buthler, "Implementation Of A LTE Inspired Transceiver On A USRP Platform", 2011-2012, thèse de master, Aalborg University, ES, PL, M2. Situation actuelle des diplômés : M. Buhl : Application Software Developer at Doms ApS, Danemark. J. L. Buthler : doctorant industriel, Intel Mobile Communications, Danemark.

27. B. G. Knudsen, M. Jensen, "Implementation and Optimization of an Aircraft Transponder Tracking Payload on FPGA for a Miniaturized Satellite", 2011-2012, Aalborg University, thèse de master, EP, PL, M2. En coopération avec Gomspace ApS. Situation actuelle des diplômés : B. G. Knudsen : Brüel & Kjær, Danemark ; M. Jensen : Embedded Developer, GomSpace, Denmark.

26. J. Johansen, "Automatic Bit-Width Minimization Framework based on Information Theoretic Degradation Measures", 2012, thèse de master, ES, PC, M2. Situation actuelle du diplômé : inconnue.

25. H. G. Grøn, "Acquisition for GPS Receivers: An Investigation of Parallelism for FPGA Implementation", 2010, Aalborg University, thèse de master, EU, PC, M2. Situation actuelle du diplômé : chargé de cours, University College Nordjylland, Danemark.

24. E. B. Poulsen, "Implementation and Evaluation of an Ultra-low Delay Audio Codec on an Embedded Platform - Constrained Energy Lapped Transform", 2009-2010, Aalborg University, thèse de master, EU, PL, M2. En coopération avec RTX A/S, Danemark. Situation actuelle du diplômé : System Architect, Dynaudio A/S, Danemark.

23. M. Pastorelli, "Capacity Analysis and FPGA Implementation feasibility Preanalysis for Multi User MIMO Precoding", 2010, Aalborg University, thèse de master, ES, PC, M2. Situation actuelle du diplômé : VoIP Engineer, Trentino Network, Italie.

22. S. Sepman, 'Design of a Cost-effective and Expandable Assistive Domotics Wireless Transceiver Unit', 2010, Aalborg University, thèse de master, EU, PC, M2. En coopération avec TKS A/S, Danemark. Situation actuelle du diplômé : Ingénieur développeur, Proekspert, Estonie.

21. B. Paul, S. Marcombes, "HAMAC: Home Assistive and Mobile Access Controller, part 2", 2010, Aalborg University, thèse de master, EP, PC, M2. En coopération avec TKS A/S, Danemark. Situation actuelle des diplômés : B. Paul : ingénieur développeur, Coppernic, France ; S. Marcombes : directeur général fondateur, Lima, France.

20. A. Birklykke, "High Dynamic GPS Signal Acquisition: A Case Study in GPS Receivers on Nano-Satellites in LEO", 2009-2010, Aalborg University, thèse de master, EP, PL, M2. Situation actuelle du diplômé : ingénieur applications FPGA, Rohde & Schwarz Technology Center A/S, Danemark.

19. P. Silpakar, "An Investigation of Power Consumption of Computational Efficient DFT Algorithm: a Comparison Between Non Pruned and Pruned Version of Split-radix FFT", 2010, Aalborg University, thèse de master, ES, PC, M2. Situation actuelle du diplômé : ingénieur développeur, Sipradi Trading Pvt Ltd, Nepal.

18. K. D. Hansen, K. L. Jakobsen, "Feed-Forward Quadrature Phase Shift Keying Frequency Offset Correction: The Development of a Hardware-Implementable Phase Error Estimator Algorithm for Use in Third Generation Mobile Telephony Systems", 2009-2010, Aalborg University, thèse de master, EU, PL, M2. En coopération avec Rohde & Schwarz Technology Center A/S, Danemark. Situation actuelle des

diplômés : K. D. Hansen : post-doc, Aalborg University, Denmark ; K. L. Jakobsen : ingénieur développeur, Rohde & Schwarz Technology Center A/S, Danemark.

17. M. Filippi, "SDR Implementation of a OFDM-MIMO Receiver, part 2", 2009, Aalborg University, thèse de master, ES, PC, M2. Situation actuelle du diplômé : responsable recherche et développement, Photonics Power Srl, Italie.

16. O. Tonelli, "Analysis of Selected Implementation Issues of a Dynamic Spectrum Allocation Algorithm for OFDM Systems, 2009, Aalborg University, thèse de master, ES, PC, M2. Situation actuelle du diplômé : ingénieur développeur, Cobham Satcom, Danemark.

15. J. Leresteux, U. Cerasani, "Design & Implementation of a Control Unit for a Tongue Control System", 2009, Aalborg University, thèse de master, EP, PC, M2. En coopération avec TKS A/S, Danemark. Situation actuelle des diplômés : J. Leresteux : ingénieur développeur, Freelance Computer Services, Russie ; U. Cerasani : inconnue.

14. M. Daniel, "WPA Password Cracking: Parallel Processing on the Cell-BE", 2008, Aalborg University, thèse de master, EU, PC, M2. Situation actuelle du diplômé : attaché technico-commercial, E-Tronics s.r.o, République Tchèque.

13. M. Portalski, "Hardware Aspects of Fixed Relay Station Design for OFDM(A) Based Wireless Relay Networks", 2008, Aalborg University, thèse de master, ES, PC, M2. Situation actuelle du diplômé : ingénieur développeur, Service2Media, Pays-Bas.

12. S. Lal, S. Warar, "SDR Implementation of a Multi-carrier Transmitter with Link Adaptation", 2007, Aalborg University, thèse de master, ES, PC, M2. Situation actuelle des diplômées : S. Lal : inconnue ; S. Warar : ingénieure développeuse, The Weather Network, Canada.

11. A. Saramentovas, P. Ruzgys, "Analysing and Implementing a Reed Solomon Decoder for Forward Error Correction", 2007, Aalborg University, thèse de master, EP, PC, M2. En coopération avec RTX A/S, Danemark. Situation actuelle du diplômé : Saramentovas, A. : développeur infrastructure, Barclays, Lituanie ; Ruzgys, P. : Ingénieur, Telekonta, Lituanie.

10. H. Thakur, D. Uppudi, "Implementation of MISO Time Reversal Transmission", 2007, Aalborg University, thèse de master, ES, PC, M2. Situation actuelle des diplômés : H. Thakur : développeur assurance qualité, GN ReSound, Danemark ; D. Uppudi : inconnue.

9. S.H. Chitti, G. Kulkarni, "Flexible Reconfigurable Implementation of Link Adaptation", 2007, Aalborg University, thèse de master, EP, PC, M2. Situation actuelle du diplômé : Chitti, S.H : Ingénieur sénior, Telenor Norge AS, Norvège ; Kulkarni G. : Ingénieur sénior, Motorola Solutions, Danemark

8. D. Idris, "Digital Compensation of PA Non-linearity for WIMAX Transmitter", 2007, Aalborg University, EP, PC, M2. En coopération avec Motorola Denmark, Danemark. Situation actuelle de la diplômée : inconnue.

7. M. S. Kristensen and S. B. Sørensen, "Suppression of MPEG Artifacts - Algorithm Development and Constrained Implementation", 2005-2006, Aalborg University, thèse de master, EU, PL, M2. Situation actuelle des diplômés : M. S. Kristensen : ingénieur développeur, Oticon A/S, Danemark ; S. B. Sørensen : ingénieur développeur, Grundfos, Danemark.

6. M. Y. Appiah, "An efficient FPGA implementation of High-Speed JPEG-2000 Encoder and Decoder", 2005-2006, Aalborg University, thèse de master, EU, PL, M2. Situation actuelle du diplômé : Situation actuelle du diplômé : consultant senior, Mappiah Consulting, Danemark.

5. S. Munagala and R. Muchanthula, "HSPDA-Adaptive Modulation and Coding Acceleration on FPGA", 2005, Aalborg University, thèse de master, EP, PC, M2. En coopération avec Rohde & Schwarz Technology Center A/S, Danemark. Situation actuelle des diplômés : S. Munagala : concepteur systèmes, Motorola Solutions, Danemark ; R. Muchanthula : ingénieur, Intel Mobile Communications, Danemark.

4. B. Maiz, "OFDM Implementation on MicroBlaze", 2005, Aalborg University, thèse de master, EU, PC, M2. Situation actuelle du diplômé : responsable d'unité, Ansaldo STS, France.

3. A. Veiverys and V. P. Goluguri, "Hardware/Software Co-Design of a Multipath Jakes" Channel Simulator for an OFDM System", 2005, Aalborg University, thèse de master, EP, PC, M2. Situation actuelle des diplômés : A. Veiverys : ingénieur développeur, Deeper UAB, Lituanie ; V. P. Goluguri : consultant solutions, iCITA, Australie.

2. A. Rashid, "Implementation of Reed Solomon Error Decoder for DVB-H Terminals", 2005, Aalborg University, thèse de master, EU, PC, M2. Situation actuelle du diplômé : consultant technologies de l'information, NxP Software, Belgique.

1. W. Chengpen and P. Cao, "Evaluation of Digital Implementation Methodologies: A Rake Receiver Case Study", 2005, Aalborg University, thèse de master, EP, PC, M2. Situation actuelle des diplômés : W. Chengpen : directeur des ventes, German Bio Energy Technology, Chine ; P. Cao : inconnue.

Encadrement ou co-encadrement de projets semestrialisés depuis 2004

Total : 31 étudiants ou groupes (5 à TUT, 25 à AAU), 1 semestres M1 ou M2.

31. G. Venin, "Déploiement, test et évaluation de performance d'un système modulaire 6LoWPAN Wireless Body Area Sensor Network appliqué à la supervision médicale", 2015, Tallinn University of Technology, EP, PC. En coopération avec CEBE, Estonie.

30. M. El-Sayed, S. Lund, "An FPGA-friendly Compressed Sampling Engine for WSN-based Heart Rate Monitoring", 2015, coopération Tallinn University of Technology/Aalborg University, EP, PC.

29. A. Basnet, "6LowPAN-based Wireless Sensor Network", 2014, Tallinn University of Technology, EU, PC.

28. C.D. Okeke, "Simulations tools for 6LowPAN-based Wireless Sensor Network", 2014, Tallinn University of Technology, EU, PC.

27. A. Jademi, S.M.J. Hassan, A. Samir, A.K. Rimal, "6LowPAN-based Wireless Sensor Network", 2014, Tallinn University of Technology, EU, PC.

26. Y. Lecat, "PTP synchronization in 6LowPAN-based Wireless Sensor Networks", 2014, Tallinn University of Technology, EU, PC. En coopération avec CEBE, Estonie.

25. H. J. Pedersen, T. F. Pedersen, E. Ricciardi, M. K. Rævdal, "Image Coding using Wavelets on an FPGA Platform", 2012, Aalborg University, SS, SP.

24. E. Dhiver, V. Fouillard, Romain Herry, Y. Li, "Low-cost 3D Scanner, Part 1", 2011, Aalborg University, EP, PC. En coopération avec CreateIT Real ApS, Danemark.

23. J. Johansen, "FPGA-oberon - Design and Implementation of a System Description Language based on the Oberon Language", 2011, Aalborg University, EU, PC, M2

22. F. Hyfing, C. Le Bars, C. Terasa, "Analysis and Derivation of a Synchronization Algorithm for ADS-B Signal Reception - A Recursive Maximum-Likelihood Approach", 2011, Aalborg University, EP, PC, M2. En coopération avec Gomspace ApS, Danemark.

21. J. Johansen, S. Enevoldsen, V. Pucci, "Analysis and Architectural Mapping of an FFT Algorithm into an Already Existing FPGA Firmware of a Low-cost COTS SDR Peripheral", 2011, Aalborg University, EP, PC, M1.

20. J. Buthler, T. Jessen, M. Buhl, R. Simonsen, "Turbo codes and OFDM Implementation for LTE Mobile Systems", 2011, Aalborg University, ES, PC, M1.

19. F. Marot, T. Beyou, J. Morand, "Cross-Talk Cancellation Implementation on the Cell-BE processor", 2010, Aalborg University, EP, PC, M2. En coopération avec Rohde & Schwarz Technology Center A/S, Danemark.

18. B. Paul, S. Marcombes, "HAMAC: Home Assistive and Mobile Access Controller, part 1", 2009, Aalborg University, EP, PC, M2. En coopération avec TKS A/S, Danemark.

17. H. G. Grøn, "Acquisition for GPS receivers: an Investigation for FPGA Implementation", 2009, Aalborg University, EP, PC, M2.

16. P. Silpakar, B. Sala, "Implementation of Costas Loop Algorithm for Compensating Carrier Frequency Offset on FPGA", 2009, Aalborg University, EU, PC, M2.

15. J. Leresteux, J-M. Lory, O. Le Jacques, "FFT Parallelization for OFDM Systems", 2009, Aalborg University, EU, PC, M2.

14. U. Cerasani, "Class F Power Amplifier Linearization Using LUT Based Predistortion", 2009, Aalborg University, ES, PC, M2.

13. E. B. Poulsen, P. Silpakar, T. T. Østeraa, A. Corneliussen, "Evaluation of FPGA Based Turbo Coding Implementations", 2008, Aalborg University, EU, PC, M1.

12. A. R. Jensen, N. T. K. Jørgensen, K. Laugesen, "Non-Data Aided Carrier Offset Compensation for SDR Implementation", 2008, Aalborg University, EP, PC, M2.

11. N. Cothereau, G. Delaite, E. Gourdin, "Intelligent IP Camera - An FPGA Motion Detection Implementation", 2008, Aalborg University, EU, PC, M2.

10. S. R. Søndergaard, M. B. Sørensen, M. Daniel, C. Borlot, J. Kjeldsen, "Surveillance Camera - MPEG2 Implementation on FPGA", 2008, Aalborg University, EU, PC, M1.

9. J. T. Kristensen, P. A. Simonsen, "DS-CDMA Procedures with the Cell-BE Processor", 2008, Aalborg University, EP, PC, M2. En coopération avec Rohde & Schwarz Technology Center A/S, Danemark.

8. F. Lozach, A. Quemere, E. Aulnette, A. Ballouard, "Implementation of Image Processing Algorithms on FPGA: Designing a Digital Photo Frame", 2008, Aalborg University, EU, PC, M2.

7. S. H. Chitti, S. Lal, G. Kulkami and S. Warar, "SDR Implementation of Link Adaptation Algorithm for OFDM based Mobile Broadband Wireless Access (Wimax)", 2007, Aalborg University, ES, PC, M2.

6. D. Uppudi, D. Idris and H. S. Thakur, "Implementation of Time Reversal Transmission", 2007, Aalborg University, ES, PC, M2.

5. A. Saramentovas and P. Ruzgys, "Guidance for the Implementation of HSDPA-AMC by Means of Design Space Exploration with Design-trotter", 2007, Aalborg University, EP, PC, M2.

4. M. Portalski and H. S. D. Lebreton, "Implementation Aspects of Cooperative Diversity Schemes for Wireless Communications", 2007, Aalborg University, ES, PC, M2.

3. M. M. Alam and M. U. R. Awan, "Image Tracking Through Kalman Filter and its Implementation on FPGA", 2006, Aalborg University, EU, PC, M2.

2. C. Leroux and E. Baud, "Power Consumption Estimation of a Multithreaded Processor", 2004, Aalborg University, EU, PC, M1.

1. W. Chengpen and P. Cao, "Design Space Exploration for TD-SCDMA Receiver", 2004, Aalborg University, ES, PC, M2.

6.2.2 Encadrement au niveau licence depuis 2004

Total : 2 groupes à AAU.

2. A. S. Pedersen, J. Filsø, K. H. Hansen, M. B. Madsen, M. S. Madsen, M. E. Cook, 'SMS-styret Sommerhus', 2009, EU, PC, L2

1. M. Hostrup, S. Thestrup, J. S. Damgaard, S. S. Villerup, S. J. Jensen, 'Sommerhusstyring', 2009, EU, PC, L2

Partie II

Sélection de travaux

Chapitre 7

Projet "Methods for Accelerated Design to FPGA Technology"

7.1 Contexte

Le projet "Methods for Accelerated Design to FPGA Technology" est né d'une coopération entre Center for Embedded Software Systems (CISS) (puis Center for Software Defined Radio (CSDR)) à Aalborg University et l'entreprise danoise ETI A/S (rachetée par le britannique BAE Systems Applied Intelligence en 2010). Le projet était co-financé par Danish Ministry of Higher Education and Science d'une part et ETI A/S d'autre part.

À l'époque (2005), ETI A/S disposait d'un groupe "Accelerated Processing" (je remercie au passage messieurs René Bastrup Knudsen et Karl-Ejner Christensen) qui travaillait sur des solutions matérielle pour l'implantation de systèmes de diagnostiques et d'analyse de données. Leurs produits étaient vendus aux agences gouvernementales (lutte contre le crime) et aux entreprises de télécommunication (trouble-shooting).

Bien que la nature sensible de ces applications et clients ne m'autorisent pas à donner plus de détails dans ce document, les ingénieurs du groupe susnommé étaient confrontés à des défis typiquement liés à l'utilisation de cibles hétérogènes tels que le partitionnement logiciel/matériel, mais aussi d'autres défis liés notamment à l'estimation du temps de conception et d'implantation de tels systèmes hétérogènes.

Ceci constitue le point de départ du projet "Methods for Accelerated Design to FPGA Technology" et du doctorat de Rasmus Abildgren ("Implementation Effort and Parallelism — Metrics for Guiding Hardware/Software Partitioning in Embedded System Design"). Les co-encadrants étaient Peter Koch (AAU) et Jean-Philippe Diguet (Lab-STICC/UBS) qui a accueilli et co-encadré Rasmus à deux reprises. J'en profite pour remercier l'Université Européenne de Bretagne pour avoir co-financé l'un de ces deux séjours de recherche.

7.2 Problématiques

L'une des observations les plus marquantes faite sur la pratique à ETI A/S jusqu'à lors était que l'étape de partitionnement logiciel/matériel était entièrement manuelle et

uniquement basé sur l'expérience des développeurs. Elle ne faisait appel à aucun outil (commercial ou académique) d'exploration de l'espace des solutions, soit parce que ces outils étaient trop onéreux, soit parce qu'ils étaient trop complexes. De ce fait, les concepteurs avaient des difficultés à garder une vue d'ensemble des systèmes qu'ils avaient à concevoir et développer, et tout changement du partitionnement en cours de projet se traduisait par des cycles de développement trop longs.
De plus, les concepteurs et leurs responsables étaient trop souvent dans l'impossibilité d'estimer suffisamment précisément le temps nécessaire à l'implantation de ces systèmes. Comme pour le partitionnement, ils faisaient uniquement appel à leur expérience et n'utilisaient pas d'outil permettant d'automatiser ce processus.

7.2.1 Problématique du partitionnement logiciel/matériel

La première problématique abordée dans ce projet concernait donc cette étape importante du flot de conception. L'objectif était de faciliter et automatiser l'étape de partitionnement logiciel/matériel tout en gardant le flot de conception relativement simple, léger et rapide.
Pour ce faire, nous avons choisi une approche de haut-niveau (parfois appelée niveau système), en continuité avec mes travaux de thèses. Dans cette approche, la spécification des algorithmes (en langage C) est représentée sous forme de graphes de contrôle et de flot de données hiérarchiques (HCDFG) et diverses métriques permettant de caractériser cette spécification sont calculées aux différents niveaux du graphe. Sans entrer dans les détails, ces métriques incluent le degré de parallélisme potentiel, l'orientation mémoire et l'orientation contrôle des différentes fonctions de la spécification. Ces métriques permettent de guider le partitionnement (fonctions parallèles sur FPGA, fonctions orientées contrôle sur GPP, etc.)
Nous avons proposé une extension de nos travaux précédents. Nous proposons une version normalisée de la métrique de parallélisme potentiel développée pendant ma thèse et la combinons avec l'approche dite métrique d'affinité. La métrique résultante permet de mieux caractériser le parallélisme potentiel et ainsi de mieux guider l'étape de partitionnement logiciel/matériel. Cette contribution est présentée dans la section 7.3.

7.2.2 Problématique de l'estimation de l'effort d'implantation matérielle

La deuxième problématique considérée dans ce projet était celle de l'estimation de l'effort (principalement le temps) d'implantation matérielle, en pratique sur cible FPGA. Le but était de fournir une méthode et un outil permettant d'automatiser ce processus, et tout comme pour l'étape de partitionnement, de garder le flot de conception simple, léger et rapide.
Pour ce faire, nous avons identifié trois grandes catégories de facteurs pouvant influencer le temps d'implantation (liste non-exhaustive) :

- Facteurs humains
  - Perte de productivité des concepteurs lorsqu'ils travaillent sur plusieurs projets en parallèle;
  - Gain de productivité des concepteurs lorsqu'ils travaillent sur des tâches connues et/ou des outils connus et exploitent leur expérience, et à contrario perte

de productivité lorsqu'ils doivent développer de nouveaux algorithmes ou architectures et/ou apprendre à maitriser de nouveaux outils;

- Influence de l'atmosphère sociale aux seins des équipes sur l'efficacité positive ou négative de la coopération entre les développeurs.

- Facteurs algorithmiques
  - Nombre et types de contraintes (p. ex. temps réel) qui peuvent augmenter la difficulté, et donc l'effort, d'implantation;
  - Taille du projet, en particulier le nombre de composants/fonctions peut augmenter la difficulté;
  - Nombre de signaux d'entrées/sorties, idem;
  - Complexité, en particulier le nombre de relations entre les différents composants/fonctions augmente la difficulté;
  - La nécessité de développer de nouveaux IP augmente le temps de conception par rapport à la réutilisation d'IP existantes (sauf si problèmes d'intégration).

- Facteurs architecturaux
  - Complexité de l'architecture cible et exploitation de celle-ci (parallélisme, communication, VLIW, superscalaire, reconfiguration, etc.;)
  - Disponibilité d'instructions spécialisées ou d'accélérateurs matériels pouvant simplifier l'implantation;
  - Efficacité des outils de compilation, synthèse, test, etc. pouvant réduire les temps de d'implantation (p. ex. optimisation du temps d'exécution);
  - Temps de compilation et/ou de synthèse pouvant ralentir le temps d'implantation.

Étant donnée la difficulté à mesurer certains de ces facteurs (surtout humains), l'approche développée dans ce projet se concentre sur la complexité de l'application et l'expérience des développeurs. La complexité est évaluée à partir du nombre de chemins indépendants dans l'application (représentation HCDFG) ; l'impact de l'expérience des développeurs est quant à elle estimée à partir d'un modèle obtenu à travers la collecte de données du temps nécessaire à l'implantation matérielle de divers projets par des personnes avec différents niveaux d'expérience. Cette contribution est présentée en section 7.3.

7.3 Résumé des contributions

7.3.1 Contribution au partitionnement logiciel/matériel

Pour mieux comprendre cette première contribution, il est nécessaire de revenir d'une part sur la définition de la métrique d'affinité [1] et d'autre part sur celle de la métrique de parallélisme potentiel γ proposée dans ma thèse doctorat.
La métrique d'affinité consiste de trois valeurs ($A_{GPP}, A_{DSP}, A_{FPGA}$) qui indiquent le degré supposé d'adéquation (affinité) entre un algorithme et les trois classes d'architectures. Sans entrer dans les détails, les trois valeurs sont obtenues à partir de 14 autres métriques qui mesurent la corrélation entre des motifs types dans le code source et des motifs équivalents dans les architectures. Ces motifs sont calculés de manières statique, directement sur

le code source; comme par exemple le ratio entre le nombre total de ligne de code et celles comprenant des opérations de type MAC, des données d'un certain type, la manipulation au niveau bit, etc.
La métrique γ est quant à elle calculée comme le ratio entre le nombre d'opérations (NOP) sur le chemin critique (CP) et la longueur de ce chemin (ceci à tous les niveaux des HCDFG constituant l'application) et exprime le parallélisme potentiel à ces différents niveaux :

$$\gamma = \frac{NOP}{CP}$$

Cependant, γ n'est pas normalisée, ce qui rend les comparaisons peu aisées.
Aussi, nous avons dans un premier temps normalisé celle-ci sous la forme :

$$\gamma' = 1 - \frac{CP}{NOP}$$

Ensuite, nous avons proposé une métrique applicable directement sur le code source (voir la publication C.43 dans le chapitre 5 pour plus de détails) mesurant le ratio entre le nombre de lignes de code sur le chemin critique (S_{CP}) et le nombre total de lignes de code (S_m) telle que :

$$\theta = 1 - \frac{S_{CP}}{S_m}$$

Une étude de cas (également détaillée dans la publication C.43) montre que la nouvelle métrique indique une toute autre affinité entre un décodeur Reed-Solomon et le FPGA (0,806) que la métrique d'affinité originelle (0,205), ce qui se traduit par une latence fortement réduite, c.à.d. 244 μs pour une implantation parallélisée contre 2278 μs pour une implantation naïvement séquentielle (puisque indiqué comme telle par la métrique originelle).

7.3.2 Contribution à l'estimation de l'effort d'implantation matérielle

La mesure et l'estimation de l'effort d'implantation sont des thèmes bien plus explorés dans le domaine logiciel (notamment pour les grands systèmes) que dans celui du matériel. Dans ce projet nous avons donc cherché à adapter certaines métriques et techniques existantes du monde logiciel à celui du matériel (FPGA).
L'idée de départ est d'estimer la taille du projet et d'en déduire l'effort (temps) d'implantation, comme par exemple dans les projets COCOMO 81 [2] et COCOMO II [3], sous la forme générale :

$$Effort = A * taille^b$$

où A et b sont des variables d'ajustement fonctions du type de projet, des ressources humaines allouées au projet et de l'expérience des développeurs.
Pour estimer la taille du projet, nous avons proposé d'utiliser la notion de complexité cyclomatique [4]. La métrique de complexité cyclomatique calcule le nombre de chemins linéairement indépendants dans un algorithme à partir d'une description comportementale de celui-ci. Bien qu'elle n'est pas initialement prévue pour mesurer la taille d'un projet, il a été suggéré [5] qu'il existe une relation linéaire entre la métrique de complexité

cyclomatique et le nombre de lignes de code (LOC). Dans sa version simplifiée, elle est exprimée sous la forme :

$$P(G) = \pi + 1$$

où G est le graphe représentant l'algorithme à analyser et π le nombre de nIJuds de conditions dans G.
Dans nos travaux nous calculons celle-ci de manière hiérarchique sur les parties contrôles du HCDFG représentant l'application, c.à.d. les structures *if, switch, for et while/do-while* et les arrangements séquentiels et parallèles (voir la publication J.12 dans le chapitre 5 pour les détails).
Ensuite, afin de prendre en compte l'expérience des développeurs, nous avons utilisé un modèle de courbe d'apprentissage tel que :

$$\eta_{experience}(Dev) = \frac{1}{\alpha\, log(Experience(Dev) + \beta)}$$

où α et β sont utilisés pour ajuster la courbe et $Experience$ représente le temps (en semaines) d'expérience du développeur Dev avec le langage et l'architecture.
Nous avons ensuite utilisé une approche de type validation croisée pour ajuster le modèle. Pour cela nous avons considéré deux applications (camera intelligente et suite cryptographique) pour lesquels des codes de référence en langage C étaient disponibles et à convertir en VHDL par des développeurs assez peu expérimentés. Une fois les résultats analysés (voir publication J.12 dans le chapitre 5), l'effort (temps) d'implantation est modélisé sous la forme :

$$Effort = A.\eta_{experience}(Dev).P(n_{HCDFG_{Alg}})^b$$

où, pour nos applications, $A = 0.226$ et $b = 1.103$ (en pointillé rouge dans la figure 7.1).

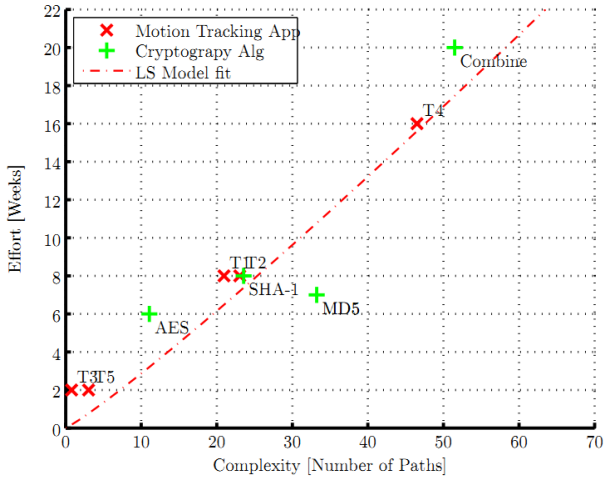


Figure 7.1: Phase d'ajustement. Relation entre l'effort d'implantation (semaines) et la complexité en tenant compte de l'expérience des développeurs. Extrait de la publication J.12.

Pour valider ce modèle nous avons ensuite utilisé des applications (orientées réseaux) fournies par notre partenaire industriel ETI A/S et pris en compte l'expérience de ses développeurs. Les résultats montrent que l'effort (temps) d'implantation estimé pour

les données de validation tombe dans l'intervalle de confiance de 95% (voir figure 7.2) : l'erreur moyenne est de 0,2 semaines et la variance de 8.

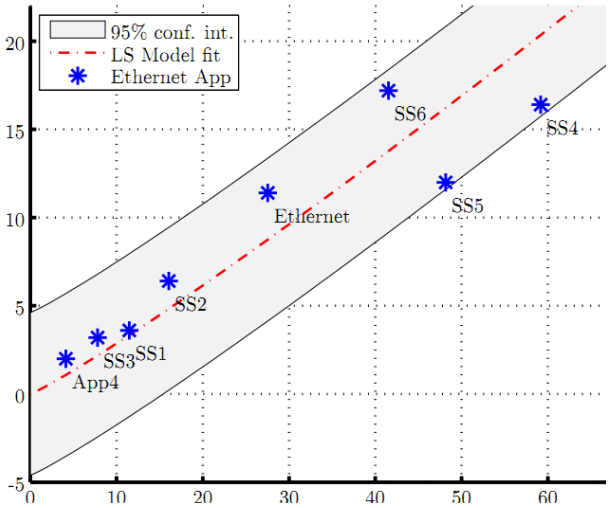


Figure 7.2: Phase de validation. Pointillé rouge : modèle, étoiles bleues : application de validation, zone grisée : intervalle de confidence de 95%. Extrait de la publication J.12.

7.4 Publications et commentaire

Ce projet de recherche a donné lieu à cinq publications (voir J.12, C.43, C.41, C.36 et R.1 dans le chapitre 5), ainsi qu'à la thèse de doctorat de Rasmus Abildgren [6].
C.36 est la version conférence sur laquelle a été construite la version longue (journal) de J.12, R.1 est un rapport technique présentant une extension de l'estimation de l'effort aux systèmes temps-réels et C.41 présente les résultats préliminaires du projet (méthode d'estimation statiques et exploration de l'espace de conception pour un système HSDPA). Les pages qui suivent reproduisent les deux publications qui à mon sens représentent la quintessence du projet et présente en détail les deux contributions présentées précédemment. La première (C.43, *Algorithm-Architecture Affinity — Parallelism Changes the Picture*) présente la métrique pour l'évaluation du parallélisme et affinité vers le matériel (FPGA) ; la seconde (J.12, *A Priori Implementation Effort Estimation for Hardware Design Based on Independent Path Analysis*) détaille les métriques et la méthode pour l'estimation de l'effort (temps) d'implantation matérielle (FPGA).

Enfin, une piste pour donner une éventuelle suite à ces travaux serait d'aller vers une collaboration inter-disciplinaire avec des spécialistes des sciences humaines et sociales (apprentissage et acquisition de compétences au sein des entreprises et influence sur l'expérience des développeurs, comportement des équipes et influence sur le temps de travail nécessaire, etc.).

Authors' version of the paper that appears in the proceedings of DASIP 2007.

Algorithm-Architecture Affinity – Parallelism Changes the Picture

Rasmus Abildgren*, Aleksandras Šaramentovas†, Paulius Ruzgys†, Peter Koch†, and Yannick Le Moullec†

*Center for Embedded Software Systems(CISS)
Aalborg University, Selma Lagerlöfs Vej 300,
DK-9220 Aalborg East, Denmark
rab@es.aau.dk

†Center for Software Define Radio (CSDR)
Aalborg University, Fredriks Bajers Vej 12,
DK-9220 Aalborg East, Denmark
{aleksara,paulius,pk,ylm}@es.aau.dk

***Abstract*—Reducing the time-to-market factor is a challenge for many embedded systems designers. In that respect, hardware-software partitioning is a key issue which has been studied during the last two decades. In this paper we present an extension to recent works dealing with metrics for guiding the hardware-software partitioning step. This extension builds upon and complement our own work with metrics in the Design Trotter project, and is combined with the affinity metric approach. We show that the proposed extension improves the original affinity metric in terms of parallelism detection, and thus can help system designers to make wiser hardware-software partitioning decisions, which in turn reduces the time-to-market factor.**

I. Introduction

In order to achieve more advanced and faster services in embedded systems, increasingly sophisticated algorithms are used. To keep abreast with the increased need for processing power, heterogeneous multiprocessor platforms are introduced, which includes GPPs, DSPs and FPGAs.

Introducing this variety of processing elements (PEs), not only increases the computational capacity of embedded systems but also adds various computational properties. To exploit this increased capacity and properties, the designer needs to find the best suited PEs for the different system functionalities. By considering these facts together with all the system constraints (Area, Time, Power, Price, Development Time), it becomes a non-trivial task to decide how the system functionality should be mapped on the architecture.

To handle this task system level design methodologies have been developed, including structured design space exploration (DSE). A suite of academic DSE frameworks, e.g. [1]–[3], as well as commercial tools have been proposed, in order to provide the design engineer with qualitative information for partitioning.

Exploring the design space with optimising for different constraints is known to be $\mathcal{NP}$ hard [4]. The DSE in these frameworks is therefore carried out as heuristic simulations, which still can be a time-consuming but necessary task for state-of-the-art large scale products. Large companies can usually find these resources and keep up with their competitors.

However, small and medium enterprises (SMEs), which typically sell state-of-the-art products of much smaller volumes, must also stay on the competitive edge. They are also restricted by the time-to-market factor, and can also benefit from using system level design methodologies (SLD) and tools. Unfortunately, many SMEs can not afford tools and specialists like big companies, and therefore have problems with changing their design methodology into SLD methodology.

We have examined the design methodology of a high-tech company in Denmark and found that the design space exploration phase in their overall design trajectory is limited in the sense that their partitioning depends on prior design, designers intuition and experience, and in rare cases on ad hoc analysis. Danish Technological Institute, a consulting company helping many SMEs incorporating new research results, agrees on that picture in most SMEs [5].

As a consequence of sticking to ad hoc design methodologies, SMEs development often run into situations where redesigning part of the system is necessary and therefore increases the time-to-market.

In this paper we propose an extension to the existing affinity metric proposed in [6] for guiding the partitioning of the system specification, and help making the DSE faster and easier. The rest of the paper is organised as follows. In section II, the existing affinity metric is presented and examples for the need of an extension to the original metric are shown. In section III the new proposed metric for parallelism is presented. The benefits of the proposed parallelism metric are illustrated in section IV by means of a Reed-Solomon decoder case-study. Finally we conclude in section V.

II. Affinity Metric

This section summarises the affinity metric proposed by D. Sciuto et.al. in [6], [7], and argues for the need of an extension of this metric. The affinity metric is designed to guide the design partitioning of system specification between general purpose processors, DSP processor, and FPGA/ASIC. The metric consists of a triplet of values $(A_{GPP}, A_{DSP}, A_{FPGA})$ indicating the match between the processing elements and the examined code. The individual values in the metric are calculated based on 14 other metrics which are designed to measure the source code for certain patterns highly correlated with architectural properties. The measurement is a static analysis of the source code and the metrics are defined as ratios between lines with specific properties, e.g., the ratio between lines with a condition and the total number of lines, or

defined as the number of assignment of a special type related to the total number of assignments. The metrics measure properties such as data types, Harvard architecture patterns, MAC patterns, and bit manipulation.

To illustrate how the affinity metric works on a real life example, we have applied it onto c-code (Fig 3) calculating a matrix multiplication. The results of the different metrics are shown in table I:

TABLE I
THE AFFINITY VALUE FOR THE MATRIX MULTIPLICATION ALGORITHM, WHERE A_{xxxx} INDICATES THE MATCH BETWEEN THE PROCESSING ELEMENT TYPE AND THE CODE. 0 =NO MATCHING, 1 =PERFECT MATCH.

A_{GPP}	A_{DSP}	A_{FPGA}
0.89	0.96	0.39

The normalised metric values indicate that the best architecture matching the algortihm is a DSP architecture, which the designer could easily rely on. An in-depth analysis of the code shows that besides the already extracted properties from the affinity metric, a high degree of inherent parallelism is present in the matrix multiplication algorithm. This is further discussed in section III. A high degree of inherent parallelism indicates that the algorithm is suited for parallel execution. This is one property of a FPGA architecture, and the original affinity metric does not consider it.

III. PARALLELISM METRIC

From the analysis of the matrix multiplication shown in Fig 3, we see that the inherent parallelism of an algorithm is an important parameter. Therefore it would be beneficial to measure the degree of inherent parallelism in the algorithm and use this in calculating the A_{FPGA} value of the affinity metric.

One of the first metrics considering the parallelism is Amdahl's speedup metric [8]. Here the potential execution speedup of an algorithm is defined as the ratio between the sequential execution, and the fully parallelised execution. What determines the fully parallelised execution is the critical path in the algorithm.

This is also the case for more recent parallelism metrics e.g. [9], [10], so let us consider the critical path by looking at precedence graphs.

Definition 1: Let $G = (N, E)$ represent the precedence graph of a method, m, where N represents the set of nodes n_i and E is the set of edges $e_{i,j}$. A node n_i can have a source node and a destination node. If the node does not have a source node, it is defined as a start node, and if the node does not have a destination, it is a sink node. If a dependency between two nodes; the parent node, n_i and the child node, n_j, exists, it is connected with an edge $e_{i,j}$. The node, n_j, cannot execute before it has obtained data from its parent(s).

Using definition 1, we can now express the critical path of algorithm using the following definition:

Definition 2: The critical path, CP, is a set of nodes $n_{start}, \ldots, n_i, \ldots, n_{sink}$ and associated edges $e_{start,h}, \ldots, e_{i,j}, \ldots, e_{k,sink}$ forming a path, p, from a start node, n_{start}, to a sink node, n_{sink}, for which the sum of costs are a maximum:

$$CP = \max cost(\{n_{start}, e_{start,h}, n_h, \ldots, n_i, e_{i,j}, n_j, \ldots \\ \ldots, n_k, e_{k,sink}, n_{sink}\}) \quad (1)$$

A way to measure the inherent parallelism that uses the critical path is the γ metric developed in our previous work [9] which is defined as:

$$\gamma = \frac{N}{CP} \quad (2)$$

where we consider the nodes to be atomic, meaning that N represents the total number of operations in the precedence graph.

The metric described in (2) expresses the level of inherent parallelism of the algorithm by calculating the ratio between the number of operations in the algorithm, and the number of operations in the critical path. In this case, where we consider all nodes as basic operations, N is equivalent with the total number of nodes N. This metric is organised such that with no inherent parallelism its gives the value 1. The metric value increases along with the inherent parallelism.

The affinity metric [7] on the other hand is in comparison a normalised measure, where zero indicates the worst match and one indicates a perfect match between the algorithm and the architectural property. Using the γ for expressing the inherent parallelism will lead to non-comparable results. A metric expressing the parallelism together with the affinity metric should have the same normalised properties. To suit these properties we can rewrite the γ metric into a normalised metric:

$$\gamma' = 1 - \frac{CP}{N} \quad (3)$$

The affinity metric is based on textural analysis of the source code and therefore does not refer to the number of operations, critical path or any of the terms used above for γ and γ'. Instead it operates with source lines which contain certain patterns.

In order to cope with the parallelism measure inside this source line based framework, we propose a new metric, θ, inspired by the γ' metric. θ is defined as:

$$\theta = 1 - \frac{S_{CP}}{S_m} \quad (4)$$

where S_{CP} is the number of source lines included in the critical path and S_m is the total number of source lines in the code. To emphasise the weight of the critical path, a loop unrolling is need to be performed before measuring S_m and S_{CP} of the θ metric.

This way of expressing the parallelism is not equivalent with γ' since every source line in a high level language will usually lead to more than one atomic operation. The danger is that the number of atomic operations highly depends on the programmers coding style. A compact code will result in

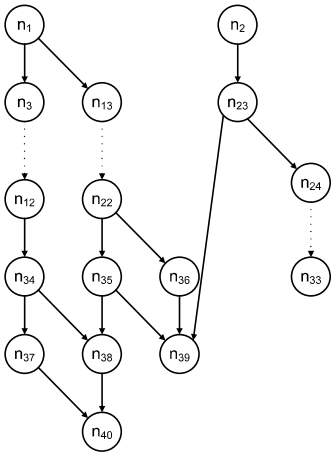


Fig. 1. Precedence graph of random 1 algorithm.

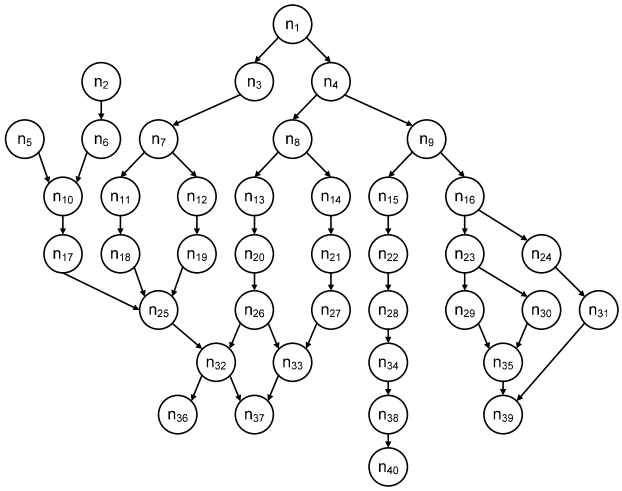


Fig. 2. Precedence graph of random 2 algorithm.

more operations per source line than a fragmented code with many intermediated/temporary variables which come close to one operation per code line. It is therefore impossible to obtain the same precision, as the modified and normalised γ' metric.

To examine their differences, extreme cases, i.e. a purely sequential and a fully parallel execution as well as two random cases have been considered. The two random execution graphs are shown in Fig 1 and Fig 2. Comparing the γ' metric and the θ metric on these cases provides us with the results shown in the four first lines of table II. We here consider $N = 40$ in the precedence graphs, where a source line on average corresponds to four nodes. The sequential execution gives, as expected, the same result for both metrics i.e., 0. The fully parallel execution however, gives a slightly different result for the two metrics, $\gamma' = 0.975$ and $\theta = 0.9$. None of them reach the value 1 for a full parallel execution, because of the way CP is defined. But we notice that θ gives a lower score than the γ' metric. This is due to the smaller number of code lines compared with the number of nodes, which influences the ratio. For the random case there are larger differences (0.65 vs. 0.56) and (0.7 vs. 0.75).

TABLE II
DIFFERENCES BETWEEN THE γ' AND θ METRIC.

	γ'	θ
Sequential:	0	0
Parallel:	0.975	0.9
Random 1	0.65	0.56
Random 2	0.7	0.75
Matrix Multiplication:	0.999	0.989

Even though the θ metric and the γ' metric do not give similar results, θ still gives a good indication of the algorithms affinity to a parallel architecture. Let us discuss this issue by re-considering the matrix multiplication case given by:

$$\mathbf{C} = \mathbf{AB} \tag{5}$$

```
int matrixMul(static int A[X*Y],
              static int B[Y*Z],
              static int C[X*Z])
{
  int *p_a = &A[0] ;
  int *p_b = &B[0] ;
  int *p_c = &C[0] ;

  int f ;
  int i ;
  int k ;

  for (k = 0 ; k < Z ; k++)
    {
      p_a = &A[0] ;  /* point to the beginning of array A */

      for (i = 0 ; i < X; i++)
         {
           p_b = &B[k*Y] ;     /* take next column */
           *p_c =  0 ;

           for (f = 0 ; f < Y; f++) /* do multiply */
              *p_c += *p_a++ * *p_b++ ;

           *p_c++ ;
         }
    }
  return(&C[0]) ;
}
```

Fig. 3. Matrix multiplication example.

where $\mathbf{C} \in \mathbb{R}^{X\times Z}$, $\mathbf{A} \in \mathbb{R}^{X\times Y}$, $\mathbf{B} \in \mathbb{R}^{Y\times Z}$ are matrixes where X, Y, Z denotes the dimensions. Here the dimensions are $X = Y = Z = 10$. The c-code taken from the DSPstone project [11] is shown in Fig 3, and we see that the kernel of the algorithm consists of multiplications, memory reads and writes together with some indexing controls. A precedence graph of the kernel of the algorithm is shown in Fig 4. The results of the examination of the algorithm with the two metrics are also shown in table II. From this we see that there is an insignificant difference between the two metrics (i.e., 0.999 and 0.989), which is due to the high number of nodes and unrolled source lines. From these cases it appears that the newly proposed metric θ serves its purpose of indicating parallelism.

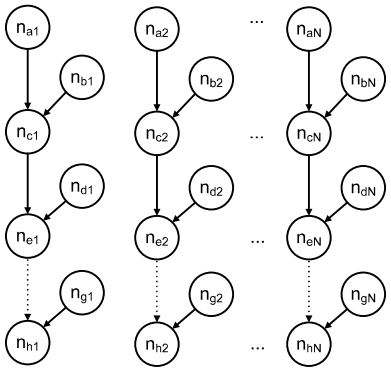


Fig. 4. Precedence graph of the kernel of the matrix multiplication example.

TABLE III
THE ORIGINAL AFFINITY METRIC VALUES FOR GPP, DSP, AND FPGA AND THE PROPOSED METRIC (FPGA&θ) FOR THE REED-SOLOMON DECODER ALGORITHM. THE PERFORMANCE (LATENCY) OF THE DIFFERENT ARCHITECTURES ARE ALSO SHOWN.

	GPP	DSP	$FPGA$	$FPGA\&\theta$
Affinity	0.717	0.795	0.205	0.806
Latency $[\mu s]$	-	514	2278	244

IV. CASE STUDY

In this section we present a case study, which expresses the benefits of the introduced metric, before selecting the architecture for a Reed-Solomon decoder.

A. Reed-Solomon Decoder

Reed-Solomon codes are a forward error correction codes used in many modern communication systems. The decoder is able to detect and correct some bit errors which have occurred doing the transmission. It is an algorithm which involves many conditional branches in order to detect and repair errors.

The algorithm has been examined with the affinity metric, and the results are shown in table III. The table shows the original affinity metric values for GPP, DSP and FPGA architectures and the affinity metric for FPGA with our new extension (added as an extra parameter for FPGA metric before normalisation as in [7]). We see that the Reed-Solomon decoder has the highest score (0.795) on a DSP architecture with the original affinity metric, however, the score for FPGA architecture increases significantly (from 0.205 to 0.806) when including our extension, and thereby gets the highest score. To verify the results, the algorithm has been implemented on a Analog Devices TigerSHARK ADSP-TS201 DSP and a Xilinx Virtex II FPGA, in high-level languages (C and Handel-C, respectively). The latency for decoding one block was measured on both platforms. The FPGA implementation was done in two steps: first, a version without exploiting the parallelism, which corresponds to the original affinity metric interpretation, and second, a version exploiting the inherent parallelism. These latencies are also shown in table III.

Inspecting the results shows that the best performance is obtained by the parallelised FPGA implementation, with a latency of $244\mu s$. We can then deduce that using the original affinity value for FPGA in this case will not disclose the architectures potential for the Reed-Solomon algorithm. Without considering the parallelism, the designer would make an inefficient partitioning choice.

Using the extended metric that we propose gives a better indication of the affinity between algorithm and FPGA architecture, thus helps the designer to make wiser partitioning decisions.

V. CONCLUSION

In this paper we have proposed an extention of the affinity metric [6], in order to improve the capability to measure the algorithm-architecture affinity for FPGA. The extension consists of a new metric derived from some of our previous work [9]. This new metric provides a mean for measuring the inherent parallelism of the algorithm inside the source code. We have shown that adding this new metric to the original affinity metric improves its score for FPGA matching.

REFERENCES

[1] C. Hylands, E. A. Lee, J. Liu, X. Liu, S. Neuendorffer, Y. Xiong, Y. Zhao, and H. Zheng, "Overview of the ptolemy project," Technical memorandum ucb/erl m03/25, Department of Electrical Engineering and Computer Science, University of California, Berkeley, California 94720, July 2003.

[2] C. Erbas, A. D. Pimentel, M. Thompson, and S. Polstra, "A framework for system-level modeling and simulation of embedded systems architectures," *EURASIP Journal on Embedded Systems*, 2007.

[3] J. Riihimäki, P. Kukkala, T. Kangas, M. Hännikäinen, and T. D. Hämäläinen, "Interfacing uml 2.0 for multiprocessor system-on-chip design flow," in *Proceedings of International Symposium on System-on-Chip*, November 2005, pp. 108 – 111.

[4] Z. Á. Mann and A. Orbán, "Optimization problems in system-level synthesis," in *Proceedings of the 3rd Hungarian-Japanese Symposium on Discrete Mathematics and Its Applications*, 2003.

[5] T. S. Olesen, "Private conversation about Danish SMEs design methodologies," August 2006.

[6] C. Brandolese, W. Fornaciari, L. Pomante, F. Salice, and D. Sciuto, "Affinity-driven system design exploration for heterogeneous multiprocessor soc," *Computers, IEEE Transactions on*, vol. 55, no. 5, pp. 508–519, May 2006.

[7] D. Sciuto, F. Salice, L. Pomante, and W. Fornaciari, "Metrics for design space exploration of heterogeneous multiprocessor embedded systems," in *Hardware/Software Codesign, 2002. CODES 2002. Proceedings of the Tenth International Symposium on*, 6-8 May 2002, pp. 55–60.

[8] G. M. Amdahl, "Validity of the single processor approach to achieving large scale computing capabilities," in *AFIPS spring joint computer conference*, 1967.

[9] Y. Le Moullec, N. Ben Amor, J-Ph. Diguet, M. Abid, and J-L. Philippe, "Multi-granularity metrics for the era of strongly personalized SOCs," in *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition*, 2003.

[10] G. C. Sih and E. A. Lee, "A compile-time scheduling heuristic for internocnection-constrained heterogenous processor architectures," *IEEE Transactions on Parallel and Distributed Systems*, vol. 6, no. 4, 1993.

[11] "DSP Stone," 1995, http://www.ert.rwth-aachen.de/Projekte/Tools/DSPSTONE/dspstone.html.

Hindawi Publishing Corporation
EURASIP Journal on Embedded Systems
Volume 2008, Article ID 280347, 12 pages
doi:10.1155/2008/280347

Research Article

A Priori Implementation Effort Estimation for Hardware Design Based on Independent Path Analysis

Rasmus Abildgren,[1] Jean-Philippe Diguet,[2] Pierre Bomel,[2] Guy Gogniat,[2] Peter Koch,[3] and Yannick Le Moullec[3]

[1] *CISS, Aalborg University, Selma Lagerlöfs Vej 300, 9220 Aalborg East, Denmark*
[2] *Lab-STICC (UMR CNRS 3192), Université de Bretagne Sud, Centre de recherche, BP 92116, 56321 Lorient Cedex, France*
[3] *CSDR, Aalborg University, Fredriks Bajers Vej 7, 9220 Aalborg East, Denmark*

Correspondence should be addressed to Rasmus Abildgren, rab@es.aau.dk

Received 15 March 2008; Revised 30 June 2008; Accepted 18 September 2008

Recommended by Markus Rupp

This paper presents a metric-based approach for estimating the hardware implementation effort (in terms of time) for an application in relation to the number of linear-independent paths of its algorithms. We exploit the relation between the number of edges and linear-independent paths in an algorithm and the corresponding implementation effort. We propose an adaptation of the concept of cyclomatic complexity, complemented with a correction function to take designers' learning curve and experience into account. Our experimental results, composed of a training and a validation phase, show that with the proposed approach it is possible to estimate the hardware implementation effort. This approach, part of our light design space exploration concept, is implemented in our framework "Design-Trotter" and offers a new type of tool that can help designers and managers to reduce the time-to-market factor by better estimating the required implementation effort.

1. INTRODUCTION

1.1. Discussion of the problem

Companies developing embedded systems based on high-end technology in areas such as telecommunication, defence, consumer products, healthcare equipment are evolving in an extremely competitive globalised market. In order to preserve their competitiveness, they have to deal with several contradicting objectives: on one hand, they have to face the ever-increasing need for shorter time-to-market; and on the other hand, they have to develop and produce low-cost, high-quality, and innovative products.

This raises major challenges for most companies, especially for small- and medium-sized enterprises (SMEs). Although SMEs are under pressure due to the above-mentioned factors, they are either not applying the latest design methodologies or cannot afford the modern electronic system level (ESL) design tools. By limiting themselves to traditional design methodologies, SMEs make themselves more vulnerable to unforeseen problems in the development process, making the time-to-market factor one of the most critical challenges they have to deal with. A survey released at the Embedded Systems Conference (ESC 2006) [1] indicated that more than 50% of embedded design projects are running behind schedule (i.e., 25% are 1-2 months late, 18% 3–6 months). In the 2008 version of the survey [2], it is again shown that meeting the schedule is the greatest concern for design teams.

Moreover, a workshop [3] held for Danish SMEs working in the domain of embedded systems clearly indicates that there is a need for changing and improving their design trajectories in order to stay in front of the global market. More specifically, this calls for setting modern design, that is, hardware/software (HW/SW) codesign, and ESL design into actual practice in SMEs, so that they can reduce their time-to-market factor and keep up with their competitors by being more efficient in producing embedded systems.

Although HW/SW codesign and ESL design tools (both commercial and academic) have been available for several years, there are several barriers that, so far, have prevented their wide adoption such as the following:

(i) difficulty in transferring the methods and tools developed by academia into industry, because they are mostly developed for experimenting, validating, and proving new concepts rather than for being used in companies; therefore adapting and transferring these methods and tools require additional and tedious efforts, delaying their adoption;

(ii) financial cost in terms of tool licenses, training, and so forth that many SMEs cannot afford, since the cost of a complete commercial tool chain can exceed in excess of 150 k€ per year;

(iii) training cost and knowledge management issues, meaning that switching to a new design trajectory also involves the risk of loosing momentum, that is, loosing time and efficiency because of the training needed to master the new methods and tools;

(iv) many modern design flows are not mature enough to generate efficient and automatic real-time code, and combined with the previous item, cause potential adopters to wait until it is safe to switch.

Considerable research has been undertaken to estimate implementation factors such as area, power, and speed up that are subsequently used in HW/SW partitioning tools with different focuses related to granularity, architecture model, communication topology, and so on. All of these research projects do not include the man-power cost which is the most critical one for many companies, and especially SMEs. This work takes its outset in a research framework facilitating the HW/SW partitioning step for SMEs. It focuses on a light design space exploration approach called "DSE-light" that combines the advances in terms of design methodologies found in academia and the ease of integration required by SMEs, that is, lowering the above-mentioned barriers.

The contribution presented in this paper is the development of a method for estimating the man-power cost (i.e., development time) for implementing hardware components and the integration of this method into our framework, so that HW/SW partitioning decisions can be wiser. A method that used iteratively and systematic will form the engine for precise development schedules. The following subsections present the rationale for this work and the idea enabling this contribution.

1.2. Parameters that influence the implementation effort

A common problem in both SMEs and larger companies is that of estimating the amount of time required to map and implement an algorithm onto an architecture given parameters such as [4, 5] the following :

(i) manpower, that is, the available development team(s) and their size(s),

(ii) quality of the social interactions between the team members and the teams,

(iii) experience of the developers (e.g., years of experience, previously developed projects, novelty of the current project, etc.),

(iv) skills of the developers, that is, their ability to solve problems (this is not the same as experience, which only reflects how often one has tried before),

(v) availability of suitable and efficient tools and how easy they are to learn and use,

(vi) availability of SW/HW IP code/cores,

(vii) involvement of the designers, that is, are they working on other projects simultaneously?

(viii) design constraints, that is, real-time requirements,

This work addresses the issue of adding man-power cost parameter into the cost function and thereby guiding the HW/SW partitioning. More specifically we concentrate on the mapping process, that is, the process of mapping a given algorithm onto a given architecture and the implementation effort (i.e., time) related to the complexity of that algorithm. Our framework also addresses other issues of HW/SW partitioning, for example, [6].

1.3. Idea

In order to understand what makes an algorithm difficult to implement, five semistructured interviews have been conducted with engineers (hardware developers) with very little to 20 years of experience. (Semistructured interview is an information-gathering method of qualitative research. It is also an adequate tool to capture how a person thinks of a particular domain [7].)

From the interviews, it was deduced that several parameters influence on the hardware design difficulty. The hardware developers stated that available knowledge about worst cases, dependencies between variables, and the completeness of the design description of the entire system including all communications are important for the design time. However, according to them, the major parameter influencing a hardware design is the number of connections and signals between the internal components. This should be viewed in the way in which every time a signal enters a component, it means that the component needs to act on it. More signals bring more parameters into the component and that very often leads to an increased complexity.

Based on the interviews, we form our hypothesis, which is that a strong relation exists between what renders an algorithm complex to implement and the number of components as well as the number of signals/paths in the algorithm.

To ensure that not only the number of paths are counted but also that a high number of components is present, we choose to only measure the number of linear-independent paths. Furthermore, this insures that components occuring several times during the execution are counted only once, which better reflects the actual implementation efforts.

The remainder of the paper is organised as follows: Section 2 gives an overview of the state-of-the-art methods for estimating the implementation effort both for software and hardware designs and indicates the need for further work for hardware design. In Section 3 a new metric for estimating the development time is defined and combined with our

research tool "Design-Trotter." Section 4 presents some test cases used to investigate the validity of the above-mentioned hypothesis and of the proposed metric. Furthermore, the experimental results are analysed. Finally we conclude in Section 5.

2. STATE OF THE ART

2.1. Software

Most research about estimating implementation effort is found in the software domain, especially within the COCOMO project [8]. The problem of estimating the implementation effort is twofold. First, a reasonable measure needs to be developed for being able to quantify the algorithm. Second, a model needs to be developed, describing a rational relation between the measure and the implementation effort.

2.1.1. COCOMO

To start with the model, a typical power model has been proposed inside the COCOMO experiment [8, 9]:

$$\text{Effort} = A \times \text{Size}^{b}, \tag{1}$$

where Size is an estimate of the project size, and A and b are adjustable parameters. These parameters are influenced by many external factors which we previously discussed in Section 1.2, but can be trained, based on previous project data.

To use this COCOMO measure, there is a need for expressing the size of the project. Inside the software domain, the dominating metric is lines of code (LOC). Using LOC is not without difficulties, for example, how is a code line defined? Reference [10] discusses this issue and states that LOC is not consistent enough for that use; this is also supported by [11]. Using the LOC metric also has several difficulties, for example, it is not a language independent metric. Furthermore, hardware developers also tend to disapprove this measure, since they do not feel that it is a representative measure for hardware designs.

However, we do not claim that there is no relation between LOC and the implementation effort. It is impossible to write 10 k lines in one day, but for VHDL the relation is not always straightforward. In the experiments that we have performed (data shown in Table 1) there is no unambiguous relation between the LOC in VHDL and the development time.

Reference [11] describes that making "a priori" determination of the size of a software project is difficult especially when using the traditional lines of code measure; instead function points-based estimation seems to be more robust.

2.1.2. Function points analysis

The function points metric was first introduced by Albrecht [12] and consists of two main stages: The first stage is counting and classifying the function types for the software. The identified functions need to be weighted reflecting their complexity, that is determined on the basis of the developers' perception. The second stage is the adjustment of the function points according to the application and environment, based on 14 parameters. The function points can then be converted into an LOC measure, based on an implementation language-dependent factor, and, for example, [11] reports that the function points metric can be used as an implementation effort estimation metric. The function points analysis has been criticised of being too heuristic and [10] has proposed the SPQR/20 function points metric as an alternative. Reference [13] has compared the SPQR/20 and the function points analysis and found their accuracy comparable even though the SPQR/20 metric is simpler to estimate.

2.2. VHDL function points

To the knowledge of the authors, limited research has been carried out in the field of estimating the implementation difficulty of hardware designs.

Fornaciari et al. [14] have taken up the idea from the function points analysis and modified it to fit VHDL. By counting the number of internal I/O signals and components, and classifying these counts into levels, they extract a function point value related to VHDL. They have related their measure to the number of source lines in the LEON-1 processor project, and their predictions are within 20% of the real size. However, as stated previously, estimating the size does not always give an accurate indication of the implementation difficulty, and the necessary implementation time.

By measuring the number of internal I/O signals and components, their work goes along the same road as our initial observations indicate. However, our approach is pointing towards estimating the implementation effort, based on a behavioural description of the algorithm in the C-language. Furthermore, it also takes the designer's experience into account.

3. METHODOLOGY

The proposed flow for estimating the implementation effort is illustrated in Figure 1. It takes its outset in a behavioural description of the algorithm, in C-language (including library function source code), which is intended to be implemented in hardware. From this description, we use the design-Trotter framework to generate a hierarchical control data flow graph (HCDFG) which is then measured to identify the number of independent paths. The resulting measure, combined with the experience of the developers, gives an estimate of the required implementation effort. The method is self-learning in the sense that after each successful implementation, new knowledge about the developers involved can be integrated, and improve the accuracy of the estimates. The HCDFG and the approach for modelling the developers experience are covered later in this section but initially we investigate how the number of paths can be measured.

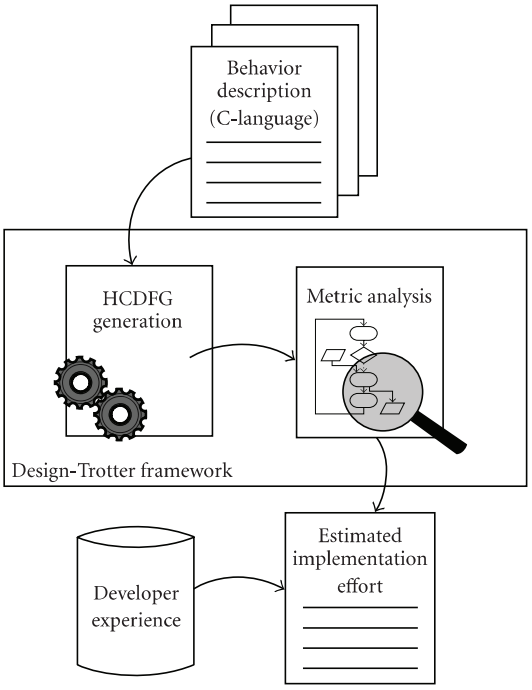
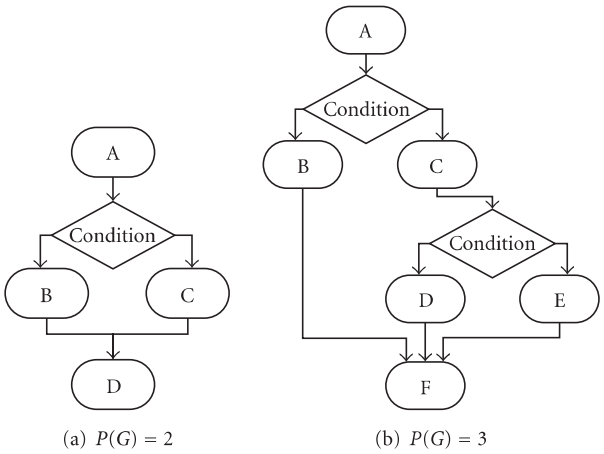


Figure 1: The flow of estimating the required implementation effort. The starting point is a behavioural description in C of the algorithm to be implemented in hardware (e.g., via VHDL). From this description, an HCDFG is generated and measured to identify the number of independent paths in the algorithm. This measure, combined with the experience of the developers, gives an estimate of the required implementation effort (expressed in time).

3.1. Cyclomatic complexity

As described in Section 1.3, the number of independent paths is expected to correlate with the complexity that the engineers are facing when working on the implementation. Therefore, finding a method to measure the number of independent paths in an algorithm could help us investigating this issue. A metric measuring is the cyclomatic complexity measure proposed by McCabe [15] which measures the number of linear-independent paths in the algorithm.

The cyclomatic complexity was originally invented as a way to intuitively quantify the complexity of algorithms, but has later found use for other purposes especially in the software domain. The cyclomatic complexity has been used for evaluating the quality of code in companies [16], where quality covers aspects from understandability over testability to maintainability. It has also been shown [17] that algorithms with a high cyclomatic complexity more frequently have errors than algorithms with lower cyclomatic complexity. The cyclomatic complexity has furthermore been used for evaluating programming languages for parallel computing [18], where languages that encapsulate control statement in instructions are receiving higher scores. All use the cyclomatic complexity measure under the assumptions that the complexity has significant influence on the number of paths the developers need to inspect, its correlation to the number of paths that needs to be tested, or a combination of the two.



(a) $P(G) = 2$ (b) $P(G) = 3$

Figure 2: Two examples of graphs for which the cyclomatic complexities have been calculated.

In the domain of hardware, the cyclomatic complexity has also found use, judging the readability and maintainability in the SAVE project [19]. It is worth noticing that they use a misinterpreted [20] definition of the cyclomatic complexity [21].

All these projects utilise the cyclomatic complexity's ability to measure the number of independent paths and relate them to their individual cases:

$$P(G) = \pi + 1, \tag{2}$$

where π represents the number of condition nodes in the graph G representing the algorithm being analysed. Figure 2 shows two examples of graphs and the corresponding cyclomatic complexity.

In this work, we propose an adapted version of the cyclomatic complexity definition to estimate, a priori, the number of independent paths on a hierarchical control data flow graph (HCDFG), defined in the following section. The cyclomatic complexity for an HCDFG is obtained by examining its subgraphs as explained in Section 3.3.

3.2. HCDFG

For this work we use the hierarchical control data flow graphs (HCDFGs), which are introduced in [22, 23]. The HCDFGs are used to represent an algorithm with a graph-based model so the examination task of the algorithm is eased. Control/Data Flow Graphs (CDFGs) are well accepted by designers as a representation of an algorithm where data flow graphs represent the data flow between different processes/operations, and the control flow layer, encapsulating these data flows and adding control structures to the graphical notation. The hierarchy layered structure is added to help representing large algorithms as well as to enable the analysis mechanism to identify functions/blocks in the graph. Such an identified block can then be seen as a single HCDFG that can be instantiated several times.

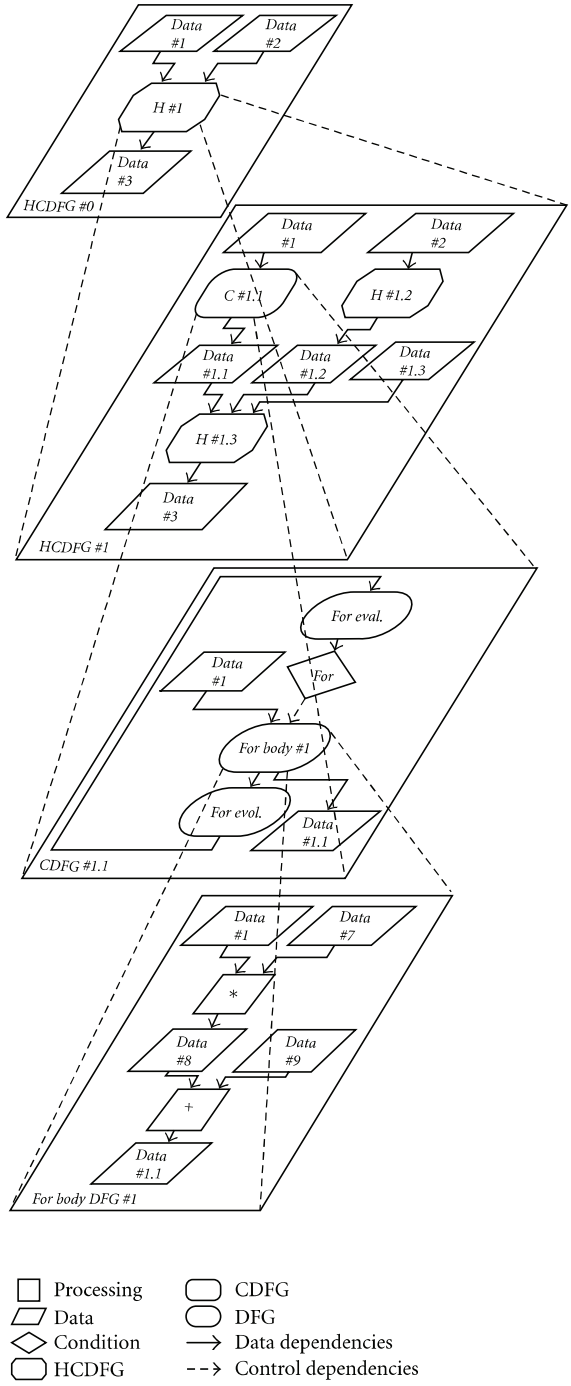


Figure 3: An overview of how the hierarchy in an HCDFG allows analysis of an algorithm on different levels and how the levels are related.

Figure 3 shows an example of a hierarchical control data flow graph.

In this work the design space exploration tool "Design-Trotter" is used as an engine for analysing the algorithms. The HCDFG model is used as "Design-Trotter's" internal representation.

The hierarchy of an HCDFG is shown in Figure 3. An HCDFG can consist of other HCDFGs, Control/Data flow graphs (CDFGs) and data flow graphs (DFGs) as well as elementary nodes (processing, memory, and control nodes). An HCDFG is connected via dependency edges. In this work we only explore the graph at levels above the DFGs, and therefore only concentrate on these when we define the graph types in what follows.

Let us consider the hierarchical control data flow graph, $G_{\text{HCDFG}} = (N_{\text{HCDFG}}, E_{\text{HCDFG}})$, where N_{HCDFG} arethe nodes denoted by $N_{\text{HCDFG}} = \{n_{\text{HCDFG}_1}, \ldots, n_{\text{HCDFG}_m}\}$ and the nodes are $N_{\text{HCDFG}} \in \{G_{\text{HCDFG}} | G_{\text{CDFG}} | G_{\text{DFG}} | \text{Data}\}$, meaning that the nodes in the G_{HCDFG} can be instances of its own type, encapsulated control data flow graphs, G_{CDFG}, encapsulated data flow graphs G_{DFG}, or data transfer nodes, *Data*. The last one is introduced to avoid the duplication of data representations in the hierarchy, when data is exchanged between the graphs. Thereby, data are only represented by their nodes and not by edges as it is common in many other types of DFGs.

The edges, E_{HCDFG}, connect the nodes such that $E_{\text{HCDFG}} = \{e_{n_{\text{HCDFG}_i}, n_{\text{HCDFG}_j}}\}$, where $i \neq j$ and represent the indexes of the nodes, $E_{\text{HCDFG}} \in \{\text{DD}\}$ and where every node can have multiple input and/or output edges. For the G_{HCDFG}, only data dependencies, DD, are allowed, and no control dependencies, CD.

In this way the HCDFG forms a hierarchy of encapsulated HCDFGs, CDFGs, and DFGs, connected via exchanging data nodes. The HCDFG can be seen as a container graph for other graph types such as the CDFG.

We can define the CDFG as $G_{\text{CDFG}} = (N_{\text{CDFG}}, E_{\text{CDFG}})$, where N_{CDFG} are the nodes denoted by $N_{\text{CDFG}} = \{n_{\text{CDFG}_1}, \ldots, n_{\text{CDFG}_m}\}$ and the nodes are $N_{\text{CDFG}} \in \{\text{CC} | G_{\text{HCDFG}} | G_{\text{DFG}} | \text{Data}\}$, where $\text{CC} \in \{\text{if}|\text{switch}|\text{for}|\text{while}|\text{do-while}\}$. In this way the G_{CDFG} is able to describe common control structures, where the actual data processing is encapsulated in either DFGs or HCDFGs. Again, the data exchange nodes are used to exchange data between the other nodes.

The edges, E_{CDFG}, connect the nodes such that $E_{\text{CDFG}} = \{e_{n_{\text{CDFG}_i}, n_{\text{CDFG}_j}}\}$, where $i \neq j$ and represent the indexes of the nodes. If $n_{\text{CDFG}_i} \in \text{CC}$ and $n_{\text{CDFG}_j} \in \{G_{\text{HCDFG}} | G_{\text{DFG}}\}$, then $\{e_{n_{\text{CDFG}_i}, n_{\text{CDFG}_j}}\} \in \{\text{CD}\}$, else $\{e_{n_{\text{CDFG}_i}, n_{\text{CDFG}_j}}\} \in \{\text{DD}\}$.

Beneath the control data flow graphs G_{CDFG}, the data flow graphs G_{DFG} exist but they are of no use in this work so we will not define them further here.

3.3. Calculating the cyclomatic complexity on CDFGs

Now that the HCDFG has been defined, we explain our proposed method for measuring the cyclomatic complexity on the CDFGs.

Since the cyclomatic complexity only considers the control structure in finding the number of independent paths in the algorithm, the DFG part of the algorithm is, as mentioned earlier, of no interest for this task because it only gives a single path. On the other hand, what is of interest is how the cyclomatic complexity is measured on the CDFGs

and HCDFGs which are built by the tool Design-Trotter. This leaves us with the following cases which are described in detail afterwards:

(i) If constructs,

(ii) Switch constructs,

(iii) For-loop,

(iv) While/do-while loops,

(v) Functions,

(vi) HCDFGs in parallel,

(vii) HCDFGs in serial sequence.

3.3.1. *If constructs*

"If constructs" case is represented as CDFGs, G_{CDFG}, where one node is a control node of type *if* (see Figure 4(a)). Before arriving at the control node, a condition evaluation node $n_{\text{eval}} \in \{G_{\text{HCDFG}} | G_{\text{DFG}}\}$ is traversed to calculate the boolean variable stored in n_{Data} (to maintain simplicity, these are not shown in Figure 4(a)) that is used in the condition node. If the variable is true, the algorithm follows the path through the true body node, $n_{\text{true}} \in \{G_{\text{HCDFG}} | G_{\text{DFG}} | \varnothing\}$. Else it goes to the false body node $n_{\text{false}} \in \{G_{\text{HCDFG}} | G_{\text{DFG}} | \varnothing\}$. Note that in some cases, either the true body or the false body does not exist, but it still gives a path. In this case, according to the cyclomatic complexity measure, the number of independent paths is

$$P(n_{\text{if}}) = P(n_{\text{true}}) + P(n_{\text{false}}) + P(n_{\text{eval}}) - 1. \quad (3)$$

The last part of (3), $+P(n_{\text{eval}}) - 1$ is included in case the evaluation graph is an HCDFG node.

3.3.2. *Switch constructs*

"Switch constructs" case is represented as CDFGs, G_{CDFG}, and is almost the same flow as the "if constructs" case discussed above. One node is a control node of switch type. Before arriving to the control node, a condition evaluation node $n_{\text{eval}} \in \{G_{\text{HCDFG}} | G_{\text{DFG}}\}$ is traversed. Depending on the output, the switch node leads the algorithm flow to the selected case node: $n_{\text{case}_i} \in \{G_{\text{HCDFG}} | G_{\text{DFG}}\}$. An example is shown in Figure 4(b). According to the cyclomatic complexity measure, the number of independent paths is as follows :

$$P(n_{\text{switch}}) = P(n_{\text{eval}}) - 1 + \sum_{i=1}^{N} P(n_{\text{case}_i}), \quad (4)$$

where N represents the number of cases, i the index to the corresponding node on which the paths are measured.

The same argument goes for the $P(n_{\text{eval}}) - 1$ part of (4); it is included in case the evaluation graph is an HCDFG node, but else it is omitted.

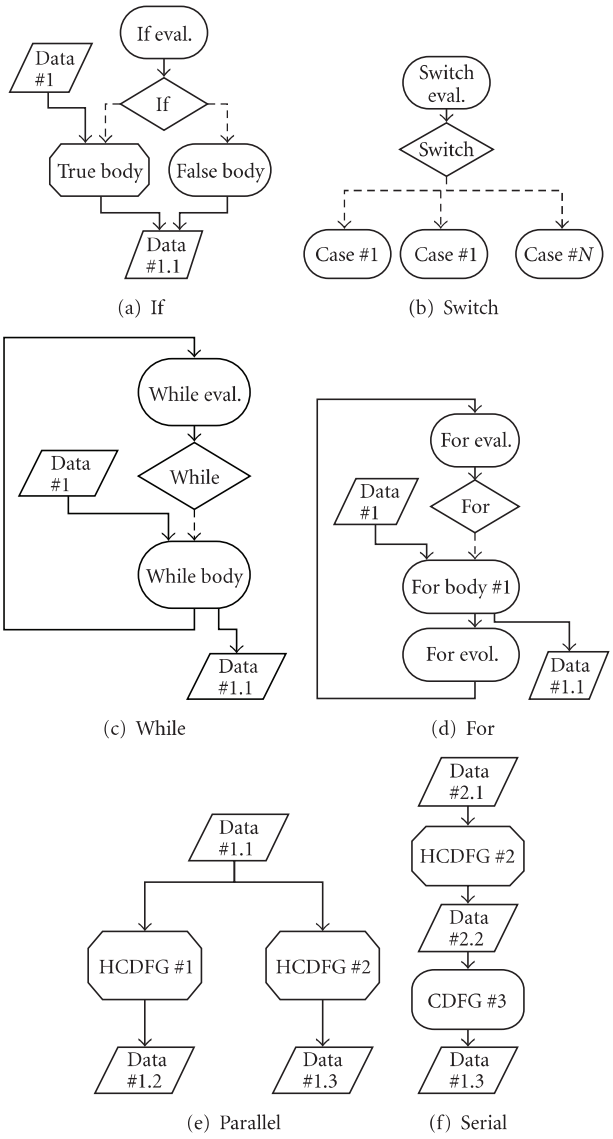


Figure 4: Overview of the different CDFGs and combined HCDFGs, on which the cyclomatic complexity values are measured. Between the (HC)DFGs there is a set of data exchange nodes which are here left out for simplicity. The symbols are similar to those presented in Figure 3.

3.3.3. *For-loop*

"For-loop" case is the most complex of the control structures. Strictly speaking, a "for loop" consists of three different parts: the evaluation body, the evolution body, and the for body, n_{eval}, n_{evol}, and $n_{\text{for-body}}$, respectively. The control node nfor, determines, based on the output from the evaluation graph, whether the flow should go into the "for loop" or leave it. The evolution node updates the indexes. Since each iteration of the graph needs to pass through the evaluation and evolution nodes, the number of independent paths is calculated as

$$P(n_{\text{for}}) = P(n_{\text{for-body}}) + P(n_{\text{eval}}) - 1 + P(n_{\text{evol}}) - 1. \quad (5)$$

In many cases, the evaluation and evolution part of the "for loop" are quite simple indexing functions, meaning that $n_{\text{eval}} \in \{G_{\text{DFG}}\}$, $n_{\text{evol}} \in \{G_{\text{DFG}}\}$, will leave $P(n_{\text{for}}) = P(n_{\text{for-body}})$. The "for loop" is illustrated in Figure 4(d).

3.3.4. While loops and do-while loops

"While loops" and "do-while loops" cases are described jointly since it is only the entry to the loop structure that separates them and their cyclomatic complexity are equivalent. The "while loops" consist of two main parts: the while body $n_{\text{while-body}} \in \{G_{\text{HCDFG}}|G_{\text{DFG}}\}$, and the while evaluation $n_{\text{eval}} \in \{G_{\text{HCDFG}}|G_{\text{DFG}}\}$. This is illustrated in Figure 4(c). Deciding whether to continue looping is decided by the control node $n_{\text{while}} \in \{\text{while}\}$ based on the output of the n_{eval}. Similarly to the "for loop," each iteration of the graph needs to pass through the evaluation nodes, so the number of independent paths can be calculated as

$$P(n_{\text{while}}) = P(n_{\text{while-body}}) + P(n_{\text{eval}}) - 1. \tag{6}$$

In many cases, the evaluation part of the while loop is a set of simple test functions, meaning that $n_{\text{eval}} \in \{G_{\text{DFG}}\}$, which leaves the $P(n_{\text{while}}) = P(n_{\text{while-body}})$.

3.3.5. Functions

The goal is to identify the number of independent paths in the algorithm/system. For this, reuse in terms of functions/blocks of code is important. When all independent paths through a function are known, reuse of this function does not change the number of independent paths in the system. From an implementation point of view, such functions represent an entity where the paths only need to be implemented once. In HCDFGs, a function/block can be seen as an encapsulated G_{HCDFG}. Therefore, the number of independent paths in function/blocks of reused code should only count once. The paths can be calculated as

$$P(n_{\text{HCDFG}_{\text{function}}}) = \begin{cases} 0 & \text{if reuse,} \\ P(n_{\text{HCDFG}}) & \text{else.} \end{cases} \tag{7}$$

3.3.6. HCDFGs in parallel and serial

Knowing how to handle all the HCDFGs that are identified for reuse (function), together with all the CDFGs, does not give it all. How the hierarchy of graphs should be combined is also of interest. For a parallel combination of two or more HCDFGs/CDFGs, as shown in Figure 4(e), the increase in the number of independent paths is then additive. The number of paths can be calculated as

$$P(n_{\text{HCDFG}_{\text{Parallel}}}) = \sum_{i=1}^{N} P(n_{\text{HCDFG}_i}), \tag{8}$$

where N represents the number of nodes in parallel, i the index to the corresponding node where the paths are measured.

For serial combination of two or more HCDFGs and/or CDFGs, the number of independent paths is a combination of the independent paths of the involved HCDFGs/CDFGs. Remembering that there always needs to be one path through the system, the number of independent paths in a serial combination, is given as

$$P(n_{\text{HCDFG}_{\text{Serial}}}) = \sum_{i=1}^{N} P(n_{\text{HCDFG}_i}) - (N-1), \tag{9}$$

where N represents the number of nodes in serial, i the index to the corresponding node where the paths are measured.

An example of serial combination is shown in Figure 4(f). The number of independent paths for the entire algorithm, $(P(n_{\text{HCDFG}_{\text{Alg}}}))$, is equivalent to the top HCDFG node which includes all the independent paths of its subgraphs.

3.4. Experience impact

The experience of the designer has an impact on the challenge that he/she is facing when developing a system. A radical example is when a beginner and a developer with ten years of experience are asked to solve the same task. They will not see equal difficulty in the same task, and thereby do not need to put the same effort into the development.

Experience is influenced by many parameters but in this work we only focus on the time the developer has worked with the implementation language and the target architecture.

The impact of experience is a factor that slowly decreases over time: consider a new developer, the experience that he/she obtains in the first months working with the language, and architecture improves his/her skills significantly. On the other hand, a developer who has worked with the language and architecture for five years, for example, will not improve her/his skills at the same rate by working an extra year. The impact from the experience is therefore not linear but tends to have a negative acceleration or inverse logarithmic nature, with dramatic change in impact in the beginning, progressing towards little or no change as time increases.

In literature, for example, [24], many studies try to fit historical data to models. An example of a model is a power function with negative slope or a negative exponential function. From the vast variety of models that has been proposed over the years, the only conclusion that can be drawn is that there are multiple curvatures, but they all appear to have a negative accelerating slope, which tends to be exponential/logarithmic.

In order to get the best possible outset for predicting the implementation effort, it is of vital importance to obtain some data of the developers' experiences, and also how they performed in the past. The parameters involved in the experience curve can then be trimmed to create the best possible fit. However, it has not been the purpose of this work to select the perfect nature for a learning curve nor to evaluate the accuracy of such one. The learning curve will be adapted to the individual developers, and as the model is used in subsequent projects, its accuracy will progressively improve. As a consequence, the experience here is only

intended as an element in modelling the complexity and thereby a means for more accurate estimates.

For the experiments in this study we have chosen to use the following model:

$$\eta_{\text{experience}}(\text{Dev}) = \frac{1}{\alpha \log(\text{Experience}(\text{Dev}) + \beta)}, \quad (10)$$

where α and β are trim parameters which can be used to optimise the curve to fit reality, Experience is the number of weeks which the developer, Dev, has worked with the language and architecture. Figure 5 depicts the shape of the experience model.

In this work,our initial experiments have shown that setting $\alpha = 1$ and $\beta = 1$ makes our model sufficiently general, and therefore we have not further investigated the tuning of these two parameters.

4. RESULTS

In order to verify the hypothesis, a classical test has been conducted. The test is dual phased and consists of (i) a training phase using a first set of real-life data, during which the hypothesis is said to be true, and (ii) a validation phase during which a second set of real-life data is used to evaluate whether the hypothesis holds true or not.

4.1. *Phase one—training*

The real-life data used as training data originate from two different application types that are both developed as academic projects in universities in France. The first application is composed of five different video processing algorithms for an intelligent camera, which is able to track moving objects in a video sequence. The second application is a cryptographic system, able to encrypt data with different cryptographic/hashing algorithms, that is, MD5, AES and SHA-1. The system consists of one combined engine [25] as well as individual implementations. These projects were selected since they all follow the methodology of using a behavioural specification in C, as a starting point for the VHDL implementation. Common to this data is that none of the developers has made the behavioural specification in C. For the cryptographic algorithms the behavioural specification comes from the standards, and the video algorithms were based on a previous project.

Using the behavioural description as the starting point of the experiment, the exercise consists of studying the relationship between the complexity of the algorithms (as defined in Section 3) and the implementation effort (i.e., time) required to implement them in VHDL (including testbed and heuristic tests).

The developers involved in these projects have all been Master and Ph.D. students with electrical engineering backgrounds but no VHDL background other than what they obtained during their studies, see Table 2. All developers were taught VHDL by other instructors than the authors, but at our university. Table 3 summaries the training data.

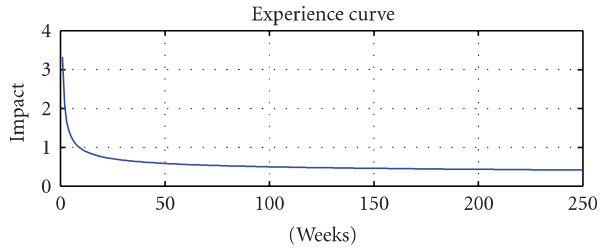


Figure 5: An example of how the lack of experience impacts the difficulty the engineers are facing.

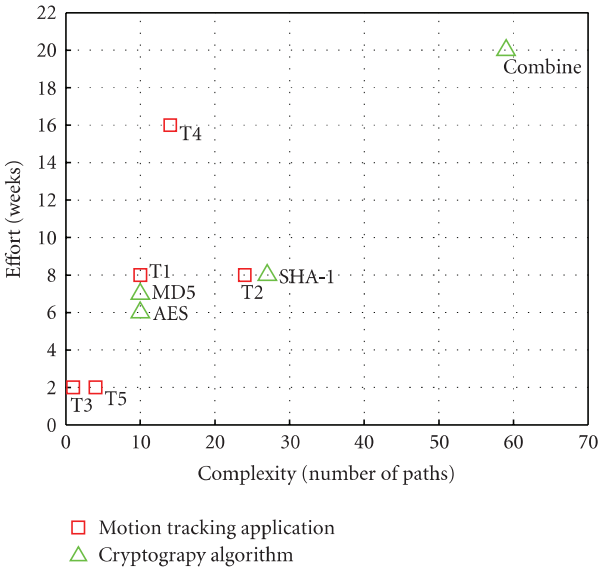


Figure 6: Relation between the implementation effort (number of weeks) and the not corrected complexity (as defined in Section 3).

Figure 6 shows the relation between the implementation effort and the measured complexity for the individual algorithms. Please note that in this graph the complexity values are not yet corrected for the designers' experience.

A first examination of the data points indicates a possible relation between some of them. However many other points are located far away from any relation. These data are not corrected for the designers' experience and, as earlier mentioned, we strongly believe that the experience of the individual designer has a nonnegligible influence on the development time. If we inspect the data more thoroughly, it is clear that the points of greatest divergence are those implementations where the developers have very limited knowledge and experience with the VHDL language.

Applying the proposed equation (10) (nonlinear) experience transform onto the data, results in a significantly different picture as depicted in Figure 7. A clear trend toward a relation is now visible in the plotted data. From the COCOMO II project [8], it is known that the relationship between the implementation time and the complexity measure (in their case lines of code, LOC) can be expressed as a

Table 1: Line of code, area, and time constraints for the validation data.

Algorithm	SS1	SS2	SS3	SS4	SS5	SS6	Ethernet	App 4
Dev. Time (weeks)	3.6	6.4	2.4	16.4	12	17.2	16	2
LOC-VHDL	994	1195	776	1695	760	2088	3973	232
Slices	564	2212	382	888	372	2171	3372	750
FlipFlops	913	2921	1290	1366	1208	2077	6149	942
LUTs	997	3157	6453	1569	6443	3458	18255	567
Time Constraint. (ns)	112	128	360	112	360	248	696	56

Table 2: Facts about the developers. Developers for training data (top) and validation data (bottom).

Developer	Education	Years in the domain
Dev 1	Ph.D. stud.	0
Dev 2	Stud. (EE)	0
Dev 3	Stud. (EE)	0
Dev 4	Stud. (EE)	0
Dev 5	BSc.EE.	9
Dev 6	MSc.EE.	15
Dev 7	MSc.EE.	9
Dev 8	MSc.EE.	8
Dev 9	MSc.EE.	8

Table 3: Training data (top) and validation data (bottom). Algorithms are related to the developers and their experience at the given time. Complexity is not corrected.

Algorithm	Complexity	Developer	Dev. Exp.
T1	10	Dev 1	2
T2	24	Dev 1	10
T3	12	Dev 1	18
T4	14	Dev 2	1
T5	4	Dev 1	20
MD5	10	Dev 3	1
MD5	10	Dev 4	1
AES	10	Dev 4	8
SHA-1	27	Dev 4	14
Combined	59	Dev 4	14
SS1	25	Dev 6, 7	150
SS2	35	Dev 5	150
SS3	17	Dev 5, 6, 7, 8	150
SS4	50	Dev 6	6
SS5	29	Dev 7	3
SS6	25	Dev 5, 6, 7	3
Ethernet app	60	Dev 5, 6, 7, 8, 9	150
App 4	9	Dev 6	150

power function with a weak slope. We showed its nature in (1), and with correction for experience it becomes

$$\text{Effort} = A \times \eta_{\text{experience}}(\text{Dev}) \times P\left(n_{\text{HCDFG}_{\text{Alg}}}\right)^{b}. \quad (11)$$

The parameters A and b are found, via a least square (LS) fit on our training data, to be $A = 0.226$ and $b = 1.103$. In Figure 7 the dashed line illustrates the relationship, with the parameters given above.

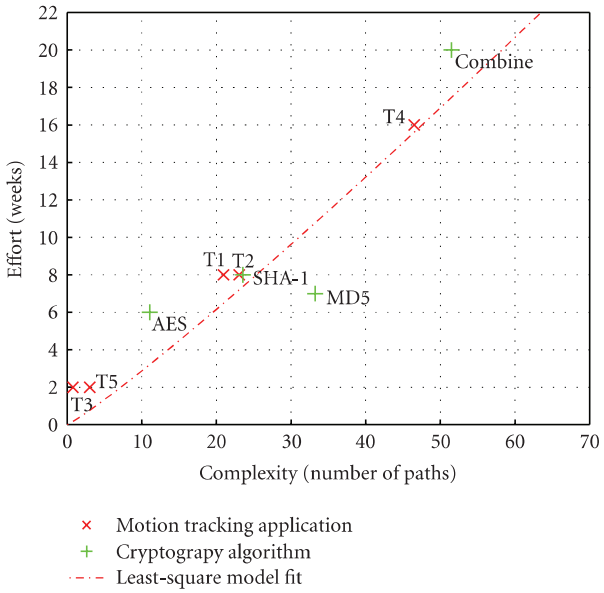


Figure 7: Relation between the implementation effort (number of weeks) and the complexity corrected according to the designers' experience model as shown in Figure 5.

4.2. *Phase two—validation*

After having elaborated on a model based on the training data, we proceeded with the validation of its correctness. For this, a new set of data provided by ETI A/S, a Danish SME, is used. The dataset originates from a networking system and consists of Ethernet applications that have been implemented on an FPGA, as well as corresponding testbeds. This Ethernet application is part of an existing system with which it requires interaction. Table 1 shows additional implementation information with regards to these applications. The system is a real-time system with hard-time constraints and all algorithms were implemented as to meet these constraints. Similar to the training data, the development flow for this application has been as follows: a behavioural C++ model of the application has been constructed before the implementation on the FPGA architecture. The behavioural model has been developed by developers separate to those undertaking the implementation. The developers responsible

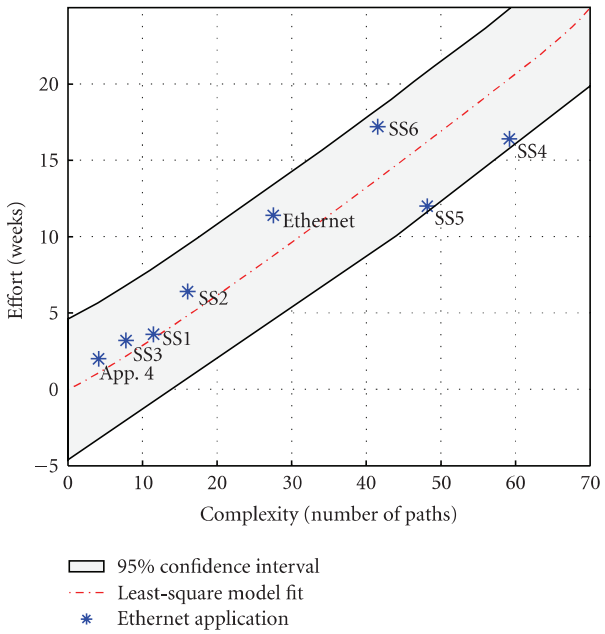


FIGURE 8: Validation data plot: relation between implementation effort (number of weeks) and complexity, corrected according to the designers' experience model.

for the implementation have obtained their skills in VHDL from a professional course with no relation to our university in Denmark.

The time spent on the implementation process covers: the design and implementation of the VHDL code of the functionalities and testbed as well as the tests of the different modules in the applications. This data is shown in the lower part of Table 3. The time data originate from the company's internal registration for the project, and correspond therefore to the effective time used.

The relation between implementation effort and complexity is plotted in Figure 8. It can be seen that this data, corrected for the designers' experience ($*$) closely follows the model derived from the training data (dashed line). Figure 8 also shows the 95% confidence interval, indicating that, with 95% confidence, future predictions of implementation effort will lie within this interval, given that the model holds true.

Comparing the predicted effort (dashed line) to the real effort ($*$), indicates that there is an estimation error. The values are also shown in Table 4, The average estimation error is 0.2 week with a variance of 8. In the next section, we discuss the validity of the model.

4.3. Validity discussion

Estimating the effort required in implementing an algorithm into hardware involves many parameters. We discussed a number of these parameters in Section 1.2, but could not include them all in this study. The proposed model is therefore devised from the idea of the relation between implementation effort and number of linear-independent paths.

TABLE 4: Development time and estimated development time measured in weeks together with the error.

Algorithm	Dev. time	Est. dev. time	Error
SS1	3.6	3.3	0.3
SS2	6.4	4.8	1.6
SS3	3.2	2.2	1
SS4	16.4	20.3	−3.9
SS5	12	16.2	−4.2
SS6	17.2	13.8	3.4
Ethernet app	11.4	8.8	2.6
App 4	2	1.1	0.9
Mean (variance):			0.2 (8)

To validate the model, a classical two-phased hypothesis test has been performed and the validity of this test depends on the following important factors: (i) the independence between training and validation data; (ii) the volume and variety of the experiments.

In the first instance, not only different applications were used for training and validation data, but in addition the developers had no relation in terms of education, nationality, work, and so forth. Moreover, the validation data has not been measured before the model was trained. All this strengthens the validity of the results. The only potential connection is that some of the developers who have been involved in the implementation of the training and validation data have also been included within those interviewed. However, this accounts for a minority and we see this as a minimal risk.

Secondly, we should ideally have had a large volume and variety of experimental data for training and validation. However, our set of data originates from a single company and a few developers. So strictly speaking we can only conclude that this model applies to the specific SME setup involved in the study and partially to the academic environment studied.

In order to generalise our model, more cases of validation are needed. However, obtaining all the statistical data for this new methodology is time consuming. We would therefore like to remind the reader that this paper proposes a methodology for estimating implementation effort and the validation of the model concentrates on illustrating its usefulness. Looking at the graphs, we can determine a clear trend in the results. The curve identified in the training data is sustained for the validation data as well: they both fall in line with the underlying rationale, and we are quite confident in the strength of the proposed model.

The results clearly show the necessity for the proposed correction function; the proposed logarithmic nature works well, even though the correction function has not been trimmed to fit the individual developers due to the lack of available data. In this light, our approach must be seen as the engine of a global methodology for the management of design projects, that impose a systematic registration of man-power. With such a registration, a database of the developers' experience can easily be constructed and the

correction function can be trimmed to fit the companies' individual designers. Several iterations of this process would provide convergence towards a more precise estimation of the implementation effort.

The limited data set on which the model is constructed also limits the complexity window to which this model can be applied: having no algorithm with a corrected complexity value larger than 51, extrapolating the model further would weaken the current conclusion. More training data, from larger and more varied projects would allow for a more refined model.

Nevertheless, the results described in this paper are very encouraging with all the real-life cases that we have examined and we are reasonably confident that this model can easily be applied to other types of applications.

5. CONCLUSION

The contribution presented in this paper is a metric-based approach for estimating the time needed for hardware implementation in relation to the complexity of an algorithm. We have deduced that a relationship exists between the number of linear-independent paths in the algorithm and the corresponding implementation effort. We have proposed an original solution for estimating implementation effort that extends the concept of the cyclomatic complexity.

To further improve our solution, we developed a more realistic estimation model that includes a correction function to take into account the designer's experience.

We have implemented this solution in our tool design Trotter of which the input is a behavioural description in C language and the output is the number of independent paths. Based on this output and the proposed model, we are able to predict the required implementation effort. Our experimental results, using industrial Ethernet applications, confirmed that the data, corrected for the designers' experience, follows the derived model closely and that all data falls inside its 95% confidence interval. Using this method iteratively paves the way for an implementation effort estimator of which the accuracy improves continuously after each project.

REFERENCES

[1] D. Blaza, "Embedded systems design state of embedded market survey," Tech. Rep., CMP Media, New York, NY, USA, 2006.

[2] R. Nass, "An insider's view of the 2008 embedded market study," Tech. Rep., CMP Media, New York, NY, USA, 2008.

[3] "Workshop for Danish smes developing embedded systems co-organized by the Danish technological institute, the center for software defined radio (csdr) and the center for embedded software systems (ciss)," Nyhedsmagasinet Elektronik & Data, Nr.1 2008, Aarhus, Denmark, 2008.

[4] O.-H. Kwon, "Keynote speaker: perspective of the future semiconductor industry: challenges and solutions," in *Proceedings of the 44th Design Automation Conference (DAC '07)*, San Diego, Calif, USA, June 2007.

[5] S. McConnell, *Software Estimation: Demystifying the Black Art*, Microsoft Press, Wasthington, DC, USA, 2006.

[6] R. Abildgren, A. Saramentovas, P. Ruzgys, P. Koch, and Y. Le Moullec, "Algorithm-architecture affinity—parallelism changes the picture," in *Proceedings on the Design and Architectures for Signal and Image Processing*, Grenoble, France, November 2007.

[7] M. A. Honey, "The interview as text: hermeneutics considered as a model for analysing the clinically informed research interview," *Human Development*, vol. 30, pp. 69–82, 1987.

[8] B. W. Boehm, C. Abts, A. W. Brown, et al., *Software Cost Estimation with COCOMO II*, Prentice-Hall, Upper Saddle River, NJ, USA, 2000.

[9] B. W. Boehm, *Software Engineering Economics*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1981.

[10] T. Jones, *Programming Productivity*, McGraw-Hill, New York, NY, USA, 1986.

[11] G. C. Low and D. R. Jeffery, "Function points in the estimation and evaluation of the software process," *IEEE Transactions on Software Engineering*, vol. 16, no. 1, pp. 64–71, 1990.

[12] A. J. Albrecht, "Measuring application development productivity," in *Proceedings of the IBM Application Development Symposium*, pp. 83–92, Monterey, Calif, USA, October 1979.

[13] D. R. Jeffery, G. C. Low, and M. Barnes, "Comparison of function point counting techniques," *IEEE Transactions on Software Engineering*, vol. 19, no. 5, pp. 529–532, 1993.

[14] W. Fornaciari, F. Salice, U. Bondi, and E. Magini, "Development cost and size estimation starting from high-level specifications," in *Proceedings of the 9th International Symposium on Hardware/Software Codesign*, pp. 86–91, Copenhagen, Denmark, April 2001.

[15] T. J. McCabe, "A complexity measure," *IEEE Transactions on Software Engineering*, vol. 2, no. 4, pp. 308–320, 1976.

[16] D. L. Lanning and T. M. Khoshgoftaar, "Modeling the relationship between source code complexity and maintenance difficulty," *Computer*, vol. 27, no. 9, pp. 35–40, 1994.

[17] T. J. Walsh, "Software reliability study using a complexity measure," in *Proceedings of the National Computer Conference (NCC '79)*, vol. 48, pp. 761–768, AFIPS Press, New York, NY, USA, June 1979.

[18] S. P. VanderWiel, D. Nathanson, and D. J. Lilja, "Complexity and performance in parallel programming languages," in *Proceedings of the 2nd International Workshop on High-Level Programming Models and Supportive Environments (HIPS '97)*, pp. 3–12, Geneva, Switzerland, April 1997.

[19] M. Mastretti, M. L. Busi, R. Sarvello, M. Sturlesi, and S. Tomasello, "VHDL quality: synthesizability, complexity and efficiency evaluation," in *Proceedings of the European Design Automation Conference with EURO-VHDL (EURO-DAC '95)*, pp. 482–487, IEEE Computer Society Press, Brighton, UK, September 1995.

[20] I. Feghali and A. H. Watson, "Clarification concerning modularization and mccabe's cyclomatic complexity," *Communication of the ACM*, vol. 37, no. 4, pp. 91–94, 1994.

[21] B. Henderson-Sellers, "Modularization and mccabe's cyclomatic complexity," *Communication of the ACM*, vol. 35, no. 12, pp. 17–19, 1992.

[22] Y. Le Moullec, N. B. Amor, J.-P. Diguet, M. Abid, and J.-L. Philippe, "Multi-granularity metrics for the era of strongly personalized SOCs," in *Proceedings of the Conference on Design, Automation and Test in Europe (DATE '03)*, vol. 1, pp. 674–679, Munich, Germany, March 2003.

[23] Y. Le Moullec, J.-P. Diguet, N. B. Amor, T. Gourdeaux, and J.-L. Philippe, "Algorithmic-level specification and characterization of embedded multimedia applications with design

trotter," *The Journal of VLSI Signal Processing*, vol. 42, no. 2, pp. 185–208, 2006.

[24] A. Heathcote, S. Brown, and D. J. Mewhort, "The power law repealed: the case for an exponential law of practice," *Psychonomic Bulletin & Review*, vol. 7, no. 2, pp. 185–207, 2000.

[25] S. Ducloyer, R. Vaslin, G. Gogniat, and E. Wanderley, "Hardware implementation of a multi-mode hash architecture for MD5, SHA-1 and SHA-2," in *Proceedings on the Design and Architectures for Signal and Image Processing Workshop (DASIP '07)*, Grenoble, France, November 2007.

Chapitre 8

Projet "Methodologies for Mapping Multiple Functionalities to Reconfigurable Heterogeneous Architectures"

8.1 Contexte

Le projet "Methodologies for Mapping Multiple Functionalities to Reconfigurable Heterogeneous Architectures" est le fruit de discussions entre Center for Embedded Software Systems (CISS) (puis Center for Software Defined Radio (CSDR)) à Aalborg University et Rohde & Schwarz Technology Center A/S, Danemark. Ce projet était financé par Danish Ministry of Higher Education and Science.

Lors de nos discussions avec Rohde & Schwarz Technology Center A/S (que je remercie), ses responsables et ingénieurs avaient mis en avant le souhait d'évaluer les approches de calcul reconfigurable pour les applications de type radio logicielle, notamment via les possibilités relativement récentes des FPGA, principalement quelques modèles de Xilinx à l'époque (2007). Les deux grandes questions qui avaient émergé de ces discussions étaient de savoir d'une part dans quelles conditions une implantation reconfigurable (globale ou partielle dynamique) est faisable et intéressante par rapport à une version statique, et d'autre part comment passer des applications sur des cibles hétérogènes logicielle/matérielle comprenant de telles parties reconfigurables.

Ceci constitue le point de départ du projet "Methodologies for Mapping Multiple Functionalities to Reconfigurable Heterogeneous Architectures" et du doctorat d'Andreas Popp ("Mapping Framework for Heterogeneous Reconfigurable Architectures : Combining Temporal Partitioning and Multiprocessor Scheduling"), co-encadré par Peter Koch. Pour information, une partie de ces travaux a été effectuée en coopération avec OFFIS, Institute for Information Technology en Allemagne où Andreas Popp a effectué un séjour de recherche entre février et juin 2009 (je remercie Kim Grüttner et Andreas Herrholz pour avoir accueilli et co-encadré Andreas) ainsi qu'avec Telecom Bretagne que moi-même et Andreas avons visité respectivement en juin et septembre 2009 (je remercie Christophe Jégo pour son accueil chaleureux et pour nous avoir rendu visite).

8.2 Problématiques

Les ingénieurs de Rohde & Schwarz Technology Center A/S étaient intéressés par les approches de calcul reconfigurable car celles-ci offrent, potentiellement, des avantages par rapport à des implantations sur ASIC ou GPP. Tout d'abord elles rendent possibles le partage temporel des ressources et permettent ainsi de réduire les besoins en surface. Elles permettent aussi de réduire la consommation énergétique statique des circuits (dont la part est grandissante dans la consommation totale) car moins de surface est nécessaire et/ou certaines parties des circuits peuvent être mis en veille. Enfin, ces approches offrent de la flexibilité car les circuits peuvent être reconfigurés, soit globalement, soit partiellement et dynamiquement, promettant ainsi le meilleur des mondes matériel et logiciel. Cependant, l'exploitation de telles approches soulèvent de nombreux problèmes liés à la faisabilité et aux méthodes de conception et l'implantation de tels systèmes, tels ceux discutés dans ce qui suit.

8.2.1 Problématique de l'évaluation de la faisabilité d'implantations reconfigurables

De nombreuses publications donnent des exemples ou l'utilisation d'une cible reconfigurable permet d'atteindre des gains substantiels sur le temps d'exécution (p. ex. jusqu'à 500x en temps d'exécution et and 70% de réduction de la consommation énergétique sur certaines applications [7]). Cependant, lorsque ces travaux ont commencé, il n'y avait pas vraiment de méthode disponible pour évaluer la faisabilité et l'intérêt d'une implantation reconfigurable par rapport à une version statique.
Le but était de proposer une méthode pour évaluer cette faisabilité (d'un point de vue temps d'exécution/ressources) de la reconfiguration (complète et partielle) d'applications sur cibles FPGA. Au vue des discussions avec notre partenaire industriel, l'objectif était de garder le temps de conception relativement court en évitant de se lancer dans des analyses fines nécessitant des modèles très détaillés (généralement lent à simuler et pas forcément disponibles tôt dans le flot de conception). Nous avons donc décidé de proposer une méthode reposant sur une modélisation gros grain de l'application, de l'architecture, et des temps d'exécution, de reconfiguration et d'éventuelles communications. La contribution correspondante est détaillée dans la section 8.3.

8.2.2 Problématique de la conception de systèmes sur cibles reconfigurables

L'une des grandes difficultés liées aux systèmes de calcul reconfigurables est la conception de ceux-ci. Ici nous nous sommes intéressé au passage des applications vers des cibles reconfigurables hétérogènes logicielles/matérielles (nous supposons que les cibles et technologies correspondantes sont disponibles).
La complexité de conception augmente avec la complexité de la cible, p. ex. une seule unité de calcul matérielle, une seule unité de calcul matérielle et une seule unité de calcul matérielle, plusieurs unités de calcul matérielles et plusieurs unités de calcul matérielles) et l'utilisation de la reconfiguration (globale ou dynamique partielle).
Parmi les principaux problèmes à résoudre se trouvent le groupement des tâches en modules reconfigurables, le partitionnement logiciel/matériel et l'ordonnancement. Bien que diverses méthodes de conception étaient disponibles lorsque ces travaux débutaient, elles

n'étaient soit pas adaptées aux cibles reconfigurables hétérogènes, ou alors complexes (p. ex. algorithmes génétiques).

Nous avons donc proposé un flot de conception relativement simple reposant sur l'extension et la modification de méthodes légères existantes de manière à couvrir le cas des cibles reconfigurables hétérogènes à plusieurs unités de calcul logicielles et matérielles. Nous avons utilisé le partitionnement temporel pour générer les groupes de tâches matérielles et modifié une méthode issue du monde multiprocesseurs pour ordonnancer ces groupes de tâches et leurs configurations tout en tenant compte des communications.
Comme la précédente, cette contribution est détaillée dans la section 8.3.

8.3 Résumé des contributions

8.3.1 Contribution à l'évaluation de la faisabilité d'implantations reconfigurables

Le modèle d'architecture reconfigurable considérée pour cette première est illustrée dans la figure 8.1.
Notre approche se distingue de l'état de l'art en ce que le modèle reflète le temps d'exécution plutôt que le débit (voir p. ex. [8]). Ainsi notre modèle permet l'évaluation du partage temporel des ressources (surface) et de la reconfiguration partielle avec une fonction de coût de type produit temps-surface (voir la publication C.30 dans le chapitre 5).

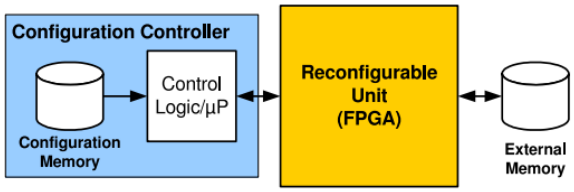


Figure 8.1: Modèle d'architecture reconfigurable considérée pour l'évaluation de faisabilité. Extrait de la publication C.30.

Le coût d'une implantation statique est exprimé par :

$$C_{static} = A_{static} \cdot T_{static} \text{ [s} \cdot \text{slices]}$$

où A_{static} est la surface totale en slices CLB et T_{static} est le temps total d'exécution en secondes.

Le coût d'une implantation reconfigurable globale (voir figure 8.2 pour le schéma d'exécution) est quant à lui exprimé par :

$$T_{exec} = \sum_i t_{exec,i}$$

$$T_{reconfig} = \sum_i t_{reconfig,i}$$

$$T_{transfer} = \sum_i t_{read,i} + \sum_i t_{write,i}$$

$$T_{global} = t_{exec} + T_{reconfig} + T_{transfer}$$

où I est le nombre total de configurations, i l'index de configuration, T_{global} le temps d'exécution en mode reconfiguration globales, T_{exec} le temps total d'exécution effective, $T_{reconfig}$ le temps total de reconfiguration, $T_{transfer}$ le temps de communication de données, $t_{exec,i}$ i, $t_{reconfig,i}$, $t_{read,i}$ et $t_{write,i}$ respectivement les temps d'exécution, de reconfiguration, de lecture vers, et d'écriture depuis de la configuration i.

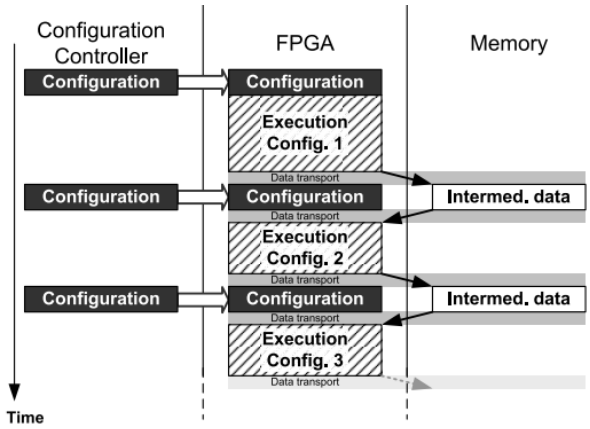


Figure 8.2: Schéma d'exécution version reconfiguration globale. Extrait de la publication C.30.

Le coût d'implantation reconfigurable globale est donné par :

$$C_{global} = A_{global} \cdot T_{global} \text{ [s} \cdot \text{slices]}$$

Le coût d'une implantation reconfigurable partielle dynamique est légèrement plus compliqué à estimer car au lieu de simplement multiplier le temps d'exécution total et la surface totale il faut multiplier la somme de temps d'exécution et de reconfiguration par la surface consommée par chaque module reconfigurable. Nous considérons que le transfert de données est effectué par les Bus Macros (technologie Xilinx) et que les délais afférents sont négligeables. Par contre leur surface ne l'est pas.
Le coût d'une implantation reconfigurable partielle dynamique (voir figure 8.3 pour le schéma d'exécution) est donnée par :

$$C_{partial} = C_{partial,proc} + C_{partial,comm} \text{ [s} \cdot \text{slices]}$$

avec

$$C_{partial,proc} = \sum_j A_j \cdot t_{exec,j} + \sum_j A_j \cdot t_{reconf,j}$$

et

$$C_{partial,comm} = A_{busregs} \cdot T_{partial}$$

où j est l'index de module reconfigurable, A_j la surface du module j [slices], $A_{busregs}$ la surface des Bus Macros [slices], $t_{exec,j}$ le temps d'exécution du module j [s], $t_{reconfig,j}$ le temp de reconfiguration du module j [s], $T_{partial}$ le temps total d'exécution effective, $C_{partial,proc}$

le coût d'exécution et de reconfiguration [s · slices], $C_{partial,comm}$ le coût de communication [s · slices], et $C_{partial}$ le coût total de la reconfiguration partielle dynamique [s · slices].

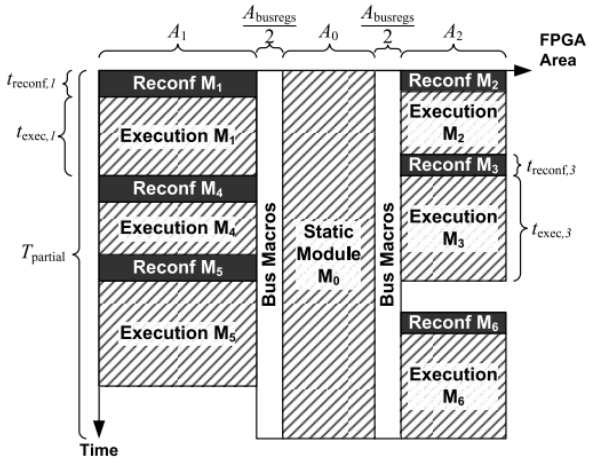


Figure 8.3: Schéma d'exécution version reconfiguration partielle dynamique. Extrait de la publication C.30.

Afin d'évaluer la faisabilité, la deadline de l'application $T_{deadline}$ doit être connue ou déterminée (pour le cas de l'implantation statique T_{static}=$T_{deadline}$). Pour la version reconfigurable globale il est nécessaire de décomposer l'application en configurations pouvant s'exécuter séquentiellement ; pour la version reconfigurable partielle dynamique il est nécessaire de regrouper les tâches pour former les modules reconfigurables. Ceci peut être effectué manuellement ou automatiquement (p. ex. via [9]). Dans ce dernier cas il faut connaître ou estimer les temps d'exécution et l'utilisation des ressources, par exemple via une étape de synthèse logique.

Enfin, une fois les coûts calculés, l'une ou les deux formes d'implantations reconfigurables sont considérées comme faisables (et intéressantes) si les deux conditions suivantes sont remplies :

$$C_{global}, C_{partial} \leq C_{static}$$

et

$$T_{global}, T_{partial} \leq T_{deadline}$$

Nous avons évalué l'approche proposée ci-dessus sur un cœur FFT et un récepteur DAB (voir publication C.30 pour les détails). De manière générale, les deux études montrent que le temps de reconfiguration est pénalisant, et peut annuler les gains en surface des versions reconfigurables. Dans le cas du cœur FFT (dynamique globale vs. statique), aucune des conditions n'est respectée (45,8 vs. 35,8 s *cdot* slices et 7,4 ms vs. 1s). Dans le cas du récepteur DAB (dynamique globale vs. statique), bien que la condition de coût soit respectée (307 vs. 340 s *cdot* slices), celle du temps d'exécution ne l'est pas (22,13 ms vs. 20 ms).
D'un autre côté, la version reconfigurable dynamique partielle permet de remplir les deux conditions : facilement pour le coût (28,4 vs. 307 s *cdot* slices) et de justesse pour le temps d'exécution (20 ms vs. 20 ms).

Depuis la publication de ces travaux, d'autres se sont intéressés à l'aspect consommation, comme par exemple [10].

8.3.2 Contribution à la conception de systèmes sur cibles reconfigurables

Foncièrement, le passage et l'ordonnancement d'une application sur une cible reconfigurable hétérogène logicielle/matérielle est un problème d'optimisation contraint. Les contraintes et les coûts sont typiquement décrits par une ou plusieurs fonctions. Dans notre cas nous visons à minimiser le temps d'exécution total (*makespan*) tout en satisfaisant les contraintes, notamment de ressources, de la cible (ce qui correspond plutôt bien avec le flot Xilinx pour la reconfiguration).
La méthode proposée est illustrée dans la figure 8.4 et résumée dans ce qui suit.

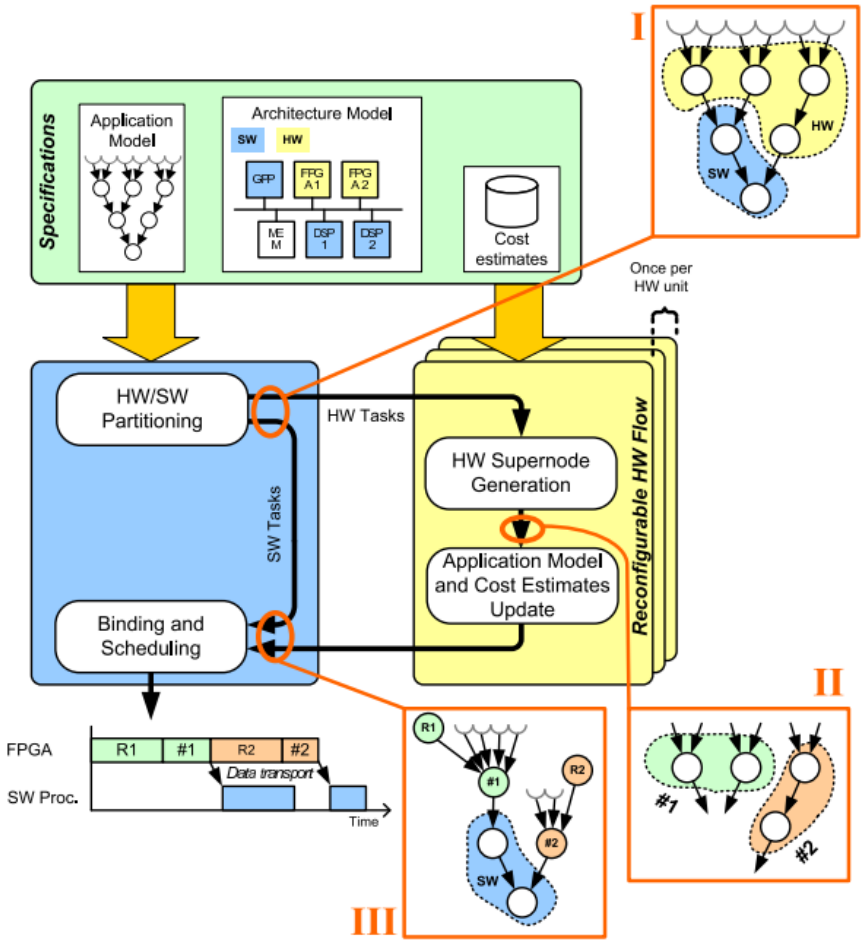


Figure 8.4: Méthode de conception systèmes sur cibles reconfigurables. Extrait de la thèse de doctorat d'Andreas Popp.

Le point d'entrée du flot est constitué des spécifications de l'application, de l'architecture et de leurs relations via une bibliothèque d'estimations des coûts. La bibliothèque de coûts contient les coûts associés à différentes associations (logicielles/matérielles) pour chaque tâche, correspondant à un point dans l'espace des solutions. Un tel point correspond à un ensemble temps d'exécution, ressources et éventuellement blocs DSP et blocs mémoire dans le FPGA.

Dans la première étape (HW/SW partitioning), des groupes de tâches logicielles et matérielles sont formés (boîte I dans la figure). Les groupes de tâches matérielles (correspondant à des ressources matérielles, typiquement une unité de calcul reconfigurable) sont traités par la partie flot matériel reconfigurable (Reconfigurable HW Flow) à l'aide d'une méthode de partitionnement temporel pour générer des supernœuds matériels (HW Supernode Generation). Chaque supernœud correspond à une configuration complète de la partie reconfigurable du système (boîte II dans la figure). La création de ceux-ci est reflétée dans le graphe représentant l'application et dans la table des coûts (Application Model and Cost Estimates Update). Cette partie flot matériel reconfigurable est répétée pour chaque unité de calcul reconfigurable. Ceci résulte dans le nouveau graphe d'application (boîte III dans la figure).
La décision de partitionnement peut être effectuée selon plusieurs stratégies. La plus simple est de mettre une tâche en matériel si le temps d'exécution correspondant est plus court qu'en logiciel. De manière un peu plus fine, il est aussi possible de prendre en compte l'utilisation des ressources (surface), dans ce cas une tâche peut rester en logiciel si elle prend trop de place en matériel. On peut aussi tenir compte des communications logicielles/matérielles et regrouper les tâches qui communiquent souvent (ou beaucoup de données ponctuellement) sur une même unité de calcul. Enfin, il est aussi possible de réduire le temps d'exécution total en accélérant les tâches parallélisables sur des unités matérielles.
Dans nos travaux nous avons proposé une version étendue de la méthode à base de liste de Purna and Bhatia [11] car elle est de complexité réduite par rapport à d'autres (nous en avons évaluées plusieurs telles que level-based ou cluster-based).
Bien que les meilleures performances pourraient être obtenues en effectuant le partitionnement et l'ordonnancement en même temps, nous avons séparé ces deux aspects afin de réduire la complexité du flot.

Ensuite l'association et l'ordonnancement sont effectués par un algorithme d'ordonnancement pour architectures hétérogènes. Il traite les supernœuds comme des tâches logicielles, mais uniquement pour la partie reconfigurable. De plus, l'algorithme doit s'assurer de la cohérence et de l'ordre entre la reconfiguration et l'exécution des tâches. Nos travaux s'appuient sur l'algorithme Extended Dynamic Level Scheduling (EDLS) [12] de complexité relativement faible tout en tenant compte des temps de communications inter-processeurs. Cet algorithme calcule une valeur de niveau dynamique pour chaque combinaison nœud/processeur et ordonnance la combinaison avec la plus haute valeur en priorité. Nous avons modifié l'algorithme de sorte que le lorsqu'une unité matérielle est configurée pour une tâche spécifique, le niveau dynamique est ajusté pour que la combinaison correspondante soit sélectionnée. Nous avons aussi modifié l'algorithme afin de tenir compte du fait que plusieurs zones reconfigurables peuvent exister au sein d'un FPGA et partagent donc le même port de configuration : ce dernier est représenté par une unité de calcul spéciale et un nœud correspondant dans le graphe.

Le point de sortie du flot est un ordonnancement qui décrit quelles sont les tâches qui doivent être groupées dans une configuration, sur quelle unité elles doivent s'exécuter et dans quel ordre (la figure montre un exemple simplifié avec un seul FPGA et un seul microprocesseur).

Nous ne cherchons pas les solutions optimales (le temps d'exploration pouvant devenir

prohibitif) afin de maintenir des temps de conception cours : pour ce faire nous utilisons notamment des modèles applicatifs et architecturaux gros grains, en acceptant les limitations inhérentes.
Dans la publication C.33 nous avons entre autres évalué la performance de la méthode d'association et l'avons comparée avec une solution ILP binaire, ceci pour le temps d'exploration des solutions et pour le temps d'exécution des applications. Les résultats, obtenus pour des graphes générés aléatoirement, montrent que dans 90% des cas, la combinaison partitionnement temporel level-based et extended dynamic level scheduling donne les temps d'exécutions les plus courts. De plus, l'approche ILP binaire ne permet d'obtenir une solution que dans 20% des cas, et souffre d'un temps d'exploration prohibitif dans les autres cas.

Dans la thèse de doctorat d'Andreas Popp (voir le chapitre 'Contribution D') nous fournissons une description formelle de ces algorithmes et quantifions leur complexité. Celle de l'algorithme complet est $O(P\ M\ N^3)$ pour des grandes valeurs de M et N, à comparer à celle de l'algorithme itératif de Chatha and Vemuri [13] $O(N^4\ B\ +\ N^3\ B2\ +\ N^3\ B\ M)$ où N est le nombre de nœuds dans le graphe de tâches, M le nombre d'arcs et B le nombre maximum de points solution pour chaque tâche.

De plus, une étude de cas portant sur un égaliseur MMSE pour un système MIMO implanté sur un FPGA Xilinx Virtex-5 illustre certaines des difficultés liées à l'exploitation effective des architectures multiprocesseurs hétérogènes reconfigurables, à savoir le niveau de parallélisme intrinsèque des algorithmes et le temps de reconfiguration. En effet, le coût d'implantation le plus faible est obtenu pour une implantation logicielle (1 seul cœurs Microblaze, pas de reconfiguration) car il a relativement peu de parallélisme à exploiter et l'accélération obtenue pour la version reconfigurable ne justifie pas nécessairement le surcoût en temps de reconfiguration.

8.4 Publications et commentaire

Six publications ont résulté de ces travaux (voir C.29, C.30, C.33, C.35, P.1) ainsi que la thèse de doctorat d'Andreas Popp [14].
C.29 présente une méthode d'estimation automatique du cout d'implantation de graphes de flots de données synchrones sur architectures hétérogènes logicielles/matérielles à partir d'une spécification SystemC-AMS, C.35 donne une vue d'ensemble de l'avancement de ces travaux et P.1 se concentre sur l'aspect partitionnement temporel.
C.30 et C.33 constituent le cœur du travail et j'ai donc choisi de les reproduire ici ; elles détaillent les contributions résumées plus haut. La première C.30 (*Fast Feasibility Estimation of Reconfigurable Architectures*) présente la méthode pour estimer la faisabilité de la reconfiguration sur FPGA. La seconde C.33 (*Scheduling Temporal Partitions in a Multiprocessing Paradigm for Reconfigurable Architectures*) détaille la méthode combinant algorithmes d'ordonnancement multiprocesseurs et de partitionnement temporel. Un complément aux travaux présentés dans C.33 est disponible dans la thèse de doctorat d'Andreas Popp (voir le chapitre 'Contribution D' dans celle-ci) mais n'a pas fait l'objet d'une publication séparée.

Environ six ans après la fin de ces travaux nous pouvons noter que même si il a eu des

améliorations en ce qui concernent les difficultés liées au temps de reconfiguration et au flot de conception, elles existent toujours ; ainsi la reconfiguration partielle/dynamique semble principalement utilisée dans des cas très spécifiques (p. ex. *self-reconfiguration, self-repair*, sécurité). Néanmoins, les systèmes reconfigurables (de manière plus générale) ont toujours le vent en poupe, p. ex. les architectures SoC *many-core* reconfigurables et l'introduction d'éléments reconfigurables dans les systèmes de type calcul haute performance.

Fast Feasibility Estimation of Reconfigurable Architectures

Andreas Popp, Yannick Le Moullec, and Peter Koch Center for Software Defined Radio & Technology Platforms Section,
Department of Electronic Systems, Aalborg University
Fredrik Bajers Vej 7A, 9220 Aalborg Øst, Denmark
{anp,ylm,pk}@es.aau.dk

***Abstract*—Reconfigurable architectures are often said to be able to exploit the possibilities of resource savings by means of hardware time-sharing. However, existing literature does not point clearly at which conditions must be fulfilled for considering a reconfigurable architecture for the implementation of signal processing applications. Therefore, we propose a fast method to perform high-level pre-implementation feasibility-based evaluation of a reconfigurable hardware implementation. The method is based on a light architectural model to compute costs of a static reference as well as costs for globally and partially reconfigurable architectures. Two case studies have been performed for an FFT and an FPGA-based DAB application. Our results show that implementation on reconfigurable architectures is only feasible when the reconfiguration time is low, which generally means that a dynamically partially reconfigurable solution is preferred.**

***Index Terms*—Field programmable gate arrays, reconfigurable architectures, performance evaluation, feasibility**

I. Introduction

Reconfigurable hardware architectures have been introduced as a possibility to provide an intermediate solution between Application Specific Integrated Circuits (ASIC) or Application Specific Instruction-set Processors (ASIP) and Digital Signal Processors (DSP) [1]. Reconfigurable hardware is known to offer the opportunity of resource and energy savings for some applications due to the possibility of time-sharing of the hardware resources, as well as run-time circuit specialization allowing an accelerator that is ultimately customized to the task executing at any given moment of operation.

One of the most utilized reconfigurable architectures is the Field Programmable Gate Array (FPGA), where an example is the Xilinx Virtex series with the improved version of Dynamic Partial Reconfiguration (DPR) [2] in the newest Virtex-4 and 5 series. In DPR, also noted *partial reconfiguration* in the rest of this text, parts of the logic can be reconfigured while maintaining operation on the other parts. The application of the inherent flexibility of DPR has been demonstrated especially in the field of Software Defined Radio (SDR), among others by Delahaye et al. [3] and Ihmig et al. [4] where DPR allows the implementation of several functionalities without having to perform parallel implementations of all functionalities. Furthermore, extensive research efforts in both academia and industry have been put into *i)* synthesis tools and methods to perform scheduling of algorithms onto reconfigurable architectures by e.g. Bobda [5], and *ii)* technical solutions to reduce the reconfiguration overhead suggested by e.g. Hauck [6].

However, even though the use of reconfiguration in FPGA architectures seems promising, it is important for the designer to realize and remember that it is associated with certain costs to provide and use reconfiguration capabilities. Firstly, reconfiguration takes time (known as reconfiguration overhead) and consumes power. Secondly, reconfigurable architectures generally also consume more power, area, and have longer execution time than non-reconfigurable or static solutions. Finally, development time is also longer than for non-reconfigurable architectures, as reconfigurable hardware requires the developer to spend more time on design as well as test and debugging of the implementation.

Shoa & Shirani [8] have given a survey on reconfigurable systems in the context of digital signal processing operations. One conclusion is that FPGA implementation is suitable for data-intensive operations like FIR-filters, FFT and DCT transforms. In traditional FPGA implementations the reconfiguration capabilities are not utilized. However, the inherent lack of flexibility during run-time is a motivation for considering reconfigurable architectures. The survey concludes that reconfigurable architectures should be considered due to possibilities of run-time circuit specialization and logic resource savings by time-sharing among hardware resources.

Although many applications of reconfigurable architectures based on FPGAs have been built, there is, to the best of our knowledge, a lack of clear pointers in the direction of determining when a reconfigurable implementation is feasible. In this paper, feasibility is defined as a non-reduction in performance as compared to a static implementation. Thereby, the study does not include the implementation effort in terms of development costs. The ability to derive a pre-implementation estimate before conducting the final implementation is considered an important basis for deciding whether it is worth the man-hours to perform the implementation in reconfigurable hardware architectures versus non-reconfigurable hardware. Therefore, we have posed the question:

"What high-level characteristics of the application must be fulfilled, and in which conditions is it feasible to make an implementation using reconfigurable hardware?"

This is the author's version of the paper published in
4th IEEE Conference on Industrial Electronics and Applications, ICIEA, 2009

Previous approaches to answer similar questions have mainly been focused on developing a full implementation and comparing it to another implementation in static hardware or programmable processors. Typically, solutions are compared by means of a cost-function or metric that weighs time, silicon area or resource usage, energy or power consumption, and other factors such as numerical properties. Such a cost-metric is used by Wirthlin & Hutchings [7], who provide an estimation method to evaluate the feasibility of a reconfigurable implementation. The evaluation is based on functional density, D, which is a throughput oriented cost metric including area, A, and total operation time, T, and combining these by the expression $D = \frac{1}{A\,T}$. Feasibility is determined from

$$I_{\max} \geq f \quad ,$$

where $I_{\max} = \frac{D_{\text{reconfigurable}}}{D_{\text{static}}} - 1$ is the improvement in functional density over a static implementation and f is the configuration ratio defined as the relation between total time spent on configuration and the total time spent on execution. This means that in case the area A is reduced by a factor of two, a two-fold increase in execution time, T, gives exactly the same functional density, D. This leads to an improvement $I_{\max}$ of 0%, thus if any time is spent on reconfiguration, f will become greater than 0, and a reconfigurable solution is deemed infeasible despite that the static and reconfigurable solution have equal functional density D. Similarly, if execution time is only increased by a factor of 1.5, then as long as 33% or less of the execution time is spent on reconfiguration, the throughput will not be degraded compared to the static reference. While the work evaluates feasibility of reconfigurable architectures, it has two limitations:

- The throughput oriented metric does not reflect the possibility of time-sharing resources and thereby reduction of the area-costs.
- Partial reconfiguration cannot be evaluated, as DPR is not directly reflected in the configuration ratio.

Manet et al. [9] evaluated dynamic partial reconfiguration for non-consumer applications based on selected scenarios where DPR could be advantageous. The evaluation shows that DPR has clear advantages when changes occur in environment or functions (denoted "mission change"). Furthermore, advantages are shown by the use of hardware time-sharing to obtain hardware resource reduction. However, the evaluation of the advantages of DPR is subjective and an objective measure is desired.

A. Contribution

In this work we develop a light architectural model for globally and partially dynamically reconfigurable architectures. The model describes high-abstraction level characteristics of the architecture. The characteristics are considered adjustable to the architecture under consideration. The feasibility estimation method consists of two subsequent steps:

1) **Analysis** of the application from a high level of abstraction to determine execution patterns for the reconfigurable architecture.

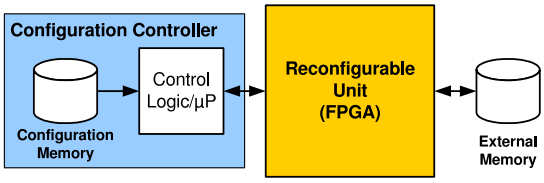


Fig. 1. The basic reconfigurable architecture. The controller can be on-chip or off-chip, but the control resources are separated from the computational resources. The external memory is used to save intermediate data that is used in subsequent configurations.

2) **Logic synthesis** of parts or modules to estimate costs. The costs can also be estimated based on a cost-library for basic functions. The estimates are input to the architectural model to evaluate the feasibility by means of a *cost-function*.

The focus of the cost-function is put on the time and area trade-off that is made possible by time-sharing of resources and not on the flexibility that is provided for applications. Area is counted based on logic resources and the costs of software processors or controllers are not considered.

In the following, the method is presented. This is followed by two case-studies and presentation of the result of these studies. Finally, the results are discussed followed by the conclusion.

II. Method

The method consists of a conveniently light architectural model to describe the characteristics of the architecture. This is followed by a description of the application analysis.

A. Architectural Model

The architectural model is limited to consider a reconfigurable unit, a controller with configuration memory, and external memory as depicted in figure 1.

The proposed architectural model is the basis of two cost models, describing globally reconfigurable architectures and partially reconfigurable architectures. The models describe the capabilities of the architectures from a high-level point of view and capture time and resource parameters. Time costs are categorized on the basis of time spent on execution/computation, reconfiguration, or data transfer. There are many possible area parameters for quantifying resource costs, such as Configurable Logic Block (CLB) and DSP slices, reconfiguration resources, and RAM/memory resources for data, configuration and intermediate data representation. However, in this work area resources are only counted in CLB slices allocated for execution, based on the results from logic synthesis.

The improvement or degradation caused by a reconfigurable implementation is based on a comparison to a static implementation. The static costs, C_{static}, are expressed by

$$C_{\text{static}} = A_{\text{static}} \cdot T_{\text{static}} \quad [\text{s} \cdot \text{slices}] \quad , \qquad (1)$$

where A_{static} is the total area in CLB slices of the architecture, and T_{static} is the total execution time in seconds. The area

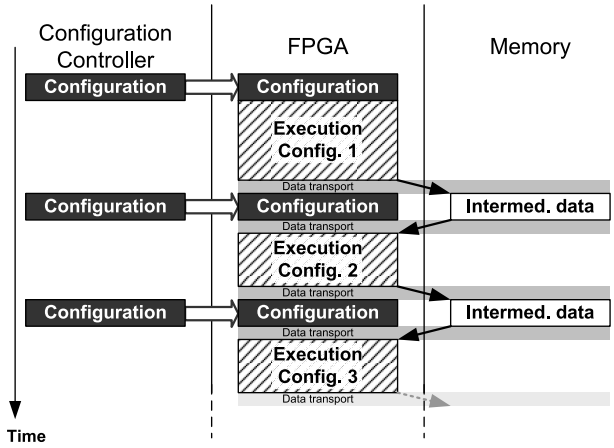


Fig. 2. Execution flow in global reconfiguration.

is inherently two-dimensional thus the costs are conveniently described in three dimensions.

In our proposed cost model, time and area are given equal weight in order to fully reflect the area-time trade-off in time-sharing of resources. In case certain area or time constraints must be fulfilled, these constraints are evaluated externally to the cost evaluation.

B. Dynamic Global Reconfiguration

In the case of dynamic global reconfiguration, it is assumed that reconfiguration and execution cannot be overlapped, which is a general assumption for globally reconfigurable FPGAs. The execution flow is illustrated in figure 2 and proceeds as follows: First, the controller configures the FPGA. Then the FPGA executes the tasks of configuration 1 and stores intermediate data in the external memory. Then configuration 2 is programmed into the FPGA, followed by reading the intermediate data from memory. This process repeats itself until all configurations have been executed.

Time costs can easily be described by the sum in (2) that describes time-consuming parts of execution in a globally reconfigurable system:

$$
\begin{aligned}
T_{\mathrm{exec}} &= \sum_i t_{\mathrm{exec},i} \\
T_{\mathrm{reconf}} &= \sum_i t_{\mathrm{reconf},i} \\
T_{\mathrm{transfer}} &= \sum_i t_{\mathrm{read},i} + \sum_i t_{\mathrm{write},i} \\
T_{\mathrm{global}} &= T_{\mathrm{exec}} + T_{\mathrm{reconf}} + T_{\mathrm{transfer}} \quad , \qquad (2)
\end{aligned}
$$

where the symbols are defined as in table I.

The total cost of the globally reconfigurable solution is given by multiplying equation (2) by the area in CLB slices, A_{global}, of the globally reconfigurable architecture:

$$
C_{\mathrm{global}} = A_{\mathrm{global}} \cdot T_{\mathrm{global}} \quad [\mathrm{s} \cdot \mathrm{slices}] \quad , \qquad (3)
$$

which can then be compared to C_{static}, (1).

TABLE I
DEFINITION OF SYMBOLS IN (2).

I	Total number of configurations
i	Configuration index, $i \in \{1, 2, \ldots, I\}$
T_{global}	Total time spent in the global reconfiguration scenario [s]
T_{exec}	Total time spent on execution [s]
T_{reconf}	Total time spent on reconfiguration [s]
T_{transfer}	Total time spent on data transfer [s]
$t_{\mathrm{exec},i}$	Execution time of configuration i [s]
$t_{\mathrm{reconf},i}$	Reconfiguration time of configuration i [s]
$t_{\mathrm{read},i}$	Memory read-time for input to configuration i [s]
$t_{\mathrm{write},i}$	Memory write-time for output from configuration i [s]

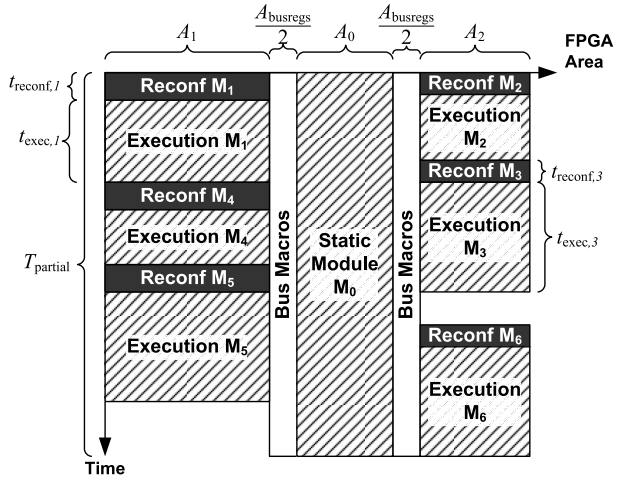


Fig. 3. Execution flow in dynamic partial reconfiguration.

C. Dynamic Partial Reconfiguration

The partial reconfiguration model is basically similar to the model for global reconfiguration. However, instead of multiplying the time and total area for global configurations, the sum of reconfiguration and execution time is multiplied by the resources consumed by each reconfigurable module.

The model assumes that transfer of data between modules is performed by special bus registers, so called Bus Macros in Xilinx tool flows [2], and the transfer delay across Bus Macros is assumed negligible. However, the bus registers consume area during the whole operation. Furthermore, the placement of bus macros is assumed fixed during operation, as this is similar to current DPR implementation in Xilinx FPGAs [2]. The execution flow is illustrated in figure 3. The figure has one static module, M_0, that is active during the whole execution T_{partial}. There are two bus macros that handle communication of data between the reconfigurable modules and the static module. The configuration of the static module and the bus macros is not included in the costs, as it is assumed being a part of the general start-up of the FPGA. The six static modules M_1-M_6 are reconfigured prior to their execution. As indicated in the figure, there are periods where some of the resources are unused for execution. This is not included in the costs, as the area is theoretically available for other functionalities.

TABLE II
DEFINITION OF SYMBOLS IN (4), ALSO ILLUSTRATED IN FIGURE 3.

j	Module index
A_j	Area of module j [slices]
A_{busregs}	Area of the bus registers [slices]
$t_{\text{exec},j}$	Execution time of module j [s]
$t_{\text{reconf},j}$	Reconfiguration time for loading module j [s]
T_{partial}	Total execution time [s]
$C_{\text{partial,proc}}$	Cost of processing and reconfig. in DPR [s·slices]
$C_{\text{partial,comm}}$	Cost of communication in DPR [s·slices]
C_{partial}	Total cost of dynamic partial reconfiguration [s·slices]

The total cost of a partially reconfigurable implementation, C_{partial}, is expressed by

$$C_{\text{partial}} = C_{\text{partial,proc}} + C_{\text{partial,comm}} \quad [\text{s} \cdot \text{slices}] \quad (4)$$
$$C_{\text{partial,proc}} = \sum_j A_j \cdot t_{\text{exec},j} + \sum_j A_j \cdot t_{\text{reconf},j}$$
$$C_{\text{partial,comm}} = A_{\text{busregs}} \cdot T_{\text{partial}} \quad ,$$

where the symbols are defined as in table II. C_{partial} can be compared to C_{static}, (1), as well as C_{global}, (3).

D. Application Analysis and Logic Synthesis

The application analysis is performed by an examination of the application to demonstrate how to extract the parameters of the architecture model described in the previous section.

From a high level of abstraction the application and specifications are analyzed to determine the deadline, T_{deadline}, at which the task-set, (i.e. all operations), must be finished. The task-set can either be a one-time running application or periodic tasks. For periodic tasks, the deadline is equal to the longest period of the tasks. Since the static reference occupies all resources from execution start to deadline, T_{static} is set to T_{deadline}.

In the second part of the analysis, the application is examined to determine whether it can be divided into configurations that can be executed sequentially thus suitable for global reconfiguration. It may, however, be that it is judged more suitable for partial reconfiguration, and tasks are then grouped or defined by modules. The process can either be performed manually, or by an automated scheduling approach including temporal partitioning and placement similar to Bobda [5]. The latter does however, require knowledge or estimates of execution time and resource usage. Those estimates have to be acquired by logic synthesis, as described in the next paragraph.

The determined configurations or modules are provided as input to a synthesis program to obtain estimates of execution time and area consumption. The reconfiguration time, $t_{\text{reconf},i}$, is estimated by dividing the bitstream size estimate by the speed of the configuration interface (up to 100 MHz using the Xilinx Virtex-4 SelectMAP interface [2]) as in

$$t_{\text{reconf},i} = \frac{W + 1312}{100} \quad [\mu\text{s}] \quad , \quad (5)$$

where W is the configuration array size of 147600, 726520, and 426810 for the LX15, LX80, and SX35 Virtex-4 FPGAs respectively [11]. In a similar manner, the data transferred between the configurations are quantified and divided by the read/write speeds of the external memory.

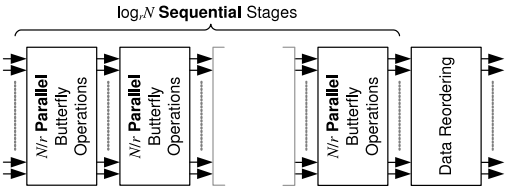


Fig. 4. Organization of an FFT.

Finally, the costs are calculated, and the use of reconfigurable architectures is deemed feasible if the conditions (6) and (7) are satisfied. The left hand side arguments in curly braces indicate that only one argument is considered at a time; This is determined by the selection of global or partial reconfiguration:

$$\{C_{\text{global}}, C_{\text{partial}}\} \leq C_{\text{static}} \quad \text{AND} \quad (6)$$
$$\{T_{\text{global}}, T_{\text{partial}}\} \leq T_{\text{deadline}} \; , \quad (7)$$

which ensures that the total cost is lower than or similar to the static implementation, and that the deadline is fulfilled.

In case T_{global} or T_{partial} are lower than T_{deadline}, it may be considered to utilize reconfiguration capabilities even further i.e. trade off execution time for area reduction, or select an architecture with a lower clock speed as idle resources are available.

III. CASE STUDIES

The previous sections described our proposed architecture model and how to do the application analysis. This is demonstrated by two case-studies in this section. The first study considers global reconfiguration, whereas the second study considers both global and partial reconfiguration.

A. Fast Fourier Transform

The Fast Fourier Transform (FFT) algorithm is widely used in multimedia applications and communications systems. In the latter case it is known as an efficient implementation of orthogonal frequency-division multiplexing (OFDM). An FFT is composed of parallel butterfly operation blocks that are executed sequentially followed by data reordering as illustrated in figure 4. N is the number of points in the FFT and r is the radix of the butterfly operations. The computation consists of $\log_r N$ sequential blocks of $\frac{N}{r}$ parallel radix-r butterfly operations.

The case is selected to be a 32 point radix-2 FFT ($N = 32$ and $r = 2$) operating at 16 bit resolution. The static reference is a fully parallel implementation with constant twiddle-factor multipliers synthesized for a Virtex-4LX80 FPGA with 35840 CLB slices [10] executing an FFT-operation at a rate selected to be 1 kHz. The reordering of data at the output is not considered for the static reference.

TABLE III
DETAILS OF CONFIGURATIONS FOR THE FPGA-BASED DAB RECEIVER [4]:

Configuration i	$t_{\text{exec},i}$ [ms]	Content
0	2.26	Mixer, FIR filter, and fine frequency offset correction
1	1.14	Fast Fourier Transform
2	0.48	Coarse freq. offset correction, demodulator, frequency and time deinterleaving
3	0.11	Viterbi decoding and energy dispersion

The alternative implementation is a globally reconfigurable solution at which each stage is implemented as a full configuration, theoretically making it possible to reduce the hardware area by a factor of $\log_r N = 5$. Reconfiguration time is estimated by (5) for a Virtex-4LX15 FPGA containing totally 6144 CLB slices [10], as the number of CLB-slices in this FPGA is close to 5 times smaller than the Virtex4-LX80.

The memory read and write times are estimated based on SDRAM memory running at 266 MHz, by using the expression

$$t_{read} = t_{write} = \frac{n_{\text{bytes}}}{4\text{bytes} \times 266\text{MHz}} + \frac{3}{266\text{MHz}} \quad , \quad (8)$$

where the last part of the sum is based on the latency of the memory. In this case study, the transferred amount of data, n_{bytes}, was 128 Bytes.

The static and globally reconfigurable implementations were synthesized in Xilinx ISE 9.1 based on VHDL code to obtain the necessary estimates.

B. FPGA-based DAB Receiver

The second case is based on a study of the results by Ihmig et al. [4]. The work consists of a digital audio broadcasting (DAB) receiver that is investigated for combining the tasks in a sequential execution on a Xilinx Virtex-4 SX 35 FPGA. The reference is a pipelined architecture consisting of 10 stages running at multiple rates, and is characterized by having a very relaxed latency requirement. The authors investigate a solution where the 10 stages are partitioned into four configurations, listed in table III, that are executed sequentially at a higher clock frequency (100 MHz) than the pipelined architecture (8.2 MHz).

The buffered sequential implementation is assumed based on a 50 Hz cycle, which determines the time, T_{static}, of the static implementation. In their work, the read/write time for external memory is not listed, and is therefore estimated as in (8). The transferred amount of data between configurations, n_{bytes}, is conservatively assumed based on the maximum data rate of 8192 kbytes/s, which gives 8192/50 kbytes between each configuration.

The static and global area were both determined by the size of the FPGA to 15360 CLB slices [10] and the reconfiguration time was estimated as described in (5).

The above referenced work also considers partial reconfiguration, where the four configurations are set to the size of the largest configuration of 2048 CLB slices. Reconfiguration time was given to be 750 μs, and $C_{\text{partial,comm}}$ was estimated by multiplying the memory controller area (668) by the total period of 20 ms.

TABLE IV
RESULTS: FAST FOURIER TRANSFORM:

Static Implementation (Virtex-4LX80)		
Time	$T_{\text{static}} = T_{\text{deadline}}$	1 ms
Area	$\{A_{\text{static}}$,synthesis$\}$	{35840, 24516} slices
Cost	C_{static}	35.8 s·slices
Globally Reconfigurable Implementation (Virtex-4LX15)		
Time	$\{T_{\text{exec}}, T_{\text{reconf}}, T_{\text{transfer}}\}$	{27.0E-6, 7.4, 1.32E-3} ms
Area	$\{A_{\text{global}}$,synthesis$\}$	{6144, 5815} slices
Cost	C_{global}	45.8 s·slices

TABLE V
RESULTS: FPGA-BASED DAB RECEIVER:

Static Implementation (Virtex-4SX35)		
Time	$T_{\text{static}} = T_{\text{deadline}}$	20 ms
Area	A_{static}	15360 slices
Cost	C_{static}	307 s·slices
Globally Reconfigurable Implementation (Virtex-4SX35)		
Time	$\{T_{\text{exec}}, T_{\text{reconf}}, T_{\text{transfer}}\}$	{4.4, 17.1, 0.63} ms
Area	A_{global}	15360 slices
Cost	C_{global}	340 s·slices
Partially Reconfigurable Implementation (Virtex-4SX35)		
Time	$t_{\text{reconf},0}, \ldots, t_{\text{reconf},3}$	750 μs
	$\{t_{\text{exec},0}, \ldots, t_{\text{exec},3}\}$	{2.26,1.14,0.48,0.11} ms
	T_{partial}	20 ms
Area	$A_0, \ldots, A_3$	2048 slices
	A_{busregs}	668 slices
Cost	$C_{\text{partial,proc}}$	15.1 s·slices
	$C_{\text{partial,comm}}$	13.4 s·slices
	C_{partial}	28.4 s·slices

IV. RESULTS

The results were obtained as described in section III. The results from the FFT case study are shown in table IV. A_{static} and A_{global} are the actual values for the FPGAs [10], whereas "synthesis" is the synthesis result obtained by ISE 9.1. In addition to CLB slices, DSP48 resources were also utilized. However, these are not included in the cost-model, thus not showed in the table.

From the synthesis results, it is clear that one FFT-stage only consumes 16% of the FPGA's resources in the full static implementation. However, due to the high reconfiguration overhead, the costs and time are higher, 28% and 540% respectively, than for the static reference.

For the second case of the DAB receiver, the results are shown in table V. The results are a combination of extracts from [4] and the estimates described in section III-B.

V. DISCUSSION

For the investigated FFT-case, the results clearly showed that a globally reconfigurable implementation had significantly higher costs than a static implementation, in spite of the possibility of HW sharing. The cost can be reduced by packing

more operations into each configuration, and thereby reduce the number of reconfigurations. However, this will not make the reconfigurable solution feasible for this case, as (6) and (7) still cannot be fulfilled. It can be argued that the investigated fully parallel FFT implementation is not a realistic reference and is inefficiently implemented in the FPGA. However, we find that the suggested scenario describes the problem of feasibility estimation for block-processing applications in an illustrative and easily understandable way.

For the investigated DAB-receiver case, the global reconfiguration did not fulfill the conditions (6) and (7), and it was thereby concluded that a globally reconfigurable implementation is not feasible compared to a static solution. This is mainly caused by the long time spent on reconfiguration as shown in table V. However, the reconfiguration time can be decreased by selecting a smaller FPGA - thus reducing the cost of the globally reconfigurable implementation.

The partially reconfigurable solution for the DAB-case did show a significant reduction in cost and only 9.3% of the resources were utilized. The rest of the resources can either be utilized for other functionalities or a smaller FPGA can be selected. The feasibility conditions (6) and (7) were fulfilled, so a partially reconfigurable implementation is feasible for this application.

The investigated cases show that a reconfigurable implementation may be feasible and may satisfy the time-constraints either due to a very relaxed deadline, or by running the reconfigurable architecture at a higher clock-speed than the non-reconfigurable implementation. Increasing the clock-speed leads to an increased power-consumption, thus we suggest extensive evaluation of power-consumption for future work.

An advantage of the methodology is that it is relatively simple to obtain the estimates and set up the feasibility conditions. However, it requires that the designer performs the partitioning of the application into configurations or modules and performs logic synthesis of these configuration or modules. The partitioning of the application can be performed by an automatic scheduling approach as suggested in section II-D.

So far, our methodology does only consider the CLB slices, but other conditions are currently being investigated for memory blocks and DSP slices.

VI. Conclusion

In this work we propose a method to evaluate the feasibility of implementing signal processing applications in reconfigurable architectures.

A general condition for feasibility of a globally reconfigurable architecture is closely related to the reconfiguration time and thus the size of the reconfigurable area. The size must be carefully selected so that the reconfiguration time does not exceed the execution time of the static configuration reference. However, as the reconfiguration time is potentially significantly smaller for partially reconfigurable implementations than for globally reconfigurable implementations it is generally preferable to choose a partially reconfigurable solution.

We conclude that the proposed cost-metric makes it possible to evaluate the feasibility considering area-usage and timing. An observation is that timing constraints may be fulfilled by adjusting the clock-speed, thus consideration of power-consumption in the cost-metric is suggested as future work.

References

[1] R. Hartenstein, "Trends in reconfigurable logic and reconfigurable computing," *9th International Conference on Electronics, Circuits and Systems*, vol. 2, September 2002, pp. 801–808.

[2] P. Lysaght et al., "Enhanced architectures, design methodologies and cad tools for dynamic reconfiguration of xilinx fpgas," *International Conference on Field Programmable Logic and Applications*, 2006.

[3] J. P. Delahaye et al., "Software radio and dynamic reconfiguration on a dsp/fpga platform," *3rd Karlsruhe Workshop on Software Radios*, 2004.

[4] M. Ihmig et al., "Resource-efficient sequential architecture for fpga-based dab receiver," *5th Karlsruhe Workshop on Software Radios*, 2008.

[5] C. Bobda, "Synthesis of dataflow graphs for reconfigurable systems using temporal partitioning and temporal placement," Doctor's dissertation, Faculty of Computer Science, Electrical Engineering and Mathematics of the University of Paderborn, May 2003.

[6] S. Hauck, "Configuration prefetch for single context reconfigurable coprocessors," *ACM/SIGDA sixth international symposium on Field programmable gate arrays*, pp. 65–74, 1998.

[7] M. J. Wirthlin and B. L. Hutchings, "Improving functional density using run-time circuit reconfiguration," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 6, no. 2, pp. 247–256, June 1998.

[8] A. Shoa and S. Shirani, "Run-time reconfigurable systems for digital signal processing applications: a survey," *Journal of VLSI Signal Processing Systems*, vol. 39, no. 3, pp. 213–235, 2005.

[9] P. Manet et al., "Evaluation of dynamic partial reconfiguration in professional electronics applications," *DASIP Workshop on Design and Architectures for Signal and Image Processing*, November 2007.

[10] Xilinx Inc., "Virtex-4 FPGA User Guide," *UG070*, June 2008.

[11] Xilinx Inc., "Virtex-4 FPGA Configuration User Guide," *UG071*, April 2008.

Scheduling Temporal Partitions in a Multiprocessing Paradigm for Reconfigurable Architectures

Andreas Popp, Yannick Le Moullec, and Peter Koch
Center for Software Defined Radio & Technology Platforms Section,
Department of Electronic Systems, Aalborg University
Aalborg, Denmark
{anp,ylm,pk}@es.aau.dk

***Abstract*—In this paper we describe a mapping methodology for heterogeneous reconfigurable architectures consisting of one or more SW processors and one or more reconfigurable units, FPGAs. The mapping methodology consists of a separated track for a) the generation of the configurations for the FPGA by level-based and clustering-based temporal partitioning, and b) the scheduling of those configurations as well as the software tasks, based on two multiprocessor scheduling algorithms: a simple list-based scheduler and the more complex extended dynamic level scheduling algorithm. The mapping methodology is benchmarked by means of randomly created task graphs on an architecture of one SW processor and one FPGA. The results are compared to a 0-1 integer linear programming solution in terms of exploration time as well as the finish-time of all tasks of the application. The results show that, in 90% of the investigated cases, the combination of level-based temporal partitioning and extended dynamic level scheduling gives the best performance in terms of finish-time of the full task-set.**

***Keywords*-Reconfigurable Hardware; Heterogeneous Reconfigurable Architectures; Temporal Partitioning; Multiprocessor Scheduling**

I. Introduction

Most signal processing architectures are both reconfigurable and heterogeneous, consisting of several software processors as well as configurable hardware, typically Field-Programmable Gate Arrays (FPGAs). Moreover, FPGAs provide reconfiguration during runtime, either for the full FPGA area - or for a portion of the area, noted Dynamic Partial Reconfiguration (DPR). Such systems have the possibility to provide better performance than compile-time configured systems in terms of total execution time, logic resource usage, and power consumption [1]. However, in order to obtain such performance benefits, it is necessary to have efficient scheduling techniques and methods which we denote "mapping methods" in the following.

Existing solutions for mapping applications to reconfigurable heterogeneous architectures target architectures consisting of a software processor connected to a reconfigurable FPGA via a common bus. The software processor serves as the host, either being 1) a simple configuration controller for the reconfigurable hardware, or 2) a processor that utilizes the reconfigurable hardware for acceleration of computationally heavy tasks.

In case 1 where the processor works solely as a configuration controller, approaches for temporal partitioning have been suggested by, among others, Kaul&Vemuri [2] and Purna&Bhatia [3]. Temporal partitioning is the task of dividing a large application into partitions that are mutually exclusive in time, and thus can be executed sequentially on a device that is smaller than needed for fully parallel implementation of the entire application.

In case 2, approaches have been suggested by, among others, Banerjee et al. [4] that formulated the solution as a 0-1 Integer Linear Programming (ILP) problem to obtain the minimum cost in terms of overall execution time. Noguera&Badia [5] proposed a HW/SW partitioning algorithm where tasks are moved between HW and SW until a minimum overall execution time is obtained. The method considers prefetching of configurations to reduce the reconfiguration overhead. The computational complexity of both works is high (non-polynomial for the first), leading to prohibitively long execution times of exploration algorithms, which we from hereon will denote "exploration times". An approach with lower computational complexity has been proposed by Chatha&Vemuri [6]. The work consists of an algorithm of five steps: a) HW/SW partitioning, b) temporal partitioning of HW tasks, c) scheduling of HW and SW tasks, d) scheduling of HW reconfigurations, and e) scheduling of communications.

However, as these three approaches do cover a subset of heterogeneous reconfigurable architectures, they are not suited for architectures consisting of several units, both in HW and SW.

Mapping methods for homogeneous SW architectures have been well studied for some time. One of the well known methods is Dynamic Level Scheduling (DLS) by Sih&Lee [7], who in the same connection propose an extended DLS algorithm for heterogeneous architectures.

The previously mentioned approaches do not cover heterogeneous reconfigurable architectures consisting of several processing units, thus in this work we combine the known temporal partitioning algorithms with multiprocessor scheduling algorithms in a scheduling methodology for heterogeneous reconfigurable architectures. The methodology is inspired by Chatha&Vemuri [6] that starts with an initial HW/SW partitioning followed by creation of temporal partitions for HW nodes. The temporal partitions are then treated as super-nodes in a multiprocessing framework - where the super-nodes are tied to a particular unit, the reconfigurable HW unit.

This is the author's version of the paper published in
NASA/ESA Conference on Adaptive Hardware and Systems, 2009

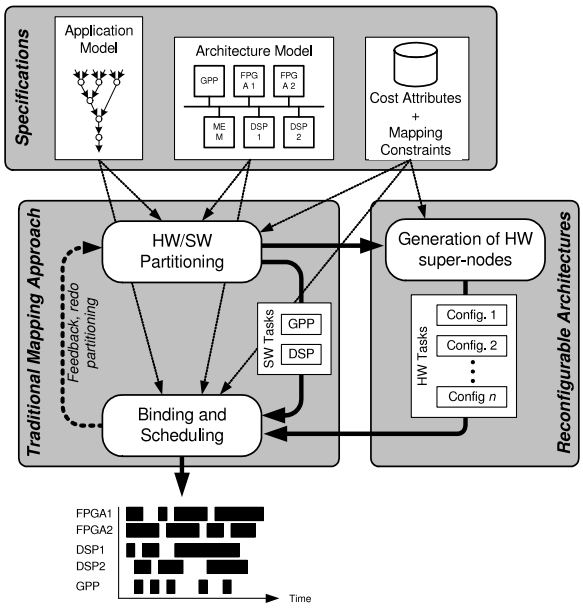


Fig. 1. The proposed mapping methodology. The first step is the specification of the application, architecture, and their interrelation via a cost-library. This is followed by a partitioning between HW and SW tasks. The HW tasks are sent to the HW-flow, where the tasks are partitioned into temporal partitions of HW tasks. The HW tasks and their reconfiguration are each considered super-nodes of tasks, which are fed to the multiprocessor binding and scheduling process.

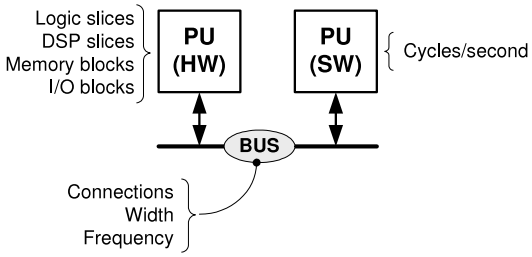


Fig. 2. General Architecture Model. The attributes for each architecture element is found by studying the data sheets of the architecture.

This paper describes the suggested methodology in section II, including the underlying application and architecture model. This is followed by a series of experiments in section III where the mapping results are compared to a 0-1 ILP solution that serves as a lower boundary reference. The results are presented in section IV, followed by a discussion and a conclusion in section V and VI, respectively.

II. Mapping Methodology

The proposed mapping methodology is a combination of multiprocessor scheduling and temporal partitioning for reconfigurable architectures, and is outlined in figure 1. The starting point is the specifications of the application, architecture, and cost-library which are all expanded in section II-A. Following the specification, the application's tasks are partitioned between HW and SW units, and between several HW units. This is fed back to the original SW multiprocessor scheduling flow, as described in section II-D.

A. Specifications and Modeling

The application is specified as a directed acyclic task-graph, consisting of nodes and edges. The nodes represent tasks, whereas the edges represent data dependencies. The edges are assigned a width, describing the amount of data transferred between the nodes. The task granularity can vary, being both single algorithmic operations as well as larger blocks of operations. The general architecture model is illustrated in figure 2. The model is a composition of Processing Units (PUs), memories, and ports, all connected via buses. PUs are again either SW or HW. SW units have a certain number of cycles per second, whereas HW has a number of resources, each corresponding to the number of logic slices, DSP resources, memory blocks etc. Buses are described by the units they connect, their direction, width, and frequency.

The cost-library binds the application and the architecture together. It contains the cost of various implementation alternatives for each task, i.e. execution time for SW and execution time, reconfiguration time, and resource usage for HW. Reconfiguration time is derived from the size of the reconfigurable HW. The cost-library is derived by sample implementations of each task, without having to perform the full implementation of the application. Another option is to provide estimates based on previous experiences.

B. Partitioning

The partitioning approach is based on the values of the cost-library: t_i^{sw} is the SW execution time of task i, t_i^{hw} is the hardware execution time of task i, and $t_{\mathrm{reconf}}^{\mathrm{hw}}$ is the full reconfiguration time of the HW unit. The HW/SW partitioning is based on the principles described in the list below:

1) If logic slice resource usage for task i is larger than the capacity of the HW unit, then partition to **SW**
2) Else If $t_{\mathrm{reconf}}^{\mathrm{hw}} + t_i^{\mathrm{hw}} < t_i^{\mathrm{sw}}$, then partition to **HW**
3) Else $t_{\mathrm{reconf}}^{\mathrm{hw}} + t_i^{\mathrm{hw}} \geq t_i^{\mathrm{sw}}$ is true, so partition to **SW**

As seen in the partitioning scheme, reconfiguration time is included in the HW execution time, assuming that each HW execution must be preceded by reconfiguration.

C. HW Flow

The partitioning is followed by an extraction of the HW tasks from the application graph.

The task-set is then temporally partitioned, following the two list-scheduling temporal partitioning algorithms by Purna&Bhatia [3]. However, it is a requirement to the execution scheme that temporal partitions do not start execution before all inputs are ready. Thus, there must not be a path through other nodes or partitions from an output to an input in the same partition. Therefore, the temporal partitioning algorithms are extended with a search for paths outside the current partition. If such a path exists, a new partition is created, and the current node is placed in that new partition.

The result of the HW flow is fed to the binding and scheduling by performing an application graph and cost-table update. In the application graph update, the temporal partitions are considered as HW super-nodes, and are fed to the SW flow

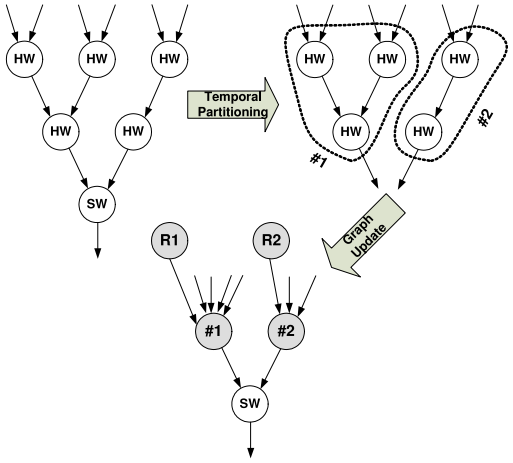


Fig. 3. Illustration of the application graph update. Firstly, HW nodes are temporally partitioned. Secondly, nodes in temporal partitions are replaced by super-nodes, followed by insertion of reconfiguration nodes for each super-node.

as super nodes. The cost-table entries for the HW tasks are removed and replaced by cost-table entries for the super-nodes.

The application graph and cost-table update follows the scheme as described below and refers to the illustration of the application graph update in figure 3:

1) All nodes in the same temporal partition are replaced by a single super-node (**#1** and **#2** in figure 3). This is performed for all temporal partitions. All edges going to/from those nodes are being redirected to the corresponding super-nodes, preserving the direction of the edge.
2) Reconfiguration nodes (**R1** and **R2**) are added to all the new super-nodes. The reconfiguration nodes have no predecessors, and their only successor will be the corresponding super-node.
3) The cost-table is updated by firstly removing the entries for the nodes that are replaced by super-nodes. Secondly, entries are added for each super-node. The execution time is the maximum execution time of the tasks in the super-node. The resource cost is the sum of all tasks in the super-node.
4) Similarly, entries are added for the reconfiguration nodes. The execution is similar to the reconfiguration time of the unit, and the resource cost is similar to the super-node that is reconfigured.

D. SW Flow

The SW scheduling flow is based on two approaches:

1) A simple list-scheduler where nodes are scheduled in the order given by the finish-time of their predecessor as well as their mobility, such that the node with the lowest mobility is scheduled first.
2) The extended DLS algorithm by Sih&Lee [7] for heterogeneous processor systems.

For both approaches additional constraints have been included in order to ensure that reconfiguration and execution sequences are performed in the right order, without interruption by other tasks. The two approaches have been implemented in order to be able to compare two SW scheduling algorithms, thus they are both used for scheduling.

For both algorithms, we use a light communication model based on communication time. Communication time between tasks executed in the same unit is assumed to be zero. The transfer of data over the connecting bus is associated with a certain communication time based the the amount of data transferred, the bus width, and the bus frequency.

The extended DLS algorithm has been selected due to its ability to handle heterogeneous multiprocessing architectures consisting of several HW and SW units taking interprocessor communication costs into account. Heterogeneity is represented by varying execution times of tasks, which are included in the Dynamic Level (DL) computation. If a task-processor combination is invalid, its execution time is infinity, leading to a DL of minus infinity. This prevents that combination from being selected. The state of the communication resources are modeled as occupied slots of communication. The state is included in two steps of the algorithm:

- **DL computation:** If the communication resource is free to provide communication from the predecessor to the current node, the communication time is assumed to take place right after finishing the predecessor, else the communication is moved to the next free communication slot. Both possibilities influence the Data Available (DA) time, thus the computation of DL.
- **Scheduling:** When the node with the highest DL is scheduled, it is performed based on the calculated start time in the previous step. This is followed by an update of the state of the communication resources.

TABLE I
DESCRIPTION OF TASK-GRAPHS FOR THE EXPERIMENTS. CP DENOTES THE LENGTH OF THE CRITICAL PATH IN TERMS ON NUMBER OF NODES.

Experiment	Tasks	Edges/Task	CP [nodes]
1	5	1.2	3
2	5	1	4
3	10	1.6	3
4	10	0.8	4
5	10	1.2	5
6	10	1.8	5
7	15	0.8	5
8	15	1	8
9	15	1.2	6
10	15	1.53	6

III. MAPPING EXPERIMENTS

Several mapping experiments have been performed during the development of the framework, they are explained in this section. The experiments were performed as a series of mapping experiments for various task-graphs. The task-graphs had the number of nodes $\{5, 10, 15\}$, with varying numbers of edges and length of the Critical Path (CP). The graphs are described in table I. All graphs have only a single sink node.

TABLE II
ALGORITHM OPTIONS FOR THE MAPPING EXPERIMENTS

No	Temporal Partitioning	Multiprocessor Scheduling
1	Level-based	Simple list-based
2	Clustering-based	Simple list-based
3	Level-based	Extended DLS
4	Clustering-based	Extended DLS
5	0-1 ILP-based Optimal Reference	

The architecture for all experiments was the same, a HW/SW architecture consisting of one SW processor and one HW unit. The HW unit had 15 logic slices, and the reconfiguration time was 10 cycles. Reconfiguration was assumed not to overlap with HW execution, but has no influence on the SW execution. We assumed a constant transfer time of two cycles between the SW and HW units. This transfer was assumed not to interrupt HW nor SW execution.

The SW and HW execution times as well as the HW-cost were randomly created to each task, based on random distributions in the given intervals.:

- SW execution time: $[1; 20]$
- HW execution time: $[1; 10]$
- HW Cost: $[1; 15]$

The experiments were performed for four combinations of our mapping framework as well as the optimal 0-1 ILP reference as indicated in table II. The ILP problem formulation is outlined in the next section III-A. The results were compared in terms of makespan (defined as the total execution time of the task-set) and the exploration time (defined as the execution time of the exploration algorithm). The mapping framework was executed in Matlab® on a standard PC.

A. ILP Formulation of Optimal Mapping

The optimal mapping reference is performed by an 0-1 ILP formulation of the problem. The formulation is a light version of the work by Banerjee et al. [4] and is described below. The major difference between their work and our work is that we only consider the area and have disregarded HW placement constraints that Banerjee et al. use to make sure that tasks that span several columns are placed in consecutive columns. Furthermore, we have added the precedence constraint for reconfiguration in equation (4), such that a HW area is reconfigured before its tasks are executed. The formulation of the problem allows partial reconfiguration, thus potentially a lower makespan than for the global reconfiguration case. First some binary variables are described, indexed by i as the task-index, $i \in \{0, \ldots, n_{\text{tasks}} - 1\}$, and j as the time-step, $j \in \{0, \ldots, n_{\text{timesteps}} - 1\}$. The variables are:

- $x_{i,j}$ is 1 if task T_i starts execution in HW at timestep j, 0 otherwise.
- $y_{i,j}$ is 1 if task T_i starts execution on the SW processor at timestep j, 0 otherwise.
- $r_{i,j}$ is 1 if the reconfiguration for task T_i starts execution at timestep j, 0 otherwise.
- in_{i_1,i_2} is 1 if the communication along the edge between task T_{i_1} and T_{i_2} incurs a communication delay, 0 otherwise.

Furthermore, the costs are given by the symbols:

- t_i^{sw} is the SW execution time of task T_i.
- t_i^{hw} is the HW execution time of task T_i.
- c_i^{hw} is the HW resource cost of task T_i.
- $t_{\text{reconf}}^{\text{hw}}$ is the time is takes to reconfigure the HW.
- C_{FPGA} is the full logic capacity in terms of CLB logic slices of the FPGA.
- ct_{i_1,i_2} is bus data transfer time from task T_{i_1} to T_{i_2}.

The variables are subject to a series of constraints:

1) Uniqueness Constraint: Every task executes only once:

$$\forall i, \sum_j (x_{i,j} + y_{i,j}) = 1 \tag{1}$$

2) SW Processing Constraint: At each time, at most one task is executing on the SW processor:

$$\forall j, \sum_i \sum_{m=j-t_i^{\text{sw}}+1}^{j} y_{i,m} \leq 1 \quad , \tag{2}$$

where the sum over m is performed to include $y_{i,m}$ over all time-steps where a SW task can occupy the SW processor.

3) Reconfiguration Constraint: For each task, there is at most one configuration, expressed as mutual exclusiveness of SW execution and reconfiguration:

$$\forall i, \sum_j (y_{i,j} + r_{i,j}) \leq 1 \tag{3}$$

Furthermore, if the task is performed in HW, reconfiguration must precede execution:

$$\forall i, \sum_j j \cdot r_{i,j} + \sum_j t_{\text{reconf}}^{\text{hw}} \cdot r_{i,j} - \sum_j j \cdot x_{i,j} \leq 0 \tag{4}$$

4) FPGA Resource Constraint: For the FPGA, the sum of resources used for execution or reconfiguration at any timestep must not exceed the full size of the FPGA. A sum over m is included similarly to (2):

$$\forall j, \sum_i \Big(\sum_{m=j-t_i^{\text{hw}}+1}^{j} c_{\text{i}}^{\text{hw}} \cdot x_{i,m} + \sum_{m=j-t_{\text{reconf}}^{\text{hw}}+1}^{j} c_{\text{i}}^{\text{hw}} \cdot r_{i,m} \Big) \leq C_{\text{FPGA}} \tag{5}$$

5) Communication Constraint: Communication on the bus should only be performed when tasks connected by edges are performed on different units:

$$\forall \text{edges}(i_1, i_2), \sum_j y_{i_1,j} + y_{i_2,j} + in_{i_1,i_2} = \{0, 1\} \tag{6}$$

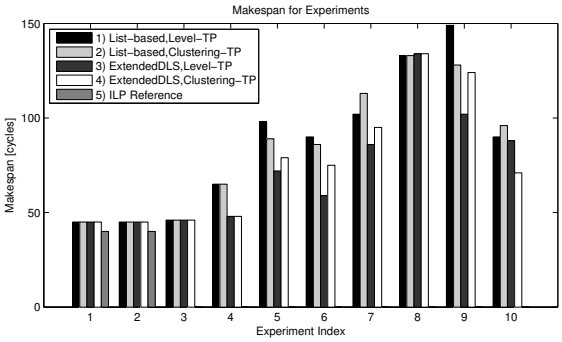


Fig. 4. Results in terms of makespan. The bars 1-4 for each experiment are for the proposed framework. Bar 5 is an adapted version of [4]. In 90% of the cases, the Extended DLS algorithm gave better or equally good results compared to the list-based scheduling. Out of those cases, 44% showed additional improvement in makespan by using the level-based temporal partitioning.

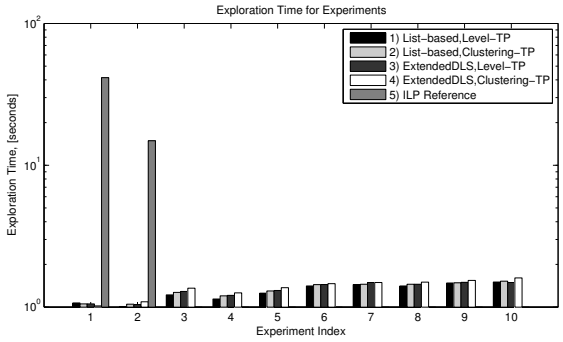


Fig. 5. Resulting in terms of exploration time. The bars 1-4 for each experiment are for the proposed framework. Bar 5 is an adapted version of [4], described in section III-A. The 0-1 ILP solution was only obtained for 20% of the cases, while it had prohibitively long exploration time for the rest of the cases. The results clearly showed that the 0-1 ILP solution is not a viable alternative, whereas the variation in exploration times in the four options of the proposed mapping framework was insignificant.

6) Precedence Constraint:

$$\forall \text{edges}(i_1, i_2), \sum_j \big(j \cdot x_{i_1,j} + j \cdot y_{i_1,j}\big) + \quad (7)$$

$$\sum_j \big(t^{\text{hw}}_{i_1} \cdot x_{i_1,j} + t^{\text{sw}}_{i_1} \cdot y_{i_1,j}\big) + \quad (8)$$

$$ct_{i_1,i_2} \cdot in_{i_1,i_2} - \sum_j \big(j \cdot x_{i_2,j} + j \cdot y_{i_2,j}\big) \quad \leq \quad 0 \quad (9)$$

The optimization goal is given by minimization of the finish-time of the last task, which can be formulated as:

$$\min \sum_j \big(j \cdot x_{n,j} + j \cdot y_{n,j} + t^{\text{hw}}_i \cdot x_{n,j} + t^{\text{sw}}_i \cdot y_{n,j}\big) \quad , \quad (10)$$

where n is the index of the last task (sink node).

Having the ILP-problem defined, it was passed to the solver, `glpsol` version 4.35, from the GNU Linear Programming Kit (GLPK) [8]. The `glpsol` was executed on a standard Linux PC. The results were compared to the result of the mapping framework as described in the previous section III.

IV. Results

The results of the mapping experiments are given by makespan and exploration times shown in the figures 4 and 5, respectively. For the cases where ILP experiments have been performed, the results are shown in the graphs as a rightmost grey bar for each task graph. The optimal ILP solution was only found for 20% of the task-graphs, as the exploration time was simply too long, going beyond more than eight hours for even relatively simple task-graphs with only 10 nodes.

Furthermore, we have included the resulting schedule for task-graph 6 in figure 6, as an illustration of the outcome of the mapping framework.

V. Discussion

When comparing the results presented in figure 5 it is clear that the ILP reference has a significantly higher exploration time than the framework that we propose in section II of this paper. However, when looking at the makespan results in figure 4, the mapping framework resulted in a slightly higher (12.5%) makespan for the experiments 1 and 2. However, the lower makespan of the optimal reference was made possible due to overlaying of HW execution and reconfiguration in dynamic partial reconfiguration.

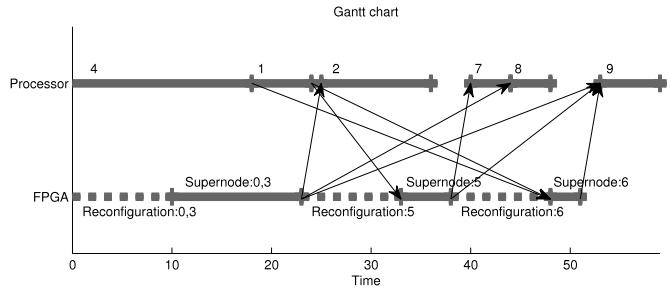


Fig. 6. Resulting schedule of task-graph 6, obtained by level-based temporal partitioning and the Extended DLS scheduling. The dotted lines indicate reconfiguration of the HW, and the arrows represent data transfer on the bus.

When the results are compared for the four different combinations for the presented mapping framework, the results were less clear. For 9 of the 10 cases, the Extended DLS algorithm gave better or equally good results compared to the list-based scheduling. Out of those 9 cases, 4 of them showed that the level-based temporal partitioning gave better results than the clustering-based. Only in 1 of those 9 cases, the level-based performed worse than the clustering-based temporal partitioning. This was surprising since the level-based algorithm would normally lead to more connections to outside partitions, which could potentially increase the HW/SW communication delay. However, the level-based algorithms are less likely to create paths from output to input of the same partition that go through other partitions, thus leading to fewer partitions than the clustering-based approach.

However, it is beneficial to run all four algorithms and compare the results. Such runs do only take short time as seen in figure 5, but gave a highest-to-lowest makespan reduction between 0% and 34%.

The performance of the proposed mapping framework is highly dependent on the early HW/SW partitioning, and it is therefore relevant to consider if this can be improved. First, the reconfiguration time is included for each HW task, even though it may cover reconfiguration of several tasks in parallel (for HW supernodes). This may be improved by weighting the HW reconfiguration time relative to the logic resource usage. However, the partitioner may then not be aware of the risk that small tasks may still require their own partition as described in section II-C. Second, there has not been incorporated any feedback loop into the partitioning as indicated in figure 1. This may be beneficial especially for the partitioning cases where the HW and SW execution times are close to each other.

VI. Conclusion

In this paper we presented a mapping framework for reconfigurable heterogeneous architectures consisting of a SW processor and a HW unit with global reconfiguration capability. Our main contribution is that the framework has been developed with the explicit goal to be able to handle heterogeneous reconfigurable architectures consisting of multiple HW and SW units. The framework is based on an application and architecture description, related through a cost-library that provides information of implementation alternatives of each task. The mapping framework performs HW/SW partitioning, and uses temporal partition algorithms to create HW partitions that can be handled by a scheduling and binding algorithm for heterogeneous multiprocessor architectures.

Mapping experiments were performed for ten task-graphs, with four combinations of two temporal partitions algorithms and two multiprocessor scheduling algorithms. The results showed that the mapping framework had very short exploration time as compared to the (existing) ILP approach, but that the selection of a specific mapping method (out of the four combinations) had an impact of up to 34% compared to the worst performing method. For 90% of the cases, the Extended DLS algorithm in combination with level-based temporal partitioning had the best performance.

We conclude that the proposed mapping methodology is promising and that it can provide designers with a tool for rapid exploration of scheduling strategies for reconfigurable heterogeneous architectures. In order to further improve the methodology, we will conduct the following as future work: a) improve the HW/SW partitioning algorithm, and b) add a feedback loop from the multiprocessor scheduler. Furthermore, future work will also include experiments that cover architectures consisting of multiple SW and HW units.

References

[1] A. Shoa and S. Shirani, "Run-time reconfigurable systems for digital signal processing applications: a survey," *Journal of VLSI Signal Processing Systems*, vol. 39, no. 3, pp. 213–235, 2005.

[2] M. Kaul and R. Vemuri, "Optimal temporal partitioning and synthesis for reconfigurable architectures," in *Proceedings of the conference on Design, automation and test in Europe*, 1998, pp. 389–397.

[3] K. M. G. Purna and D. Bhatia, "Temporal partitioning and scheduling data flow graphs for reconfigurable computers," *IEEE Trans. Comput.*, vol. 48, no. 6, pp. 579–590, Jun. 1999.

[4] S. Banerjee, E. Bozorgzadeh, and N. D. Dutt, "Integrating physical constraints in hw-sw partitioning for architectures with partial dynamic reconfiguration," *IEEE Trans. VLSI Syst.*, vol. 14, no. 11, pp. 1189–1202, Nov. 2006.

[5] J. Noguera and R. M. Badia, "A hw/sw partitioning algorithm for dynamically reconfigurable architectures," in *Proceeding of Design, Automation and Test in Europe*, Mar. 2001, pp. 729–734.

[6] K. S. Chatha and R. Vemuri, "Hardware-software codesign for dynamically reconfigurable architectures," in *9th International Workshop on Field-Programmable Logic and Applications*, 1999, pp. 175–184.

[7] G. C. Sih and E. A. Lee, "A compile-time scheduling heuristic for interconnection-constrained heterogeneous processor architectures," *IEEE Trans. Parallel Distrib. Syst.*, vol. 4, no. 2, pp. 175–187, Feb. 1993.

[8] GNU, "Gnu linear programming kit (glpk)," http://www.gnu.org/software/glpk/.

Chapitre 9

Projet "Adaptive Tongue-Controlled Interface"

9.1 Contexte

Le projet "Adaptive Tongue-Controlled Interface" trouve ses origines dans un thème de recherche développé au Department of Health Science and Technology (HST), Aalborg University. L'idée centrale de ce thème est le développement d'une solution pour l'aide au handicap visant la commande d'objets (aussi varié qu'un fauteuil roulant, qu'un ordinateur ou qu'un système d'éclairage) à partir d'un système inductif activé par la langue.

Comme le montre la figure 9.1, le système initial est composé des trois parties : i) un dispositif d'entrée placé contre le palais de l'utilisateur (un ensemble de capteurs intra-buccaux à mini-bobines), ii) un dispositif d'activation de ces capteurs intra-buccaux (un piercing ferromagnétique doux sur la langue) et iii) une interface à microcontrôleur pour la commande des équipements externes. Les signaux générés par les capteurs intra-buccaux sont transmis à l'interface via une solution sans fil 2.4 GHz ISM (industriel, scientifique, et médical). Cette interface applique un traitement numérique au signal reçu pour le mettre en forme et interprète les commandes qui sont ensuite retransmises vers les équipements externes (liaison filaire pour la commande d'un fauteuil roulant et liaison Bluetooth pour la commande d'un ordinateur personnel).
Il faut souligner que le système intra-buccal développé par HST est quasi-invisible, critère déterminant pour le succès de la solution auprès des utilisateurs (qui souhaitent minimiser le sentiment de différence).

Les premiers résultats de ces travaux ont encouragé ses responsables à créer la société TKS A/S afin de commercialiser des produits utilisant la technologie qu'ils ont développée lors de ces recherches.
C'est dans ce contexte que HST et TKS A/S ont approché le Center for Software Defined Radio et le Department of Electronic Systems, Aalborg University afin de rendre le système existant plus flexible (c.à.d. permettre la cohabitation de plusieurs standards de communication sans fil tels que Bluetooth, Zigbee,...) et sensible au contexte (c.à.d. capable de s'adapter à l'environnement de l'utilisateur : détection des objets disponibles, mise à disposition de ceux-ci sur l'interface utilisateur graphique, etc.).
Les travaux de thèses de master de Bastien Paul et Séverin Marcombes (Department of Computer Science) et de Siim Sepman (Department of Electronic Systems) ont été réal-

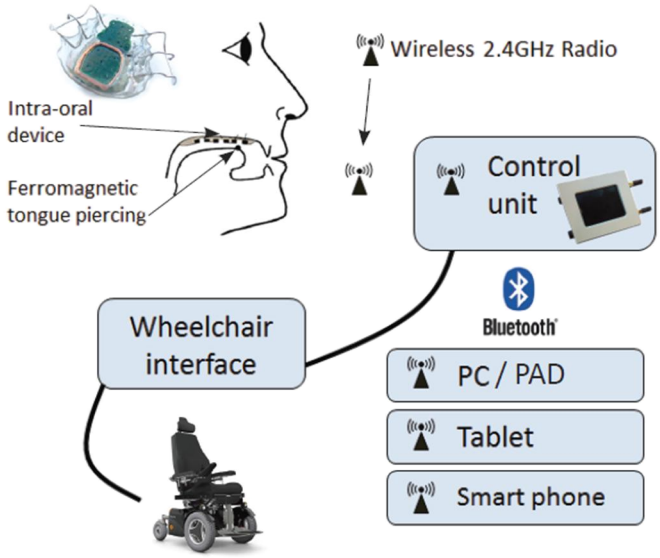


Figure 9.1: Illustration du système initial développé par HST et TKS A/S. Extrait de la brochure commerciale du produit iTongue de TKS A/S.

isés dans le cadre de ce projet.

9.2 Problématique

Les deux objectifs principaux de nos travaux étaient de permettre la découverte et la connexion aux objets connectés présents dans l'environnement de l'utilisateur et de lui donner accès aux services de contrôle de ces objets, et ce de manière dynamique. Afin d'atteindre ces objectifs, les verrous technologiques suivants ont dû être traités :

- Reconnaissance de l'environnement (*context awareness*) : afin de faciliter la vie des utilisateurs, l'interface proposée doit offrir des mécanismes de découverte des objets et des services proposés par ceux-ci. Ceci doit être effectué de manière aussi transparente que possible pour l'utilisateur afin de minimiser le nombre de manipulations 'informatiques' ; il est en effet fortement souhaitable de minimiser l'intervention de personnes tierces pour augmenter l'autonomie des utilisateurs visés.

- Co-existence de standards de communication sans fil : les objets connectés peuvent utiliser différents standards de réseaux personnels (*Personnal Area Network (PAN)*) ; l'interface proposée doit non seulement permettre des connexions simultanées via ces différents standards mais aussi une gestion de type multitâche pour que l'utilisateur puisse basculer entre les objets connectés correspondants.

- Évolutivité : pour faire face à l'apparition de nouveaux standards de réseaux personnels, l'interface proposée doit être suffisamment souple pour faciliter sa mise à jour.

- Maîtrise de la consommation électrique : l'interface étant destinée à être installée sur des fauteuils roulants électriques alimentés par batteries, il est nécessaire de minimiser la consommation de la première afin de limiter son impact sur l'autonomie en énergie des seconds.
- Ergonomie : enfin, la manipulation de l'interface doit rester simple et fluide avec différent types de mécanismes d'entrées, notamment le système intra-buccal. La navigation dans les menus et l'activation des commandes via l'interface graphique doit être pensée de manière à éviter la fatigue de l'utilisateur.

La figure 9.2 illustre le concept de l'interface que nous avons développée afin de rendre possible les points évoqués plus haut. À noter que dans une version commerciale, les deux interfaces (l'existante et la nouvelle) pourraient être avantageusement implantées sur une seule puce.

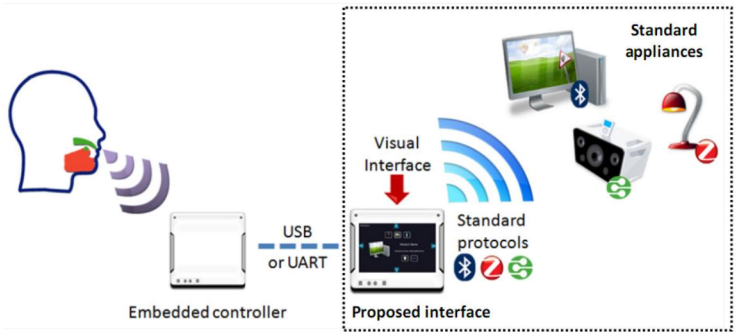


Figure 9.2: Illustration du concept de l'interface développée (dans le cadre en pointillé) venant compléter le système initial. Celle-ci permet i) la cohabitation de plusieurs standards de communication sans fil tels que Bluetooth, Zigbee, etc., ii) la sensibilité au contexte et iii) l'affichage sur interface utilisateur graphique.

9.3 Résumé des contributions

9.3.1 Conception

En ce qui concerne l'aspect télécommunication, une approche SOA (Software Oriented Architecture) a été développée. Elle permet de combiner différentes couches protocolaires à des fin d'interopérabilités (permet p. ex. de gérer des appareils équipés Bluetooth ou Zigbee) et de simplicité (découverte et accès aux services). De plus, pour la couche application, nous avons choisi DLNA car il est répandu dans beaucoup de produits grand public.

La structure de l'interface est modulaire et composée de trois catégories d'éléments : appareils, protocoles et fonctionnalités. Chacune de ses catégories inclue des classes, des plugins et des extensions (voir publication J.5 dans le chapitre 5). Une vue d'ensemble de

cette structure est donnée dans la figure 9.3 : les éléments clefs de celle-ci sont résumés dans ce qui suit.

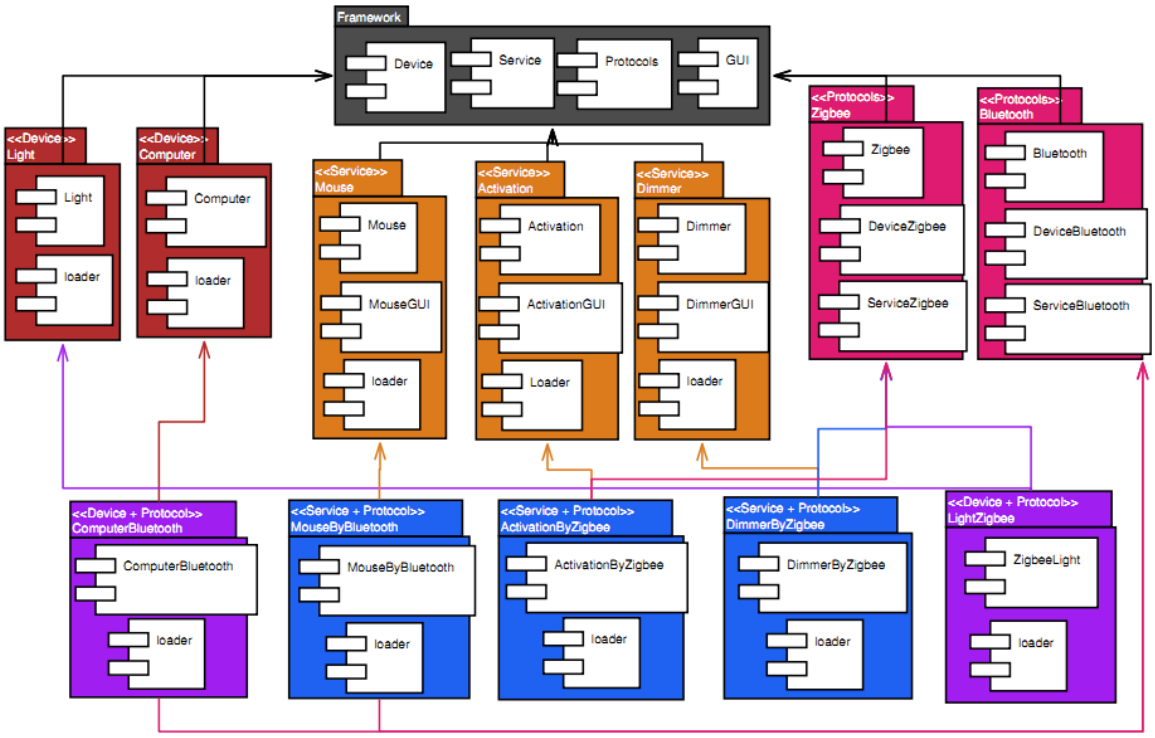


Figure 9.3: Structure de l'interface. Extrait de la publication J.5.

Les appareils sont représentés au moyen de la classe 'Device' (nom, emplacement, ID unique, icone, type (luminaire, ordinateur, etc.)). La classe 'Service' définie une fonctionnalité en fonction du protocole auquel elle est associée et de sa famille. La classe 'ConcurrentList' est utilisée pour ajouter, accéder à ou retirer des appareils. Chaque objet 'Device' contient une liste des 'Services' (c.à.d. des fonctionnalitées) qu'il propose.

Pour le confort des utilisateurs, la découverte des appareils disponibles doit se faire de manière transparente. Ceci est possible grâce à la classe 'Protocol' (qui implémente aussi les protocoles de communication), notamment via les fonctions de découverte offertes par les protocoles.

Un autre aspect est que les activités de communication ou de l'interface graphique ne doivent pas ralentir ou interdire les actions utilisateur ; ce besoin en multitâche est rendu possible par l'utilisation de threads.

La modularité est implantée via un mécanisme de plugins qui permet d'ajouter de nouvelles technologies (plugins Protocol-Dependent) et de nouvelles fonctionnalités (plugin Functionality). Dans la figure 9.3 les boîtes 'Devices' (rouge) représentent les appareils, 'Services' (orange) les services et les éléments de l'interface graphique associés, 'Protocols' (rose) la connectivité (en liaison avec 'Device Definition' (mauve) et 'Service Definition' (bleu). Les plugins sont composés de trois classes principales, et en option, un autre ensemble de classes pour leurs implantations : 'Implementation', 'Plug-in' et 'Loader'. Un

gestionnaire de plugins les charge en mémoire, gère les dépendances, et les retire de la mémoire.

Enfin, afin de réduire la consommation énergétique, un module de gestion de l'énergie (Power Management System (PMS)) a été conçu. Au niveau matériel, il permet d'appliquer les techniques suivantes : gestion dynamique de la fréquence et de la tension du processeur, mise en veille de l'écran tactile, contrôle du rétro-éclairage, mode basse consommation de la RAM, gestion des modes basses consommations des protocoles de communication, mise en veille dynamique des composants.
Par exemple, en ce qui concerne les modules de communication sans fils, plusieurs techniques sont aussi appliquées : limitation des possibilités de découverte et de connectivité, évitement de collision radio, communication en mode basse consommation, mise en veille en fonction de probabilités d'utilisation, modulation de la fréquence et du type de découvert selon la probabilité de découverte d'appareils dans l'environnement de l'utilisateur.
Au niveau logiciel, un système de drapeaux et de timers détermine où et quand appliquer ces techniques, évitant ainsi de contrôler la consommation en permanence.

9.3.2 Prototype et test

Le prototype construit dans le cadre de projet est constitué des éléments suivants : (visibles dans la figure 9.4)

- Kit d'évaluation Atmel AT91SAM9263-EK avec microcontrôleur AT91SAM9263 et un module LCD 3.5" tactile rétroéclairé;
- Kit Texas Instrument CC2530 pour la connectivité Zigbee avec deux modules CC-2530EM, deux kits SmartRF05EB et un dongle USB CC2531;
- dongle Bluetooth générique;
- Système d'exploitation Ångström Linux;
- Interface graphique Qt pour Linux embarqué;
- Bibliothèque Pthread.

Toute une batterie de tests (voir publication J.5 dans le chapitre 5 pour plus de détails) a été effectuée pour évaluer le bon fonctionnement et la performance du prototype. Ces tests concernaient la découverte et la commande d'appareils (ordinateurs via Bluetooth, éclairage via Zigbee), la commande de et le basculement entre plusieurs appareils via le même protocole de communication, la même chose mais avec des protocoles différents, l'adaptation dynamique à l'environnement (présence et absence d'appareils), et des mesures de consommation.
Ces tests ont montré à la fois le bon fonctionnement et la bonne fluidité de l'interface. Les mesures de consommation montrent 2400 mW au démarrage, 2244 mW en utilisation normale (rétroéclairage et fréquence processeur max, 1656 mW sans rétroéclairage, 1620 mW à fréquence processeur mini, 1080 mW quand les deux sont appliquées simultanément. Bien entendu il serait possible de réduire la consommation via une carte personnalisée excluant tous les composants non-utilisés du kit de développement.

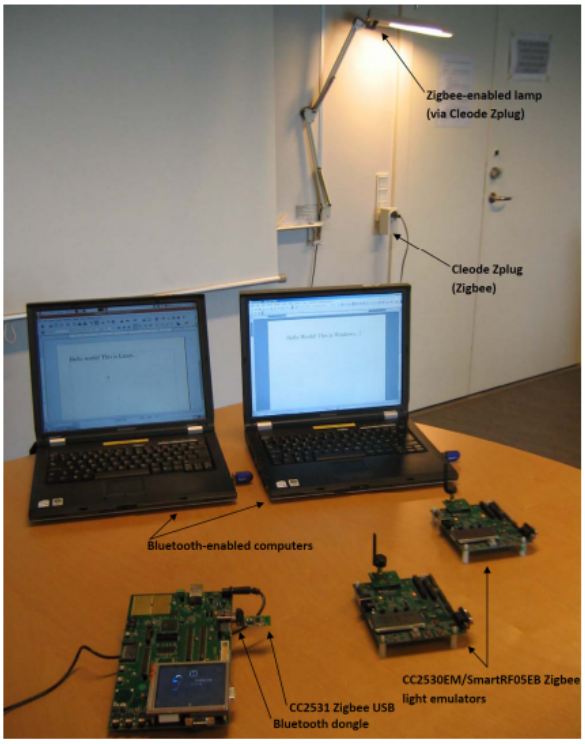


Figure 9.4: Prototype de l'interface et appareils (ordinateurs, éclairage) contrôlés via celle-ci. Extrait de la publication J.5.

9.4 Publications et commentaire

Ces travaux ont donné lieu à la publication de l'article J.5 dans le chapitre 5 ainsi qu'aux thèses de master de Bastien Paul et Séverin Marcombes et celle de Siim Sepman.
La publication J.5 *A Context-Aware User Interface for Wireless Personal-Area Network Assistive Environments* présente en détail la conception et l'implantation de l'interface introduite plus haut ; elle est reproduite dans les pages qui suivent.

Une suite intéressante à ces travaux aurait été de tester l'interface développée sur le public visée et de l'intégrer sur un fauteuil roulant. Pour le premier point il aurait été possible de travailler avec les patients du partenaire hospitalier de TSK A/S, tandis que pour le second il nous aurait fallu définir l'autonomie souhaitée et si besoin optimiser la consommation de l'interface en enlevant tout le superflu afin d'obtenir cette autonomie.

A context-aware user interface for wireless personal-area network assistive environments

Bastien Paul[1], Séverin Marcombes[1], Alexandre David[1], Lotte N.S. Andreasen Struijk[2], Yannick Le Moullec[3]

[1] *Department of Computer Science, Aalborg University, Denmark*

[2] *Department of Health Science and Technology, Aalborg University, Denmark*

[3] *Technology Platforms Section, Department of Electronic Systems, Aalborg University, Denmark*

Corresponding author:

Yannick Le Moullec, Fr.Bajers Vej 7, A3-216, DK-9220 Aalborg Ø, Denmark.

ylm@es.aau.dk

+45 9940 7507

Abstract The daily life of people with severe motor system impairments is challenging and thus often subordinated to extensive external help; increasing their level of self-support is thus highly desirable. Recent advances in wireless communications, in particular in wireless personal-area networks, serve as technological enablers well suited for implementing smart and convenient assistive environments which can increase self-support. This paper presents the design and prototyping of a versatile interface for such wireless assistive environments. We propose a modular framework that can accommodate several wireless personal-area network standards. The interface is built upon this framework and is designed in such a way that it can be controlled by various types of input devices such as a touch screen or a tongue-control unit. The interface can automatically discover consumer appliances (e.g. Zigbee and Bluetooth enabled lights and computers) in the user's environment and display the services supported by these devices on a user-friendly graphical user interface. A demonstrator is prototyped and experimental results show that the proposed interface is context-aware, i.e. it successfully detects available appliances, adapts itself to the changes that occur in the user's environment, and automatically informs the user about these changes. The results also show that the proposed interface is versatile and easy to use, i.e. the user can easily control multiple devices by means of a browser menu. Hence, the proposed work illustrates how assistive technology based on wireless personal-area networks can contribute to improving the quality of life of motor system impaired persons.

Keywords Context-aware control interface, wireless assistive environments, service discovery, motor system impairments, tongue-controlled interface.

This is the authors' version of the paper which appears in Wireless Personal Communications
DOI: 10.1007/s11277-012-0582-x

1. Introduction

Spinal Cord Injuries (resulting in quadriplegia), brain injury and other sources of motor system impairments raise severe barriers for individuals who are subject to these. Very often, their daily lives are challenging and activities that we take for granted can be or can become out of scope for these individuals. These activities include, among others, attending school, having a job, and socializing. In such situations, life is subordinated to extensive or continuous external help; however, there are cases where technology makes it possible to increase the level of self-support of these individuals, and in turn can improve their Quality of Life (QOL).

Over the last two decades, the amount of work carried out in the multidisciplinary domains of Assistive Technology (AT) and e-health [1] has been increasing, encouraged by the evolution of available technologies such as sub-micron VLSI digital circuits, digital signal processing platforms, and, more recently, advances in wireless communications, in particular in terms of Wireless Personal-Area Networks (WPAN) based on e.g. Zigbee and Bluetooth. The combination of these technologies pave the way for advanced systems which can improve the QOL of people with motor system impairments by enabling them to control their environments (e.g. in smart homes) and to access modern communication channels (e.g. email, web, audio/video calls). This is a domain that has witnessed many research and development efforts [2]. Indeed, even when living with severe impairments such as quadriplegia, people are often still able to move body parts such as jaws, eyes, head, and tongue. Research efforts have resulted in several types of adapted control systems that enable disabled people to control their environment, e.g. their wheelchair, lights, TVs, computers. Although many promising concepts have been demonstrated, only a limited fraction of

This is the authors' version of the paper which appears in Wireless Personal Communications
DOI: 10.1007/s11277-012-0582-x

these environmental control systems have become widely accepted by the users; price, ease of control, aesthetics, and social cost are essential issues that can determine the adoption and success of such systems [3-4].

Home medical equipment is expensive and not always affordable without health insurance compensation. Any added feature that increases the overall price too much has reduced chances to become widespread. Typically, AT relies on three major parts [2]. The first one, the "access pathway" is composed of the physical sensors or input devices actuated by the user; their outputs are then converted into electrical signals which are further analyzed by means of digital signal processing techniques. In the context of severe motor system impairment, access pathway examples include control systems based on the eye, head, brain, or tongue actuation. The second part is the actual "user interface" which analyzes the digitally processed input signals and converts them into output signals. These output signals are then used to enable the third part, i.e. "functional activities" such as controlling appliances in the user's environment.

Ideally, the user interface should be versatile so that it enables the user to interact with as many types of devices located in his/her environment as possible, as transparently as possible. Moreover, aesthetics is too often underestimated; in many cases it is highly desirable to minimize the feeling of being "different" so that the control system can be accepted and adopted by the users [5]. Two of the above mentioned issues, namely aesthetics and access pathway, have been investigated at Aalborg University and have resulted in a fully integrated wireless inductive tongue control system [6-7]. The work presented in the current paper deals with the design and prototyping of a user interface, i.e. a versatile interface which could be used together with (but is not limited to) the above mentioned tongue control system. The proposed interface exploits the fact that wireless features are increasingly found in daily appliances and that inexpensive and versatile embedded

This is the authors' version of the paper which appears in Wireless Personal Communications
DOI: 10.1007/s11277-012-0582-x

systems (e.g. microcontroller-based Linux platforms) are more and more widespread. This interface automatically discovers wireless-enabled appliances (e.g. Zigbee-enabled lights, Bluetooth-enabled computers), displays their services on the visual interface and lets the user control them by means of an access pathways (input) device, which could be, for example, a touch screen (part of the proposed interface) or a tongue control system (e.g. [6-7]). The AT scenario considered for this work is illustrated in Figure 1, where the work presented in the current paper is delimited by the dashed box.

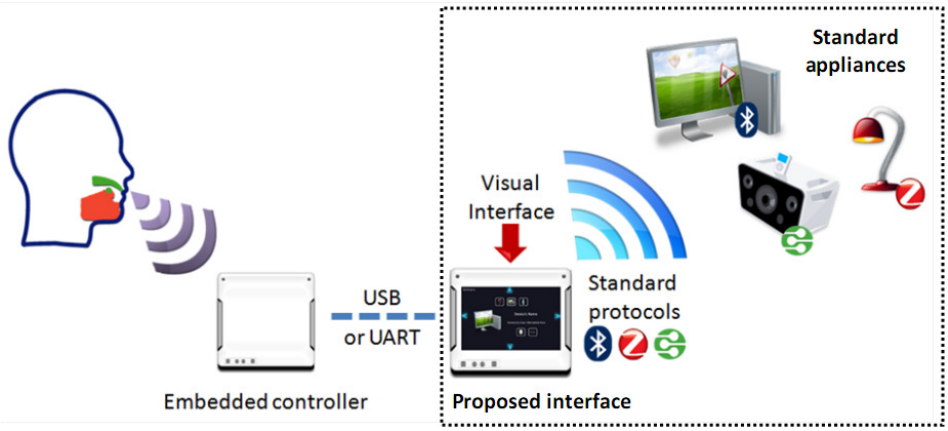


Figure 1 Illustration of the proposed interface for controlling appliances in a wireless assistive environment. The work presented in this paper is delimited by the dashed box.

The remainder of this paper is organized as follows. Section 2 briefly reviews related works in terms of i) assistive technologies for motor system impaired persons and ii) discovery, multi-protocol compatibility and context awareness. It also summarizes our contributions. Section 3 presents the design of the proposed user interface and the underlying discovery and connectivity mechanisms. Section 4 presents the prototyping of the proposed interface and Section 5 presents the experimental setup and results. Finally, Section 6 discusses the results and suggests directions for future work.

This is the authors' version of the paper which appears in Wireless Personal Communications
DOI: 10.1007/s11277-012-0582-x

2. Related Work and Contributions

2.1. Related work

Earlier works in terms of assistive technologies for motor system impaired persons include, among others, [8] that proposes a palatal tongue controller consisting of touch-sensitive pads mounted on the surface of the dental plate. A transmitter embedded in the dental plate transmits switch activations remotely to external devices fitted with receivers. Computers in the user's environment can be controlled via a mouse emulator whereas other appliances are controlled via infrared signals; however, its users may have to face challenges due to the limitations of that specific technology (e.g. line-of-sight and code learning). [9] describes a Hall effect-based approach for controlling appliances through tongue movement. Besides being rather complex and physically large, this system requires cables going from the mouthpiece unit to the processing unit. Moreover, no description of the interfacing/control methods to the devices in the user's environment is provided. More recently, a ZigBee-based wireless intra-oral control system for quadriplegic patients has been proposed in [10]. Their system is composed of a wireless intra-oral module, a wireless coordinator, and distributed ZigBee wireless controllers. The intra-oral module communicates wirelessly with the coordinator which itself communicates wirelessly with the Zigbee controllers to activate external devices, depending on the requests made by the user via a GUI. Although their concept and ours share a few similarities, they differ significantly on the interfacing aspects: their interface for controlling external devices is quite un-versatile since it relies on a static GUI that runs on either a PC or a pocket PC and it can only communicate with Zigbee-enabled appliances. [11] reports on an external tongue controlled device that communicates with a PC through a proprietary 2.4-GHz wireless link. An application executing on the PC enables the user to control a powered wheelchair. That paper also suggests the opportunity to control other

This is the authors' version of the paper which appears in Wireless Personal Communications
DOI: 10.1007/s11277-012-0582-x

types of appliances by means of a Wi-Fi/Bluetooth enabled PDA, but this is not implemented nor discussed. Finally, [12] explores an EOG-based eye tracking technique combined with infrared and Bluetooth connectivity for controlling appliances in the user's environment. Although sufficiently compact for being mounted on a wheelchair, their interface is neither designed for tongue control nor versatile enough for accommodating extra wireless protocols.

Clearly, the above works have paved the way for improved self-support in terms of environment control; however, there is still room for advancing the opportunities made possible by technological advances in wireless communications and context-aware computing. For our application, the ideal interface should not only be wireless, but should also feature mechanisms such as service discovery, multi-protocol compatibility and context awareness. Works related to these features are found in e.g. [13], where service discovery for mobile network is surveyed, while distance-sensitive service discovery is addressed in [14]; service reconfiguration for ensuring service availability in assistive environments is considered in e.g. [15]. Multi-protocol compatibility is investigated in works such as [16-22]. For example, [16] proposes a layered middleware that enables multiple protocol to coexist and [17] considers Wi-Fi/Zigbee coexistence. [18] and [19] present translation strategies for KNX-Zigbee and infrared-Zigbee, whereas [20] proposes a more universal approach by means of a dynamic device integration manager that enables transparent service discovery. [21] describes a so-called adaptive-scenario-based reasoning system where adaptive history scenarios are used to collect and aggregate user habits in smart homes; similarly, [22] suggests a dynamic service composition system for coordinating Universal Plug and Play (UPnP) services in smart homes and identifying devices that can work together, taking user habits into account.

This is the authors' version of the paper which appears in Wireless Personal Communications
DOI: 10.1007/s11277-012-0582-x

Finally, [23] suggests guidelines and recommendations for constructing robust context awareness applications; specific context-aware wireless network systems are discussed in e.g. [24] that describes CANE, a Context-Aware Network Equipment for multimedia applications that adapts dynamically to the user's environment by taking user's preferences, network status, as well as service requirements and policies into account.

2.2. Contributions

To overcome the limitations of [8-12] we exploit concepts similar to that of [13-24]. We present the design of a context-aware and versatile framework upon which the user interface is built. The paper details the mechanisms used so that the framework can i) discover and connect seamlessly to WPAN enabled appliances in the user's environment and ii) give the user access to the control and communication services of these appliances. The novelty of the work is twofold. Firstly, in contrast to existing assistive user interfaces, the one that we propose is not limited to a single communication standard or a predefined, fixed set of standards. To do so, we propose a discovery and multiprotocol connectivity mechanism combined with a plug-in system that makes the framework versatile enough to accommodate a wide range of existing and future WPAN standards. Secondly, the features needed for such a versatile interface are implemented as an embedded system with hard constraints in terms of size, power, and price so that it could be mounted on e.g. a wheelchair. As opposed to desktop implementations or less constrained embedded implementations, the proposed user interface can be implemented by means of relatively inexpensive and physically small components with low computational capabilities. Furthermore, a specific power management system is designed to reduce the power footprint of the user interface. A demonstrator is prototyped and used to experimentally test the proposed approach. The experiments show that the proposed interface is context-aware, i.e. it successfully detects available

This is the authors' version of the paper which appears in Wireless Personal Communications
DOI: 10.1007/s11277-012-0582-x

appliances, adapts itself to changes that occur in the user's environment, and automatically informs the user about these changes. Furthermore, the proposed interface is versatile and easy to use, i.e. the user can easily control multiple devices by means of a browser menu.

3. Design

3.1. Design considerations

The purpose of the proposed interface is to provide the necessary features for i) enabling the connection of various access pathway (input) types, such as e.g. a touch screen or a tongue control unit, with multiple WPAN-enabled appliances in the user's environment and ii) enabling the user to interact with those appliances. To implement the above, the following design aspects have been considered:

- The interface must be versatile enough to accommodate a significant set of appliances; as the number of wireless standards and home automation products is expanding, the interface must be flexible and upgradable.
- Using appliances through the interface must be as easy as it would with their native interfaces. In particular, the user should not feel the need for having someone to help him or her when interacting with the appliances. Similarly, the interface must provide an “easy first-time installation” procedure.
- The interface must provide the user with the possibility to switch easily and quickly between the appliances, without disrupting their operation. Support for some form of multitasking is thus needed.

This is the authors’ version of the paper which appears in Wireless Personal Communications
DOI: 10.1007/s11277-012-0582-x

- Moreover, the interface must be aware of its environment (i.e. able to detect available appliances) and inform the user about it (i.e. maintain a list of available appliances and their respective services and present them visually to the user).
- Finally, as the interface is expected to be on mounted on e.g. wheelchairs, the size and power factors are critical and should be kept as low as possible.

Regarding the networking aspect, this work combines various standard protocol stack layers; this makes that interoperability with multiple protocols stacks only requires that WPAN enablers, such as Bluetooth or Zigbee, are already available or installed on the appliance. Protocols using Service Oriented Architectures (SOA) are used here since they enable service discovery and service access. Examples of possible combinations of standard protocols providing both a physical layer and an application layer, as targeted in this work, are listed in Table 1. Although many SOA-based protocols implementing an application layer would be suitable for this work, the DLNA (Digital Living Network Alliance) specification is currently the only one broadly implemented in consumer appliances. Besides supporting UPnP, DLNA also enables the targeted users of the proposed interface to enjoy various audio and video media. The six combinations shown in Table 1 are the ones that have been considered. However, as the proposed interface is modular, this list can easily be extended.

To make it versatile and expendable with future standards, the framework has been designed following a modular approach. Its constituting elements fall into three main categories: devices, protocols, and functionalities. Each of these categories includes classes, plug-ins and extensions, as described in the following sections.

This is the authors' version of the paper which appears in Wireless Personal Communications
DOI: 10.1007/s11277-012-0582-x

Table 1 Examples of stack combinations for the proposed interface

#	Application stack	Network stack	Physical stack
1	Bluetooth Profiles + SDP	Bluetooth	802.15.1 (Bluetooth)
2	Zigbee Profiles + SDP	Zigbee	802.15.4
3	DLNA + UPnP	TCP/IPv4	802.11.X (Wi-Fi)
4	DLNA + UPnP	TCP/IPv6	802.11.X (Wi-Fi)
5	DLNA + UPnP	6LoWPAN (TCP/IPv6)	802.15.4
6	DLNA + UPnP	TCP/IP + Bluetooth	802.15.1 (Bluetooth)

3.2. Listing controllable appliances

Remote appliances are represented by means of the 'Device' class; each device has a name, a location, a unique ID, an icon, as well as a type (light, computer…); thus remote appliances are represented by means of objects derived from both the ‘Device’ and ‘Type’ classes.

3.3. Listing accessible functionalities

'Service' is the most generic class that can define a functionality, depending on the protocol it belongs to and its family. All devices are aggregated, through the ‘Element’ class, in the ' ConcurrentList' class that is used to add, access, or remove a device. Objects derived from the ‘Service’ class represent supported functionalities. Each ‘Device’ contains a list of the ‘Services’ (i.e. functionalities) it offers. Functionalities are referred to as ‘services’. Each functionality is therefore represented by an object derived from the ‘Service’ class. A summary of the connections between these classes is shown in Figure 2.

This is the authors’ version of the paper which appears in Wireless Personal Communications
DOI: 10.1007/s11277-012-0582-x

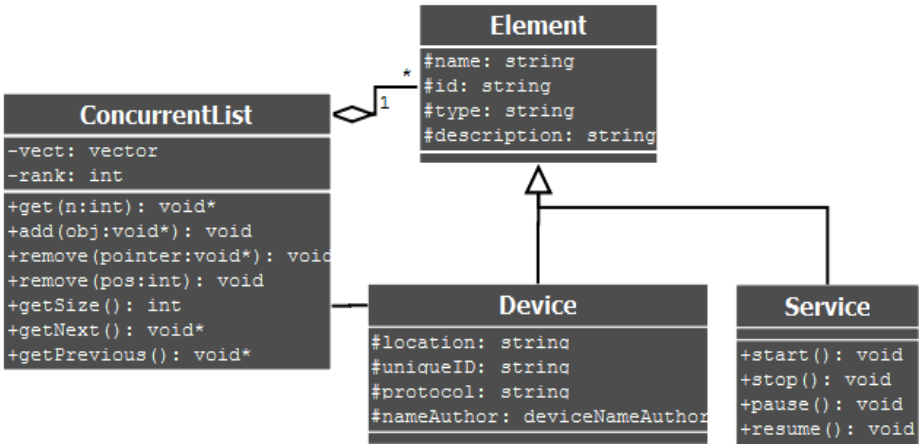


Figure 2 The 'ConcurrentList' class aggregates the 'Device' and 'Service' classes through the ‘Element’ class

3.4. Awareness of and adaptation to the user's environment

When the user moves from place to place or when appliances are displaced, the appliances seen by the framework enter or leave its coverage range. For usability sake, device discovering must take place without any command from the user. This is supported by the 'Protocol' class. The 'Protocol' class defines all protocol-related attributes and methods the framework can use. Adding new protocols is rather easy since it only requires a new class derived from ‘Protocol’ to describe the given protocol; ‘Protocol’ plug-ins also have to define extensions to the ‘Device’ class. Another aspect is that communication activities should not prevent the system from being responsive to the user’s actions. This is supported by means of threads: the communication activities are performed by one or several threads while the GUI is performed by another one, thus the framework is reactive and can continuously adapt to the user's environment. Adaptation is achieved by means of the discovery features of the considered protocols. The 'Protocol' class is responsible for discovering devices and for adding or removing them from the list. Moreover, the

This is the authors’ version of the paper which appears in Wireless Personal Communications
DOI: 10.1007/s11277-012-0582-x

GUI has to have access to the list of 'Element' just like 'Protocol' that is implemented in another thread. To avoid any conflict, 'ConcurrentList' provides a mutex mechanism to protect the access to the list of devices. These elements are connected through the 'ProtocolManager' class, as shown in Figure 3.

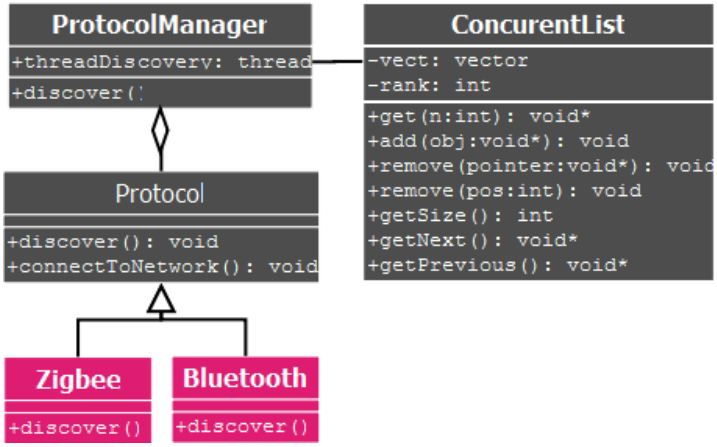


Figure 3 'ProtocolManager' connects the 'Protocol' and 'ConcurrentList' classes by aggregating the protocols (e.g. Zigbee and Bluetooth)

'ProtocolManager' aggregates all the protocols and 'ConcurrentList' is implemented in a thread where all the protocols are to be executed. 'ProtocolManager' manages a thread dedicated to device and service discovery. To represent a remote appliance, the 'Protocol' plug-in instantiates a subclass of the 'Device' class adapted to its needs in terms of e.g. data storage. These instances are then attached to the object that represents the appliance.

3.5. Switching between devices

Switching rapidly from one device to another is made possible by not terminating open connections when performing the switch. Instead, these connections are kept alive for a certain duration, or set in an idle mode that is less energy consuming, depending on the needs of the services and on the properties of the protocols. When a protocol does not support multiple

This is the authors' version of the paper which appears in Wireless Personal Communications
DOI: 10.1007/s11277-012-0582-x

connections, switching between devices may imply terminating an active connection: in this case, the termination is done in such a way that the current state of the appliance is not modified.

3.6. Modularity

Modularity is achieved by means of a plug-in mechanism which supports new technologies (protocol-dependent plug-ins) and new functionalities (functionality plug-ins). A plug-in is a library used by the framework and contains the definition of protocol-dependent and type-dependent devices as well as the definition of specific protocols. Figure 4 shows how the plug-in mechanism relates to the framework (in grey) and the dependencies between the plug-ins. 'Devices' (red) represent appliances such as computers and lights. 'Services' (orange) define the supported services and their associated GUI items. 'Protocols' (pink) bring connectivity support in conjunction with 'Device Definition' (purple) and 'Service Definition' (blue). Plug-ins are composed of three main classes and, optionally, a set of other classes providing the plug-in implementation: the 'Implementation' class that depends on the plug-in type, the 'Plug-in' class which creates an instantiation of the implementation class and contains the dependency information between plug-ins, and the 'Loader' class which registers the plug-in class in the framework. Finally, a plug-in manager loads all plug-ins. When loaded, plug-ins register themselves automatically in the plug-in manager; then the manager checks the dependencies. The manager is also responsible for unloading plug-ins.

This is the authors' version of the paper which appears in Wireless Personal Communications
DOI: 10.1007/s11277-012-0582-x

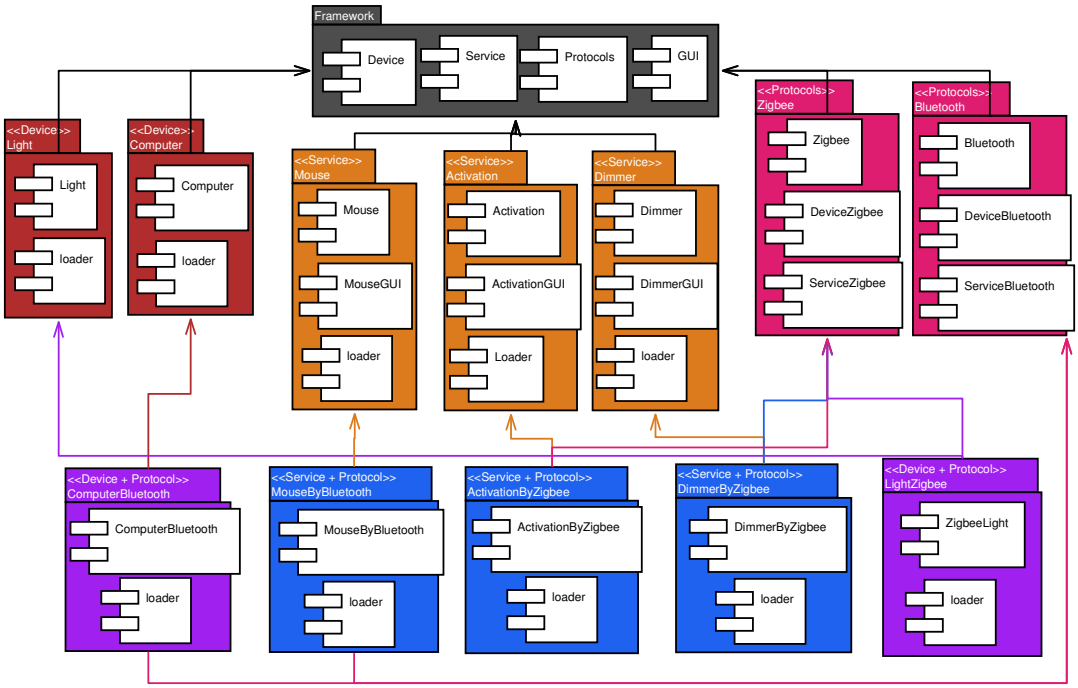


Figure 4 Plug-ins dependencies: Framework (grey), Devices (red), Services (orange), Protocols (pink), Device Definition (purple), and Service Definition (blue)

In the current AT scenario, it is expected that several types of access pathways (input) devices could be used for navigating through the GUI, for example a touch screen or the tongue-control system described in [1] . In order to avoid user's fatigue, the GUI has been designed so that the number of required movements and clicks is minimized. Figure 5 illustrates how it is possible to select the appliance to be controlled (e.g. a light), to configure and activate it, and to access its features (e.g. turn on/off) by means of the left, right, up, down, click, and the escape commands. The GUI is integrated with the framework as shown in Figure 6. When the user switches to a device, a service or a configuration panel, the displayed object (e.g. device, service or configuration) is associated to the corresponding class. This class then reads the required information and displays them on the screen.

This is the authors' version of the paper which appears in Wireless Personal Communications
DOI: 10.1007/s11277-012-0582-x

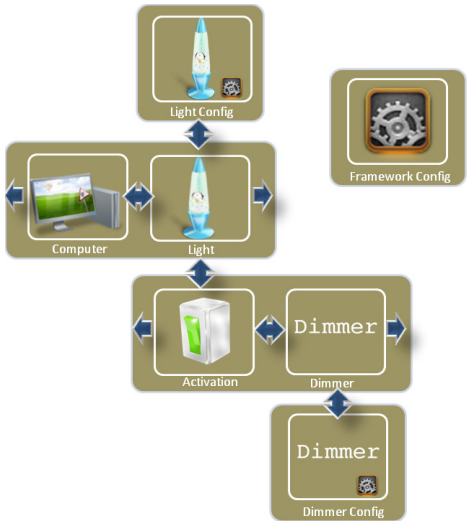


Figure 5 Illustration of the GUI architecture. It is possible to select the appliance to be controlled (e.g. a light), to configure and activate it, and to access its features (e.g. turn on/off) by means of the left, right, up, down, click and escape commands available on e.g. the tongue-control unit and the touch-screen

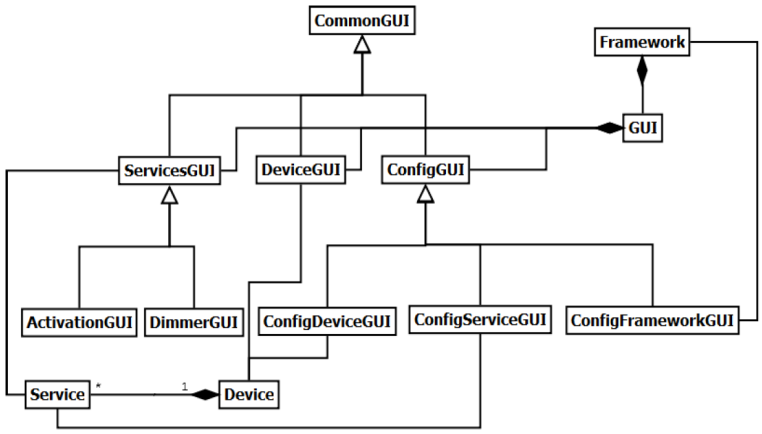


Figure 6 Integration of GUI with the framework: 'Service' is associated with 'ServicesGUI' and 'ConfigServiceGUI', 'Device' with 'DeviceGUI' and 'ConfigDeviceGui', and 'Framework' with 'ConfigFrameworkGUI'

3.7. Managing the power

A power management system (PMS) has been designed and implemented. The role of the PMS is to minimize power by combining several techniques, both at the physical (HW) and framework (SW) levels. The following techniques are applied at the hardware level: i) dynamically scaling the

This is the authors' version of the paper which appears in Wireless Personal Communications
DOI: 10.1007/s11277-012-0582-x

frequency and voltage of the CPU, ii) turning off the touch panel when unused, iii) diming or turning off the backlight when unused, iv) using the RAM low power mode, v) exploiting the low power facilities provided by communication protocols, and vi) turning off the various chips when unused. At the software level, flags and timers are used to determine how and when to apply the above techniques. This removes the need for monitoring the system usage and eases the detection of service requests. As seen in Figure 7, the core of the PMS (i.e. for the CPU/memory) consists of four modes ('Run', 'Conservative', 'Sleep', 'Shutdown'). When the period of time a user is inactive reaches a first threshold value, the system switches from 'Run' to 'Conservative' where the voltage and frequency of the CPU are decreased. If the inactivity period exceeds a second, larger threshold value, the system switches to 'Sleep' where the memory is set to self-refresh and the wireless chips in stand-alone modes. Any activity while in the 'Conservative' or 'Sleep' modes makes the system to switch back to 'Run'. Finally, all the components are turned off when the system enters 'Shut down'. A flag is used so that when a service needs the CPU computational power, the system does not switch to the 'Conservative' (and thus 'Sleep') modes. Similarly (not shown in Figure 7), the backlight is dimmed or turned off when inactivity thresholds are reached. A flag is used so that when a service needs to display visual feedback, the system does not switch to the dimmed and off modes.

This is the authors' version of the paper which appears in Wireless Personal Communications
DOI: 10.1007/s11277-012-0582-x

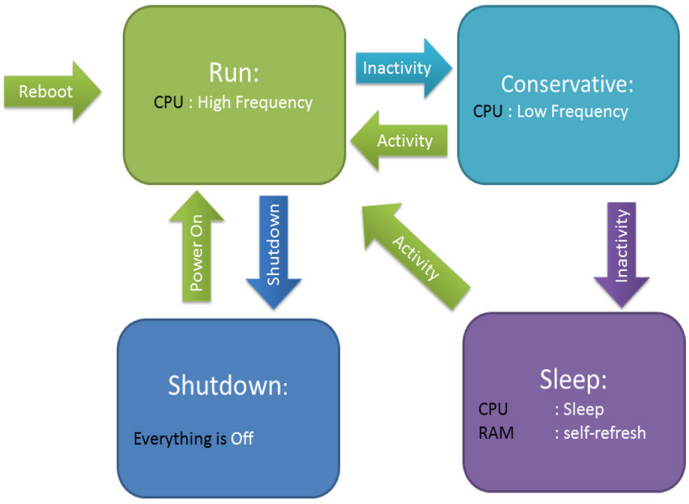


Figure 7 The core of the power management system consists of four modes. As user inactivity reaches thresholds, the system is switched to the Conservative and Sleep modes; user activity makes the system return to the Run mode

Regarding the wireless modules, several strategies are used. These include limiting discoverability and connectivity, performing radio collision avoidance, communicating in low power modes whenever possible, switching to stand-by modes based on usage probabilities, and modulating the period and type of discovery according to the probability of finding devices around the user.

4. Prototyping

4.1. Prototyping Platform

The demonstrator has been prototyped on a platform composed of the following elements. The core of the platform is Atmel's AT91SAM9263-EK Evaluation Kit [24] featuring among others, an AT91SAM9263 micro-controller and a 3.5" 1/4 VGA TFT LCD module with touch screen and backlight. For demonstration purposes, two WPAN standards (Zigbee and Bluetooth) have been implemented on the interface using a Texas Instrument CC2530 Development Kit [26] and a

This is the authors' version of the paper which appears in Wireless Personal Communications
DOI: 10.1007/s11277-012-0582-x

generic Bluetooth dongle, respectively. The CC2530 Development Kit contains, among others, two CC2530EM evaluation modules, two SmartRF05EB Evaluation Boards (on which the CC2530EM evaluation modules are plugged), and one CC2531 Zigbee USB dongle. The CC2530EM evaluation modules and the CC2531 USB dongle are essentially composed of a 2.4-GHz IEEE 802.15.4/Zigbee RF transceiver, an 8051 micro-controller core, and 256KB Flash/8KB SRAM. The selected operating system executing on the AT91SAM9263 micro-controller is Ångström Linux [27]. Finally, the GUI is implemented by means of Qt for Embedded Linux [27], a compact, memory efficient windowing system for Linux.

4.2. Prototyping of the Framework

In order to keep the implementation modular, the framework has been implemented by means of three packages, namely 'Core', 'GUI' and 'Communication'. 'Core' is composed of three of the classes introduced in Section 3: 'ConcurrentList', 'Device' and 'Service'. 'Device' and 'Service' are stored in Vector containers supported by the Standard Template Library (STL) library. 'ConcurrentList' handles these containers by means of concurrency protection via mutex synchronization. 'GUI' implements the classes related to the graphical user interface. It makes use of the signal-and-slot mechanism supported by Qt: a signal is emitted when an action occurs (e.g. user click) and a slot (a method) is executed in response to the emitted signal. 'Communication' relates to the 'Protocol' class and its derived classes ('Zigbee' and 'Bluetooth' in the demonstrator). In the demonstrator, threads are implemented by means of the Pthread Library. Qt events are used as a messaging system between the threads, and mutexes are used to protect data and as a synchronization mechanism. Moreover, a notification system exploits Qt's event system and the Pthread library so that when devices are discovered, the 'Protocol' classes can notify the GUI. These are added to 'ConcurrentList'. Subsequently a notification is sent to 'GUI' by

This is the authors' version of the paper which appears in Wireless Personal Communications
DOI: 10.1007/s11277-012-0582-x

'ConcurrentList'. Then, the protocol lists the devices functionalities: it creates a service for each functionality and adds them to the corresponding devices. Finally, 'ConcurrentList' sends notification to 'GUI' so that the user can see on the screen which functionalities are available.

4.3. Zigbee module

The Zigbee module implements a generic mechanism for Zigbee communication. A plug-in enables this module to support the Zigbee Home Automation Profile. The lower and upper layers of the Zigbee module are shown in Table 2. The hardware consists of the CC2531 Zigbee USB dongle connected to the AT91SAM9263-EK board. The firmware executing on the CC2531 is composed of Z-stack, a Zigbee stack provided by Texas Instruments, and a custom application developed specifically for the implementation of the demonstrator. Z-stack, based on a simple event-based OS, provides an OS Abstraction Layer (OSAL) Application Programming Interface (API) for interfacing with custom applications. Moreover, Z-stack provides a hardware abstraction layer API, so that multiple hardware components can be exploited generically.

The CC2531's Z-stack firmware searches for a Zigbee network to join, both upon power up and periodically once executing. Profile advertising is carried out upon joining a network: the CC2531 device describes itself to let the other Zigbee devices know which profiles it supports and the functions it provides. The first operation performed by the custom firmware application is to provide this information to Z-stack in order to advertise that the device is compatible with the Home Automation Profile and can act, in the demonstrator, as a remote light switch.

Service discovery is taken care of by the custom firmware application that requests Z-stack to search for a device providing a Zigbee cluster. In return, Z-stack sends a Zigbee “Descriptor Matching Request” to the network. This request is sent to a router or to the coordinator of the network, and is propagated to all routers of the network. If a device with a matching cluster is

This is the authors’ version of the paper which appears in Wireless Personal Communications
DOI: 10.1007/s11277-012-0582-x

found on the network, the router or the coordinator to which the device is connected sends a response packet back. Upon reception of this packet, Z-stack calls the custom firmware application by sending an OSAL event. The addresses of the Zigbee devices resulting from the search are then sent one by one to the interface framework software. Afterwards, the latter can request the custom firmware application to bind itself to the devices that have been discovered.

Connection management is taken care of by the custom firmware application that handles connections to other devices by sending, upon request, the list of available Zigbee networks and the channels that they use to the Zigbee communication module of the interface framework software. The interface framework software can also request the custom firmware application to connect to a certain network and, upon success, to search for compatible devices. It can also request the custom firmware application to disconnect from a certain network at any time.

4.4. Bluetooth Module

The Bluetooth module supports the Bluetooth Human Interface Device (HID) Profile, enabling the proposed interface to emulate the mouse and keyboard of a Bluetooth-enabled computer. It also provides most of the generic features of the communication module, such as service discovery, multi-tasking and connection management. The lower and upper layers of the Bluetooth module are shown in Table 2. The hardware used in the demonstrator consists of a low-cost Bluetooth USB dongle. The implementation of the Bluetooth module is composed of several software components. The firmware of the Bluetooth USB Dongle manages the low-level part of the Bluetooth stack, while the framework software takes care of the high-level part. The two parts of the stack interact with each other by means of the Host Controller Interface (HCI) and the appropriate Linux drivers for Bluetooth USB dongles. The higher level of the Bluetooth stack is itself composed of several blocks.

This is the authors' version of the paper which appears in Wireless Personal Communications
DOI: 10.1007/s11277-012-0582-x

The generic features of the Bluetooth stack are implemented with Blue-Z, the Linux De facto Bluetooth stack. The specific features that interact with Blue-Z to provide an application layer usable by the framework software are custom implemented. Our implementation makes use of the Blue-Z drivers for Bluetooth USB dongles, Blue-Z mechanisms for managing connections with other devices and the configuration of the local Bluetooth device, and the Blue-Z Service Discovery Protocol (SDP) server. These elements are accessed and used through the Blue-Z C/C++ library. The core components of the Bluetooth communication module are responsible for managing Bluetooth communications transparently for the interface framework by means of derivations of the 'Protocol', 'Device' and 'Service' classes.

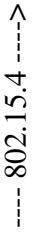
Service advertisement is managed by the 'SDPManager' class together with the Blue-Z SDP server that listens continuously for service discovery requests from the other Bluetooth devices. The list of services itself is modified upon loading and unloading the plug-ins of the Bluetooth communication module. Service and device discovery are performed periodically by means of a threaded implementation of the 'BluetoothServiceDiscoverer' class. Discoveries are only performed if at least one service plug-in requested the class to be called upon discovery of a certain type of device or service, by registering a filter. Moreover, 'BluetoothServiceDiscoverer' can be configured to warn a service upon the availability of a certain device by means of the paging mechanisms.

The Bluetooth communication module manages its connections with the remote Bluetooth devices through the 'BluetoothConnectionManager' class. This class is used by the Bluetooth services to listen for incoming connections from specific devices on specific ports, and to establish connections with remote devices. It also keeps track of the active connections, in a centralized way, which enables the coordination of the service plug-ins.

This is the authors' version of the paper which appears in Wireless Personal Communications
DOI: 10.1007/s11277-012-0582-x

Table 2. The lower and upper layers of the Bluetooth and Zigbee modules implemented on the interface.

	OSI layers	Bluetooth module	Zigbee module
Upper layers	Application	Blue-Z stack and custom application	Zstack stack, custom application, and OSAL API
	Presentation		
	Session		
	Transport		
		HCI	OSAL
Lower layers	Network	Generic Bluetooth USB dongle and its firmware	CC2531 Zigbee USB dongle and its firmware
	Data link (MAC)		
	Physical		



4.5. Power management

The 'PMS' class implements the various techniques and strategies presented in Section 3. The timers are based on an "alarm" mechanism by which a signal is sent to the relevant process on time-outs. The hardware modules are controlled by accessing their Linux drivers. This is achieved through the virtual file system interface maintained by the kernel. For example, the "/sys/class/backlight/atmel/" directory contains files for configuring the brightness of the backlight and the "/sys/devices/system/cpu/cpu0/cpufreq/" files for scaling the frequency and voltage of the CPU.

4.6. Reactivity considerations

This subsection discusses possible sources of delays in the proposed framework and their impact on the reactivity of the interface. The selected operating system (Ångström Linux) as well as the selected GUI (Qt for Embedded Linux) are both lightweight and have been configured so as to

This is the authors' version of the paper which appears in Wireless Personal Communications
DOI: 10.1007/s11277-012-0582-x

minimize their respective overheads in terms of CPU usage and memory footprint. Doing so ensures that sufficient resources are available for the services that take care of the communication protocols, device management, and user input management. Regarding the interface reactivity, we consider the following two cases.

The first case is when a device (appliance) appears or reappears in the user's environment. The delay (latency) for a device to be discovered and to become available for the user depends on several factors. The first one, enumeration, is protocol dependent. For example, the theoretical enumeration latency for Bluetooth devices is 10.24s and that of Zigbee is about 30ms. Note that the actual time for a device to be discovered also depends on its distance to the interface, the physical obstacles that might attenuate the radio signal between the device and the interface, as well as the number of devices to be discovered 'simultaneously'. Furthermore, the discovery times are typically shorter (e.g. ca. 3s instead of ca.10.24s for the Bluetooth case) when a device has already been discovered in the past and its parameters have been stored in the interface.

Furthermore, as indicated in Section 3.7, the energy saving strategy that we have implemented makes use of the energy saving facilities supported by the protocols. These have an impact on the discovery and availability delays. In the current implementation it is possible to set the radio modules of the interface into sleep or idle modes, and to tune the discovery period.

When setting a radio module into sleep mode, one has to consider the wakeup times (ca. 3s for Bluetooth and ca. 15ms for Zigbee). A radio module is set to sleep only when it is not used in any active connection. Alternatively, a radio module can be set to idle (i.e. a low power mode) since the wake-up time when exiting the idle mode is lower as compared to when exiting the sleep mode. Regarding the discovery period, for the targeted users the environment is not

This is the authors' version of the paper which appears in Wireless Personal Communications
DOI: 10.1007/s11277-012-0582-x

expected to change very rapidly, so in order to save energy it is possible not to search for new devices continuously. In the current implementation the discovery period can be either automatically tuned according to several factors such as the number of recently discovered devices and how often they are used, or it can be manually set by the user.

The second case is when switching from one device to another one. In this case, the main source of possible delay is related to radio collision and the protocol/plugin manager. Radio collision can happen when several radio modules are operating next to each other, using the same frequencies. However, most protocols provide mechanisms for specifying which frequencies to use or to avoid. In the current implementation, the protocol plugin takes advantage of these mechanisms and provides a negotiation mechanism that reduces radio collision. The negotiation phase introduces some delay, mostly depending on how many devices the user wants to switch among; however, negotiation should not take place very often as the number of devices in the user's environment does not change very rapidly for the application scenario. Also note that since the various functionalities of the interface are implemented as lightweight threads, switching between devices is easily handled by the selected microprocessor.

The various delays introduced by the operating system/GUI, the communication protocols, and the energy saving strategy are relatively small, and given the application scenario, we consider that these various delays are either negligible or not penalizing seen from the targeted user's point of view. This is confirmed by our observations reported in Section 5.

This is the authors' version of the paper which appears in Wireless Personal Communications
DOI: 10.1007/s11277-012-0582-x

5. Experimental results

5.1. Experimental Setup

The experimental setup is composed of the interface (i.e. the platform described in Section 4) and of several devices placed in the environment of the user. These devices are: 1) the two CC2530EM Zigbee evaluation modules fitted on the SmartRF05EB boards; their respective LEDs are used to emulate Zigbee-controlled lights; 2) a lamp connected to a Cleode Zplug (a ZigBee power plug), 3) a Linux-based laptop (Blue-Z Bluetooth stack); and 4) two Windows-based laptops with Widcomm and Microsoft Bluetooth stacks, respectively. A photograph of the setup (except the laptop running the Microsoft Bluetooth stack) can be seen in Figure 8.

Please note that the following experiments have been conducted by abled bodied subjects, using the touch screen as the access pathway. Clinical trials with motor system impaired persons, using the tongue control unit as the access pathway, have yet to be conducted and are not described in this paper.

This is the authors' version of the paper which appears in Wireless Personal Communications
DOI: 10.1007/s11277-012-0582-x

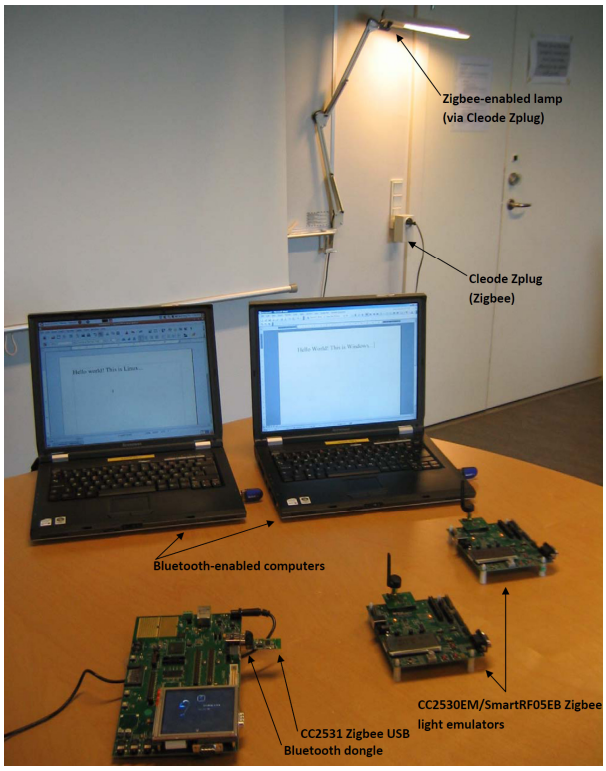


Figure 8. Photograph of the setup. The CC2531 Zigbee USB dongle provides Zigbee connectivity to the interface prototype. The two CC2530EM Zigbee modules/SmartRF05EB boards are used as a Zigbee-enabled light emulators and a desk lamp is Zigbee-enabled by means of a Cleode Zplug. Bluetooth connectivity is provided by the Bluetooth dongles.

5.2. Experimental Results

The following experiments have been conducted to verify that the interface effectively allows the user to discover and control the above-mentioned devices (appliances).

5.2.1. Discovering and controlling individual appliances

Controlling a Bluetooth-enabled computer: the goal of this experiment is to verify that it is possible to discover, connect to, and control a Bluetooth-enabled computer by using the Bluetooth HID Profile to emulate a Bluetooth mouse and keyboard. The HID Profile makes the Bluetooth HID peripherals to provide a service to the Bluetooth HID hosts; thus, the initial setup begins on the computer side. The three computer configurations listed in Subsection 5.1 have been used. For

This is the authors' version of the paper which appears in Wireless Personal Communications
DOI: 10.1007/s11277-012-0582-x

each configuration, the user activates the HID service displayed in the list of services of the target computer. If the computer has never been connected to before, the user sees a popup message requesting a security password (this password is displayed on the interface screen). Once authenticated, a 'virtual' keyboard or mousepad is displayed on the prototype's screen: the two devices are now connected and ready to use.

The user can control the mouse cursor or type some text. We observe that the three (four when authentication is carried out) phases of the experiment are successful on the three configurations. The interface is discovered by the three Bluetooth stacks and can connect to the computers. Using these stacks, the user can successfully control the corresponding computers. We also observe that there is no visible delay between the commands being inputted on the interface and their effects on the computer; this is consistent with the discussion in Section 4.6.

The next experiment consists in turning Zigbee-controlled lights on and off: the goal is to verify that the proposed interface can be used to turn Zigbee-enabled lights on and off. The two CC2530EM/SmartRF05EBs are programmed with Texas Instruments' Zigbee light emulator (compatible with the Zigbee Home Automation profile). Moreover, a desk lamp is Zigbee enabled by means of a Cleode Zplug. Once the three lights have been detected and added to the list of available appliances, the lights are repetitively switched on and off at non-steady intervals. The lights respond smoothly to the commands entered by the user on the interface and no visible delay can be observed; this is also consistent with the discussion in Section 4.6.

5.2.2. Controlling several appliances using the same protocol

The goal of this set of two experiments is to verify that the interface allows the user to control several appliances using the same protocol simultaneously, i.e. the user can switch rapidly between the controls of several appliances without affecting their respective states. The setup for the first

This is the authors' version of the paper which appears in Wireless Personal Communications
DOI: 10.1007/s11277-012-0582-x

experiment consists of one computer with Microsoft Windows XP/Widcomm Bluetooth stack and one computer with Linux Ubuntu/Blue-Z Bluetooth stack. The setup for the second experiment consists of the three lights used in the previous experiment. In the first experiment, the user alternatively takes control of the two computers through the proposed interface as follows: connect to the first computer and use it, switch to the icon of the second one on the GUI, connect to the second one and use it, switch back to the first one, and so on and so forth. The experiment shows that the user can easily switch between the two Bluetooth-enabled computers and that there is no visible delay related to the switching operation; this is consistent with the discussion in Section 4.6. Similarly, in the second experiment the user alternatively takes control of the three Zigbee lights. Again, the user can easily switch between the appliances and there is no visible delay related to the switching operation; this is also consistent with the discussion in Section 4.6.

5.2.3. Controlling several appliances using different protocols

With this experiment, we seek to verify that the interface can detect, connect to and control several appliances using different protocols simultaneously, without affecting their respective states. Moreover, the experiment illustrates how the interface abstracts, for the user, the various appliances and their services, irrespectively of their protocols. In this experiment, we use the two Bluetooth-enabled computers and the three Zigbee lights. The user alternatively takes control of the computers and of the lights by connecting to and using one of the computers, switching to one of the lights and turning it on and off, switching to the other computer, and so on and so forth. The experiment shows that the user can easily switch between all the appliances, and that there is no visible delay related to the switching operation; this is consistent with the discussion in Section 4.6.

This is the authors' version of the paper which appears in Wireless Personal Communications
DOI: 10.1007/s11277-012-0582-x

5.2.4. Dynamic Adaptation to the user's environment

Finally, the last set of experiments consists in verifying that the interface can dynamically adapt itself to the user's environment, i.e. update the list of appliances and services when e.g. the user moves from one location to another one, or when an appliance is removed by a third party from his/her vicinity. The experiments consist in turning the appliances on and off, as well as modifying the distance between the interface and the appliances without modifying their current states (e.g. on/off, service configurations). These are repeated at non-steady intervals: short (ca. 10 seconds), moderate (3 minutes) and long (10 minutes) ones. The experiments show that whenever one of the five appliances is turned on and is within reach, it is successfully detected by the interface and that a notification informs the user about availability of the appliance and of its services. Similarly, whenever one of the five appliances is turned off or becomes out of reach, it and its services are successfully removed from the list. Moreover, the user is informed by means of a non-blocking notification message. On the contrary, whenever an appliance is removed while it is being used by the user, a blocking message informs the user about the unavailability of the corresponding service(s).

5.2.5. Power measurements

In order to evaluate the impact of the PMS we have measured the power on the main board in different modes. At boot-time, the power reaches 2400mW. Once the system has booted, and with the backlight turned on and maximum CPU frequency, the power is stable at 2244 mW. Turning off the backlight or scaling to the minimum frequency brings the power down to 1656mW and 1620mW, respectively. Finally, combining the two above techniques brings the power down to 1080mW, i.e. 2.22 less than in the full mode. Note that these measures include components (e.g. audio) that are not used for the interface; hence a custom-made board would require less power.

This is the authors' version of the paper which appears in Wireless Personal Communications
DOI: 10.1007/s11277-012-0582-x

6. Discussion and Outlook

The experiments show that the proposed interface effectively enables its user to remotely and easily control several types of appliances available in his/her environment through different types of WPANs. The interface enables the user to easily switch between appliances and their respective services, without interfering with each other. Moreover, the interface is context-aware, i.e. it successfully adapts itself to the user's environment by detecting remote appliances that enter or leave the interface coverage range and informs the user about those changes by means of the graphical user interface. The prototype, constructed as an embedded system, is relatively inexpensive, small, and power efficient. Suggestions for future work are listed in what follows. Firstly, other plug-ins (e.g. Wi-Fi, 6LowPan) could be added and experiments carried out to evaluate how the interface handles more than two protocols simultaneously. Secondly, experiments should be conducted with more appliances from multiple vendors to verify the robustness and compatibility of the proposed interface. Thirdly, although usability has been considered during the design of the interface, in-field clinical experiments with the targeted user group, using the mouth control unit as the access pathway, should be conducted. Finally, converting and merging the development kits into a custom board would result in a further optimized system in terms of size, power, and price.

References

[1] Simunic D. & and Djurek M. (2009). Transdisciplinarity of smart health care: transmedical E-volution. Wireless Personal Communications, 51(4), 687-695.

[2] Tai K., Blain S. & Chau, T. (2008). A review of emerging access technologies for individuals with severe motor impairments. Assistive Technology, 20(4), 204-219.

This is the authors' version of the paper which appears in Wireless Personal Communications
DOI: 10.1007/s11277-012-0582-x

[3] Craig A., Tran Y., Mcisaac P. & Boord P. (2004). The efficacy and benefits of environmental control systems for the severely disabled. Med Sci Monit, 11(1), RA32-39.

[4] Louise-Bender Pape T., Kim J. & Weiner B. (2002). The shaping of individual meanings assigned to assistive technology: a review of personal factors. Disability and Rehabilitation, 24(1-3), 5-20(16).

[5] Ville I., Crost M., Ravaud J.F. & Tetrafigap Group (2003). Disability and a sense of community belonging A study among tetraplegic spinal-cord-injured persons in France. Social Science & Medicine, 56(2), 321-332.

[6] Andreassen Struijk L. N. S. (2006). An inductive tongue computer interface for control of computers and assistive devices. *Biomedical Engineering, IEEE Transactions on.* 53(12), 2594-2597.

[7] Lontis, R. & Andreassen Struijk, L. N. S. (2010). Design of inductive sensors for tongue control system for computers and assistive devices. *Disability and Rehabilitation: Assistive Technology.* 5(4), 266-271.

[8] Clayton C, Platts R. G. S, Steinberg M., & Hennequin J.R. Palatal tongue controller (1992). *J. Microcomput. Appl.* 15(2), 9-12.

[9] Buchhold N. (1995). Apparatus for controlling peripheral devices through tongue movement, and method of processing control signals. United States Patent no. 5,460,186.

[10] Peng Q. & Budinger T. (2007). ZigBee-based wireless intra-oral control system for quadriplegic patients. *29th annual international conference of the IEEE engineering in medicine and biology society.* August 23-26, Lyon.

This is the authors' version of the paper which appears in Wireless Personal Communications
DOI: 10.1007/s11277-012-0582-x

[11] Huo X. & Ghovanloo M. (2009). Using unconstrained tongue motion as an alternative control mechanism for wheeled mobility. *IEEE Transactions on Biomedical Engineering*. 56(6), 1719-26.

[12] Kirbis M. & Kramberger I. (2009). Mobile device for electronic eye gesture recognition. *Consumer Electronics, IEEE Transactions on.* 55(4), 2127-33.

[13] Ververidis C.N. & Polyzos G.C. (2008). Service discovery for mobile Ad Hoc networks: a survey of issues and techniques. *Communications Surveys & Tutorials, IEEE.* 10(3), 30-45.

[14] Xu L., Santoro N. & Stojmenovic I. (2009). Localized distance-sensitive service discovery in wireless sensor and actor networks. *Computers, IEEE Transactions on.* 58(9), 1275-88.

[15] Lankri S., Berruet P. & Philippe J-L. (2009). Multi-level reconfiguration in the DANAH assistive system. *IEEE international conference on systems, man and cybernetics.* October 11-14, San Antonio.

[16] Bronsted J., Madsen P.P., Skou A. & Torbensen R. (2010). The HomePort system. *7th IEEE consumer communications and networking conference*. January 9-12, Las Vegas.

[17] Gill K., Shuang-Hua Y., Fang Y. & Xin L. (2009). A zigbee-based home automation system. *Consumer Electronics, IEEE Transactions on.* 55(2), 422-30. Woo Suk L. & Seung Ho H. (2009).

[18] Woo Suk L. & Seung Ho H. (2009). Implementation of a KNX-ZigBee gateway for home automation. *IEEE 13th international symposium on consumer electronics.* May 25-28, Braunschweig.

[19] Young-Guk H. (2009). Dynamic integration of zigbee home networks into home gateways using OSGI service registry. *Consumer Electronics, IEEE Transactions on.* 55(2), 470-6.

This is the authors' version of the paper which appears in Wireless Personal Communications
DOI: 10.1007/s11277-012-0582-x

[20] Guangming S., Yaoxin Z., Weijuan Z. & Aiguo S. (2008). A multi-interface gateway architecture for home automation networks. *Consumer Electronics, IEEE Transactions on.* 54(3), 1110-13.

[21] Cheng S. & Wang C. (2010). An adaptive scenario-based reasoning system across smart houses. Wireless Personal Communications, November 2010.

[22] Chou J., Weng S. & Chen C. (2009). Action patterns probing for dynamic service composition in home network. Wireless Personal Communications, 51(1), 137-151.

[23] Kulkarni D. & Tripathi A. (2010). A framework for programming robust context-aware applications. *Software Engineering, IEEE Transactions on.*36(2), 184-97.

[24] Mathieu B. (2009). A context-aware network equipment for dynamic adaptation of multimedia services in wireless networks. Wireless Personal Communications, 48(1), 69-92.

[25] Atmel (2009). AT91SAM9263-EK Evaluation Board User Guide.

[26] Texas-Instruments (2009). CC2530 Development Kit User's Guide.

[27] M. Luca (2010). Ångström Manual - Embedded Power -.

[28] Nokia (2009). Qt for Embedded Linux White Paper.

This is the authors' version of the paper which appears in Wireless Personal Communications
DOI: 10.1007/s11277-012-0582-x

Chapitre 10

Projet "FPGAs in Space - Low-Power Signal Processing Capacity for Nano-Satellites"

10.1 Contexte

Ce projet est le résultat de discussions entre Department of Electronic Systems, Aalborg University, et l'entreprise danoise Gomspace ApS (crée en 2007 par d'anciens étudiants de Aalborg University et cotée au NASDAQ OMX Nordic depuis juin 2016 via GS Sweden AB). Gomspace ApS développe et fournit des services et équipements pour les mini-satellites, allant des sous-systèmes tels que modules de calculs et de communication, panneaux solaires et caméras jusqu'à une plateforme complète.

Lors de ces discussions, les responsables et ingénieurs de Gompace ApS nous avaient fait part de leur intérêt croissant pour utiliser la technologie FPGA à bord des mini-satellites car cette technologie offre un bon rapport entre puissance de calcul, reconfigurabilité et prix (voir par ailleurs le projet "Global Air Traffic Awareness and Optimization through Space Based Surveillance (GATOSS)" décrit dans le chapitre 11 pour un cas concret).

Cependant, la puissance électrique disponible sur les mini-satellites tels que ceux visés par Gomspace ApS est limitée, typiquement 2 W en continu pour un Cubesat de deux unités, ce qui laisse peu de marge pour le FPGA une fois soustraite la consommation des équipements radio et module de correction de l'attitude. Il est donc nécessaire de faire baisser la consommation du FPGA, notamment via des techniques d'ajustement dynamique de la fréquence et de la tension.

Ces travaux visaient à mieux comprendre et à exploiter le comportement des FPGA lorsque ceux-ci sont sous-alimentés (c.à.d. tension d'alimentation inférieure à tension nominale), notamment sa nature stochastique qui est liée aux erreurs qui résultent de cette sous-alimentation. Ceci constitue le point de départ de la thèse de doctorat d'Alex Birklykke "Modeling and Predicting the Behavior of Computers Operating without Guard-Bands — An Experimental Approach Based on Voltage-Scaled FPGAs" (co-encadrée par Peter Koch (AAU), Ramjee Prasad (AAU) et Lars Alminde (Gomspace ApS)).
Je remercie aussi Rakesh Kumar du groupe PASSAT, University of Illinois, Urbana-Champaign, USA, pour avoir accueilli et encadré Alex lors de son séjour de recherche

de six mois (automne-hiver 2012-2013).

10.2 Problématiques

10.2.1 Problématique de la compréhension de la causalité des erreurs

Le premier point que nous avons étudié est l'identification des sources qui peuvent créer des erreurs dans un circuit CMOS (dans notre cas un FPGA). Nous nous sommes notamment intéressés aux erreurs liées à l'ajustement dynamique de la fréquence et de la tension (notamment sous-alimentation), au bruit de l'alimentation, et de manière beaucoup moins directe aux erreurs dues aux radiations et aux variabilités inter- et intra-puces.
Les questions abordées sont d'une part celle de la conception et de l'implantation d'un banc d'essai permettant de gérer ces facteurs et d'autre part celle du développement d'une méthode permettant d'évaluer la dépendance, au niveau logique, entre les propriétés statistiques des signaux appliqués au circuit et le taux d'erreur et l'hypothèse que l'essentiel des erreurs est de type timing (*timing errors*).

10.2.2 Problématique de la modélisation de la prédiction des erreurs

Une fois le lien entre causes et effets établi, ce deuxième point vise à proposer des modèles permettant de prédire 'quand' (c.à.d. dans quelles conditions) les erreurs se produisent dans les FPGA sous-alimentés. Nous avons notamment cherchés à développer une méthode permettant de quantifier les délais de propagations dans les circuits à implanter sur FPGA et à utiliser celle-ci pour vérifier si il est possible de normaliser les estimations obtenues à partir des outils constructeurs, l'objectif étant de permettre aux concepteurs d'estimer les délais à différentes tensions d'alimentation.

10.2.3 Problématique de la généralisation de la prédiction des erreurs

Une fois les deux points précédents traités, se pose la question de la généralisation de la prédiction des erreurs à des circuits numériques arbitraires. Les deux sous-questions auxquelles nous nous sommes intéressées sont d'une part l'analyse du comportement des circuits numériques soumis à des erreurs de timing et d'autre part un modèle et une approche permettant d'approximer le taux d'erreur pour un circuit arbitraire pour lequel on ne connaît que la fonction de transfert binaire, la tension d'alimentation et la fréquence d'horloge. Bien que la précision des résultats soit limitée par la variance des estimations, nous proposons une approche permettant de modéliser les erreurs de timing de manière déterministe et non aléatoire comme généralement fait dans la littérature.

10.2.4 Problématique de l'exploitation de la prédiction des erreurs

Veuillez noter que je n'ai pas directement contribué à cette partie ; je la mentionne pour information. Cette partie a été traitée par Alex Birklykke et une partie de l'équipe PAS-

SAT mentionnée plus haut.

Ce dernier point vise à rendre les systèmes numériques résistants aux erreurs via une approche innovante. La méthode proposée cherche à rendre les applications sous une forme de type chaîne de Markov, c.à.d. permettant de chercher une solution de manière aléatoire contrôlée. Les résultats montrent qu'une fois représentées sous cette forme, les applications sont très robustes aux erreurs (taux d'erreurs entre 10 et 20% pour un surcoût en temps d'exécution mineur). Le lecteur intéressé par cette partie peut se référer au chapitre II.E de la thèse de doctorat d'Alex Birklykke.

10.3 Résumé des contributions

Avant d'entrer dans les détails, je signale ici que dans les contributions présentées dans ce chapitre, la notion de comportement s'applique au niveau RTL.

10.3.1 Contribution à la compréhension de la causalité des erreurs

Conception et implantation du banc d'essai

Afin de mieux comprendre le comportement d'un FPGA sous-alimenté tout en tenant compte de l'impact d'autres facteurs, nous avons avons conçu un banc d'essai permettant de comparer les valeurs de sorties de deux FPGA dont les entrées reçoivent le même signal de stimulus, mais où des erreurs sont artificiellement générées pour un des deux. Nous définissons $\mathcal{F}$ comme la fonction de transfert attendue (FPGA sans erreur) et $\hat{\mathcal{F}}$ comme la fonction de transfert pour le FPGA soumis aux erreurs via les facteurs contrôlables (voir figure 10.1). Les deux FPGA sont soumis aux mêmes facteurs non-contrôlables (voir figure 10.2).

Factor	Action	Remark
Supply voltages		
- Core	Design factor	0.6-1.2 V
- IO	Held-constant	2.5 V
- Auxiliary	Held-constant	3.3 V
Operating frequency	Design factor	20-80 MHz
Input signal	Design factor	$\Pr(x = 1) \in [0, 1)$
RTL design	Design factor	Held-constant for each experiment
- Logic type	Held-constant	Synchronous
- Logic transfer function	Design factor	
Device type/vendor	Held-constant	Spartan 3E (XC3S500E-PG208)
Technology node	Held-constant	90 nm
IOB allocation	Held-constant	
CLB allocation	Allowed-to-vary	Dictated by the place-and-route algorithm in the tool-chain
Ambient temperature	Allowed-to-vary	Air-conditioned room at approx. 20 C

Figure 10.1: Facteurs contrôlables. Extrait de la thèse de doctorat d'Alex Birklykke.

Le banc de test est schématisé dans la figure 10.3 ; une photographie est disponible dans la publication C.21 listée dans le chapitre 5.

Factor	Type	Remark
Intra-die variations	Noise-factor	
Die-temperature	Noise-factor	Measurable, but not controllable
Electromagnetic interference (EMI)	Noise-factor	Controllable via RF generators and shielding but not measurable
Thermal noise	Noise-factor	Indirect control via active heating/cooling
Charged particles	Noise-factor	Indirect control via shielding/radioactive sources.
PSU noise	Controllable	By reducing thermal noise and EMI, or by purposely adding noise
Routing	Noise-factor	Tool-chain dependent
Unknown influential factors	Noise-factors	

Figure 10.2: Facteurs non-contrôlables. Extrait de la thèse de doctorat d'Alex Birklykke.

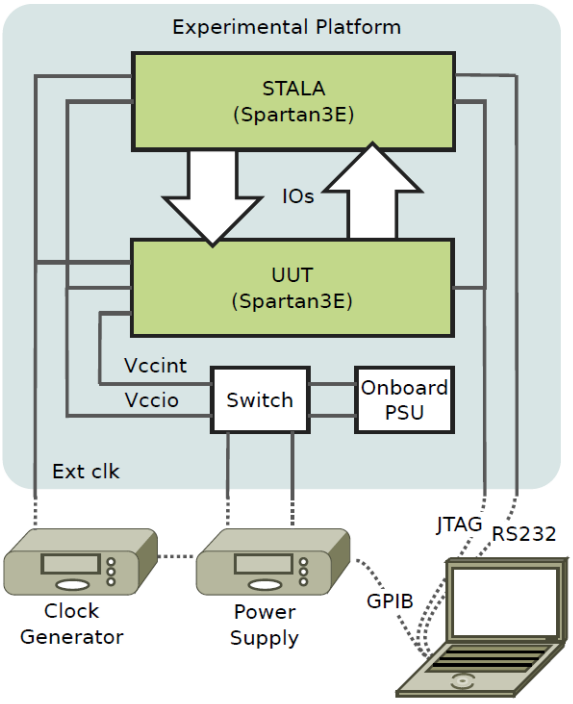


Figure 10.3: Vue schématisée du banc de test. Extrait de la publication C.21.

Le premier FPGA est celui qui est soumis aux erreurs, nous y référons en tant que UUT (Unit Under Test). Le second, que nous avons appelé STALA (pour STAtistical Logic Analyser) génère les vecteurs de test (envoyés au FPGA UUT) et la réponse sans erreur. Il effectue aussi l'analyse statistique (comparaison des résultats avec et sans erreurs). Les facteurs tension, fréquence, configuration du FPGA et propriétés statistiques des vecteurs de test sont contrôlables. La tension et la fréquence sont fournies par des sources externes commandées via le bus GPIB et la configuration du FPGA se fait via le bus JTAG. Le câblage fait que la gestion dynamique s'applique seulement au FPGA UUT ; cependant, il n'élimine pas les interférences éventuelles. De plus, le banc permet d'automatiser les expérimentations au moyen d'un programme Python qui génère les scripts pour la synthèse FPGA, le lancement des expériences et l'analyse des résultats.

Causalité des erreurs : théorie et pratique

Nous avons évalué deux modèles théoriques issue de la littérature, à savoir un modèle d'erreur de seuil (*threshold error model*) et un modèle d'erreur de timing (*timing error model*) [15], [16], [17] et les avons comparés avec les résultats pratiques obtenus avec le banc d'essai décrit plus haut.
Le modèle d'erreur de seuil représente les cas où le bruit présent dans l'environnement perturbe les signaux numériques au point où des erreurs se produisent (une valeur haute ('1') devient une valeur basse ('0') ou vice-versa). Elles sont plutôt rares lorsque la tension d'alimentation est proche de la valeur nominale, mais peuvent se produire plus facilement lorsqu'elle est abaissée.
Nous avons montré dans la publication C.20 listée dans le chapitre 5 que la si fonction de transfert $\mathcal{F}$ représente un inverseur ou une chaîne de buffers, alors le taux d'erreur d'un système avec des erreurs de seuil peut se modéliser comme suit :

$$p_e = (1-p)\alpha + p\beta$$

avec α le taux de faux positifs, β le taux de faux négatifs et $p = Pr(X = 1)$ la probabilité de valeur de bit à l'entrée du circuit.

Nous avons aussi observé que pour des valeurs de α et β, le taux d'erreur est linéairement dépendant de p. En conséquence, si le taux d'erreurs observé pour un FPGA sous-alimenté est aussi linéairement dépendant de p, alors il est probable que les erreurs de seuil soient la principale cause d'erreur.
D'un autre côté, le modèle d'erreur de timing représente les cas où une violation de timing se produit, soit à cause de la sous-alimentation, soit du bruit, soit d'une combinaison des deux. Une violation de timing se produit quand le délai du chemin (critique) excède la période d'horloge. Dans ce cas, le signal est soit lent à monter soit lent à descendre. Dans les deux cas, la conséquence est que la bascule (*latch*) qui suit la logique combinatoire mémorise l'ancienne valeur au lieu de la nouvelle et correcte. En conséquence, l'effet des erreurs de timing est que le signal devient temporairement bloque sur d'anciennes valeurs. Dans la publication C.20 nous avons montré que le taux d'erreur dans le cas d'erreurs de timing peut se modéliser comme suit :

$$p_e = p(1-p)(\alpha+\beta)(1-\rho)$$

avec α et β respectivement les probabilités que le signal soit lent à monter ou lent à descendre. ρ décrit la corrélation temporelle entre les bits adjacents dans le vecteur de test.

Contrairement au modèle précédent, nous observons que le taux d'erreur dans le case d'erreurs de timing dépend de ρ selon un polynôme du deuxième ordre. En conséquence, si le taux d'erreurs observé pour un FPGA sous-alimenté montre cette même dépendance à ρ, alors il est probable que les erreurs de timing soient la principale cause d'erreur.

Enfin, avec ces deux modèles en tête, nous avons évalué la cause des erreurs dans le FPGA UUT. La publication C.20 contient tout un ensemble de graphiques illustrant le taux d'erreur dans différents cas ; la figure 10.4 en est un exemple. Le taux d'erreur est clairement dépendent de la probabilité de bit en entrée de manière polynomiale ; nous concluons donc que les erreurs sont essentiellement de type timing.
Nous observons aussi l'augmentation soudaine du taux d'erreur (vers 50 mV), ce qui indique que le phénomène est dichotome (soit il y a des erreurs, soit il n'y en pas). De plus, le taux d'erreur est positif seulement lorsque $0 < p < 1$, c.à.d. que la violation de timing n'est qu'une condition nécessaire aux erreurs ; il faut aussi que le chemin où elle a lieu soit actif (*sensitized*). Enfin, nous observons que le timing du circuit est le facteur qui lie la tension et les erreurs. Ainsi, un prérequis pour prédire les erreurs de timing est la possibilité de prédire le timing en fonction de la tension (voir ce qui suit.)

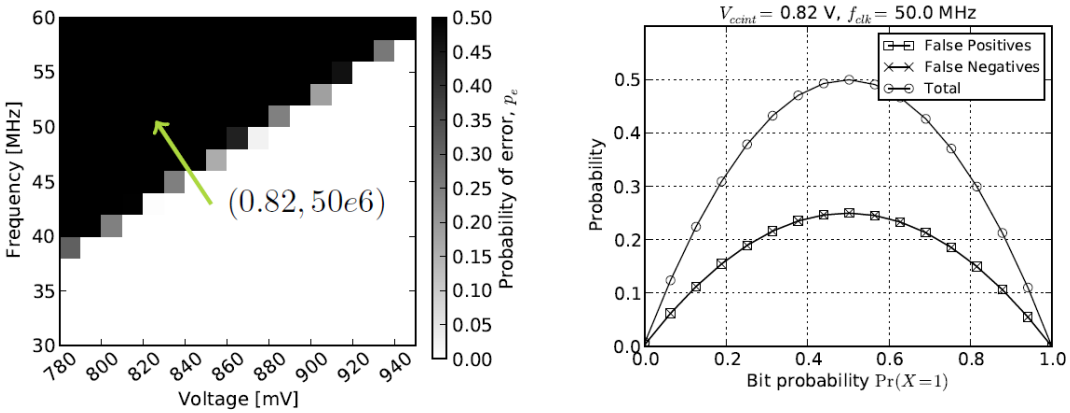


Figure 10.4: Exemple de taux d'erreur pour un FPGA sujet à l'ajustement dynamique de la fréquence et de la tension (sous-alimentation). Extrait de la publication C.20.

10.3.2 Contribution à la modélisation de la prédiction des erreurs

Spécification

Ce premier paragraphe résume la spécification (niveau RTL) utilisée dans les paragraphes suivants pour la modélisation de la prédiction des erreurs.

La structure d'un circuit logique synchrone ayant N bits en entrée et M bits en sortie est résumée à un bi-graphe complet $G = (X, Y, E)$ avec $X = (x_1, ..., x_N)$ et $Y = (y_1, ..., y_N)$ représentant les nœuds d'entrées/sorties et $E \subset X \times Y$ les arcs qui les relient. Les nœuds prennent des valeurs dans $\mathbb{B} := \{0, 1\}$, X dans $\mathbb{B}^N$ et Y dans $\mathbb{B}^M$.

Le comportement statique du circuit est défini comme la fonction de transfert binaire $\mathcal{F} : \mathbb{B}^N \to \mathbb{B}^M$ (cas sans erreur).

Son comportement dynamique est défini comme le délai de propagation pire-cas entre les nœuds d'entrée et de sortie. Pour chaque arc $e \in E$ le délai pire-cas est spécifié par le poids $d(e)$ (obtenu via une analyse de timing statique). Ainsi, les estimations des délais sont relatives à la tension d'alimentation nominale mais biaisées à cause de l'approche pire-cas.

La tension d'alimentation est notée u et la période d'horloge t_c.
La probabilité de bit en entree est notée $p_i = Pr(x_i = 1)$ et son coefficient de corrélation par $\rho_i = cov(x_i[\,k\,]\,, x_i[\,k+1\,])$.
On présume que les nœuds d'entrée adjacents sont indépendants tel que $cov(x_{i1}, x_{i2}) = 0$ quand $i_1 \neq i_2$. Ensemble, $u, t_c, \mathcal{F}$ et les propriétés statistiques des entrées constituent les quatre facteurs *design factors* de la figure 10.1.

Prédiction de timing

Dans la thèse de doctorat d'Alex Birlykke (partie II.C, non publiée en conférence ou revue) nous avons proposé une méthode de prédiction permettant d'étendre les estimations de timing fournies par les outils constructeurs à des valeurs de tensions arbitraires. L'objectif est d'obtenir une estimation $\tilde{d}(e, u)$ de la valeur réelle (dépendante de la tension) du délai $d(e, u)$. Ne connaissant que la valeur (biaisée) $d(e, u)$ pour la tension nominale u_{nom}, nous avons montré (voir II.C dans la thèse de doctorat d'Alex Birklykke pour les détails) qu'il existe une fonction dite de normalisation $h(u)$ telle que $\tilde{d}(e, u) = d(e)h(u)$. Cette fonction compense à la fois les changements de tension d'alimentation et les bias introduit par les outils d'analyse de timing statiques. Cette fonction est obtenue empiriquement en ajustant les valeurs de timings observées $d_{obs}(e, u)$ normalisées avec les estimations de timing $d(e)$ à l'aide un polynôme d'ordre 3 (qui représente $h(u)$) en utilisant une régression moindres carrés pondérée.

La fonction de normalisation est une solution au problème suivant :
$h(u) = \mathbb{E}[\,\frac{d_{obs}(e,u)}{d(e)}]$
Afin d'obtenir les observations de timing nous exploitons le fait que les erreurs de timing se produisent quand le délai est approximativement égal à la période d'horloge. La valeur de cette dernière est utilisée comme mesure lorsque la première erreur due à l'ajustement de fréquence (par rapport à une tension u et un chemin e) se produit. De plus, la fonction de normalisation obtenue pour une chaîne d'inverseurs peut aussi être utilisée pour prédire le timing de circuits plus complexes, comme par exemple pour un additionneur 4 bits (figure 10.5).
Pour une tension u et une période d'horloge t, l'ensemble des chemins avec des violations de timing est donné par :

$$C(u, t) = \{e \in E : t < d(e)h(u)\}$$

Prédiction du taux d'erreur

L'ensemble des transitions *one-step* est donné par $S = \mathbb{B}^{\mathbb{N}} \times \mathbb{B}^{\mathbb{N}}$; pour une tension u et une période d'horloge t, le sous-ensemble des transitions qui activent les chemins causant des erreurs est donné par $\Gamma(u, t) \subset S$. Le taux d'erreur peut alors être trouvé en calculant la probabilité d'observer la transitions fautives dans $\Gamma(u, t)$.
Dans la thèse de doctorat d'Alex Birklykke (partie II.D, non publiée en conférence ou revue), nous montrons que celle-ci est donnée par :

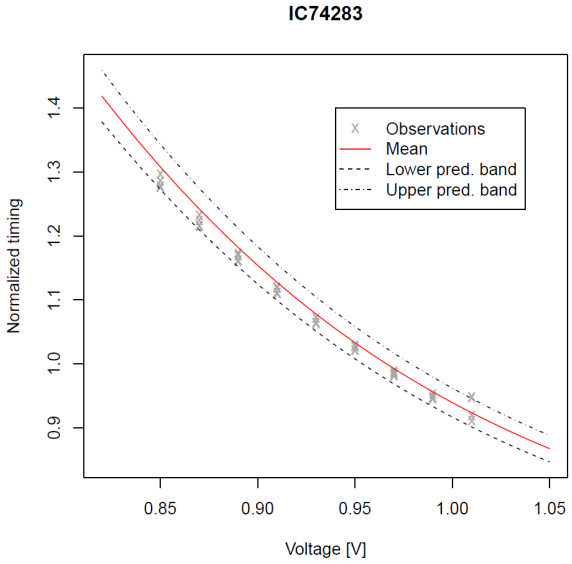


Figure 10.5: Observations de timing normalisées pour un additionneur 4 bits ajusté en tension, valeurs idéales et intervalle de prédiction à 95%. Extrait de la thèse de doctorat d'Alex Birklykke.)

$$p_e(u,t) = \sum_{(A,B)\in\Gamma(u,t)} \prod_{i=1}^{n} p_i^{a_i+b_i}(1-p_i)^{2-a_i-b_i} + (-1)^{2-a_i-b_i}\rho_i p_i(1-p_i)$$

avec $A = \{a_i : i \in (1...N)\}$ et $B = \{b_i : i \in (1...N)\}$ les vecteurs de transitions et p_i et ρ_i respectivement la probabilité de bit et le coefficient de corrélation du i^{ieme} signal d'entrée. Afin d'obtenir l'ensemble des transitions fautives, il est nécessaire de prédire le *glitching* et donc d'obtenir des informations détaillées sur la structure et le timing du circuit. Pour ce faire, deux procédures ont été proposée (voir II.D dans la thèse de doctorat d'Alex Birklykke pour les détails). Celles-ci donnent une approximation de l'ensemble des transitions fautives à partir de la fonction $\mathcal{F}$ et de l'ensemble des chemins avec violations de timing $C(u,t)$. Les deux méthodes traversent l'ensemble des transitions S et évaluent si une transition active les chemins avec des violations de timing.
Les deux méthodes ont été comparées pour des cas de timing déséquilibrés. Un exemple de résultat (figure 10.6(a) montre une dégradation approximativement linéaire de la justesse en fonction de probabilité que les glitches soient responsable des erreurs. Donc, pour le cas de circuits déséquilibrés, une prédiction très juste du taux d'erreur au niveau RTL n'est pas possible avec les deux procédures proposées.

Malgré cette limitation, les résultats restent assez encourageants lorsque l'on compare l'ensemble approché des transitions fautives avec les observations obtenues via le banc d'essai, comme montré par exemple dans la figure 10.6(b).

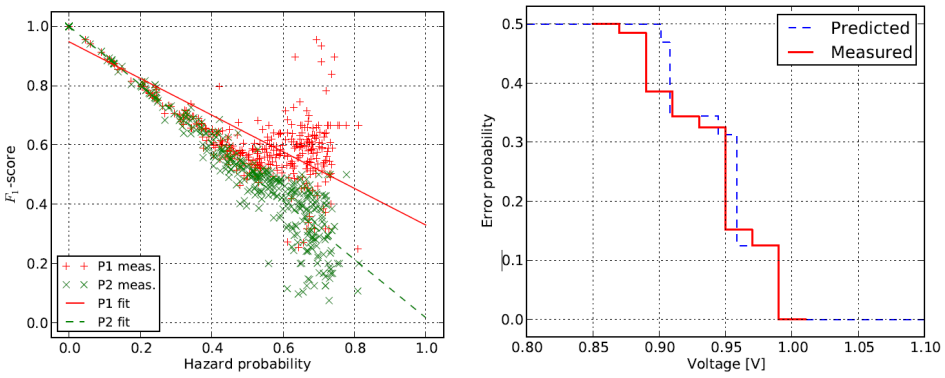


Figure 10.6: (a): Justesse des deux procédures de prédiction. (b) Taux d'erreurs prédit et mesuré pour un additionneur 4 bits synchrone ajusté en tension. Extrait de la thèse de doctorat d'Alex Birklykke.

10.3.3 Contribution à la généralisation de la prédiction des erreurs

Les modèles présentés dans les sous-sections précédentes requièrent des connaissances préalables sur le circuit et certaines propriétés de son implantation. Afin d'essayer généraliser la prédiction d'erreurs, par exemple pour pouvoir l'utiliser dans la phase de conception, nous avons cherché à mieux comprendre, d'un point de vue théorique, le comportement de circuits logiques synchrones arbitraires avec des violations de timing.
Nous avons montré qu'il est possible de décomposer l'ensemble $S = \mathbb{B}^{\mathbb{N}} \times \mathbb{B}^{\mathbb{N}}$ (voir plus haut) en quatre sous-ensembles disjoints comme représentés dans la figure 10.7). S_0 représente toutes les transitions à entrées constantes qui produisent des sorties constantes ; S_1 est vide car aucune transition d'entrée constante peut produire d'activité en sortie (en supposant pas d'erreur due au bruit) ; S_2 représente toute les transitions qui activent le circuit mais produisent des sorties constantes et S_3 représente toutes les transitions avec de l'activité en entrée et en sortie.

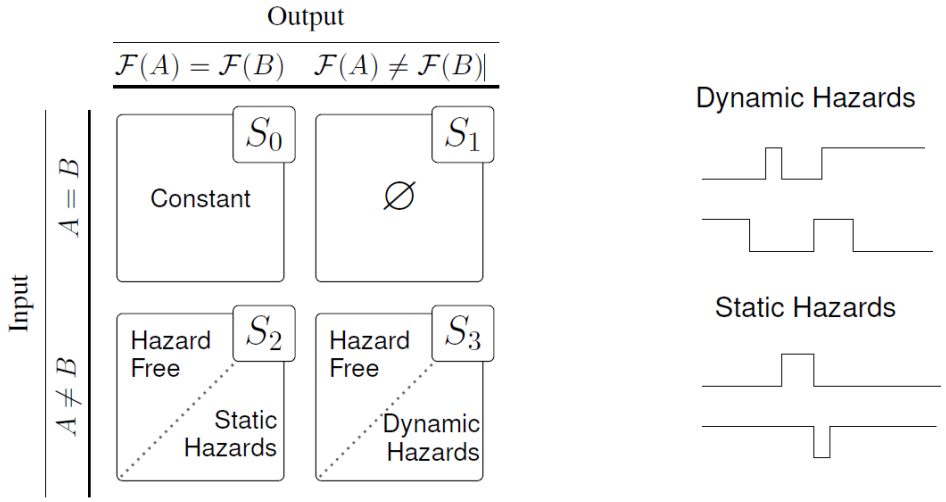
$S_2 \cup S_3$ contient toutes les transitions pouvant activer un circuit ; l'ensemble des transitions fautives en est donc un sous-ensemble lorsque que se produisent des violations de timing. De même, quand tous les chemins ont des violations de timing, l'ensemble des transitions fautives correspond à S_3 car toutes les sorties sont lues avant que le signal ne se soit propagé sur le chemin le plus court. Ainsi, et comme détaillé dans la publication thèse de doctorat d'Alex Birklykke, le contenu de l'ensemble des transitions fautives en fonction de $d_{min}(u)$, $d_{max}(u)$, t et $S1...S3$ peut être décrit comme :

$$\Gamma = \emptyset \text{ si } d_{max(u)<t} \text{ (pas de violation de timing)}$$

$$\Gamma =\subset S_2 \cup S_3 \text{ si } d_{min}(u) \leq t \leq d_{max}(u) \text{ (plusieurs violations de timing)}$$

$$\Gamma = S_3 \text{ si } d_{min} > t \text{ (tout les timings sont violés)}$$

Dans le deuxième cas, soit l'ancienne valeur est lue pour les transitions dans S_3, générant ainsi une erreur *stuck-at*, soit une erreur (hasard) statique est lue pour les transitions dans S_2, générant ainsi un *bit-flip*. Les ensembles correspondants sont notés $\Gamma_{S_3}(u,t)$ et $\Gamma_{S_2}(u,t)$ pour une tension d'alimentation et période d'horloge données.



(a) Partitioning of input transitions

(b) Logic hazards

Figure 10.7: (a) Décomposition de l'ensemble des transitions en fonction des activités d'entrée et de sortie. (b) Erreurs logiques pouvant être observées en sortie. Extrait de la thèse de doctorat d'Alex Birklykke.

Le comportement d'un circuit avec des violations de timings, dont les entrées et sorties au temps k sont $x_k \in \mathbb{B}$ et $y_k \in \mathbb{B}$, et dont la fonction de transfert est $\mathcal{F}$, peut être décrit comme :

$$y_k = \begin{cases} \mathcal{F}(x_{k-1}) \text{ si } (x_{k-1}, x_k) \subset \Gamma_{S_3}(u,t) \\ 1 - \mathcal{F}(x_k) \text{ si } (x_{k-1}, x_k) \subset \Gamma_{S_2}(u,t) \\ \mathcal{F}(x_k) \text{ sinon} \end{cases}$$

Une limitation de l'approche proposée est qu'il est généralement difficile de savoir quelles sont les transitions qui génèrent des erreurs sans expérimentation préalable car les hasards logiques dépendent à la fois de la structure et des propriétés dynamiques du circuit au niveau physique. Même si ces informations sont disponibles, il reste un certain niveau d'incertitude notamment a cause des variations processus, tension et température.

Bien qu'il soit difficile d'aller plus loin dans la généralisation les modèles ci-dessus a cause de leur dépendance à l'implantation, nous concluons ces travaux par quatre grandes observations :

- Pour qu'une violation de timing produise une erreur, le chemin où cette violation se produit doit etre actif. Une violation de timing est une condition nécessaire pas suffisante pour générer une erreur ;
- Dans le cas des erreurs de timing, soit l'ancienne valeur est lue pour les transitions dans S_3 créant ainsi une erreur *stuck-at*, soit un hasard statique est lue pour les transitions S_2 créant ainsi une erreur bit-flip ;
- Dans le cas des erreurs de timing, tous les états de sortie restent accessibles car les transitions dans S_0 ne sont pas affectées par les violations de timing. Ainsi, le

circuit est quand même capable de traiter l'information correctement tant que les signaux d'entrée sont rafraîchis. ;

- Dans le cas où tous les chemins sont sujets à des violations de timing, toutes les transitions d'entrée qui normalement produisent des activités en sorties (c.à.d. toutes les transitions de S_3) produisent des erreurs *stuck-at* à la place.

10.4 Publications et commentaire

Ces travaux ont abouti à deux publications (C.20 et C.21) ainsi qu'à la thèse de doctorat d'Alex Birklykke [18]. Pour information, la partie des travaux d'Alex Birklykke effectuée lors de son séjour dans le groupe PASSAT ont aussi donné lieu à trois autres publications (dont je ne suis pas co-auteur) sur le thème des chaines de Markov.

Les publications C.21 et C.20 sont reproduites dans ce qui suit. La première, *An Automated Test Framework for Experimenting with Stochastic Behavior in Reconfigurable Logic*, présente la plateforme expérimentale ; la seconde *Empirical Verification of Fault Models for FPGAs Operating in the Subcritical Voltage Region* présente l'étude du comportement d'un FPGA opérant de manière sous-alimentée.

À mon sens ces travaux montrent qu'il est possible de contribuer de manière originale sur un thème qui pouvait sembler déjà bien exploré et un peu bouché. En effet, en prenant pour point de départ la sous-alimentation en tension et l'acceptation d'un certain taux d'erreurs, nous avons, d'une certaine manière, remis en question les différentes causes d'erreurs dans les circuits numériques et cherché à mieux comprendre comment elles affectent la transformation de l'information. Nous avons aussi proposé une approche originale pour l'estimation des délais à différentes tensions d'alimentation et pour la modélisation des erreurs de timing.

Authors' version. Appears in the proceedings of IEEE Reconfig 2012. DOI: 10.1109/ReConFig.2012.6416729

An Automated Test Framework for Experimenting with Stochastic Behavior in Reconfigurable Logic

Alex Aa. Birklykke*, Yannick Le Moullec*, Lars K. Alminde‡, and Ramjee Prasad*

* Department of Electronic Systems
Aalborg University, Fredrik Bajers Vej 7, DK-9220 Aalborg Ø
{**alb**, ylm, prasad}@es.aau.dk
‡GomSpace APS
Niels Jernes Vej 10, DK-9220 Aalborg Ø
alminde@gomspace.com

***Abstract*—In this paper, we present an automated test framework for the characterization of stochastic behavior in logic circuits. The framework is intended as a platform for experimenting with and providing statistics on digital architectures given behavioral uncertainties at the gate-level. As an experimental platform, we propose to use an FPGA due to the proven value of reconfigurable architectures in design space exploration. We hypothesize that stochastic behavior can be introduced in FPGAs using external noise sources; a fact that is later confirmed by characterizing the behavior of an FPGA IO block subject to voltage/frequency scaling and V_{dd}-noise. The framework provides easy interfacing with laboratory equipment, design of experiment capabilities and automatic test execution, thus providing a powerful tool for characterizing stochastic behavior in reconfigurable logic.**

I. Introduction

As a result of technology scaling, device variability, timing variances and signal integrity are becoming critical design parameters in the VLSI domain. Previous efforts have mainly focused on avoiding the adverse impact of these factors using large design margins, but the continuous push towards increased circuit density or energy efficient designs have lead to the realization that some amount of errors must be accepted if further progress is to be made [1]. However, by accepting errors, we venture into a new domain where circuit behavior is best described in terms of probability and statistics. This tendency has already taken effect at the transistor-level, where statistical modeling is used extensively to handle process and timing variability [2]. Similarly, concepts such as probabilistic [3] and stochastic computing [4] that address behavioral uncertainties at the logic-level and above have emerged. One of the important steps in the further exploration of this emerging domain is to enable physical experimentation with new architecture designs. This would allow researchers to explore how behavioral uncertainties at the gate-level propagate to higher levels of abstraction and verify how this eventually affects architectural and algorithmic performances.

In this context, we see reconfigurable architectures (RAs) such as field programmable gate arrays (FPGAs) as a promising platform. With their flexibility and extensive tool-chains, RAs have proved to be a valuable tool for design space exploration. Hence, if the level of behavioral uncertainty could somehow be increased while maintaining the positive properties of RAs, it could become a valuable tool for the exploration of emerging fields such as stochastic computing or even the future development of RAs. We believe this idea is worth pursuing and is what motivates this paper.

The proposed idea has two prerequisites: 1) that behavioral uncertainties can be increased in RAs, and 2) that it is possible to obtain statistics of the consequent behavior. In this paper, we focus on prerequisite 2 by introducing an automated test framework capable of characterizing combinatorial and sequential circuits statistically. The presented framework consists of two FPGAs; the first works as a statistical logic analyzer and test vector generator and is used to characterize the second FPGA, that acts as a unit under test (UUT). To achieve prerequisite 1, we propose to use a combination of voltage and frequency scaling (VFS) as well as V_{dd}-noise to increase timing uncertainties in the UUT FPGA. Thus, the test platform is designed specifically to support VFS. However, we reserve the detailed discussion on increasing the behavioral uncertainty in RAs for a future paper.

Apart from the hardware setup, the framework features a software module that incorporates standard experimental design methods, and allows automatic execution of tests and interfacing with laboratory equipment. This gives the experimenters the freedom necessary when characterizing problems with many variants. We conclude by demonstrating the capabilities of the framework by evaluating it in terms of accuracy and precision, and we show statistical results of UUT input/output block (IOB) behavior subject to VFS with/without noise.

II. Problem description

A. Characterization of Stochastic Logic Behavior

Formally, logic entities can be described as the mapping $f : \mathbb{B}^N \to \mathbb{B}^M$, where $\mathbb{B} := \{0, 1\}$ is the binary alphabet, N the number of inputs, and M the number of outputs. Given an input vector $x \in \mathbb{B}^N$, the expected response is given by

$y = f(x) \in \mathbb{B}^M$. The equality is justified for almost all digital systems today due to their extremely low error rates. However, for stochastic logic, the mapping f might be perturbed, thus producing a faulty mapping $\tilde{f} : \mathbb{B}^N \rightarrow \mathbb{B}^M$. We expect that the behavior of $\tilde{f}$ converges to that of f as behavioral uncertainties decreases. But the question of interest in our case is rather: how does $\tilde{f}$ diverge from f as the level of uncertainties increase?

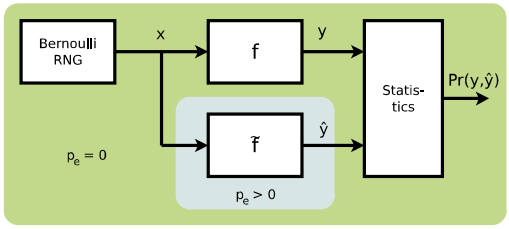


Fig. 1. Proposed characterization method for stochastic logic. The stochastic logic have an error rate p_e larger than zero, which produces the faulty mapping $\tilde{f}$. On a sample-to-sample basis, the statistics module calculates the joint probability between the deterministic and stochastic responses, thus characterizing the stochastic logic relative to expected behavior.

First we note that $\tilde{f}$ is a stochastic process that summarizes the totality of all factors that might perturb the desired behavior described by f. The stochasticity has two sources; 1) temporal noise processes such as thermal noise, V_{cc} variances and electro-magnetic interference, and 2) spatial processes such as process variabilities. Hence, the response of stochastic logic can described as $\tilde{y}(t) = \tilde{f}(x, t, \theta)$, where θ is a random variable collecting all location dependent parameters, x is the input vector, and t time, which indicates that $\tilde{f}$ is a function of the underlying temporal processes.

For a given implementation with (say) outcome θ_1, we are interested in the time-average behavior of the stochastic logic with sample function $\tilde{f}(x, t, \theta_1)$. Hence, to characterize the behavior of the stochastic logic relatively to the expected behavior, we propose using the joint probability measure $\Pr(y, \tilde{y})$. Conceptually, the joint probability statistics are obtained using the setup shown in Figure 1, which also form the basic outset for the test framework.

B. Testability

Having chosen FPGAs as a basis for the test framework, the setup presented in Figure 1 can be implemented in several ways. However, the design is dictated by the challenges related to the observation of the signal response $\tilde{y}$. If we instantiate a component designed to characterize stochastic logic as well as the stochastic logic in the same unit, both components would be subject to uncertainties; making the observer part of the observed, so to speak. Consequently, the observed (i.e. the stochastic logic with $p_e > 0$) and observer (input vector generator and statistics module) have to be isolated from each other. We achieve this by using two FPGAs: the first FPGA, running in its nominal operating range, generates input vectors and passes these to the second FPGA which might generate a faulty mapping $\tilde{f}$, depending on the type and amount of perturbation. The response is looped back to the first FPGA, which compares the response with the expected one and generates statistics.

Finally, as mentioned in the previous section, any experiment will be affected by both temporal and spatial sources of variations. Depending on the individual experiment, some factors might be considered nuisance in some cases, whereas they may be target for investigation in others. Hence, in this work, we do not target any specific source of variability, rather we aim at providing the tools necessary when experimenting in an environment with a multitude of potential nuisance factors.

III. Automated Test Framework

The proposed two-FPGA hardware setup for the experimental framework is presented in Figure 2. Generally, the test framework is designed to 1) enable the characterization of stochastic logic as described in Section II-A, 2) incorporate standard experimental designs presented in Montgomery [5], and 3) automate test execution. To achieve this adaptability, the framework is complemented by a Python software framework that allows integration with the hardware, and automates design and execution of experiments. Both aspects will be described in the following.

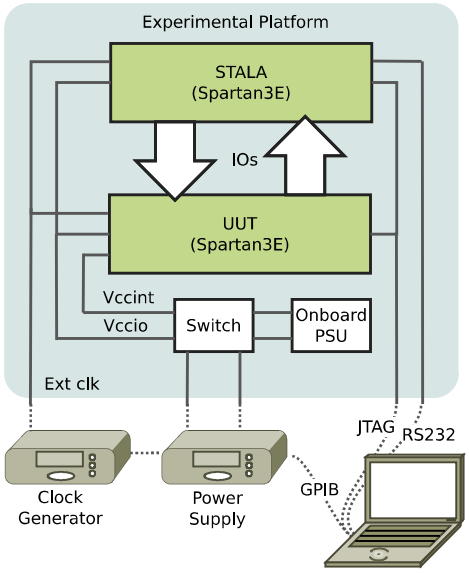


Fig. 2. Diagram of the experimental hardware platform. The platform provides communication, configuration and clock interfaces, and allows the user to switch between on-board and external power supply units for both UUT core voltage (V_{ccint}) and IO bank voltages (V_{ccio}).

In order to lower the development complexity, the current implementation is limited to provide statistics for logic entities where the number of inputs is $N \leq 4$ and the number of outputs is $M = 1$. However, 1-bit statistics can be obtained from a total of eight inputs, enabling evaluation of signal propagation through combinatorial logic, such as inverter or carry chains. Despite this limited capacity, we argue that this is sufficient as a first step toward a fully generic test framework. Later

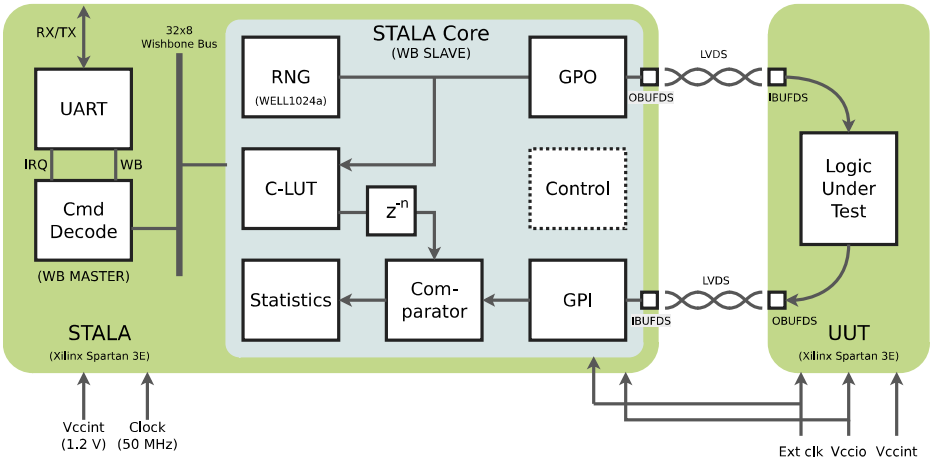


Fig. 3. Architecture overview of the statistical logic analyzer (STALA) and UUT inter-connection. All STALA modules are connected via a Wishbone bus (WB) and can be addressed through the serial interface as memory mapped registers. The STALA core and UUT are driven by the same clock, ensuring synchronous behavior when clocked circuits are characterized. General purpose in- and outputs enables tests across different IO locations.

implementation can be extended to include more advanced statistics modules that can account for e.g. multi-bit adders and multipliers, or evaluation of signal processing algorithms.

A. Hardware Setup

The design of the experimental platform is the result of a trade-off between reducing the impact of timing variabilities due to differences in IO inter-connections lengths and reducing development efforts. Ideally the timing delay between STALA output and UUT input, and vice versa, is constant for all IO locations. A fully custom design might make such a setup possible, but it would increase development efforts drastically. Instead, the selected solution consists of two Xilinx Spartan 3E break-out boards from Sparkfun [6], which provide access to all IOs at the cost of minor differences in wire lengths.

The upper FPGA in the design functions as a statistical logic analyzer (STALA) and the bottom FPGA as the UUT, as illustrated in Figure 2. The inter-connection between the FPGAs uses low voltage differential signaling (LVDS) to improve signal integrity, and the stack-connectors have been slightly modified to support external power supply units and JTAG. Finally, the stacked FPGAs are connected to a custom PCB that allows manual reconfiguration of voltage supplies via jumpers and provides standardized 4 mm power and SMA clock connectors, to allow integration with existing laboratory equipment. In order to automate the scaling of voltage and frequency, laboratory equipment is remotely controlled via a PC using GPIB. Similarly, the statistical logic analyzer is controlled using a serial interface, which allows remote configuration of test parameters and retrieval of test results. A picture of the actual test setup is presented in figure 9.

B. Statistical Logic Analyzer

The statistical logic analyzer is essentially a FPGA-based logic analyzer with a build-in signal generator and statistics module. It generates Bernoulli random variables and compares the UUT response with the expected behavior, following the formalization presented in Section II-A. Based on this comparison, each response is classified as either correct/false positive or correct/false negative. The detailed system architecture is illustrated in Figure 3.

1) Architecture and Communication: The STALA architecture is designed as a memory-mapped peripheral, where all modules are directly accessible through the serial interface (UART). The modules are inter-connected using a Wishbone bus [7], and within STALA the core modules have asynchronous registers access in order to accommodate for separate clock domains. The command decode (Cmd Decode) module parses received data and, depending on the content, automatically reads from or writes to an address. Hence, the system is fully controlled and dependent on external communication, reducing the complexity of the architecture considerably.

2) STALA Core: The STALA core maps the test framework outlined in Figure 1 to hardware. A random number generator (RNG) is used to generate four independent test-vectors. The RNG is a systolic implementation of the WELL algorithm [8]. This algorithm has been chosen as it converges rapidly to pseudo-random behavior. The 32-bit output is used to generate four independent Bernoulli random variables with 8-bit P-value resolution, which are used as test vectors. Once generated, the test-vectors are cycled through the unit under test and the response compared with the correct response in the correct response look-up table (C-LUT), which is generated and uploaded a priori by the user. The output of the comparator is categorized into false/correct negative and positives, and the statistics module simply counts the number of samples falling into each category for each of the eight inputs. General purpose input and outputs (GPI/GPO) map the test-vectors to a selection of predefined locations in one of the four IO banks. This allows experimenters to evaluate location dependence of designs. Finally, a control module provides start/stop/reset

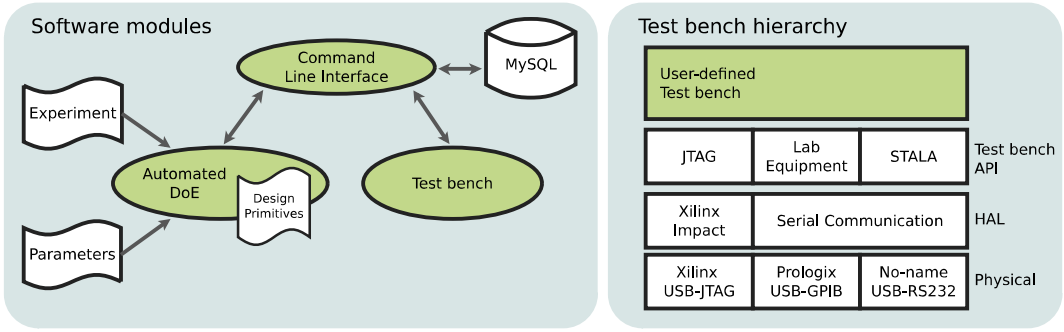


Fig. 4. Overview of the software framework modules and test bench API stack. The command-line interface acts as a proxy between the design of experiment module, the database and the user-defined test bench. The test bench API enable integration with the stacked-FPGA experimental platform.

capabilities and allows the user to set the sample size (test-vectors length), with 2^{24} being the default.

3) Skew minimization and signal integrity: Both timing skew and signal integrity in the IO inter-connection are potential nuisance factors in the desired characterization. Hence, certain precautions have been taken to minimize the effects hereof in the STALA-UUT design. In order to minimize timing skew between outputs and inputs, the STALA design uses the *Pack latches into IOBs* attribute which forces the mapping to use latches in the IO blocks. The IOB latches are connected to the low latency clock network inside the FPGA, which is driven by the external clock through a delay locked loop. In this way, the outputs should remain unaffected by internal timing skews in the STALA architecture, and, similarly, responses are sampled with minimal skew. Both inputs and outputs are latched on the rising edge of the external clock, allowing up to one cycle propagation delay in combinatorial UUT designs. As the UUT response might be noisy or poorly defined digitally, the inputs are passed through an extra latch which masks any meta-stability in the IOB latch.

C. Python Software Framework

The software framework consists of four main components written in Python; an automated design of experiment (DoE) module, a test bench application interface, a command line interface (CLI) for the handling of test execution and generation, and a MySQL database to store the generated test sequences and associated test responses. An overview of the software architecture is provided in Figure 4.

1) DoE: This module is responsible for the generation of test sequences. The module generates experiments according to common experimental designs; in the current software version randomized complete block designs (RCBD), Latin square (LS), two-level factorial (FF2N) and full factorial (FULLFACT) designs are supported. Furthermore, randomization and stacking of designs are supported as well. Each experiment and associated parameters are specified in separate configuration files, using the syntax presented in Figure 5. When executed, the module parses and evaluates the configuration files, and links the parameters to the control factors specified in the experiment configuration. Then the test sequences are generated over the full range of the specified control parameters according to the experimental design specified.

```
[fullfact_experiment]
name = "Experiment title"
description = "Description"
primitive = FULLFACT
randomize = True
replications = 4
factors = ['A', 'B']
child = None
```

(a) `experiment.cfg`

```
[A]
id = VCCINT
values = [0.8, 0.9, 1.0]
default = 1.2

[B]
id = EXTCLK
values = arange(56e6, 72e6, 0.5e6)
default = 50e6
```

(b) `parameters.cfg`

Fig. 5. Configuration example for a full-factorial experiment with core voltage and clock frequency as factors.

2) Test bench: The test bench module is responsible for evaluating the response of a system given a specific test vector. Once evaluated, the test bench returns the system response and waits for next test-vector. To ease integration with the experimental platform, a test bench application interface (API) has been developed. This allows integration and control of scientific equipment via GPIB, configuration of FPGA(s) using JTAG and an abstraction layer for the statistical logic analyzer. The GPIB interface is based on Python LabTools [9] and JTAG configuration is provided through a software wrapper for Xilinx Impact.

3) CLI: The command-line interface acts as a user-controlled proxy between the DoE module, test bench and database. It provides the user with an simple way to point to experiment and parameter configuration files, and it hides the complexity of populating and extracting test data from the database. Experiments are executed by specifying the name they have been given in the configuration file, and the name of Python test bench module. Following, the CLI will automatically parse test sequences to the test bench one by one and add the returned responses to the database.

IV. Results

A. Logic Analyzer Performance

In order to determine the accuracy and precision of STALA we consider the case illustrated in Figure 6. Test-vectors are

routed through the UUT without inferring any logic, and for each test, statistics are obtained for both test-vectors and responses. The figure of merit used is the probability of a logic *one* occurring in the output and input, denoted with p_O and p_I, respectively. In the ideal case, the difference between the specified RNG p-value, p, and p_O and p_I is zero in all tests. However, both fluctuations in the RNG p-value and uncertainties in e.g. STALA counters might add variance and bias to the individual distributions. Hence, the performance is measured by considering the difference (deviation) between the three probability values p, p_O and p_I over an ensemble of tests.

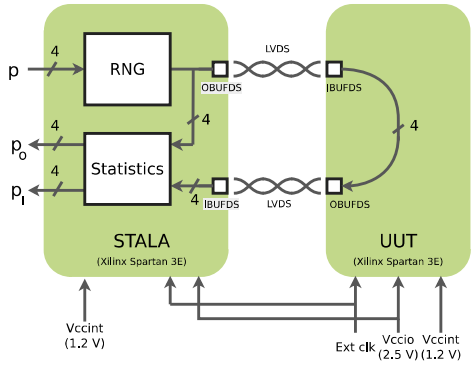


Fig. 6. Illustration of the test setup. The test-vectors are routed through the UUT and statistics of the output (test-vector) p_O and input (response) p_I distributions are obtained.

The chosen experiment type is a randomized full-factorial design with eight frequency values ranging from 40 to 75 MHz in steps of 5 MHz and three RNG p-values in the sequence $\{0.25, 0.5, 0.75\}$. Hereby, variabilities associated with either RNG p-values and/or frequency are captured. The four input/output pairs are held-constant on FPGA bank 3. For each test, the test-vectors are looped through the UUT, compared with the expected response (not shown in Figure 6) and statistics calculated for each response associated with a test-vector as well as the differences between distributions. To obtain a statistical basis, the experiment is repeated 25 times, giving a total number of tests of $8 \times 3 \times 25 = 600$.

Box-whisker plots of the ensembles over all combinations and repetitions of frequencies levels, p-values and output/input pairs are shown in Figure 7. It is evident that the RNG output is not stationary, and that this affects the accuracy and precision of STALA if left uncompensated (Response 1). However, the deviation between the measured test-vector and response distributions, p_O and p_I, is practically non-existing as seen in Response 2. This shows that the results provided by STALA must be compensated for non-stationarities in the RNG if maximum performance is sought, else minor uncertainties must be accepted; the median deviation being 2.47e-05 with an upper fence of 2.68e-4 and lower fence of -3.10e-04, as well as a few outliers.

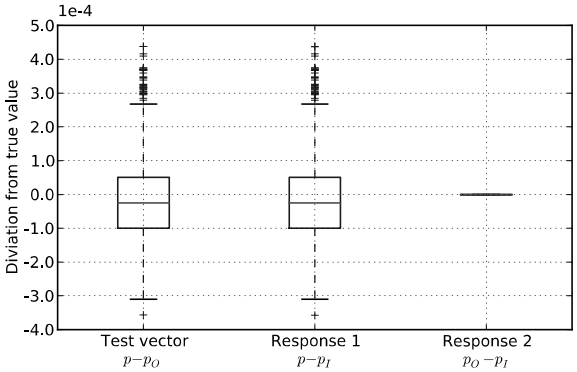


Fig. 7. Deviation from true probability values. Left) Test vector deviation from specified p-value, $p - p_O$. Response deviation: middle) uncompensated for test-vector variations, $p-p_I$, right) compensated for test-vector variations, $p_O - p_I$.

B. Demonstration of Framework Capabilities; Characterizing Signal Path Timing Errors

To further demonstrate the capabilities of the framework, we consider a more advanced test case that requires interfacing with various laboratory equipment and design of experiment with several factors. Our goal is to characterize the signal path through the UUT (input-buffer $\rightarrow$ core routing $\rightarrow$ output-buffer), in terms of error rate when subject to voltage and frequency scaling as well as PSU noise. Apart from demonstrating the framework, the test results also provide a relevant insight into the response of a digital circuits subject to VFS and PSU noise.

The test setup is similar to that in Figure 6, but with varying frequency and a noisy power supply with varying mean. The output of the noisy power supply is modeled as a narrow-band Gaussian random variable $V_{dd} \sim \mathcal{N}(\mu_{V_{dd}}, \sigma^2_{V_{dd}})$, where $\mu_{V_{dd}}$ is the mean supply voltage and $\sigma^2_{V_{dd}}$ the noise energy. For the experiment, the spectral noise density of the noisy power supply was set to $-13.5\,\mathrm{dBmV}/\sqrt{\mathrm{Hz}}$. The effective (brick-wall) bandwidth at the FPGA was measured to approx. 1 MHz, yielding a RMS noise voltage of $\sigma_{V_{dd}} = 22$ mV. Again, the experiment type is a randomized full-factorial design. The frequency has values ranging from 64 MHz to 80 MHz in steps of 2 MHz, and mean core voltage $\mu_{V_{dd}}$ has values ranging from 0.75 V to 0.96 V in 10 mV steps. With four repetitions for each parameter combination, the total number of test amounts to 1512. The RNG p-value and FPGA IO bank are held-constant at $p = 0.5$ and bank 3, respectively. For each test the statistics are obtained for each of the four input/output pairs, and the error probability calculated.

The entire experiment was executed in 4 hours, giving an average execution time of 9.6 seconds. Given a test vectors length of 2^{24} cycles, each experiment take less than a second to perform for all frequencies. Hence, the majority of run-time is spend communicating via the serial interface and is clearly something that can be improved. The numerical results of the characterization are shown in Figure 8 for the cases with and

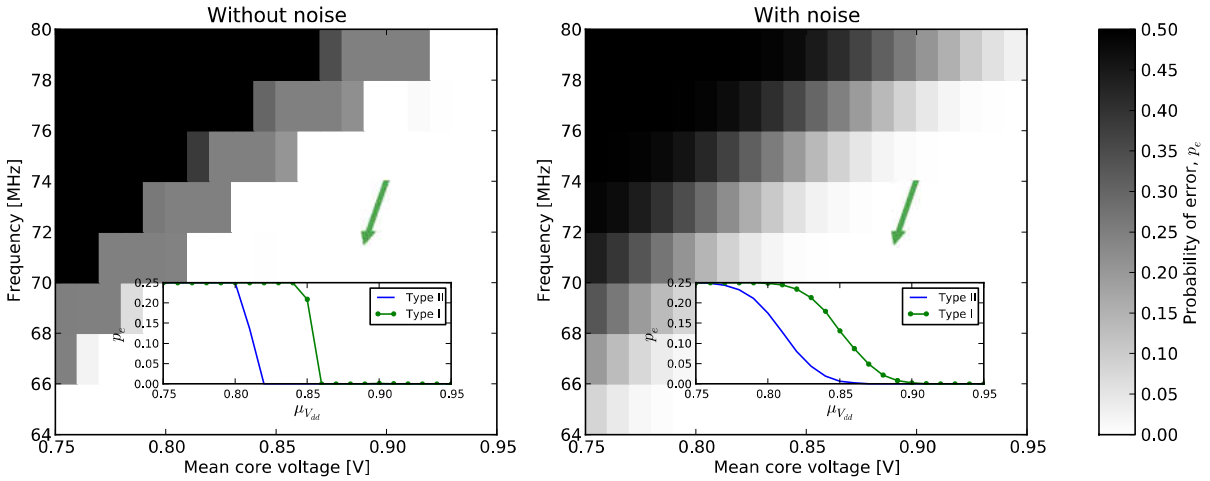


Fig. 8. Experimental results of the UUT input-to-output routing behavior when subject to voltage-frequency scaling with/without noise. The main figures show the response in terms of error probability rate, $\Pr(y \neq \tilde{y})$. The sub-figures show the raw false negatives (Type II) and false positives (Type I) data at 74 MHz.

without noise. In both cases, the results reflect the slow-down in circuit speed that occurs when the mean supply voltage is decreased, and the contour where the error rate changes from 0 to 0.5 characterizes the limit of voltage scaling for a given frequency. In the no-noise scenario it is clear that the UUT device fails abruptly during down-scaling; first with a rapid increase in false positives (Type I) followed shortly after by a rapid increase in false negative rate (Type II). In the noise scenario, the general tendency is the same but with higher variance. The bias towards false positives can be explained by asymmetries either between transition times or threshold voltages, which favor transitions from logical low to high.

V. Conclusion and Future Direction

In this paper, we have presented and demonstrated the design of an automated test framework for the characterization of stochastic behavior in reconfigurable logic. Given this capability, the next step is to continue work on noise-driven FPGAs to gain a better understanding on the error mechanisms in the UUT FPGA, and to extent the analysis to more advanced digital circuits. Also, it is highly relevant to use the results obtained from the test framework as a basis for statistical modeling and inference. Such work might then be used to derive heuristics or timing models for hardware architectures with stochastic behavior.

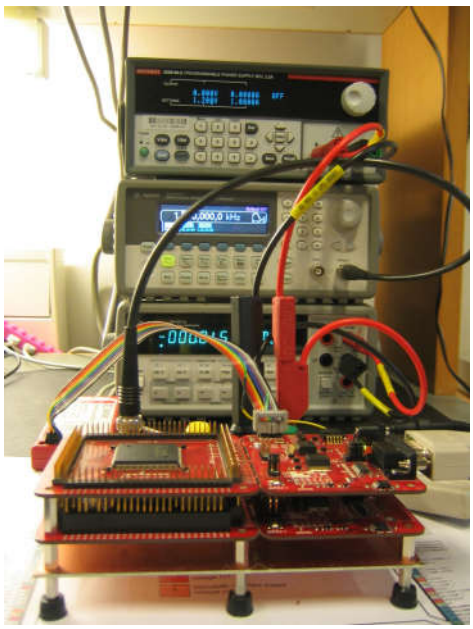


Fig. 9. Picture of the stacked-FPGA test-bed. The top board contains the statistical logic analyzer, the middle board is the unit under test and the bottom board is the custom PCB that enables integration with laboratory equipment.

References

[1] D. Lammers, "The era of error-tolerant computing," *Spectrum, IEEE*, vol. 47, no. 11, p. 15, november 2010.

[2] S. Saha, "Modeling process variability in scaled cmos technology," *Design Test of Computers, IEEE*, vol. 27, no. 2, pp. 8 –16, march-april 2010.

[3] J. George, B. Marr, B. E. S. Akgul, and K. V. Palem, "Probabilistic arithmetic and energy efficient embedded signal processing," *CASES '06*, p. 158.

[4] J. Sartori, J. Sloan, and R. Kumar, "Stochastic computing: embracing errors in architecture and design of processors and applications," in *Proceedings of the 14th international conference on Compilers, architectures and synthesis for embedded systems*, ser. CASES '11, 2011, pp. 135–144.

[5] D. C. Montgomery, *Design and Analysis of Experiments*, 7th ed. Wiley.

[6] SparkFun Electronics, "Spartan 3E Breakout and Development Board," 2012. [Online]. Available: http://www.sparkfun.com/products/8458

[7] OpenCores Organization, "WISHBONE System-on-Chip (SoC) Interconnection Architecture for Portable IP Cores," Tech. Rep., 2002. [Online]. Available: http://opencores.org/opencores,wishbone

[8] F. Panneton, P. L'Ecuyer, and M. Matsumoto, "Improved long-period generators based on linear recurrences modulo 2," *ACM Trans. Math. Softw.*, vol. 32, no. 1, pp. 1–16, Mar. 2006.

[9] P. H. Kamp, "Python LabTools for talking to GPIB instruments," 2012. [Online]. Available: https://github.com/bsdphk/pylt

Authors' version. Appears in the proceedings of IEEE PATMOS 2013. DOI: 10.1109/PATMOS.2013.6662150

Empirical Verification of Fault Models for FPGAs Operating in the Subcritical Voltage Region

Alex Birklykke*, Peter Koch*, Ramjee Prasad†, Lars Alminde‡, and Yannick Le Moullec*

*Technology Platform Section, †Center for TeleInFrastruktur
Department for Electronic Systems, Aalborg University
Fredrik Bajers Vej 7, DK-9220 Aalborg Ø
{**alb**, pk, prasad, ylm}@es.aau.dk
‡GomSpace APS, Niels Jernes Vej 10, DK-9220 Aalborg Ø
alminde@gomspace.com

***Abstract*—We present a rigorous empirical study of the bit-level error behavior of field programmable gate arrays operating in the subcricital voltage region. This region is of significant interest as voltage-scaling under normal circumstances is halted by the first occurrence of errors. However, accurate fault models might provide insight that would allow subcritical scaling by changing digital design practices or by simply accepting errors if possible. To facilitate further work in this direction, we present probabilistic error models that allow us to link error behavior with statistical properties of the binary signals, and based on a two-FPGA setup we experimentally verify the correctness of candidate models. For all experiments, the observed error rates exhibit a polynomial dependency on outcome probability of the binary inputs, which corresponds to the behavior predicted by the proposed timing error model. Furthermore, our results show that the fault mechanism is fully deterministic – mimicking temporary stuck-at errors. As a result, given knowledge about a given signal, errors are fully predictable in the subcritical voltage region.**

I. Introduction

With approximately the same flexibility as software and performance as dedicated hardware, the field programmable gate array (FPGA) has become invaluable not only as a component in industrial products but equally as a platform to facilitate experimental research. One branch of research that have seen its widespread use is on low-power methodologies for CMOS architectures based on voltage-scaling. Specifically, the critical point at which voltage scaling results in errors has served as a pivot for much work: In [1] Chow et al. investigated a *dynamic voltage scaling* method which uses a monitoring scheme to ensure that the FPGA is operating as close to, but always above, the *critical operating point* (COP), i.e., the point where further scaling will result in observable errors. In this way, a power reduction between 4 and 54% was observed for a wide range of applications. Later, a similar approach was used with success on System-On-Chips [2]. FPGAs have also been used to investigate the limits of voltage scaling in more general settings: For example; in [3], Narayanan et al. confirmed the COP hypothesis [4] which states that for VLSI systems there exists a voltage V_c, frequency f_c, and temperature T_c such that any scaling beyond the point (V_c, f_c, T_c) will lead to system failure due to a massive number of simultaneous timing violations. The authors went on to show that by changing the slack distribution in the target microarchitecture the increase in error rate could be made more gradual. In [5], Roberts et al. went beyond the COP into the *subcritical* voltage region and captured the behavior of an 18x18 Xilinx DSP block multiplier with respect to supply voltage and error rate – showing a gradual increase in errors over an approx. 200 mV range. The same line of work also helped gauge the power-saving potential of the Razor latch [6].

These examples underline the relevance of FPGAs in a research context and give a picture of the power-saving potential of FPGA-based implementations. Also, it indicates that much of the interesting work is done in the vicinity of the COP. However, although much insight can be found in the cited references regarding the fault mechanisms in voltage-scaled FPGAs, it has, to our knowledge, never been empirically illustrated and verified what happens at the bit-level. Roberts et al. [5] come close in the sense that they count the combined number of false positives and negatives on the 18-bit wide output of the multiplier given different voltage levels. This approach unfortunately hides the fault behavior at bit-level. Furthermore, the statistical properties of the test vectors are obscured due to a poorly described methodology, and the work

Authors' version. Appears in the proceedings of IEEE PATMOS 2013. DOI: 10.1109/PATMOS.2013.6662150

thus falls short at uncovering possible dependencies between the observed error-rates and statistical properties of the test data.

In this work, we therefore present statistical models for, and empirical verification of, bit-level fault models that describe the behavior of FPGAs operating in the sub-critical voltage region. Our goal is to add more detail to earlier work by exposing the inter-dependence between the statistical properties of the input signal and fault models.

Through our work, we gain a more accurate and confident understanding of how FPGAs fail during voltage-scaling. The experimental verification of fault models add further substance to assumptions made in earlier work (provided they hold) and help guide future research in the field by removing any doubt as to what happens when faults occur in over-scaled FPGAs. Finally, the probabilistic fault models constitute an effective primitive for error analysis of voltage-overscaled signal processing systems. Our work thus takes an important step towards an analytical evaluation of the accuracy-power trade-offs that exists for various DSP kernels.

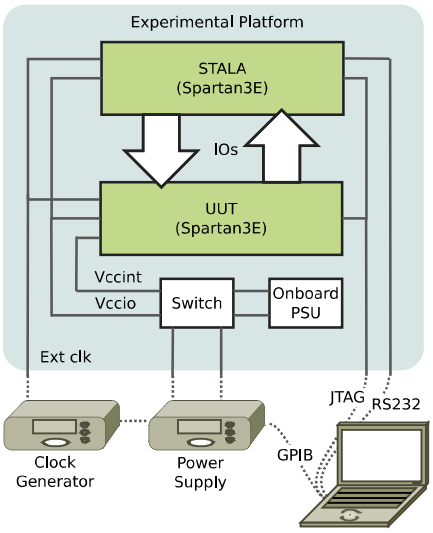


Fig. 1. The experimental platform used to observe the effects of faults on binary signals when the UUT is running in the subcritical voltage region. The full specification of the platform can be found in [7]

II. Experimental Setup

The experiments in this work are performed using the two-FPGA setup illustrated in Figure 1. The top FPGA acts as a statistical logic analyzer (STALA) that generates test vectors and analyses the response from the unit under test (UUT). The UUT is configured with the desired circuit that we would like to characterize when subject to voltage and/or frequency scaling, and the UUT behavior is characterized by the STALA unit by collecting bit-level error statistics, i.e., true and false positives, as well as negatives. The test vectors consist of independent binary random variables with adjustable outcome probability; each with a fixed length of 2^{24} bits. The randomness is sourced from a high quality random number generator based on the WELL1024a algorithm. The adjustable outputs and error statistics enable us to expose dependencies between error rates and the statistical properties of the test vectors as we operate the UUT in the subcritical voltage region. The models presented in Section IV describe the *expected* dependency between test vector properties and error rate for various fault models. Thus, by comparing the behavior of the voltage-scaled UUT to the behavior described by our models, it is possible to infer the most probable cause of the observed errors.

III. UUT Architecture Analysis

To design the experiments and identify possible fault mechanism, it is necessary to analyse the FPGA technology to pin-point inherent limitations and technology features that might cause faults or failures in the UUT during subcritical voltage operation. Figure 2 is compiled based on information from Xilinx user guides [8] and patents [9], [10], and provides a general view of the path from input to output in an FPGA. In this case, we show the FPGA configured with a simple pipeline and differential signaling with the outside world.

First we note that FPGAs are multi-V_{cc} devices, with separate voltage domain for IOs (V_{ccio}), core logic (V_{ccint}), and auxiliary circuits (V_{ccaux}) such as DCMs and JTAG components. This separation is practical since it allows us to isolate voltage scaling to the core logic, while running auxiliary and IO logic at nominal voltage levels. To ensure correct power-on and reliable behavior once the device is active, a voltage monitoring circuit is embedded in the FPGA. For Xilinx devices this is known as the power-on reset (POR) circuit. The POR insures that internal components are powered in the correct sequence during power-on. However, it also protects the user-application from faulty operation by resetting the device if the core voltage, which also supplies the SRAM, drops below a point where the configuration might get corrupted. Consequently, for voltage scaling operations, the POR trip-point constitutes a lower operational limit. For the Xilinx Spartan 3E (XC3S500E-PG208) used in our experiments, the trip-point is specified at a minimum of 400 mV with a nominal voltage of 1.2 V. But for the particular FPGA under test, the configuration is lost already at 500 mV, where the global reset is triggered. However, the `DONE` pin is unset at 550 mV, which at nominal levels indicates incorrect configuration. Subsequently, we chose 600 mV as the

Authors' version. Appears in the proceedings of IEEE PATMOS 2013. DOI: 10.1109/PATMOS.2013.6662150

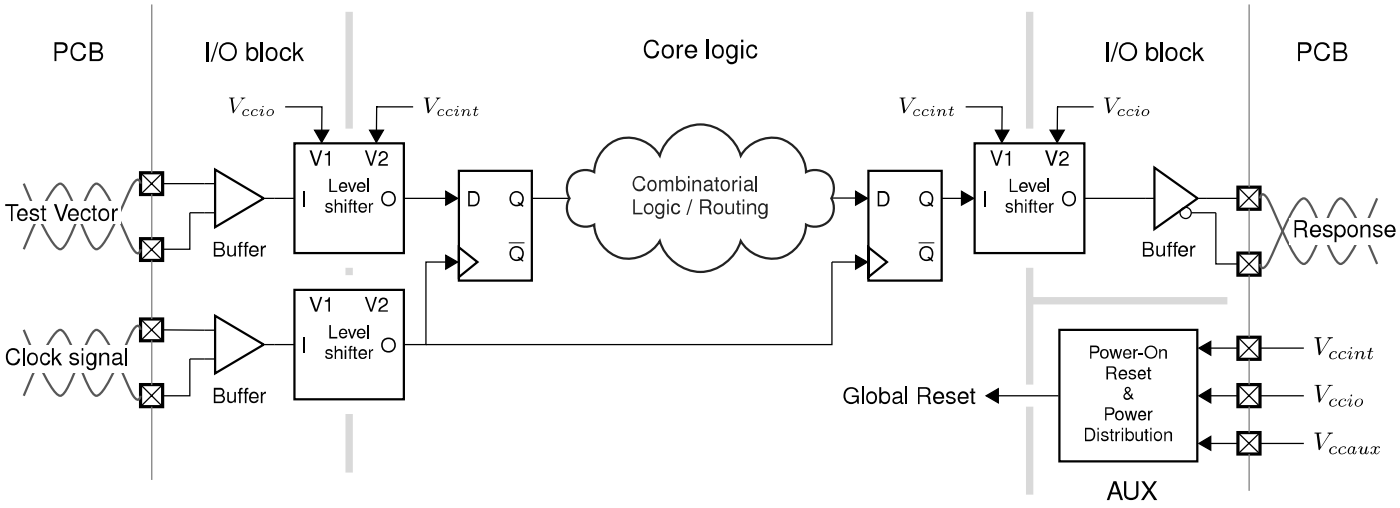


Fig. 2. Illustration of the signal path through a field programmable gate array configured with a two latch sequential circuit. The FPGA features separate voltage domains for IOs, core logic and auxiliary circuitry. While this makes voltage-scaling experimentation easy due to the natural isolation of the core circuitry, it is also a possible source of faults as the test signal has to traverse the different voltage domains.

lower limit for our experiments to avoid corrupting the configuration while allowing potential fluctuations in the supply voltage. As the POR circuit prevents SRAM corruptions, we are guaranteed that the LUTs and routing will be configured correctly for all voltages above the POR limit. However, this does not imply that the LUTs and routing operate as expected; something we have to keep in mind when analysing the experimental results.

After the differential *test vector* in Figure 2 has been converted to single-ended, it is routed to the core logic through a step-down converter. However, since the logical HIGH-level in the IO block is sufficient to trigger the core logic which runs at a lower voltage, it is unclear whether the level-shifter from V_{ccio} to V_{ccint} actually contains any circuitry or is merely a direct connection. User guides and patent searches have not provided any information on this matter, but we assume that the circuitry needed is very limited. Similarly, we do not expect this transition to be the source of any faults. In contrast, the logical HIGH-level in the core logic is not sufficient to trigger logic running at higher supply voltages. As a result, the level-shifting from core to IO voltage levels requires inventive circuits, which is evident from the patents and papers found on the subject [9], [10], [11]. In relation to voltage scaling, the step-up transition might cause faults as lowering the core voltage potentially forces the level-shifter circuitry beyond its operational limits. In [1], such events where reported as *IO errors*. That is, errors caused by faults in the core-IO transition. Given the earlier reports of these types of faults, IO errors must be considered as a potential source of faults in our experiments. The last type of fault we might encounter is due to timing violations. It is well-known that voltage and propagation delay are correlated, hence, scaling the voltage might introduce errors due to timing faults. This is considered in more detail in Section IV, where a probabilistic model for both IO and timing errors will be introduced.

Generally, both the core logic and IO blocks are significantly more complex than depicted in Figure 2. However, to limit the scope of the paper and number of experimental factors, we restrict our focus to basic features of the FPGA i.e., basic IO capabilities, routing and LUTs. Hence, the V_{dd} characterization of e.g. RAM and DSP components is reserved for possible future work.

IV. Probabilistic Modeling of Faults

In the following, we introduce the mathematical framework necessary to link the statistical signal properties and fault models to the error rate measure. Figure 3 shows the graphical abstraction upon which we base our analysis. The basic idea is to separate the correct and incorrect circuit behavior into different components. The first component behaves as expected, whereas the second probabilistic components might add errors according to the fault scenario considered. In Figure 3, the faulty component is a latch and the (zero-delay) probabilistic component is denoted by $\mathcal{F}$. We characterize the behavior of $\mathcal{F}$ using the transition probability matrix $P_{\mathcal{F}} = \left[p_{Y|X}(j|i)\right]_{i\in\mathcal{Y},j\in\mathcal{X}}$ where $\mathcal{X}$ and $\mathcal{Y}$ are the state space of the in and output, respectively, and $p_{Y|X}(j|i)$ is the probability of jumping to state i given the current state j. In our

Authors' version. Appears in the proceedings of IEEE PATMOS 2013. DOI: 10.1109/PATMOS.2013.6662150

case, $\mathcal{X}, \mathcal{Y} \in \{0, 1\}$. Thus, given a particular input state, say $X_n = 0$, the output of $\mathcal{F}$ will perform a probabilistic jump to a new state according to $p_{Y|X}(Y_n \in \mathcal{Y}|X_n = 0)$. The structure of $P_\mathcal{F}$ depends on the fault model that we assume to affect the digital systems, and our challenge is to find a model for $P_\mathcal{F}$ that accurately corresponds to a relevant fault-mechanism in the voltage-scaled UUT.

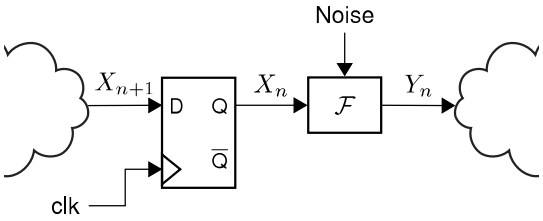


Fig. 3. Bit-level faults are modeled as a noisy communication channel proceeding the circuit (in this case a latch) where the fault have occurred.

We model the binary signals on the input of $\mathcal{F}$ as a stochastic process $X = \{X_n : n \in \mathbb{Z}^+\}$ consisting of Bernoulli random variables with probability $p = \Pr(X_n = 1)$. Furthermore, we assume that neighboring variables in X are linearly dependent. In this case the temporal correlation coefficient is given by [12]:

$$\rho = \frac{r_X(1) - p^2}{p(1-p)}$$

where $r_X(1) = E[X_{n-1}X_n]$ is the unnormalized lag-1 autocorrelation of X. Note that for the independent case, $r_X(1) = p^2$ and thus $\rho = 0$. Bit-level correlation occur as a result of correlation in physical signals which to some degree is maintained despite binary encoding [12]. However, as will be discussed later, faults also change the correlation coefficient of the signal which is why we include the property in our signal. As a result of the correlation, the joint distribution $p_X(x_0, x_1) = \Pr(X_{n-1} = x_0, X_n = x_1)$ of the stochastic process X becomes:

$$p_X(x_0, x_1) = \begin{matrix} x_0x_1 \\ 00 \\ 01 \\ 10 \\ 11 \end{matrix} \begin{pmatrix} (1-p)^2 + \rho q \\ (1-\rho)q \\ (1-\rho)q \\ p^2 + \rho q \end{pmatrix} \quad (1)$$

where $q = p(1-p)$. Given the statistical properties of the input signal and a fault model $\mathcal{F}$ defined by the transition probability matrix $P_\mathcal{F}$, the error probability $p_e = \Pr(X_n \neq Y_n)$ can be derived:

$$\begin{aligned} p_e &= \sum_{i \neq j} p_{YX}(j,i) = \sum_{i \neq j} p_X(i) p_{Y|X}(j|i) \quad (2) \\ &= \underbrace{\Pr(Y_n = 1 \cup X_n = 0)}_{\text{False positives}} + \underbrace{\Pr(Y_n = 0 \cup X_n = 1)}_{\text{False negatives}} \end{aligned}$$

That is, the rate at which $X_n = z$ does not imply $Y_n = z$, given some input $z \in \mathcal{X}$. For binary experiments the error rate is composed by the probability of false positives and false negatives, which are the statistics captured by STALA. In the following, we derive the error rate according to different fault models that we expect to observe. Later we will compare error rate of these models with the actual error rate response obtained by STALA.

A. Threshold Effect Models

One theory on how errors occur in voltage over-scaled digital circuits is based on the threshold effect [13]. In this setting, the threshold voltage V_t of the CMOS circuit is considered a hard-limit. If the signal magnitude is above the threshold voltage, the circuit considers the signal as a digital '1' and if below as a digital '0'. When the voltage difference between logical high and low grows sufficiently small, noise in the operational environment makes it harder to discriminate correctly between the states, and, eventually, logical misclassifications will occur. This behavior can be modeled by the general transition probability matrix $P_\mathcal{G}$:

$$P_\mathcal{G} = [p_{Y|X}(j|i)]_{i \in \mathcal{X}, j \in \mathcal{Y}} = \begin{matrix} i \backslash j \\ 0 \\ 1 \end{matrix} \begin{matrix} 0 & 1 \end{matrix} \begin{pmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{pmatrix}$$

where α denotes the probability of false positives and β false negatives. Using Eqn. 2, we find that the error rate of the threshold effect model is given by:

$$\begin{aligned} p_{e,\mathcal{G}} &= p_X(0)p_{Y|X}(1|0) + p_X(1)p_{Y|X}(0|1) \\ &= (1-p)\alpha + p\beta \end{aligned}$$

Two special cases of the threshold effect model are worth highlighting:

1) Symmetry, $\mathcal{S}$, $\alpha = \beta$: This model reflects the scenario when the noise on the logical states is identically distributed and the threshold voltage is assumed always to equal half the supply voltage, ie. $V_t = V_{dd}/2$. In this case, the error rate simply equals the transition probabilities:

$$p_{e,\mathcal{S}} = \alpha = \beta$$

2) One-sided Asymmetry $\mathcal{A}$, $\alpha = 0, \beta \geq 0$: If the threshold voltage does not scale with supply voltage but remains constant at some value, then it is very unlikely that the noise in the '0' state will result in false positives once we scale the supply voltage. We approximate this by letting $\alpha = 0$. Instead, false negatives start to occur as the distance between V_t and V_{dd} becomes smaller, which increases β. The error rate for this model is:

$$p_{e,\mathcal{A}} = p\beta$$

Hence, the error rate is correlated linearly with the bit-probability p on the input.

Although the symmetric model has nothing to do with timing, it is widely used at higher levels of abstraction to model the occurrence of timing errors. Here, a common approach is to randomly flip bits in e.g. the MSB of output registers to evaluate the robustness of some program. Due to the widespread application of the model, we have included it in our analysis. In contrast, the asymmetric model mimics the behavior of faults occurring in the transition from core to IO voltage in the FPGA.

B. Timing Violation Model, $\mathcal{T}$

It is well-known that reducing the supply voltage increases the latency in digital circuits. Hence, faults due to timing violations are very likely to cause errors. Figure 5 shows a pipeline section of a digital design, annotated with timing. The path slack quantifies the timing margin given by the period between the rising edges every $1/f_{clk}$ s. That is:

$$t_{slack} = (1/f_{clk}) - t_{cto} - t_{prop} - t_{su}$$

where t_{cto} denotes the time between the rising edge of the clock and when the latched state is stable on the output, t_{prop} the propagation delay of the combinatorial circuit between the latches, and t_{su} the time (setup-time) required by the D-flip-flop to latch the signal correctly. If negative slack occurs, $t_{slack} < 0$, the combinatorial signal might be captured incorrectly and result in an error. Such events are known as *transient faults* and when the delay is large enough they result in temporary stuck-at errors [14]. We can model this behavior probabilistically with the transition matrix $P_{\mathcal{T}}$:

$$P_{\mathcal{T}} = [p_{Y|X}(j|i)]_{i\in\mathcal{X}^2, j\in\mathcal{Y}} = \begin{array}{c} i \setminus j \\ 00 \\ 01 \\ 10 \\ 11 \end{array} \begin{array}{c} \begin{array}{cc} 0 & 1 \end{array} \\ \begin{pmatrix} 1 & 0 \\ \alpha & 1-\alpha \\ 1-\beta & \beta \\ 0 & 1 \end{pmatrix} \end{array} \quad (3)$$

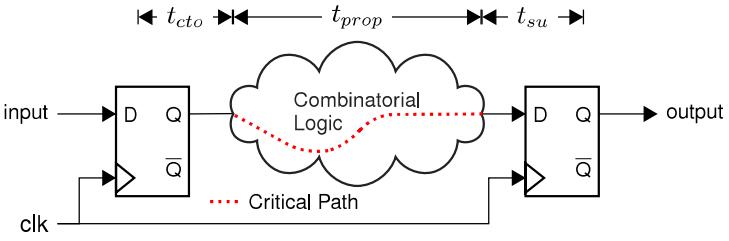


Fig. 5. Timing in sequential logic circuits

where $\mathbf{x}$ denotes the event that $(X_{n-1} = x_0, X_n = x_1)$ and $x_0, x_1 \in \mathcal{X}$, y the event that $Y_n = y$. Finally, α and β denote the probability of transient faults on transitions from $1 \to 0$ and $0 \to 1$, respectively. By combining the transition matrix with the signal distribution in Eqn. 1, we can derive an expression for the error rate due to timing faults:

$$\begin{aligned} p_{e,\mathcal{T}} &= p_X(00)p_{Y|X}(1|00) + p_X(10)p_{Y|X}(1|10) + \\ &\quad p_X(01)p_{Y|X}(0|01) + p_X(11)_X p_{Y|X}(0|11) \\ &= 0 + p(1-p)(1-\rho)\alpha + p(1-p)(1-\rho)\beta + 0 \\ &= p(1-p)(1-\rho)\left(\alpha+\beta\right) \end{aligned} \quad (4)$$

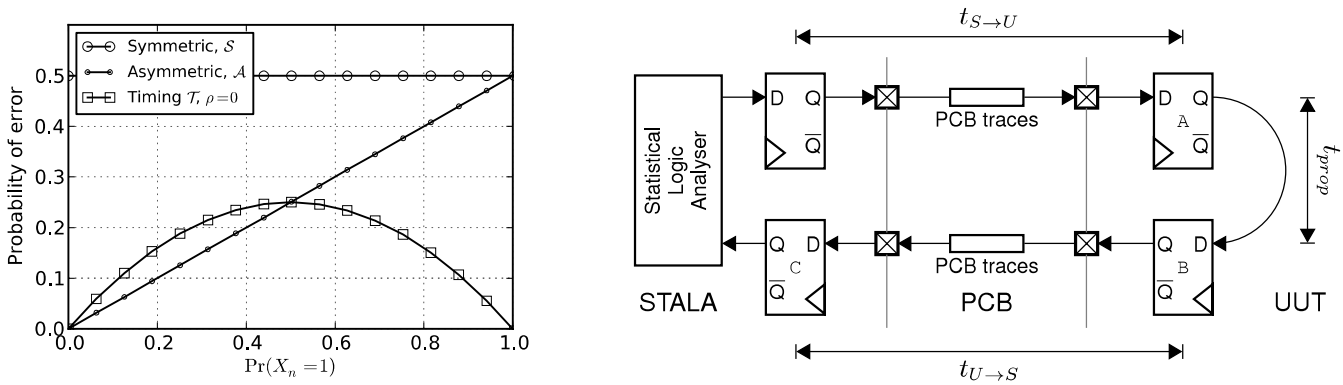
Thus for timing errors, the error rate is linearly correlated with the correlation coefficient ρ and exhibits a second order correlation with the bit-probability p.

Figure 4(a) shows the error rate relative to the bit-distribution $p = \Pr(X_n = 1)$ for models $\mathcal{S}$, $\mathcal{A}$ and $\mathcal{T}$. As evident, each model have a unique behavior, with the symmetric model being independent of p, the model asymmetric being linearly dependent, and the timing model exhibiting a polynomial dependency. Hence, by sweeping over different p values during experimentation, we expect the error rate response to correlate with one or more of these fault models. We will use this property to verify the fault models experimentally.

V. Experiments

In total three different experiments with various purposes are performed. The experiments are all variations to the setup in Figure 4(b), where some test remove certain latches, and others add additional logic in the UUT core. All experiments start with a factor screening experiment over a wide range of voltages and frequencies, and with a fixed input probability. These results expose the location of critical operation points (i.e., the point of transition between no errors and errors) and characterize the change in error rate once the voltage and frequency are scaled into the subcritical region. Based on the factor screening experiments, one or more points of interest in the subcritical region are singled out for more thorough analysis.

Authors' version. Appears in the proceedings of IEEE PATMOS 2013. DOI: 10.1109/PATMOS.2013.6662150



(a) (b)

Fig. 4. (a) Error rate for different fault models with $\alpha = \beta = 0.5$ relative to input probability. (b) Experimental setup used to characterize the inter-connection between the statistical logic analyser and UUT, as well at the core logic of the UUT

At these points, the bit-probability is swept from 0 to 1 and error statistics collected at each point. All experiments are randomized full factorial designs and each experiment is repeated 4 times to account for possible die temperature variations and other nuisance factors that might affect the experiments. The data points presented in the results section are the median of the 4 measurements. Once all data have been collected, the error rate responses are compared with the tendencies in Figure 4(a) whereby we can infer the most probable fault mechanism affecting the response.

A. Case 1 – Core-to-IO Transitions

This experiment aims at testing the transition from the core logic to the IO logic in the UUT. This is done by using the setup in Figure 4(b), but where latch `A` and `B` are removed and the input instead is connected directly to the output via the core logic. As a result, the test vector exercises the level shifter in the UUT output block (see Figure 2) which is the target of the investigation.

B. Case 2 – STALA-UUT Connection

Due to the large parasitic capacitance of the PCB traces relative to the routing in the UUT, the connection between the STALA and UUT FPGAs is associated with a significant propagation delay, Hence, the delay to and from the UUT is much greater than the delay in the UUT routing. That is, $\min\{t_{S\to U}, t_{U\to S}\} \gg t_{core}$. Also, because the level shifter circuit in the output is more complex than in the input, an additional delay will be added, and thus we assume that $t_{U\to S} > t_{S\to U}$. Consequently, in Figure 4(b), we expect to observe errors due to timing faults in the latches proceeding the traces connecting the FPGAs before internal faults occur. That is, errors will occur in latch `A` and `C` prior to latch `B`, and in `C` prior to `A`.

C. Case 3 – Subcritical Operation of UUT Core Logic

The final experiment aims at characterizing the error behavior of the UUT core logic. To achieve this, the UUT is configured with an inverter chain to increase the propagation delay such that $t_{core} > \max\{t_{S\to U}, t_{U\to S}\}$. Furthermore, to avoid timing errors in the STALA-UUT connection, the experiment is performed in the non-critical operating region of each inter-connection, whereby we are ensured that timing errors occur in the UUT and not in the inter-connections.

VI. Results

The results are shown in Figures 6 and 7. First we observe that for all points characterizing the subcritical region, except one, the error rate follows the polynomial tendency seen for timing faults in Figure 4(a). Also, the error probability curves in both cases 1 and 2 have a vertex at $(p, p_e) = (0.5, 0.25)$ with only one type of errors (false positive or negative) appearing. Using Eqn. 4 we see that this corresponds to a case where either α or β is 1, depending on whether it is false positive or negatives appearing in the measurements. Finally, for case 3, the vertex of all curves are all at $(p, p_e) = (0.5, 0.5)$ which corresponds to a case where both α and β is 1. When the transition matrix only contains ones and zeros it becomes a truth table, and the fault mechanism therefore is deterministic.

Considering Case 1 in Figure 6, we see that both false positives and negatives occur in equal proportions once the voltage is scaled sufficiently. However, prior to this, we observe a gray region which corresponds to the timing model in Eqn. 3 with $\alpha = 1$ and $\beta = 0$. That is, bits are temporary 'hanging' in the HIGH state during $1 \to 0$ transitions, but not for $0 \to 1$ transitions. This is a somewhat unexpected behavior, as transitions in CMOS circuits usually exhibits a balanced behavior (or "slow to rise" behavior

if the capacitive load is large). However, we might explain this behavior due to the low threshold of the level shifter circuit which is in the signal path. As the IO voltage in the input makes a $1 \rightarrow 0$ transition, it must drop lower than normally before the output is triggered, which causes the false negative bias.

In case 2, we again observe a gray region, but in this case corresponding to false negatives. This behavior reflects the capacitive load of the PCB traces which increases the rise time of the digital signals. However, once we scale further a right skew appears on the error rate curve. By simulating the timing error model, we found that this happens when faults composite, i.e., when faults occur more than once along the signal path. The observed behavior corresponds to a case where a latch first produces false negatives (latch A), and when passed through a second faulty latch (latch C) with $\alpha = \beta = 1$ the skew appears in the error rate curve. This behavior is due to the fact that fault models add correlation to the test vector signal relative to the value of p, α and β. As the error rate of down-stream latches depend on the correlation coefficient (see Eqn. 4), the error rate response of these might be skewed.

Figure 7 (left to right) shows the output of the inverter chain at logic levels 12,14,16 and 18. These results show that the error rate has a transient behavior at the bit-level in the core-logic of the UUT. A small gradient is hinted in some places by the gray points in the critical region, but the resolution is too low to draw any definite conclusion on whether false positive occur before negatives or vice versa.

VII. Conclusion

For all experiments, the observed error rates exhibit a polynomial dependency on bit-probability, which corresponds to the behavior predicted by the timing error model. This provides a strong indication that for FPGAs operating in the subcritical voltage region, the observed errors are caused by setup-time violations in latches. Furthermore, for the observed voltage and frequency ranges, we consider that the threshold models are falsified as the observed error rates and that predicted by the threshold models do not compare. However, we will not dismiss that the threshold models are relevant if voltage-scaling beyond the POR limit is somehow made possible. With the possible exception of a very narrow region around the COP, our results also indicate that the fault mechanism is deterministic as the fault rates belong to $\alpha, \beta \in \{0, 1\}$ for all cases. Interestingly, we observe that $\alpha = \beta = 1$ in the core logic during subcritical operation, which is equivalent to imposing an extra delay element after each latch affected by timing errors. This insight, as well as the proposed timing error model, enable an interesting outset for future research on subcritical operation of FPGAs.

Finally, it is worth commenting the generality of the results: FPGAs are complex devices and a considerable amount of hardware is activated even for simple HDL kernels. As a result, many factors influence the behavior during voltage scaling, which makes the FPGA an unlikely candidate to preserve idealized CMOS behavior. Hence, if we, as we do in the FPGA under test, identify a fault mechanism general to CMOS, then we consider it is safe to assume that the same fault mechanism will be the source of errors in many other CMOS devices subject to voltage scaling. However, some caution should be taken when comparing with other FPGA devices, as architectures differ and vendors continuously make changes here to. Hence, we recommend performing a p-value sweep to verify that timing faults are indeed still the dominant cause of errors when experimenting with other FPGAs.

References

[1] C. T. Chow, L. S. M. Tsui, P. H. W. Leong, W. Luk, and S. Wilton, "Dynamic voltage scaling for commercial fpgas," in *in ICFPT, 2005*, 2005, pp. 173–180.

[2] J. L. Nez-Yez, V. A. Chouliaras, and J. Gaisler, "Dynamic voltage scaling in a fpga-based system-on-chip," in *FPL'07*, 2007, pp. 459–462.

[3] R. K. S. Narayanan, G. Lyle and D. Jones, "Testing the critical operating point (cop) hypothesis using fpga emulation of timing errors in over-scaled soft-processors," in *IEEE Workshop on Silicon Errors in Logic*, 2009.

[4] J. H. Patel, "Cmos process variations: A critical operation point hypothesis," Online presentation, 2008.

[5] D. Roberts, T. Austin, D. Blauww, T. Mudge, and K. Flautner, "Error analysis for the support of robust voltage scaling," in *Proceedings of the 6th International Symposium on Quality of Electronic Design*, ser. ISQED '05, 2005, pp. 65–70.

[6] D. Ernst, S. Das, S. Lee, D. Blaauw, T. Austin, T. Mudge, N. S. Kim, and K. Flautner, "Razor: Circuit-level correction of timing errors for low-power operation," *IEEE Micro*, vol. 24, no. 6, pp. 10–20, Nov. 2004.

[7] A. Birklykke, Y. Le Moullec, L. Alminde, and R. Prasad, "An automated test framework for experimenting with stochastic behavior in reconfigurable logic," in *Reconfigurable Computing and FPGAs (ReConFig), 2012 International Conference on*, 2012, pp. 1–6.

[8] Xilinx Inc., "Spartan-3 generation fpga user guide," Tech. Rep., June 2011.

[9] S.-d. Z. Ronald L. Cline, Andy T. Nguyen, "High-speed, low current level shifter circuits for integrated circuits having multiple power supplies," Patent US6 842 043 B1, 01 11, 2005.

[10] J. T. J. Davies, "Dual stage level shifter for low voltage operation," Patent US 6 838 924 B1, 01 04, 2005.

[11] K. Joe, H. David, and F. Cox, "Level shifting interfaces for low voltage logic," in *9 th NASA Symposium on VLSI Design 2000*, 2000, p. 4.

[12] S. Ramprasad, N. R. Shanbhag, and I. N. Hajj, "Analytical estimation of signal transition activity from word-level statistics," *IEEE Trans. on CAD*, vol. 16, pp. 718–733, 1997.

[13] P. Korkmaz, B. E. S. Akgul, K. V. Palem, and L. N. Chakrapani, "Advocating noise as an agent for ultra-low energy computing: Probabilistic complementary metal-oxide-semiconductor devices and their characteristics," *Japanese Journal of Applied Physics*, vol. 45, p. 3307, Apr. 2006.

[14] I. Grout, *Integrated Circuit Test Engineering: Modern Techniques.*

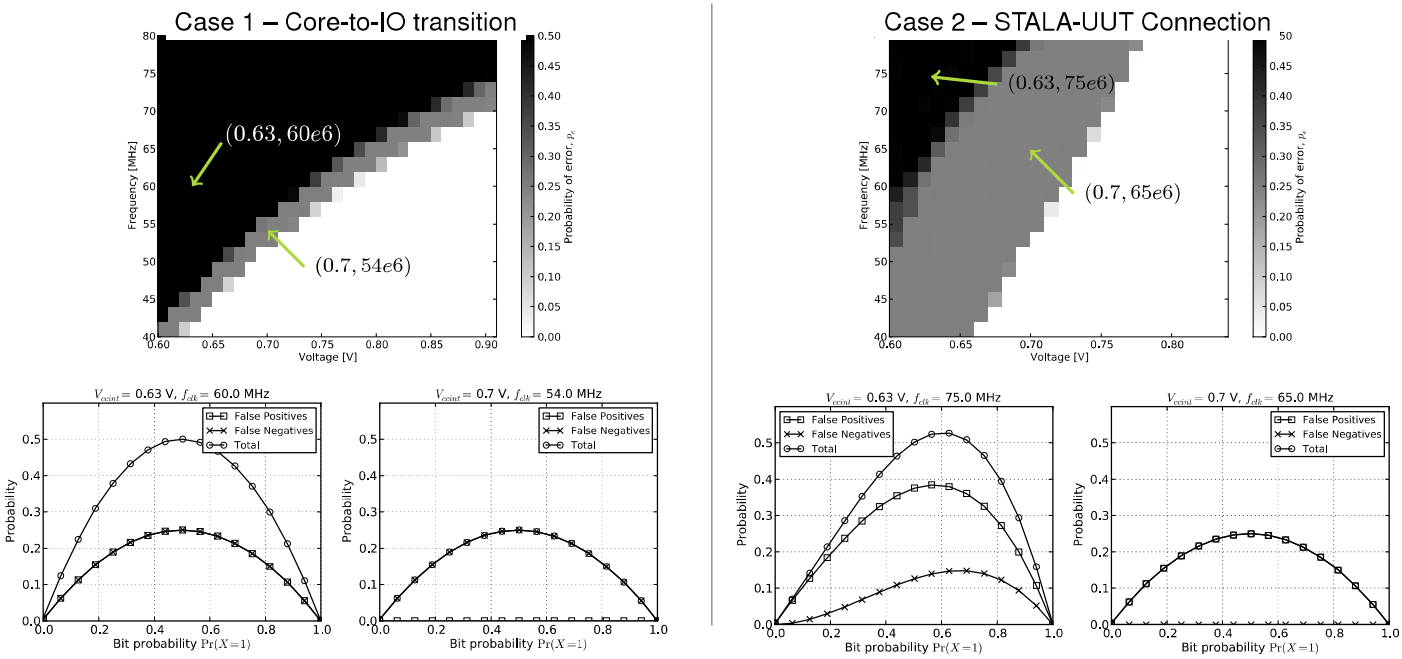


Fig. 6. Experimetal results for Case 1 and 2. The top pictures represent factor screening experiments over a wide range of voltage and frequency levels, and the bottom are bit-probability sweeps for the points marked in the top pictures.

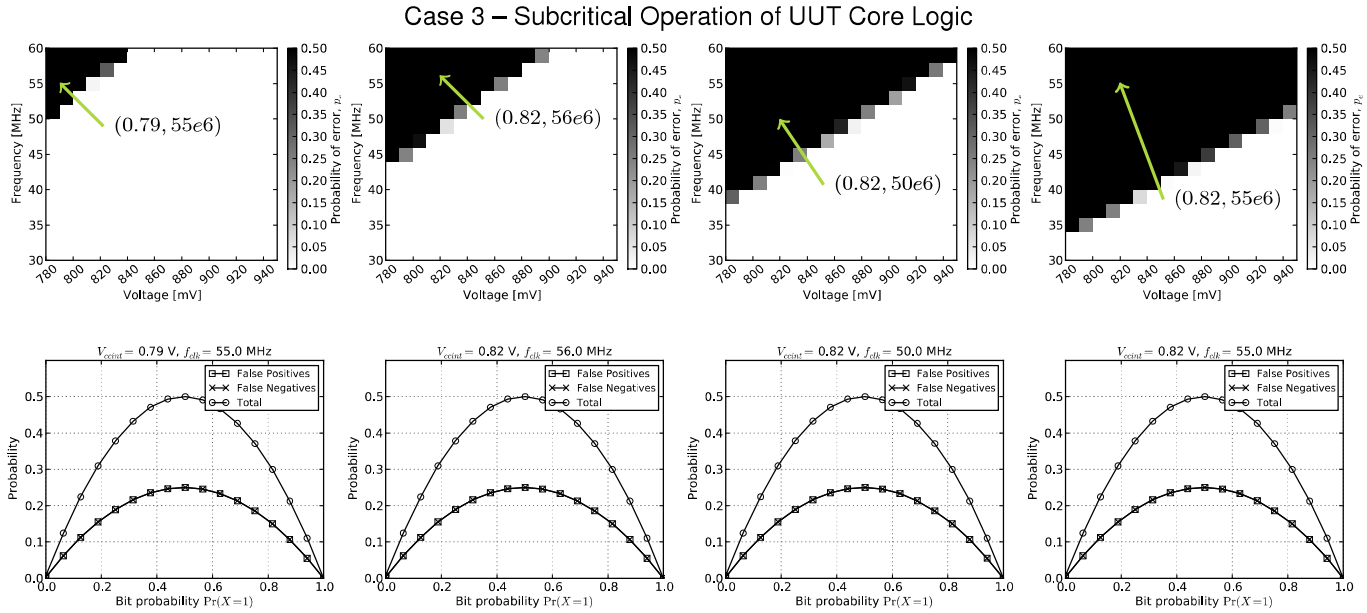


Fig. 7. Subcritical operation of an inverter chain. The upper row shows the error rate after inverter 12,14,16,18 for different voltages and frequencies. The bottom row shows sweeps over different input probabilities at selected points in the subcritical region for each inverter stage.

Chapitre 11

Projet "Global Air Traffic Awareness and Optimization through Space Based Surveillance (GATOSS)"

11.1 Contexte

Ce projet a été réalisé par Department of Electronic Systems, Aalborg University, et les entreprises danoises Gomspace ApS (voir le chapitre 10 pour plus d'informations) et DSE Airport Solutions. Il a été co-financé par Danish National Advanced Technology Foundation et les partenaires impliqués.
Deux étudiants de master, Morten Jensen et Bjarke Gosvig Knudsen, ainsi que le doctorant Alex Birklykke (voir son projet principal dans le chapitre 10), ont également participé à ce projet.

Le contrôle du trafic aérien dépends de plus en plus de la technologie *automatic dependent surveillance-broadcast* (ADS-B). Le principe de l'ADS-B est que l'électronique à bord des avions calcule leurs positions via les informations issues des systèmes de positionnement par satellite de type GPS et la rediffuse à intervalles réguliers (en fonction de la phase de vol) par radio vers les stations au sol.
Le terme (*dependent*) renvoi au fait que le système est dépendant des systèmes de positionnement par satellite, le terme *broadcast* indique que les données sont retransmises en mode diffusion (c.à.d. sans établissement de connexion)[1]

L'ADS-B est en cours de déploiement à travers le monde via, par exemple, les programmes NEXTGEN (USA), SESAR (Europe) et AIRE/ENGAGE (coopération internationale Atlantique). Les avantages de l'ADS-B par rapport aux radars et systèmes de communication traditionnels inclus des coûts infrastructurels moindres et des connaissances situationnelles améliorées.

Celles-ci sont améliorées à la fois pour les opérateurs de contrôle aérien qui ont accès à des données (positions, statuts et routes des avions) plus justes et plus précises dans la zone couverte et pour les avions équipés en ADS-B-in qui reçoivent ainsi des informations sur les avions voisins. L'ADS-B permet donc de réduire la taille des zones de sécurités entre

[1] Le site `www.flightradar24.com` propose de suivre les vols dont les données ADS-B lui sont transmises par diverses stations au sol.

les avions et ainsi d'utiliser les couloirs aériens de manière plus efficiente. D'autres réductions de coûts sont également possible car les données provenant des avions peuvent-être relayées entre les stations au sol, ce qui permet une gestion du trafic plus centralisée.

Cependant, comme l'ADS-B ne permet pas les communications de type trans-horizon, son application est limitée aux zones disposant de stations au sol ; ainsi il n'est pas possible de suivre avec justesse les vols passant, par exemple, au-dessus des régions océaniques (pas de station sur l'eau) ou désertiques (très peu de stations au sol). Malgré tous les efforts déployés jusqu'à présent, il reste donc encore une portion significative de l'espace aérien qui n'est pas surveillé de manière adéquate.

Afin d'améliorer cette situation, la vision de ce projet est le déploiement d'une constellation de petits satellites, appelés nano-satellites, chacun équipé d'un récepteur radio permettant la réception des signaux ADS-B transmis par les avions. Ces nano-satellites seraient placés en orbite basse terrestre [2], ici à une altitude comprise entre 600 et 700 km. Les données collectées par une telle constellation permettraient d'obtenir une image globale du trafic aérien. Relayées au sol, ces données apporteraient un complément significatif d'information aux sociétés chargées de la surveillance du trafic aérien.

Avec la solution envisagée dans ce projet, il serait ainsi possible d'obtenir ces données pour les zones océaniques (cf. les difficultés de localisation de l'épave des vols AF447 et MH370), celles peu peuplées ou bien encore pour certains pays en voie de développement qui ne peuvent se permettre de déployer des stations au sol.

Le principe du projet GATOSS est illustré en figure 11.1. Dans le cadre de ce projet, deux scénarios ont été envisagés.

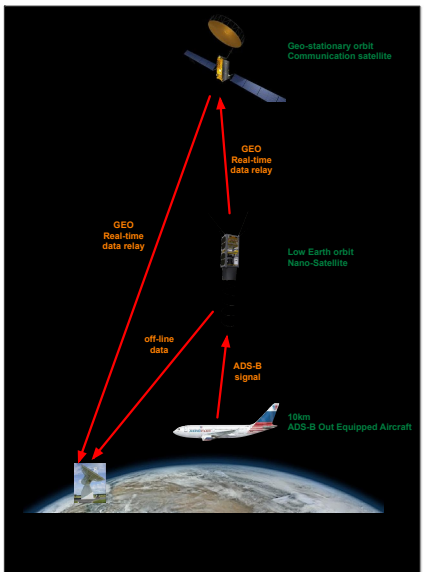


Figure 11.1: Illustration du concept du projet GATOSS. À noter que dans la mission de démonstration Gomx-1, seul le scénario 'offline' a été traité.

[2] L'orbite basse terrestre comprend les orbites dont l'altitude est comprises entre 160 km et 2000 km.

- Offline : la constellation est composée d'un nombre réduit de nano-satellites (entre trois et six). Ceci permet déjà d'obtenir une image améliorée du trafic par rapport à ce qui est possible à présent. Ces quelques nano-satellites scruteraient l'espace aérien et renverraient les données pour un traitement offline ; dans ce cas les données sont transmises lors des passes au-dessus des stations au sol, ce qui induit un certain délai.

- Online : il est estimé qu'en réduisant la séparation entre les avions lors des vols trans-océaniques il serait possible de faire circuler 16 fois plus d'avions par couloir aérien. Cela nécessiterait une constellation de nano-satellites plus conséquente (entre 40 et 70) et une retransmission des données en quasi temps-réel via des satellites de communication géostationnaires.

À noter que pour la mission appelée Gomx-1, seul le scénario 'offline' a été traité à des fins de démonstration.

L'objectif du projet étant de démontrer la faisabilité de la réception par un nano-satellite de signaux ADS-B émis par des avions en vol, la mission Gomx-1 comporte la conception, la réalisation, et la mise en orbite d'un nano-satellite embarquant un récepteur ADS-B à haute sensibilité. J'ai participé à la conception et la réalisation du récepteur ADS-B sur cible FPGA (détaillé dans l'article qui suit).
Pour information, je n'ai participé ni à la conception ni à la réalisation du nano-satellite proprement dit (tâche réalisée par Gomspace). De plus, le projet vise aussi à démontrer que les données récupérées par une station réceptrice au sol (communication VHF avec le nano-satellite) peuvent être intégrées de manière transparente avec les solutions informatiques de gestion des espaces aériens. Ce point n'est pas non plus présenté dans ce document car uniquement traité par DSE Airport Solutions et Gomspace ApS.

Une illustrations de synthèse du nano-satelitte Gomx-1 est donnée dans la figure 11.2. Le satellite est de type Cubesat, ici composé de deux unités de bases (une unité de base mesurant 10*10*10 cm). L'antenne hélicoïdale permet de capter les signaux ADS-B, les autres antennes servent à la communication VHF avec la station au sol.

11.2 Problématique

Étant donné le scénario décrit plus haut, le principal verrou technologique est la conception d'un récepteur suffisamment sensible pour capter les signaux ADS-B à une altitude comprise entre 500 km et 700 km au-dessus des avions. Le signal ADS-B est émis alternativement par des antennes supérieures et inférieures (respectivement sur le dessus et le dessous de la carlingue). La présence de l'antenne supérieure est donc un avantage ici. Néanmoins, les pertes de propagation sont significativement plus élevées que dans le scénario air-sol. De plus, le diagramme de rayonnement de l'antenne émettrice peut également se traduire par des pertes de pointage. Le niveau du signal reçu par le nano-satellite est estimé (voir publication C.16 pour le calcul) à -133 dBW, c.à.d. -103 dBm ou bien encore 50.12 femtowatt [3].

[3] Pour comparaison, -110 dBm (10 femtowatt) correspondent à la limite de réception d'un signal à étalement de spectre en téléphonie mobile

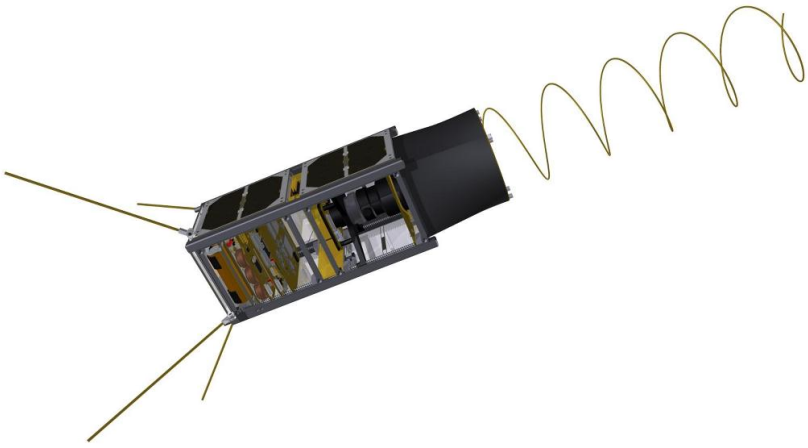


Figure 11.2: Illustration de synthèse du nano-satellite Gomx-1. Illustration fournie par Gomspace ApS.

La méthode décrite dans le standard ADS-B n'est pas adaptée au scénario envisagé. En effet, celle-ci est basée sur la détection d'énergie et lors de la détection de préambule les signaux dont le niveau est inférieur à -118 dBW (c.à.d. -88 dBm) sont rejetés. Il est donc nécessaire d'utiliser une autre méthode, point qui est résumé dans ce qui suit.

11.3 Résumé des contributions

Nos contributions consiste en la conception et la réalisation d'un récepteur ADS-B à haute sensibilité construit à partir de composants commerciaux pris sur étagère (*commercial off-the-shelf'* ou COTS). Son architecture est composé d'une tête radiofréquence à amplificateur logarithmique (*log-amp RF front-end*), d'un convertisseur analogique-numérique (*ADC*), d'un microcontrôleur (*MCU*) et d'un co-processeur FPGA. La figure 11.3 donne le schéma de principe de cette architecture.

Nous avons tout d'abord élaboré le bilan de liaison pour le scénario considéré (voir publication C.16 pour le calcul). Celui-ci indique que la réception de signaux ADS-B à partir d'un nano-satellite en orbite basse terrestre est possible mais non triviale : les pertes totale sont estimées à environ 159 dB et le niveau de signal arrivant au satellite est estimé à -133 dBW.

Deux décodeurs pour le récepteur ont été étudiés, implantés et évalués. Les détails du premier décodeurs sont fournis dans ce qui suit, ceux du second sont pour le moment encore confidentiels.

Pour le premier décodeur nous avons exploré trois approches à estimation pour la syn-

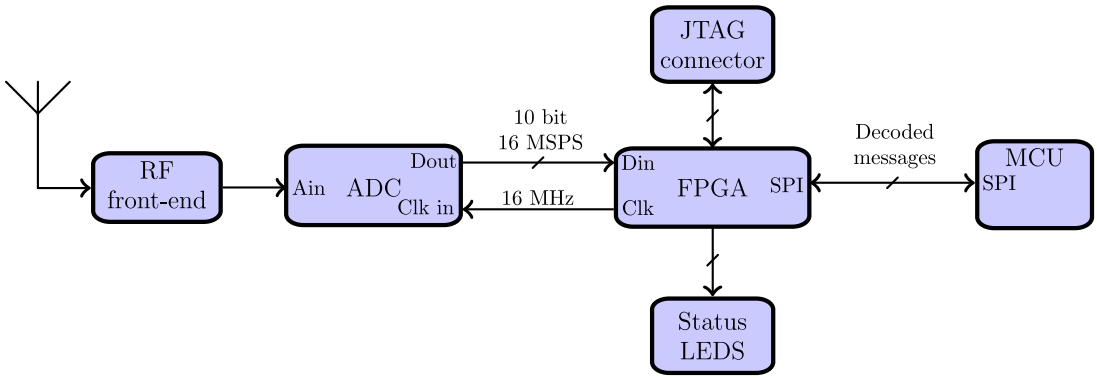


Figure 11.3: Schéma de principe du récepteur ADS-B.

chronisation et le décodage : maximum de vraisemblance, maximum de vraisemblance approximative et corrélation. Les résultats de simulation ont montré que, pour le scénario considéré, les trois approches donnent des taux d'erreurs similaires, en conséquence celle ayant la moindre complexité algorithmique (c.à.d. celle à corrélation) a été sélectionnée pour la phase d'implantation.

11.3.1 Partie RF

La partie RF étant confidentielle, je me limite à décrire son rôle, c.à.d. convertir le signal PPM (Pulse Position Modulated) 1090 MhZ ASK (Amplitude Shift Keying) en bande de base via une architecture à base d'un amplificateur logarithmique.

11.3.2 Décodeur

Le format des messages ADS-B est décrit dans le standard [19]. Pour ce premier décodeur nous proposons d'effectuer la synchronisation et le décodage des symboles dit au moyen d'une approche à corrélation. Pour la synchronisation il s'agit de calculer la corrélation-nelle entre le signal reçu et un mot de synchronisation S_w composé du préambule ADS-B et du champ DF (Data Field) = 17. Nous utilisons aussi un second mot de synchronisation (S_{wInv} = 1-S_w) et envoyons les deux résultats dans un 'discriminateur' qui calcule la probabilité de synchronisation.
Pour le décodage des symboles (les 112 bits qui suivent le préambule) nous calculons la corrélation entre le signal reçu et les mots de symboles (un '1' logique est représenté par une transition niveau bas vers niveau haut et inversement pour un '0' logique).

Dans l'implantation effectuée, le bloc de synchronisation reçoit les données 10 bit non signées à 16 MSPS et envoi le bit de synchronisation et les échantillons de données retardés à 16 MSPS par seconde vers le bloc de décodage des symboles. Ce bloc est réalisé à l'aide de filtres FIR et est compléter par un mécanisme de seuil dynamique à moyenne variable qui évite les fausses détections lorsqu'il n'y a pas d'activité.
Le décodeur symboles est constitué d'un estimateur et d'un test de seuil. Le cœur est réalisé à l'aide d'un additionneur et d'un multiplicateur. La sortie consiste en une série de bit à 1 Mbit/s correspondants aux symboles décodés.

11.3.3 Post-traitement

Le bloc de post-traitement consiste d'une mémoire tampon qui stocke les 112 symboles ADS-B, d'un module de test CRC et d'un module SPI qui envoi les données au micro-contrôleur si elles sont valides.

11.3.4 Chiffres d'implantation

Sur le FPGA Altera Cyclone IV EP4CE6, notre implantation requiert 4770 (76%) des éléments logiques, 22 (73%) des multiplicateurs, 16248 (6%) des bits mémoire et un des deux blocs PLL. Les résultats montrent que c'est le bloc de synchronisation qui demande le plus de ressources. La figure 11.4 montre une photographie du prototype.

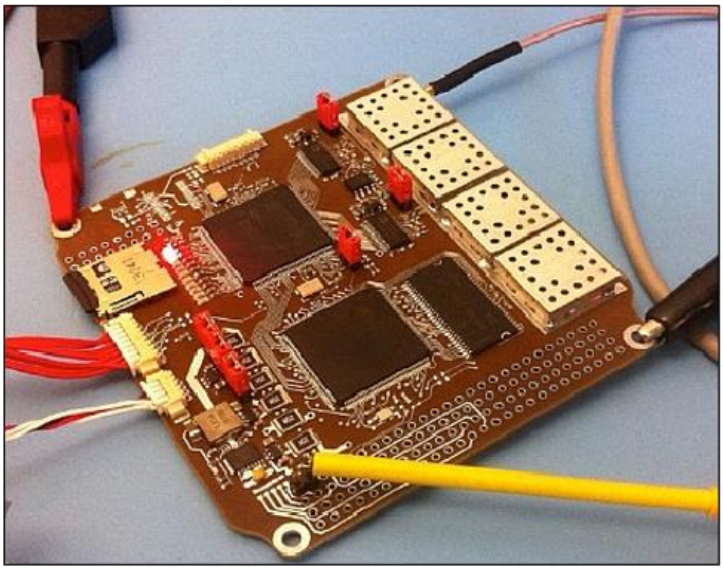


Figure 11.4: Photographie du prototype du récepteur ADS-B pour nano-satellite. Illustration fournie par Gomspace ApS.

11.3.5 Tests au sol et dans l'espace

En ce qui concerne ce premier décodeur, les résultats de différents essais effectués au sol (voir figures 11.5, 11.6, 11.7 et 11.8) montrent que sa sensibilité et sa couverture en réception sont toutes deux supérieures à celles de deux récepteurs de référence : le test de sensibilité indique qu'au minimum des signaux de -100 dBm peuvent être reçus et le test de couverture montre qu'une distance d'au moins 700 km est possible.

Le nano-satellite a été placé en orbite le 21 novembre 2013 (la mission de lancement[4], connue sous le nom 'DubaiSat-2 cluster', a permis la mise en orbite de 31 satellites par un lanceur de type Dnepr, lancé de la base de Yasny, Oblast d'Orenburg, Russie) [20].

Les résultats issus de l'analyse des données collectées sur la période du 26 novembre 2013 au 22 janvier 2014 sont très satisfaisants. Les figures 11.9, 11.10 et 11.11 montrent les

[4] À ne pas confondre avec la mission Gomx-1 proprement dite

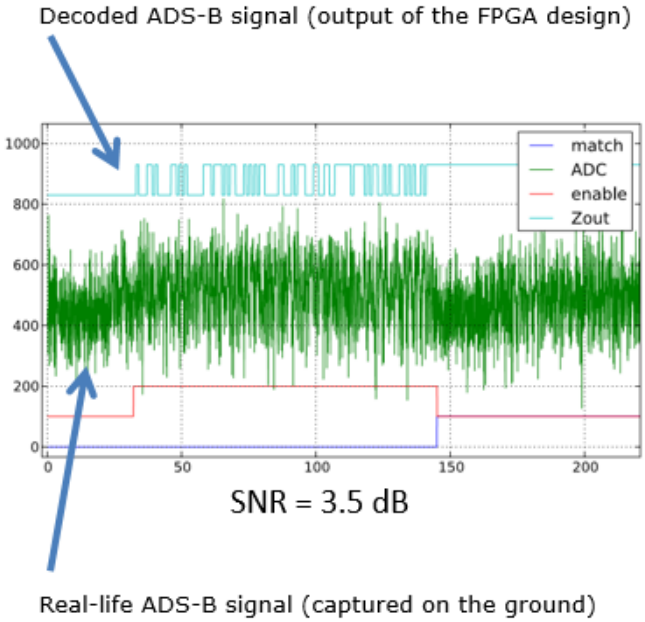


Figure 11.5: Résultats de vérification au sol, cas SNR faible (3.5 dB). Extrait de la thèse de master de Morten Jensen et Bjarke Gosvig Knudsen.

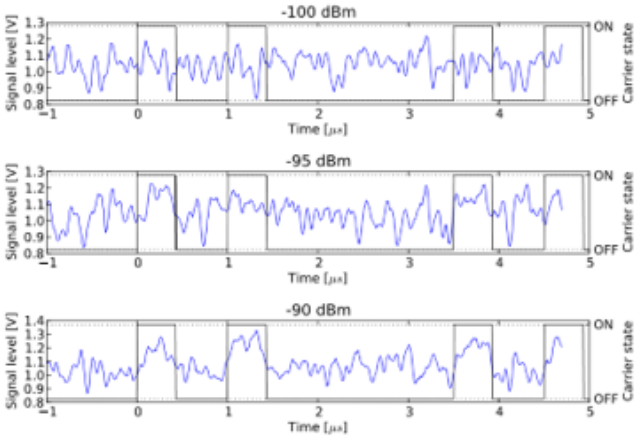


Figure 11.6: Résultats de vérification au sol pour différent niveaux de signal d'entrée. Extrait de la thèse de master de Morten Jensen et Bjarke Gosvig Knudsen.

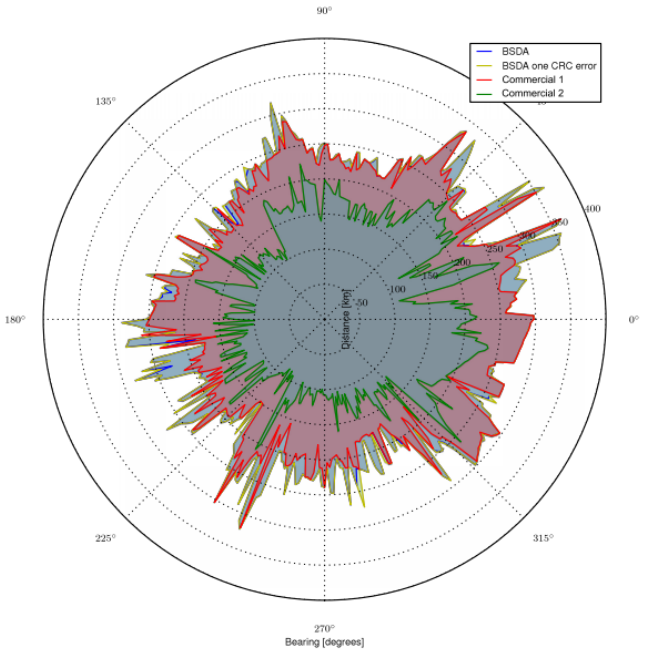


Figure 11.7: Portée max des différents récepteurs au sol. La résolution angulaire est de un degré. Extrait de la thèse de master de Morten Jensen et Bjarke Gosvig Knudsen.

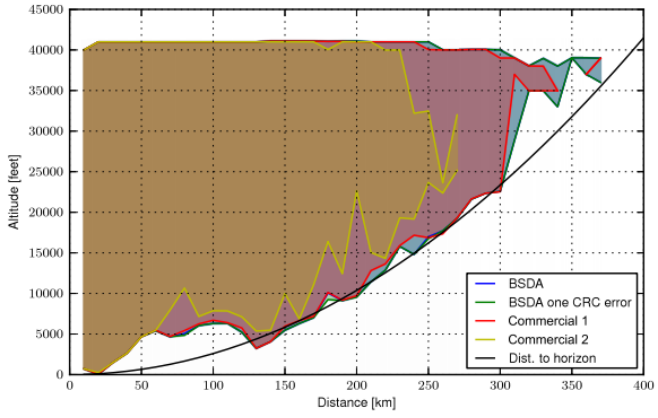


Figure 11.8: Altitudes min et max observée en fonction des bandes de distance (bandes de 10 km) au sol. La ligne noire montre la distance à l'horizon théorique en fonction de la hauteur. Extrait de la thèse de master de Morten Jensen et Bjarke Gosvig Knudsen.

résultats obtenus dans l'espace. Elles montrent clairement que de nombreuses positions ont été reçues et que celle-ci sont fortement corrélées avec les routes commerciales. Elles montrent aussi qu'il est possible de couvrir les zones dépourvues de station terrestre (ici océan atlantique et désert du Sahara).

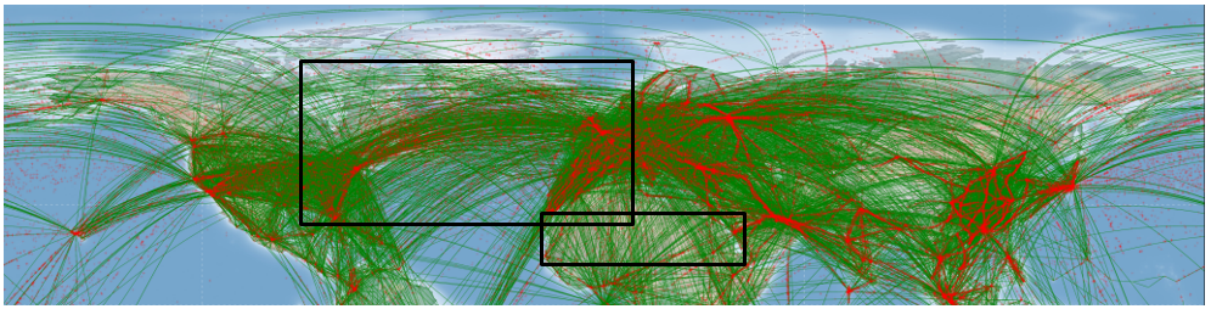


Figure 11.9: Résultats dans l'espace pour l'hémisphère nord. Les croix rouges indiquent les positions d'avions reçues, décodées et rapportées par Gomx-1 ; les lignes vertes indiquent les routes commerciales en janvier 2014 (base de données Openflights [21]). Extrait de C.16.

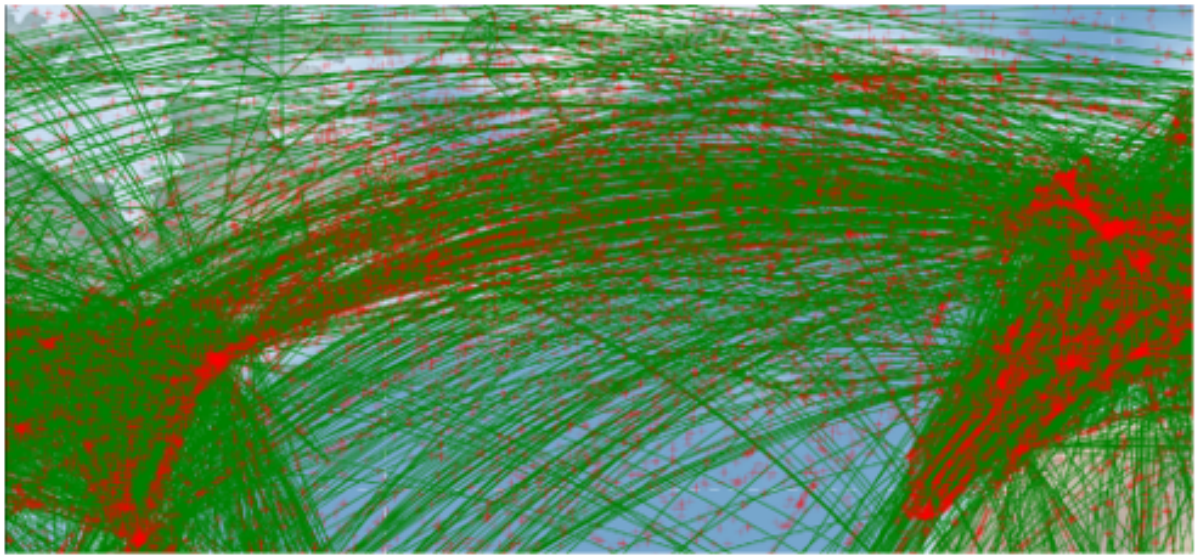


Figure 11.10: Résultat dans l'espace, zoom sur l'océan atlantique. Les croix rouges indiquent les positions d'avions reçues, décodées et rapportées par Gomx-1 ; les lignes vertes indiquent les routes commerciales en janvier 2014 (base de données Openflights [21]). Extrait de C.16.

11.4 Publications et commentaire

Ce projet a donné lieu à deux publications (C.22 et C.16 dans le chapitre 5) ainsi qu'à la thèse de master de Morten Jensen et Bjarke Gosvig Knudsen.

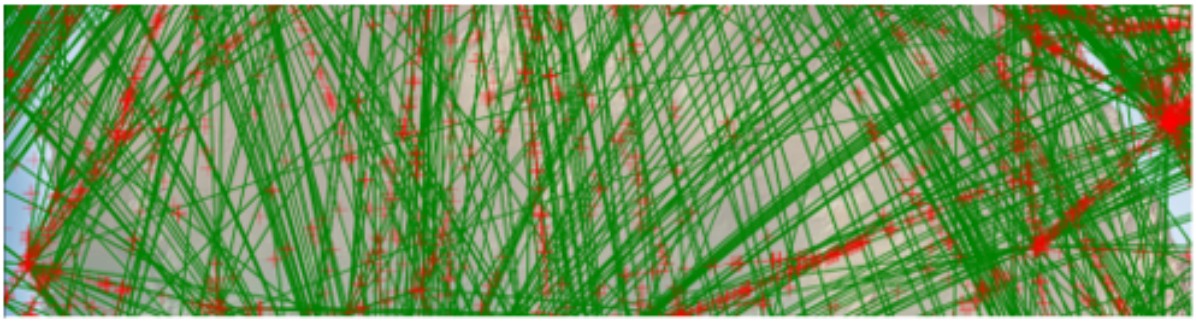


Figure 11.11: Résultat dans l'espace, zoom sur le désert du Sahara. Les croix rouges indiquent les positions d'avions reçues, décodées et rapportées par Gomx-1 ; les lignes vertes indiquent les routes commerciales en janvier 2014 (base de données Openflights [21]). Extrait de C.16.

La première (*GomX-1: A Nano-satellite Mission to Demonstrate Improved Situational Awareness for Air Traffic Control*) donne une vue d'ensemble du projet. La seconde (*ADS-B in Space: Decoder Implementation and First Results from the GATOSS Mission*) résume la conception, la réalisation, l'évaluation au sol et les résultats dans l'espace du récepteur ADS-B avec le premier décodeur. Cette deuxième publication est reproduite dans les pages qui suivent.

Au final, ces résultats ont montré la faisabilité de la vision initiale du projet. Jusqu'à ce jour, et en ce qui me concerne, c'est aussi le projet qui a donné lieu le plus d'interactions entre le milieu universitaire et l'industrie (implication significative de Gomspace et présence sur le même campus). Il est également intéressant de noter que Gomspace continue d'exploiter ces travaux dans le cadre de la mission Gomx-3 (Gomx-2 avait quant à elle échouée pour cause d'explosion de la fusée Antares 130 au décollage).

Pre-print version. The final version has been published in the proceedings of the 14th Biennial Conference on Electronics and Embedded Systems (BEC 2014). DOI: 10.1109/BEC.2014.7320555

ADS-B in Space: Decoder Implementation and First Results from the GATOSS Mission

Bjarke Gosvig Knudsen, Morten Jensen, Alex Birklykke, Peter Koch
Department of Electronic Systems
Aalborg University
Aalborg, Denmark

Johan Christiansen, Karl Laursen, Lars Alminde
Gomspace ApS
Aalborg, Denmark

Yannick Le Moullec
T.J. Seebeck Department of Electronics
Tallinn University of Technology
Tallinn, Estonia

***Abstract*—ADS-B is increasingly used for air traffic control in areas covered by terrestrial receivers; however, its limited range makes it unsuitable for other areas such as the oceans. To overcome this limitation, it has been proposed to receive ADS-B signals from low earth orbit nano-satellites and relay them to the terrestrial receivers. This paper gives an overview of the GATOSS mission and of its highly-sensitive ADS-B software-defined radio receiver payload. Details of the design and implementation of the receiver's decoder are introduced. The first real-life, space-based results show that ADS-B signals are indeed successfully received in space and retransmitted to a terrestrial station by the GATOSS nano-satellite orbiting at 700+ km altitudes, thus showing that GATOSS is capable of tracking flights, including transoceanic ones, from space.**

I. Mission Overview

Air traffic management (ATM) increasingly relies on automatic dependent surveillance-broadcast (ADS-B) [1] technology. ADS-B is currently being rolled out around the world through the NEXTGEN [2], SESAR [3] and AIRE/ENGAGE [4] programs. The advantages of ADS-B over traditional radar and radio systems include lower infrastructure costs and improved situational awareness, both for ATMs that have access to more accurate data about the position, status and routes of all aircraft within range and for ADS-B equipped aircraft that receive information about other aircraft in the nearby airspace.

However, as over-the-horizon communication is not possible, ADS-B cannot accurately track flights passing over areas without ground stations. As a result, a large part of the airspace still remains unsupervised e.g., oceanic or arctic regions [5] and areas scarcely fitted with ADS-B ground stations. To enable global ADS-B coverage, we propose a space-based ADS-B surveillance infrastructure that i) collects flight information from a constellation of low Earth orbit (LEO) nano-satellites equipped with sensitive ADS-B receivers, and ii) relays the collected flight information to ATM operators through ground stations. We refer to this as global air traffic awareness and optimization through spaceborne surveillance (GATOSS).

Two scenarios for providing space-based ADS-B service have been envisaged:

- Off-line data: a few (three to six) nano-satellites sample the airspace and provide information for off-line data processing; in this case the data is downlinked with delay when the nano-satellites pass over one or more ground stations.
- On-line data: a near real-time picture of the airspace can be achieved by means of a larger fleet (40 to 70 nano-satellites) that communicate to the ATM infrastructure via geostationary data relay satellites.

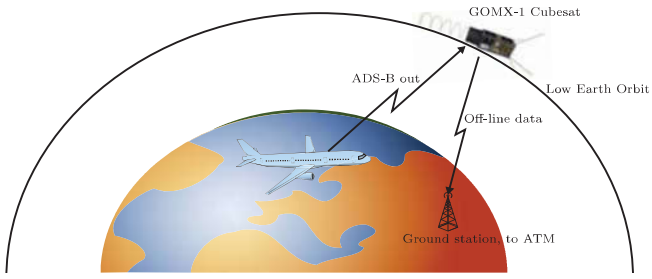


Fig. 1. GATOSS *demonstration* mission: ADS-B out signals transmitted from the aircraft are received and decoded by the proposed payload on-board the Cubesat nano-satellite and re-transmitted to ground stations. By doing so, it is possible to track flights passing over areas not equipped for ADS-B reception such as the oceans and deserts. Gomx-1 [6], the initial name of the nano-satellite, was changed to GATOSS after the launch.

To evaluate the feasibility of such an approach, a *demonstration* mission is currently taking place. Its principle is shown in Fig. 1. The GATOSS satellite is a two-unit Cubesat [7] nano-satellite constructed from commercially available subsystems combined with a custom ADS-B software-defined radio payload. The payload consists of a deployable helical antenna and a high sensitivity ADS-B receiver. In the receiver, the signal decoding is performed by a field programmable gate array (FPGA) that can be reconfigured from ground if needed.

II. Design and Implementation

A. Receiver design

In an air-to-satellite scenario the propagation loss are significantly larger than in the traditional air-to-ground scenario and radiation patterns of the transmitting antenna on-board the

The authors thank the Danish National Advanced Technology Foundation for partial funding. Yannick Le Moullec was affiliated with Aalborg University when the work began.

Pre-print version. The final version has been published in the proceedings of the 14th Biennial Conference on Electronics and Embedded Systems (BEC 2014). DOI: 10.1109/BEC.2014.7320555

aircraft might result in significant pointing loss. Thus, we first performed a link budget analysis to assess whether or not ADS-B signals can be received in space at an approximate altitude of 720 km.

The link budget for the given ADS-B scenario is summarized in Table I.

Parameter	Value
Worst case effective radiated power (ERP)	25.8 dBW (55.8 dBm)
Atmospheric gasses loss (ATH_L)	2.1 dB
Particles in the ionosphere loss (I_L)	0.2 dB
Antennas loss (AN_L)	6 dB
Slant range (S)	739 km
Channel path loss, $P_L = 20\log_{10}\left(\frac{4\pi S}{\lambda}\right)$ where λ is the wavelength	150.6 dB
Total transmission loss, $T_L = P_L + ATH_L + I_L + AN_L$	158.9 dB
Expected signal level at the satellite, $S_l = ERP - T_L$	-133.1 dBW (-103.1 dBm)

TABLE I. SUMMARY OF THE LINK BUDGET FOR THE SPACE-BASED ADS-B SCENARIO.

As shown in Table I, the expected signal level at the satellite, S_l, is only -133.1 dBW (i.e., -103.1 dBm) which means that a highly sensitive receiver is needed. The architecture of the proposed ADS-B receiver is shown in Figure 2. It is composed of five main elements:

- a deployable helical antenna,
- an RF front-end responsible for down-converting the 1090 MHz amplitude shift keying (ASK)-modulated pulse position modulated (PPM) ADS-B signal to baseband,
- a 10-bit Analog Devices AD9203 analog-to-digital converter (ADC) clocked at 16 MHz,
- an Altera Cyclone IV EP4CE6 FPGA used as a co-processor for decoding the baseband PPM ADS-B signal,
- an Atmel AT32UC3A microcontroller unit (MCU) responsible, among others, for extracting the information contained in the decoded ADS-B frames and preparing and retransmitting them to the ground stations.

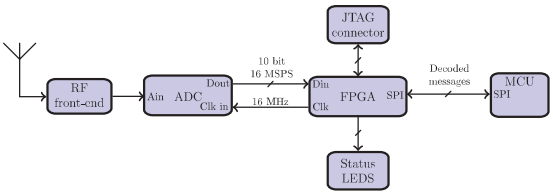


Fig. 2. Architecture of the ADS-B receiver. The FPGA is used as a co-processor for decoding the ADS-B PPM frames.

B. Decoder design and implementation

Readers unfamiliar with the ADS-B message format can get more information in e.g., [1].

In the proposed decoder, both the synchronization and actual decoding build upon a standard correlation approach, briefly discussed below.

The synchronization consists in calculating the correlation between the received signal and the synchronization word. For this demonstration mission, only civilian ADS-B messages with data field (DF) = 17 are of interest; hence using a synchronization word consisting of both the preamble and DF17 field is possible, i.e., S_w = {Preamble DF17}. Moreover, we also use the inverse of S_w as a second synchronization word, i.e., $S_{wInv} = 1 - S_w$. With two synchronization words it is possible to use a discriminator to calculate which of the two scenarios (synch. or no synch.) has most likely occurred, see Equation 1

$$Z_{synchronization} = \sum_{i=1}^{N} y_i(s_w, i - s_{wInv,i}) \quad (1)$$

Similarly, the discriminator for decoding the symbols is given in Equation 2

$$Z_{decoding} = \sum_{i=1}^{N} y_i\left(s_{1,i} - s_{0,i}\right) \quad (2)$$

The architecture of the decoder is shown in Figure 3. It consists of i) a synchronization block (standard correlation with adaptive threshold based on a moving average and an offset) that detects ADS-B messages (preamble detection), ii) a symbol decoding block (standard correlation with threshold) that decodes the 112-bit data block that follows the preamble, and iii) a post-processing block that performs a cyclic redundancy check (CRC) on the data block to verify that a valid message has been decoded properly; valid messages are passed on to the MCU over an SPI link.

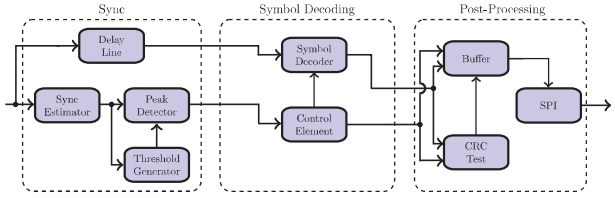


Fig. 3. ADS-B Decoder Block Diagram.

1) Synchronizer Implementation: The synchronization block receives 10-bit unsigned data at 16 Mega-samples per second (MSPS) and outputs a synchronization bit and the delayed 10-bit data samples to the symbol decoding block at 16 MSPS. It consists of a synchronization estimator, a threshold generator, a peak detector and a delay line. The synchronization estimator is implemented by means of an FIR filter of which the coefficients are +1 and -1.

The threshold generator is implemented as a moving average with an offset. This (positive) offset ensures that the generated threshold does not converge towards zero when there is no activity; hence a small amount of noise will not trigger the synchronizer. The threshold generator is given in Equation 3

$$Z_{TH} = \alpha + \frac{1}{N}\sum_{i=-N}^{0} Z_i \quad (3)$$

Pre-print version. The final version has been published in the proceedings of the 14th Biennial Conference on Electronics and Embedded Systems (BEC 2014). DOI: 10.1109/BEC.2014.7320555

where α is the offset, N the length of the moving average, and Z_i the i^{th} output of the synchronizer. The threshold generator updates the threshold value every time the synchronization estimator outputs a new value. The implementation consists of a delay line holding the 64 previously values from the synchronization estimator, 63 adders and one shift register.

The correlation peak in the synchronization estimator is present when the last sample of the synchronization word is clocked into the synchronization estimator. The last sample of the synchronization word is a part of the DF17 data field. Since the DF17 data field is a part of the 112-bit data block, it is necessary to delay the data samples to the symbol decoding block by 84 samples (80 for the DF17 field + 3 for the synchronization estimator + 1 for the peak detector).

2) Decoder Implementation: The inputs to the decoder are one synchronization bit and the delayed 10 data samples from the synchronization block. The 1 Mbit/s output of the decoder is a single bit, corresponding to the decoded symbol. The symbol decoder consists of a decoding estimator and a threshold test. The synchronization word is pre-calculated and the symbol decoder needs to output one value for each 16^{th} sample; the estimator, without its control logic, is implemented with one adder and one multiplier.

The threshold test is implemented by comparing, on every 16^{th} rising edge of the clock, the value of the decoding estimator to a threshold value of zero. If the resulting value is larger than zero, the decoded symbol is assigned a '1', otherwise a '0'.

3) Post-processing Implementation: The post-processing block consists of i) a buffer that stores the 112 symbols before they are sent to the MCU, ii) a CRC test block that calculates and verifies the checksum, and iii) an SPI module that transmits the ADS-B data block to the MCU when the checksum is correct.

4) Implementation Results: Table II summarizes the Cyclone IV EP4CE6 FPGA resource usage. The implementation requires 76% of the available logic elements (LE), 73% of the embedded multipliers, 6% of the memory bits, and one of the two phase-locked-loop (PLL) blocks. It is worth noting that it is the synchronization block that requires the most LEs, multipliers and memory bits.

	Sync.	Dec.	Post-proc.	Total	Avail.	%
LEs	4350	87	333	4770	6272	76
Multipliers	22	0	0	22	30	73
Memory bits	14200	0	2048	16248	276k	6

TABLE II. CYCLONE IV EP4CE6 FPGA RESOURCE USAGE.

III. LAUNCH AND FIRST RESULTS

The GATOSS nano-satellite, shown in Fig. 4, was launched on 21 November 2013 together with 23 other satellites on the 'DubaiSat-2 Cluster' mission [8]. The space launch vehicle was a Dnepr rocket named "RS-20". The launch was executed by the ISC Kosmotras industrial team from Yasny launch base, Russia. Fig. 5 shows a snapshot of "RS-20" Dnepr rocket during take off on 21 November 2013.

The first results of the GATOSS mission are shown in Fig. 6. The figure shows the aircraft positions (red markers) that have been received, decoded, and reported by the GATOSS nano-satellite (period: Tue, 26 Nov 2013 00:22:17 GMT to Wed, 22 Jan 2014 14:55:17 GMT) plotted against the great circle routes (green lines, extracted from Openflights database [9] in January 2014). For readability reasons we only show the northern hemisphere.

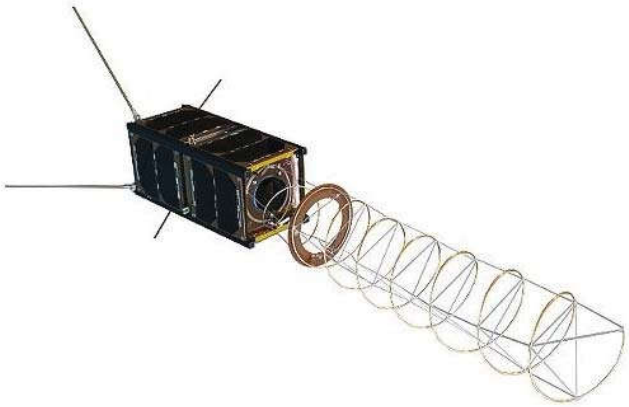


Fig. 4. Illustration of the GATOSS nano-satellite with its helical antenna deployed.

Fig. 5. DubaiSat-2 Cluster mission launch. Photo credit: EIAST

In Fig. 6, areas with intense air traffic are clearly seen (e.g. North America's East and West Coasts, Europe, East Asia). A fairly large number of aircraft positions have been spotted during the above mentioned period: 87186 positions corresponding to 12212 different plane registrations.

A closer look at Europe's West coast, North America's East Coast and transatlantic oceanic routes, Fig. 7, confirms that ADS-B signals emitted by aircraft flying over oceanic areas (with no ADS-B terrestrial stations) can be received and decoded in LEO by GATOSS' receiver.

Fig. 8 is a zoom over the Sahara Desert which has one of the lowest population densities in the world. The plot confirms that it is also possible for the receiver on-board GATOSS to receive and decode ADS-B signals emitted by aircraft flying over ground areas that are scarcely fitted with ADS-B ground stations.

Pre-print version. The final version has been published in the proceedings of the 14th Biennial Conference on Electronics and Embedded Systems (BEC 2014). DOI: 10.1109/BEC.2014.7320555

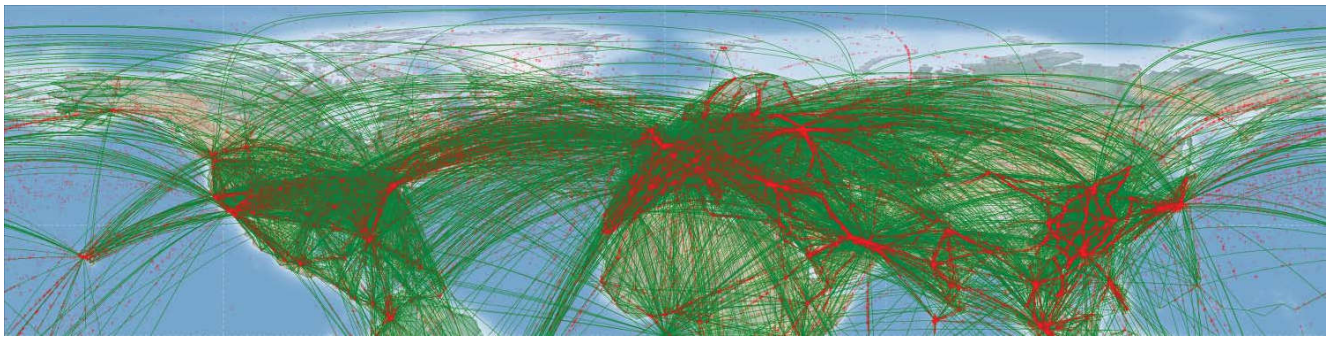


Fig. 6. First results (for readability reasons we only show the northern hemisphere). Red markers: aircraft positions received, decoded, and reported by the GATOSS nano-satellite. Green lines: great circle routes (Openflights database [9], January 2014).

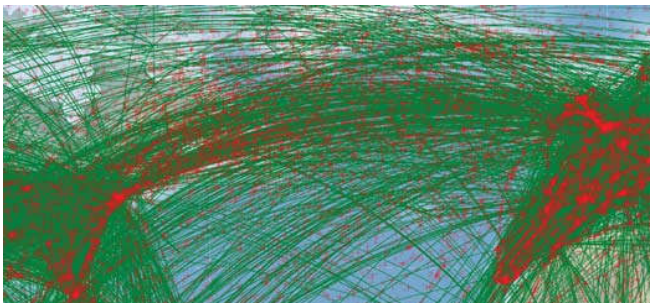


Fig. 7. First results. Zoom over Europe's west coast, North America's East Coast and transatlantic routes. Red markers: aircraft positions. Green: great circle routes (Openflights database [9], January 2014).

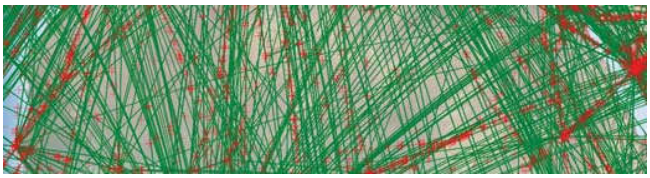


Fig. 8. First results. Zoom over the Sahara Desert. Red markers: aircraft positions. Green: great circle routes (Openflights database [9], January 2014).

Although the figures indicate that there is an overall good match between the aircraft positions and the great circle routes, some aircraft positions do not overlap with any route. This could be explained by i) the dynamic nature of the route database (routes are removed and added on a regular basis) and ii) planes do not always use a great circle route (plotted in green), they also use jet stream routes (not plotted) for efficiency purposes.

IV. Concluding Remarks

The first GATOSS results are very encouraging and thus far the demonstration mission is deemed successful. As the number of aircraft positions being collected continues to increase, it will be possible to carry out more thorough, statistically significant analyzes of the data, including, but not limited to, cross-checking the reported positions with actual flight records. Furthermore, the input signal collected by the ADC of the payload will also be analyzed to get a deeper understanding of the space propagation effects on the ADS-B signal. It would also be valuable to compare our results to that of the ADS-B receiver demonstrator fitted on ESA Proba-V miniaturized (not nano) satellite [10]. Finally, the reliability in space of the receiver (built from commercial off-the-shelf components) will be assessed.

References

[1] RTCA (Radio Technical Commission for Aeronautics), "RTCA DO-260B, minimum operational performance standards for 1090 mhz extended squitter automatic dependent surveillance broadcast (ads-b) and traffic information services broadcast (tis-b)," Tech. Rep., 2009.

[2] Joint Planning and Development Office, "Nextgen 101 addressing the nextgen challenge," visited April 2014. [Online]. Available: http://www.jpdo.gov/library/20090618_NextGen_101.pdf

[3] SESAR Joint Undertaking, "Sesar brochure: Modernising the european sky," visited April 2014. [Online]. Available: http://www.sesarju.eu/sites/default/files/documents/reports/NEW-sesar09-2011-newbassdef-4_0.pdf?issuusl=ignore

[4] SESAR Joint Undertaking, "Partnership with aire programme," visited April 2014. [Online]. Available: http://www.sesarju.eu/environment/aire

[5] R. Francis, R. Vincent, J.-M. Noel, P. Tremblay, D. Desjardins, A. Cushley, and M. Wallace, "The flying laboratory for the observation of ads-b signals," *International Journal of Navigation and Observation*, vol. 2011, no. ID 973656, 2011.

[6] L. Alminde, J. Christiansen, K. Laursen, A. Midtgaard, M. Bisgard, M. Jensen, B. Gosvig, A. Birklykke, P. Koch, and Y. Le Moullec, "Gomx-1: A nano-satellite mission to demonstrate improved situational awareness for air traffic control," in *26th Annual AIAA/USU Conference on Small Satellites*, Logan, UT, USA, August 2012.

[7] The CubeSat Program, Cal Poly SLO, "Cubesat design specification (cds) rev 13 provisional," visited April 2014. [Online]. Available: http://www.cubesat.org/images/developers/cds˙rev13˙draft˙b.pdf

[8] I. Kosmotras, "21 november 2013. dnepr cluster mission 2013," visited April 2014. [Online]. Available: http://www.kosmotras.ru/en/launch14/

[9] Openflights, "Openflights route database [r760] jan 2014 update," visited March 2014. [Online]. Available: http://sourceforge.net/p/openflights/code/HEAD/tree/openflights/data/routes.dat

[10] DLR, "Ads-b over satellite - first aircraft tracking from space," visited April 2013. [Online]. Available: http://www.dlr.de/dlr/presse/en/desktopdefault.aspx/tabid-10308/471˙read-7318/year-all/#gallery/11231

Partie III

Projets en cours, perspectives de recherche et réflexions sur l'enseignement

Chapitre 12

Projets en cours

12.1 Projet de coopération Estonie-Afghanistan

Ce projet de coopération pour le développement a été mis en place par Estonian Ministry of Foreign Affairs et son équivalent en Afghanistan. L'un des volets de ce projet concerne le développement du secteur éducationnel en Afghanistan et a abouti à une coopération entre Kabul University, Tallinn University et Tallinn University of Technology ; son implantation a commencé début 2014. Il s'agit notamment de créer des programmes de master et de doctorat liés à divers aspects des TICs. Avant ce projet, les études a Kabul University s'arrêtaient à la licence ; la première promotion de master est sortie au printemps 2016. De plus, douze doctorant(e)s afghan(ne)s sont encadré(e)s par des chercheurs et chercheuses de Tallinn University ou Tallinn University of Technology. Une fois diplômé(e)s, ces actuel(le)s doctorant(e)s auront la possibilité d'encadrer au niveau doctoral et Kabul University pourra ainsi octroyer ce titre.

C'est dans ce contexte que j'ai le plaisir de co-encadrer (avec Paul Annus) Mohammad Tariq Meeran, ce principalement à distance (les doctorants afghans ne passant que un ou deux mois en Estonie par an). Le titre de travail de son projet de doctorat est Wireless Mesh Networks Impact on Voice over Internet Protocol. Ce projet est résumé dans ce qui suit.
Un réseau maillé sans fil (*wireless mesh networks (WMN)*) est un type de réseau organisé en pair à pair qui offre de nombreux avantages par rapport aux réseaux sans fil centralisés. Ces avantages incluent un déploiement relativement aisé, souple et économique du fait de l'absence de points d'accès, assortis d'une grande tolérance aux erreurs et au pannes du fait du maillage (nombreux chemins disponibles) et du rôle que peuvent prendre les nœuds (récepteur, émetteur, relais).
Ce type de réseau est donc bien adapté à des situations telles que opérations militaires, situations de catastrophes et environnements difficiles, où la mise en place de solutions plus classiques est trop onéreuse ou techniquement trop difficile (temps de déploiement, difficultés d'accès, besoin en mobilité).
Ces réseaux peuvent être standardisés, comme par exemple dans l'amendement IEEE 802.11s.

L'un des inconvénients des réseaux maillés sans fil est que certaines applications, notamment la voix sur IP (VoIP), souffrent de l'impact négatif que ces réseaux ont sur les facteurs de délais, jitter et perte de paquets. Se pose alors la question de l'allocation

(dynamique) de ressources telles que bande passante, mémoires tampons et puissance de calcul afin de respecter les contraintes de qualité de service. Dans les travaux de thèse de Mohammad Tariq Meeran nous cherchons à mieux comprendre comment la topologie (notamment via l'ajout de routeur 'fixes') et la mobilité des nœuds affectent ces facteurs et la qualité de service.
À ce jour, un état de l'art sur les méthodes d'allocation existantes a été réalisé. Celui-ci couvre les méthodes de type placement de passerelle(s), contrôle d'admission, gestion de mobilité en fonction du trafic, protocoles de routage, le standard 802.11e, les codecs, l'agrégation de paquets, les ordonnanceurs à priorité, etc. (voir publication C.5 pour plus de détails).
De plus, nous avons effectué une comparaison d'outils pour la simulation de tels réseaux, à savoir OMNet++, NS-3, EstiNEt, Qualnet et OPNET Modeler. Notre choix final s'est porté sur Qualnet car, entre autres, il permet de simuler les standards 802.11s sur 802.11g et offre une modélisation des terrains et l'analyse *what-if* de scénarios.
Au moment de la rédaction de ce document, la simulation et l'analyse des scénarios suivants est en cours. Dans les trois scénarios, nous évaluons les cas suivants : nœuds maillés seuls et avec routeurs fixes, et ce avec trafic VoIP seul et trafic VoIP plus non-VoIP.

- Pas de mobilité. Dans ce scénario, une quarantaine de nœuds forment le réseau maillé ; ils ne se déplacent pas.
- Mobilité partielle. Ici, 50% des nœuds se déplacent.
- Mobilité complète. Tous les nœuds se déplacent.

Dans tous les cas, il s'agit d'évaluer dans quelle mesure l'ajout de routeurs fixes pourrait améliorer la qualité de service ; celle-ci est mesurée selon le modèle E de la recommandation G.107 de ITU-T.
Pour la phase suivante du doctorat de Mohammad Tariq Meeran, il est prévu d'étudier l'impact de l'intégration du standard 802.11n et de l'amendement 802.11s, éventuellement avec IPv6 sur la qualité de service.

12.2 Solutions matérielles et logicielles pour les systèmes de réseaux embarqués cognitifs

Le projet de recherche *Hardware and Software Solutions for Cognitive Embedded Networks Systems* (dont je suis le chercheur principal) est né de discussions entre Thomas Johann Seebeck Department of Electronics et Department of Computer Engineering, Tallinn University of Technology. Le financement est de type *Baseline Funding* (financement national estonien, via les organismes de recherche dont les universités).
Dans ce projet, nous nous intéressons à la conception et l'implantation de solutions matérielles et logicielles basses consommations pour les systèmes de réseaux de capteurs sans fil. Le projet traite des aspects communication (p. ex allocation en fréquence et en puissance des nœuds), récolte d'énergie (p. ex. exploration haut niveau de la faisabilité), traitement numérique du signal/architectures (p. ex. acquisition comprimée/FPGA), et les méthodes de raisonnement (p. ex. apprentissage automatique, mémoire temporelle et hiérarchique). Les applications visées sont de type e-santé.

Ce projet a aussi donné lieu à des échanges au niveau européen, avec Department of Electronics, Aalborg University (Danemark), Institute of Computer Technology, Vienna University of Technology (Autriche) et Lab-STICC, Université Bretagne Sud (France).

En plus de mon rôle de coordinateur et d'animateur de projet, j'encadre une thèse de doctorat (Tauseef Ahmed) et en co-encadre une seconde (Faisal Ahmed, co-encadré par Paul Annus) ; j'ai aussi co-encadré (à distance) deux étudiants de première année de master d'Aalborg University (Mohammad El-Sayed et Søren Lund, co-encadrés par Peter Koch).

12.2.1 Acquisition comprimée pour la surveillance du rythme cardiaque

L'application considérée dans cette activité est un système multi-utilisateurs pour la surveillance du rythme cardiaque dont l'architecture générale se compose de nœuds (capteurs et module de traitement numérique du signal (TNS)) portés par les utilisateurs et d'une unité centrale (p. ex. PC, tablette, smartphone) qui collecte les signaux et les analyse (ou les fait suivre pour analyse déportée). Celle-ci était au cœur du projet des étudiants de master Mohammad El-Sayed et Soren Lund (*An FPGA-Friendly Compressed Sampling Engine for WSN-Based Heart Rate Monitoring*).

Les deux difficultés principales sont l'énergie disponible pour les nœuds (ceux-ci fonctionnant sur batteries) et la bande passant limitée (p. ex. Bluetooth ou 6LoWPAN). Pour faire face à ces difficultés, nous avons conçu un module de compression des données, explorer son architecture matérielle et simuler son fonctionnement sur une cible FPGA Altera Cyclone III.

Le module de compression est composé de deux parties. La première est chargée d'extraire l'information du rythme cardiaque à partir du signal cardiaque (forme-R) au moyen de l'algorithme classique de Pan et Tompkins. Ceci résulte en une représentation parcimonieuse du signal, qui se prête bien à la deuxième partie, à savoir l'acquisition comprimée (*compressed sensing*). Cette dernière permet de sous-échantillonner le signal en deçà de la fréquence de Nyquist-Shannon, donc de réduire le volume de données à transmettre (permettant de mieux faire face au problème de la bande passant limitée) et aussi de réduire la consommation énergétique (moins de transmissions radio).

Le cœur de l'algorithme d'acquisition comprimée est une multiplication matrice-vecteur, pour laquelle quatre familles d'architectures ont été proposées, à savoir entièrement séquentielle, semi-parallèle, semi-parallèle modifiées et entièrement parallèle. L'espace des solutions a été exploré pour le compromis entre le nombre et le type de ressources (p. ex. utilisation ou pas de multiplicateurs matériels, chaînage ou pas des opérateurs) et le temps d'exécution.

Une fois cette exploration effectuée, les versions semi-parallèle modifiée et entièrement parallèle ont été codées en VHDL, synthétisées et simulées sur Altera Cyclone III. Cette première étape a donné lieu à la publication C.14.
L'étape suivante est de porter ces architectures sur une cible FPGA très basse consommation telle que la famille nano-FGPA IGLOO de Microsemi. Cependant, la réalisation

de cette étape dépendra des ressources humaines disponibles.

12.2.2 Récolte d'énergie et qualité de service dans les BAN

Cette partie du projet est liée aux travaux du doctorant Faisal Ahmed (titre de travail : *Energy Harvesting and QoS Solution for the Reliability of the Communication Channel in the context of WSNs - Application to Wireless Body Area Networks*). Ceux-ci prennent pour point de départ le fait que les *body area networks* (BAN), en particulier ceux sans fil, ont émergé comme des éléments clefs pour la réalisation de solutions de santé à base de TIC. Ceci est régulièrement mis en avant comme par exemple lors de l'événement International Consumer Electronic Show (CES 2016).

Afin de rendre de tels systèmes acceptables par les utilisateurs, il est nécessaire non seulement de développer une nouvelle génération de capteurs (voir par exemple les *bio-stamps* [22]) mais aussi de les rendre les plus autonomes possibles en énergie. Ce second point a engendré un intérêt pour les méthodes de récolte d'énergie (p. ex. solaire, flux d'air, radio-fréquence, thermique et mécanique) via des solutions de nano-générateurs (voir par exemple le générateur thermoélectrique souple de North Carolina State University's Center for Advanced Self-Powered Systems of Integrated Sensors [23]).

La première phase de cette partie du projet vise à fournir un outil pour évaluer la faisabilité de solutions de récolte d'énergie pour les BAN sans fil et pour en faciliter la sélection en fonction des besoins de l'application et de l'architecture sous-jacente. Afin d'éviter l'implantation réelle ou même l'utilisation de simulateurs (assez précis mais généralement plutôt lents), nous avons proposé une méthode d'exploration haut-niveau mettant en œuvre des modèles analytiques gros grains de l'application et des nœuds.

Au moment d'écrire ces lignes, l'avancement de cette partie du projet peut être résumé ainsi :
Il ressort de l'état de l'art que d'une part les simulateurs existant n'offrent pas beaucoup de modèles pour la récolte d'énergie (ou alors ils ont trop linéaires) et surtout ne permettent pas une exploration de haut niveau (rapide mais pas très précise). En conséquence, nous avons proposé un outil d'exploration au niveau système incluant des modèles gros grains de diverses méthodes de récolte d'énergie, de consommation dans les nœuds et de batteries/super-condensateurs (et la possibilité d'ajouter d'autres modèles). Nous avons aussi inclus la possibilité de modéliser l'utilisation hybride de plusieurs sources d'énergie. Enfin, nous avons aussi intégré des modèles de prédiction d'énergie afin de pouvoir explorer la chaîne complète prédiction d'énergie → récolte d'énergie → consommation d'énergie → estimation de la durée de vie des nœuds et à venir impact sur la performance et qualité de service. Cette première phase a donnée lieu aux publications J.2, C.1, C.2 et C.6.

La deuxième phase de cette partie du projet sera consacrée à la qualité de service dans les BAN sans fils lorsque ceux-ci sont alimentés via des systèmes de récolte d'énergie. Il s'agira notamment d'évaluer l'impact que la nature fluctuante de telles sources d'énergie a sur les systèmes dit *transient computing*, c.a.d des nœuds n'ayant pas de batteries ou de super-condensateurs. Comme ceux-ci peuvent rester sans énergie pendant différentes périodes de temps (et pas forcément prédictibles), on commence à fait appel à des éléments de calculs non-volatiles (p. ex. microcontrôleurs FRAM), ce qui permet de mémoriser

localement les données et l'état du microcontrôleur lorsque le niveau d'énergie devient dangereusement bas et de reprendre le traitement lorsque celui-ci remonte suffisamment. Il s'agira p. ex. d'explorer des méthodes statistiques pouvant, pour certaines applications suffisamment régulières, compenser le manque de données de certains capteurs et ainsi garantir une qualité de service minimum.

12.2.3 Gestion des ressources dans les réseaux de capteurs cognitifs

Cette autre partie du projet est liée à la thèse de doctorat de Tauseef Ahmed (titre de travail : *Advanced Radio Resource Management in Wireless Networks with Emphasis on Cognitive Radio Networks and Wireless Sensor Networks*). L'objectif principal est de transposer les méthodes d'allocation et de gestion des ressources (p. ex. exploitation des portions de spectre radio sous-utilisées) issues de la radio logicielle et cognitive aux réseaux de capteurs. La combinaison de ces deux domaines à donner naissance au terme *cognitive radio sensor network (CRSN)*. Le développement de cette approche est notamment motivée par les futurs besoins du volet internet des objets/réseaux de capteurs sous le parapluie 5G, à savoir très grand nombre d'objets, efficience spectrale, faible latence et faible consommation.

Dans la première phase de cette partie du projet nous nous intéressons plus particulièrement à la gestion du spectre radio et à son partage (notamment évitement d'interférences) entre utilisateurs primaires (ceux à qui la portion de spectre est 'officiellement' allouée) et les utilisateurs secondaires (ceux qui veulent y accéder de manière opportune lorsqu'elle est disponible).
Cette situation est d'autant plus compliquée lorsqu'il y a plusieurs réseaux secondaires, potentiellement hétérogènes entre eux. D'un côté il faut donc déployer des méthodes cognitives relativement complexes et d'un autre réussir à les implanter sur les ressources limitées de nœuds et routeurs des réseaux de capteurs sans fil.

Au moment de rédiger ce document, nous avons proposé, simulé et comparé plusieurs méthodes d'allocation de spectre basées sur l'apprentissage par renforcement pour différents scénarios (réseaux hétérogènes, réseaux non-coordonnés, présence de femtocellules, etc.). L'idée centrale de ces variantes est d'optimiser l'allocation des canaux de communication aux différents utilisateurs de manière à maintenir en priorité la qualité de service des utilisateurs primaires et d'éviter les interférences. Pour ce faire, les algorithmes cherchent à maximiser, de manière dynamique, une fonction de récompense en fonction des conditions de trafic sur le réseau et de la charge de travail des nœuds. De plus, les résultats d'allocation sont mémorisés (appris) et exploités lors des allocations suivantes.
De plus, nous avons aussi inclus une méthode d'allocation de la puissance pour les canaux assignés précédemment. Celle-ci est formulée comme un problème d'optimisation convexe qui tient compte à la fois des canaux alloués, du nombre d'utilisateurs et des interférences. Ce problème est résolu par l'utilisation d'un l'algorithme dit *water filling*.
Les résultats de cette première phase ont été publiés dans C.3, C.7, C.8 et C.9.

Pour la phase suivante, nous prévoyons d'adapter et de porter ces méthodes dans un système BAN sans fil. Il s'agira entre autres d'évaluer l'impact du passage du monde de la simulation à celui de l'embarqué (p. ex. passage d'une représentation virgule flottante

double-précision sous Matlab à virgule fixe 8 ou 16 bits sur microcontrôleur) tout en assurant un niveau de performances suffisant.

12.3 Cancer du poumon et images tomodensitométriques

Après une première incursion dans le domaine du génie biomédical (cardiographie et respirographie à impédance (co-encadrement de la seconde phase de la thèse de doctorat de Yar Mughal Mohammad intitulée A Parametric Framework for Modelling of Bioelectrical Signals)), je suis maintenant impliqué dans un projet d'imagerie médicale qui vise à améliorer la qualité de la détection des cancers du poumon à partir d'images tomodensitométriques (co-encadrement de la thèse de doctorat de Anindya Gupta dont le titre de travail est Automatic Detection of Lung Nodules in Computed Tomography (CT) Imaging).

À l'échelle mondiale, le cancer du poumon est l'une des principales causes de décès et le principal type de cancer chez les hommes. Afin de pouvoir traiter avec succès ceux qui en sont atteints, il est essentiel de le détecter le plus tôt possible. L'absence de symptômes en tout début de maladie rend la détection difficile. La présence de nodules (de petites lésions) pulmonaires est un indicateur, environ 40% des nodules sont malins (signe de cancer). Même s'ils sont difficiles à détecter et identifier, de nombreux progrès ont été réalisés ces 20 dernières années, depuis la radiographie du thorax (rayons-X) associée à la cytologie des expectorations (analyse de sécrétions), l'arrivée de la tomodensitométrie (*CT-Scan*), et maintenant les scanners CT spiraux multi-tranches qui peuvent reconstruire des images correspondant à une tranche d'épaisseur inférieure au mm (et peuvent donc détecter des nodules de moins d'un mm).
On pourrait qualifier de compétition les différents travaux de recherche menés à travers le monde qui visent à obtenir la meilleure qualité de détection et de classification de ces nodules (notamment réduire le taux de faux négatifs), tout en automatisant le processus le plus possible (évolution *Computer Aided Detection (CADe)*[1] $\rightarrow$ *Computer Aided Diagnostic (CADx)* $\rightarrow$ *Unsupervised Computer Aided Diagnostic (UCAD)*).

L'objectif de nos travaux est de développer un système UCAD qui permet de détecter différents types de nodules dans des régions segmentées des poumons et de les classifier en fonction de leurs tailles, formes et types, et si possible d'estimer leurs croissances.
Pour ce faire, nous avons proposé une approche qui segmente les éléments candidats (analyse d'image), élimine les non-nodules (apprentissage automatique supervisé) et effectue une classification phénotypique des nodules restant (apprentissage automatique supervisé dans un premier temps, apprentissage profond dans un second temps).

Pour la segmentation nous utilisons une combinaison de techniques de traitement de l'image, à savoir : masques, algorithme de remplissage par diffusion et opération morphologique *close*. La méthode de détection repose sur un procédé multi-seuil combiné avec des techniques d'extraction de propriétés. Ces méthodes ont été évaluées sur la dernière version de la base de données LIDCIDRI et contrairement aux autres travaux, nous avons considéré tous les cas présents. Nos résultats montrent que nous pouvons détecter 3058 nodules (y inclus solides, non solides, partiellement solides, juxta-vasculaires,

[1] À ne pas confondre avec *Computer Aided Design (CAD)*

juxta-pleural et bien circoncis) avec une sensitivité de 85% pour l'ensemble des cas (1010 scans CT avec 3615 nodules). Sur un ensemble plus petit (60 scans CT avec 315 nodules) nous pouvons détecter 301 nodules avec une sensitivité de 95.5%.
Cette première phase a donnée lieu aux publications C.10, C.11, C.15 et C.16.

La seconde phase est consacrée au développement d'un classifieur multi-niveaux (réseaux de neurones) visant à réduire encore plus le taux de faux positifs. Les résultats préliminaires montrent une sensitivité de 92.8% avec un taux de faux positifs de 0.3% par scan.

Enfin dans la troisième phase nous proposerons un système UCAD pour lequel nous utilisons des méthodes d'apprentissage profond (réseau neuronal convolutif 3D suivi d'une étape d'estimation de la croissance des nodules à partir de méthodes issues du domaine de la radiomique).

12.4 ESS-EtherCAT

Pour ce projet nous avons été approchés par le responsable et les ingénieurs de la division Integrated Control System (ICS) du centre source européenne de spallation[2] (*European Spallation Source (ESS)*)[3] qui est l'un des projets de type consortium pour une infrastructure de recherche européenne *European Research Infrastructure Consortium (ERIC)*. Le budget alloué pour ESS est de 1843 MEUR.

Il s'agit de conduire des recherches sur la matière à partir de techniques de dispersion des neutrons. L'installation doit permettre d'accélérer des protons et de les projeter sur une cible en tungstène via un un accélérateur linéaire, constituant ainsi une source de neutrons thermiques et froids. Ces neutrons sont ensuite guidés vers des stations d'expérimentations où des travaux de recherches couvrant une large gamme d'application (p. ex. énergie, télécommunications, technologies de l'information, biotechnologies, etc.) seront effectués.

L'installation est en cours de construction à Lund dans le sud de la Suède, le centre de traitement des données étant quant à lui installé dans la région de Copenhague au Danemark. Les prévisions font état d'une ouverture progressive à partir de 2019, d'une phase opérationnelle entre 2026 et 2066, et d'une phase de désaffection entre 2067 et 2071.

Afin de construire l'installation et les équipements, le consortium ESS a recours aux contributions en nature (*in-kind contributions*) de nombreuses entreprises, centres de recherche et universités à travers toute l'Europe.
C'est dans ce contexte que nous avons été sélectionnés pour effectuer le sous-projet ESS-EtherCAT.

EtherCAT (un réseau Ethernet temps-réel) est une des technologies utilisées dans le système de commande de ESS. Il a pour objectif de traiter des ordres de commandes pour p. ex. les systèmes de mouvements multiaxes, l'acquisition de signaux du faisceau et les

[2] Spallation nucléaire : de l'anglais *to spall*, produire des éclats. Type de réaction nucléaire. Un noyau atomique est frappé par une particule (p. ex. neutron) ou une onde électromagnétique fortement énergétique.

[3] https://europeanspallationsource.se/, à ne pas confondre avec ESS - European Social Survey (un autre ERIC).

actions de commande de celui-ci. Notre contribution consistera à concevoir, développer et tester un module matériel esclave EtherCAT, un module matériel à base de FPGA et les firmwares et logiciels associés. Ceci sera effectué en coopération avec une ou plusieurs entreprises sélectionnées sur la base d'un appel d'offre.

Le projet ne fait que commencer et les premiers résultats (qui resteront sans doute confidentiels pour une longue période) sont attendus pour début 2017.

12.5 Chaire européenne d'électronique cognitive

Le financement de ce projet est de type chaire de l'espace européen de la recherche (*European Research Area (ERA) Chair*) sous le volet WIDESPREAD de H2020.

L'objectif de ces chaires est de réduire l'écart de performance en recherche entre les pays européens : les pays éligibles sont ceux ayant rejoint l'UE à partir de 2004 (et donc l'Estonie) plus le Portugal et le Luxembourg. Les chaires sont attribuées à des chercheurs qui ont la capacité d'améliorer la qualité de la recherche, d'attirer à leur tour d'autres chercheurs de haut niveau et d'obtenir des financements de recherche européens.

En 2014 j'ai activement contribué à la rédaction de la demande de financement pour un tel projet (notamment pour les parties scientifique et implantation). Nous avons reçu une réponse favorable à l'automne 2015 et le projet a officiellement commencé en décembre 2015.
L'obtention d'un projet de ce type était primordiale pour le département. Vu le budget alloué (2,5 MEUR sur 4 ans), ce projet est pour moi une étape importante dans ma carrière. Je remercie bien entendu les collègues impliquées dans le montage du projet, sans qui ce succès n'aurait pas été possible.
Le titre de notre projet est COEL (*COgnitive ELecronics*). Les principaux axes de recherche qui seront développés concernent :

- Les réseaux de capteurs sans fil ;
- Le traitement numérique du signal dans les réseaux de capteurs sans fil, dont acquisition compressée/calcul approximatif/calcul transitoire ;
- Les communications en champs proche ;
- La radio logicielle et cognitive ;
- L'internet de l'électronique ;
- La récolte d'énergie.

Nous allons également lancer des activités pour améliorer la qualité des programmes de master et de doctorat et pour augmenter notre participation dans des projets de recherche européens.

Au moment d'écrire ces lignes, le recrutement de la personne qui occupera la chaire a été finalisé ; cette personne a pris son poste début septembre 2016. Elle prendra part aux projets en cours (p. ex. ceux mentionnés ci-dessus) et en parallèle contribuera de manière

significative au montage et à la rédaction de demande de financement de nouveaux projets de concert avec l'équipe existante (dont moi-même).

Ceci nous amène au chapitre suivant de ce document, à savoir mes perspectives de recherche à court terme et plus long terme.

Chapitre 13

Perspectives de recherche

13.1 À court terme

Mes perspectives de recherche à court terme consistent à finaliser les projets en cours ou qui commencent (voir le chapitre précédent). Il est notamment nécessaire de proposer des projets de recherche (et d'obtenir leurs financements) dans le contexte de la chaire ERA-COEL. Les lignes qui suivent résument quelques idées sur lesquelles un tel projet pourrait être construit.

Diverses barrières technologiques (p. ex. la fin de la loi de Dennard, difficultés avec la loi de Moore) appellent au développement de nouvelles applications, algorithmes et architectures afin de pouvoir tenir les promesses en performances et consommation énergétiques des différents éléments de la prochaine génération de la chaine d'information (des données issues des capteurs à la connaissance générée par et stockée sur les serveurs). Par exemple, le document de vision du partenariat publique privée 5G (qui couvre toute cette chaîne) est de diviser la consommation de manière drastique (1/10 X) tout en améliorant d'autres facteurs tels que la latence de bout en bout (1/5 X) ou la densité d'objets connectés (1000 X). [24]. En parallèle, d'autres documents tentent d'établir les grandes lignes de l'évolution de la conception matérielle/logicielle [25], des systèmes embarqués [26] et de la convergence entre ceux-ci et le calcul haute performance [27] [28].

Parmi les nouvelles approches avancées dans la littérature, il a été récemment proposé d'utiliser le calcul transitoire (*transient computing*) pour faire face à la nature intermittente de l'énergie dans les réseaux de capteurs sans fil dont les nœuds sont alimentés uniquement par récolte d'énergie (sans batterie ni super-condensateurs). Ce genre d'approche est rendue possible notamment grâce aux technologies mémoires non-volatiles FRAM et MRAM. Des exemples récents de méthodes logicielles qui exploitent cette technologie pour implanter des méthodes de sauvegarde et de reprise des calculs qui sont soit répétitives, soit ad-hoc (c.a.d initiées en fonction des variations de la tension d'alimentation) inclus [29], [30] et [31]. D'autres approches comme [32] utilisent des méthodes de gestion dynamique de la tension et de la fréquence pour traiter ce problème de manière plus proactive et fine, c.a.d ralentir les calculs (et donc diminuer la puissance requise) lorsque la puissance disponible baisse (mais reste au-dessus du seuil), permettant ainsi un mode de fonctionnement dégradé mais interrompu moins souvent.
D'autres travaux [33], [34], [35] proposent divers mécanismes au niveau matériel dans les microprocesseurs afin de réduire le surcoût logiciel des méthodes précédentes.

En parallèle, diverses approches de calcul approximatif [36] ont été mise en place, non pas pour faire face au matériel fautif (p. ex. correction d'erreurs), mais pour réduire la complexité algorithmique et architecturale (afin de réduire la consommation) au prix de résultats imparfait [37], [38].
Un exemple de méthode de calcul approximatif appliquée aux réseaux de capteurs sans fil est présentée dans [39]. Il s'agit de compresser des signaux biomédicaux de manière approximative (possible car pour certains de ces signaux il existe une certaine tolérance). Appliquée à un système de surveillance d'électrocardiogramme, la méthode permet une réduction significative du volume de données à transmettre et donc une activité radio réduite et ainsi une réduction de la consommation.
Une approche *lean sensing* est présentée dans [40]. Ici il ne s'agit pas de compresser le signal, mais d'accepter (pour certaines applications) que des données sont manquantes (p. ex. capteurs éteints par manque dŠénergie), ce qui comme pour l'exemple précédent conduit à une réduction de la consommation. La différence ici est que des méthodes de reconstruction (corrélations spatiale (avec les données des nœuds voisins) et temporelle (historique des données)) sont utilisées du côté serveur.

Il me semble opportun d'aller plus loin sur ces pistes, notamment parce que les méthodes proposées sont limitées à un ou deux niveaux d'abstraction et n'ont pas encore été exploitées de manière conjointe.
Il s'agirait donc d'identifier les types de modèles (niveau d'abstractions, précision, justesse) et de proposer des techniques permettant de combiner ces deux méthodes, dans un premier temps niveau par niveau et ensuite de manière *cross-layer*.
Il s'agirait aussi de proposer une méthode d'exploration de l'espace de conception (interaction application-algorithme-architecture, notamment la nature dynamique d'une approche conjointe de calcul transitoire et approximatif) pour comparer les solutions faisant appel aux points précédents.
Pour ce faire, il faudrait entre autres caractériser la tolérance aux erreurs et aux ralentissements et interruptions des applications, p. ex. en fonction de la précision, justesse, QoS souhaités ou requis.
À cela, j'envisage aussi d'explorer la possibilité et l'exploitation du partage coopératif et opportuniste d'énergie entre les nœuds (p. ex. transfert RF) dans un même réseau ou entre réseaux voisins [41].

Une autre question qui me semble importante est la distribution des calculs dans les réseaux, à savoir sur les nœuds ou très proches d'eux (*edge computing*), les équipements intermédiaires (p. ex. passerelles et routeurs) (*fog computing*) et les serveurs (*cloud computing*) afin de trouver des compromis entre puissance de calcul, consommation énergétique, bande passante et latence.

Enfin, et pour faire le lien entre la question précédente et mes perspectives à plus long terme, il me semble opportun de suivre les développements visant à implanter des approches d'apprentissage automatique directement sur les unités de calculs dans les systèmes embarqués [42].

13.2 À plus long terme

Sans me lancer dans des prédictions futuristes de type singularité technologique, je liste ici quelques-unes des grandes tendances dans lesquelles je pourrait éventuellement trouver, à plus long terme, des problèmes à résoudre et/ou des pistes de solutions à exploiter pour résoudre ces problèmes.

Sur le plus long terme, la recherche au sein des universités dépendra très largement (et sans doute uniquement) de l'obtention de financements externes et donc des opportunités de coopération, notamment avec les entreprises[1], ce qui explique aussi qu'à ce stade je ne rentre pas davantage dans les détails.

- Au niveau application : les grands thèmes qui me motivent sont la santé et l'environnement. Pour le premier, les applications que j'imagine viser incluent la prévention et la détection des maladies ainsi que l'aide à la personne via les TIC. Pour le second, les applications qui m'intéressent sont la réduction de la consommation énergétique au sens large (optimisation des activité humaines via les TIC) mais aussi au sens technologique (obtenir un bilan énergétique positif entre les gains obtenus et la pollution générée par les TIC elles-mêmes). Le point commun entres ces thèmes (et les autres) est la tendance marquée pour le *big data*, allant des capteurs à la connaissance[2].

- Au niveau algorithme : afin de pouvoir réaliser les applications ci-dessus, il apparaît nécessaire de développer de nouvelles méthodes et algorithmes pour traiter et exploiter les données. La tendance à ce niveau tourne autour des méthodes d'apprentissages automatiques, entre autres apprentissage profond, y inclus en version embarquée (rendue possible grâce à aux nouvelles technologies matérielles ci-dessous), potentiellement combiné avec des approches de type calcul direct sur les données compressées.

- Au niveau architecture : mise en œuvre et exploitation de nouvelles technologies pour faire face aux demandes de calculs, transmissions et stockages liées au *big data* tout en réduisant, ou du moins en maîtrisant la consommation énergétique. Ces technologies incluent : les SoC hétérogènes à technologies FinFETs, FDSOI, graphènes et suivantes ; la montée des interconnections photoniques à tous les niveaux (de l'infrastructure réseau jusqu'aux liens de communication intra et inter-puces) ; les mémoires non-volatiles et le calcul au sein des mémoires *in-memory compute*.

[1] Dont la participation est de plus en plus essentielle dans l'obtention de financements de type H2020 ; voir aussi le poids grandissant de groupes tels que Google

[2] Et de manière sans doute utopique, à la sagesse.

Chapitre 14

Réflexions sur l'enseignement

Pour clore ce document, je souhaite partager de manière succincte quelques réflexions sur la pédagogie d'apprentissage par problèmes ainsi que sur les liens entre enseignement et recherche.

Les deux modèles que je connais sont A) le traditionnel CM-TD-TP (qui dans notre domaine est généralement accompagné par des 'petits'[1] projets et/ou stages) et B) la pédagogie d'apprentissage par problèmes. J'ai une expérience des deux modèles à la fois comme étudiant (principalement du premier à UBS et Université Rennes I et un peu du second lors de deux séjours Erasmus à AAU) et comme enseignant (principalement du second à AAU et un peu du premier à UBS et TUT).

À AAU, la pédagogie par problèmes est centrée autour de projets par groupes. La complexité, durée et poids du projet augmentent au fur et à mesure des semestres (1 projet par semestre, partant de 10 crédits ECTS au premier semestre de L1 jusqu'à 20 pour le dernier semestre de L3 ; 20 pour les trois premiers semestres de master et 30 pour le dernier). À contrario, le nombre d'étudiants diminue pour atteindre un ou deux étudiants par groupe pour les deux derniers projets de master. À noter aussi que les cours dits 'cours de projets' n'ont pas leurs propres examens et sont évalués lors de la soutenance de projet (il faut donc s'assurer que les projets fassent un minimum appel aux thématiques abordés dans ces cours ; les soutenances peuvent être longues).

À mon sens, les principaux avantages de cette pratique de la pédagogie par problèmes et par projet sont le développement des capacités d'apprentissage des étudiants c.a.d apprendre à apprendre, non pas en vue d'ingurgiter des connaissances de manière efficace et de les ressortir en examen, mais en vue de savoir résoudre des problèmes (seul ou en groupe) en appliquant les préceptes de la méthode scientifique et/ou d'ingénierie, c.a.d d'analyser et/ou définir le problème, le décomposer en sous-problèmes, chercher, trouver et analyser des informations de manière approfondie, évaluer et identifier les limites des méthodes, techniques et outils existants, évaluer, comparer et critiquer son travail, communiquer et argumenter de manière détaillée et rigoureuse, travailler de manière indépendante, se comporter en professionnel responsable, etc.

Bien que ces 'savoir, savoir faire, savoir être' (ou *knowledge, skills, and competencies* comme définie dans le cadre européen des certifications ou *knowledge, skills, and abilities*

[1] petits comparés à ceux pratiqués à AAU

aux États-Unis) peuvent aussi être acquis et développer via le modèle traditionnel, la pédagogie par problèmes et par projet permet à la fois d'approfondir les sujets étudiés plutôt que de survoler un grand nombre de sujets (syndrome 'Introduction à' jusqu'en master) et de rendre les étudiants entièrement en charge de leur succès. Ils peuvent ainsi à la fois démontrer de (très) bonnes connaissances et expériences pratiques sur quelques sujets et en même temps montrer qu'ils sont capables de transposer les méthodes d'analyse et de résolution à toute une gamme de problèmes même si ils n'ont pas étudier ces sujets lors de leurs études.

Parmi les inconvénients pour les étudiants, on peut citer la parfois trop grande spécialisation (manque d'ouverture), la difficulté à se responsabiliser, la difficulté à travailler sur des projets conséquents et longs, et pour certains (typiquement étudiants internationaux qui arrivent en master et n'ont jamais connus ce modèle) l'impossibilité de compter sur le seul apprentissage par cœur et la nécessité de rédiger des rapports et thèses argumentés de manière détaillé.

Pour les institutions et les enseignants, les inconvénients incluent l'infrastructure nécessaire (en sciences dures à AAU chaque groupe dispose de sa pièce de travail accessible 24H/24), la nécessité de former les enseignants-chercheurs à ce modèle, adapter le modèle de comptage d'heures d'enseignement (réunions régulières avec les étudiants, lecture et correction des rapports et thèses, examens (à AAU, appel à des examinateurs externes qu'ils faut trouver et identifier)), suivi de qualité (besoin de retours régulier de la part des étudiants quant à l'adéquation cours/projets).

Malgré ces inconvénients, je suis partisan de la pédagogie par problèmes et par projet (telle que pratiquée à AAU) ; aussi je vise à en insuffler au moins quelques éléments dans mes enseignements à TUT. J'ai commencé à le faire dans les TP et 'petits' projets que j'encadre à TUT, et vais le faire pour les thèses de master dans le cadre du projet COEL discuté plus haut. Ceci s'inscrit également dans la réorganisation en cours (réduction du nombre de sujets mais nombre de crédits par sujet en hausse), ce qui permet d'approfondir non seulement les contenus mais aussi d'accorder plus d'importance à la méthode scientifique et/ou d'ingénierie et à la pratique. Une approche complémentaire que je souhaite explorer dans ce contexte est la classe inversée (auto-étude plutôt que CM, le temps en classe étant réservé aux TD/TP/projets).

Quel que soit le modèle et les outils (papier-crayon, multimédia, MooC, etc.), je pense qu'il est important d'expliquer et de discuter avec les étudiants, notamment de ce qui est attendu d'eux et pourquoi (pas seulement pour faire plaisir aux enseignants-chercheurs, mais pour augmenter leurs chances d'obtenir un emploi en phase avec leurs attentes ou de convaincre les investisseurs pour créer son entreprise).

Il me semble aussi important de souligner les liens qui peuvent (et doivent) exister entre enseignement et recherche. Être un chercheur actif permet non seulement d'appuyer ses enseignements sur la méthode scientifique/d'ingénierie mais permet aussi d'être au courant des dernières avancées scientifiques et techniques et de leurs applications actuelles et futures, ce qui permet à la fois de motiver son cours (pourquoi s'intéresser à un sujet) et de fournir aux étudiants des savoirs et savoir-faire à la pointe.

Inversement, enseigner peut avoir un impact bénéfique sur la recherche. En effet, pour pouvoir créer de nouvelles connaissance et avancées scientifiques il est nécessaire de maîtriser les savoirs fondamentaux (p. ex. niveau licence) et plus avancés (p. ex. niveau master) ; il est aussi de plus en plus nécessaire de communiquer les résultats de projets financés au niveaux nationaux et européens, ce qui peut se faire via l'enseignement. De plus, les questions posées et les commentaires faits par les étudiants peuvent aussi pousser les chercheurs à se remettre en question et explorer des sujets qu'ils auraient peut-être ignorés sans ces interactions.

Enfin, il est souhaitable de proposer des sujets de projets (notamment au niveau master) en lien direct avec les travaux de recherche d'un laboratoire, ce qui permet non seulement de développer les qualités scientifiques des étudiants impliqués tout en bénéficiant de leurs compétences, mais aussi de promouvoir la recherche auprès de leurs cercles de camarades, amis et famille.

Bibliographie

[1] D. Sciuto, F. Salice, L. Pomante, and W. Fornaciari, "Metrics for design space exploration of heterogeneous multiprocessor embedded systems," in *Proceedings of the Tenth International Symposium on Hardware/Software Codesign*, ser. CODES '02, Estes Park, Colorado: ACM, 2002, pp. 55–60.

[2] B. W. Boehm, *Software engineering economics*, 1st. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1981.

[3] B. W. Boehm, Clark, Horowitz, Brown, Reifer, Chulani, R. Madachy, and B. Steece, *Software cost estimation with cocomo ii with cdrom*, 1st. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2000.

[4] T. J. McCabe, "A complexity measure," in *Proceedings of the 2Nd International Conference on Software Engineering*, ser. ICSE '76, San Francisco, California, USA: IEEE Computer Society Press, 1976, pp. 407–.

[5] G. Jay, J. E. Hale, R. K. Smith, D. P. Hale, N. A. Kraft, and C. Ward, "Cyclomatic complexity and lines of code: Empirical evidence of a stable linear relationship.," *JSEA*, vol. 2, no. 3, pp. 137–143, 2009.

[6] R. Abildgren, "Implementation effort and parallelism - metrics for guiding hardware/software partitioning in embedded system design," PhD thesis, Department of Electronic Systems, Aalborg University, 2010.

[7] T. J. Todman, G. A. Constantinides, S. J. E. Wilton, O. Mencer, W. Luk, and P. Y. K. Cheung, "Reconfigurable computing: Architectures and design methods," *IEE Proceedings - Computers and Digital Techniques*, vol. 152, no. 2, pp. 193–207, Mar. 2005.

[8] M. J. Wirthlin and B. L. Hutchings, "Improving functional density using run-time circuit reconfiguration [fpgas]," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 6, no. 2, pp. 247–256, Jun. 1998.

[9] C. Bobda, "Synthesis of dataflow graphs for reconfigurable systems using temporal partitioning and temporal placement," ISBN 3-935433-37-9, Dissertation, Universität Paderborn, Heinz Nixdorf Institut, Entwurf Paralleler Systeme, Jul. 2003.

[10] T. Becker, W. Luk, and P. Y. K. Cheung, "Energy-aware optimisation for run-time reconfiguration," in *Field-Programmable Custom Computing Machines (FCCM), 2010 18th IEEE Annual International Symposium on*, May 2010, pp. 55–62.

[11] K. M. G. Purna and D. Bhatia, "Temporal partitioning and scheduling data flow graphs for reconfigurable computers," *IEEE Transactions on Computers*, vol. 48, no. 6, pp. 579–590, 1999.

[12] G. C. Sih and E. A. Lee, "A compile-time scheduling heuristic for interconnection-constrained heterogeneous processor architectures.," *IEEE Trans. Parallel Distrib. Syst.*, vol. 4, no. 2, pp. 175–187, 1993.

[13] K. S. Chatha and R. Vemuri, "An iterative algorithm for hardware-software partitioning, hardware design space exploration and scheduling," *Design Automation for Embedded Systems*, vol. 5, no. 3, pp. 281–293, 2000.

[14] A. Popp, "Mapping framework for heterogeneous reconfigurable architectures: Combining temporal partitioning and multiprocessor scheduling," PhD thesis, Department of Electronic Systems, Aalborg University, 2010.

[15] C. T. Chow, L. S. M. Tsui, P. H. W. Leong, W. Luk, and S. J. E. Wilton, "Dynamic voltage scaling for commercial fpgas," in *Proceedings. 2005 IEEE International Conference on Field-Programmable Technology, 2005.*, Dec. 2005, pp. 173–180.

[16] J. M. Levine, E. Stott, and P. Y. Cheung, "Dynamic voltage & frequency scaling with online slack measurement," in *Proceedings of the 2014 ACM/SIGDA International Symposium on Field-programmable Gate Arrays*, ser. FPGA '14, Monterey, California, USA: ACM, 2014, pp. 65–74.

[17] J. Sloan, D. Kesler, R. Kumar, and A. Rahimi, "A numerical optimization-based methodology for application robustification: Transforming applications for error tolerance," in *2010 IEEE/IFIP International Conference on Dependable Systems Networks (DSN)*, Jun. 2010, pp. 161–170.

[18] A. A. Biklykke, "Modeling and predicting the behavior of computers operating without guard-bands — an experimental approach based on voltage-scaled fpgas," PhD thesis, Department of Electronic Systems, Aalborg University, 2015.

[19] Eurocontrol, *Mode-s technical overview*, http://www.eurocontrol.int/articles/mode-s-technical-overview, Visited March 2014, 2014.

[20] Kosmotras, *21 november 2013. dnepr cluster mission 2013*, http://www.kosmotras.ru/en/launch14/, Accessed: 2013-12-10.

[21] Openflights, *Openflights route database [r760] jan 2014 update*, http://sourceforge.net/p/openflights/code/HEAD/tree/openflights/data/routes.dat, Visited March 2014.

[22] T. S. Perry, *A temporary tattoo that senses through your skin*, http://spectrum.ieee.org/techtalk/biomedical/devices/power-harvesting-sensor-patch-uses-your-body-as-a-battery, Visited January 2016, 2015.

[23] E. Ackerman, *Power harvesting sensor patch uses your body as a battery*, http://spectrum.ieee.org/techtalk/biomedical/devices/power-harvesting-sensor-patch-uses-your-body-as-a-battery, Visited January 2016, 2016.

[24] 5. PPP, *5g vision - the 5g infrastructure public private partnership: The next generation of communication networks and services*, 2015.

[25] J. Teich, "Hardware/software codesign: The past, the present, and predicting the future," *Proceedings of the IEEE*, vol. 100, no. Special Centennial Issue, pp. 1411–1430, 2012.

[26] M. Duranton, K. De Bosschere, A. Cohen, J. Maebe, and H. Munk, "HiPEAC Vision 2015," Tech. Rep., 2015.

[27] R. Barrett, S. Dosanjh, M. Heroux, X. S. Hu, S. Parker, and J. Shalf, "Toward codesign in high performance computing systems," in *Computer-Aided Design (ICCAD), 2012 IEEE/ACM International Conference on*, 2012, pp. 443–449.

[28] C. Chan, D. Unat, M. Lijewski, W. Zhang, J. Bell, and J. Shalf, "Software design space exploration for exascale combustion co-design," in *Supercomputing*, ser. Lecture Notes in Computer Science, J. Kunkel, T. Ludwig, and H. Meuer, Eds., vol. 7905, Springer Berlin Heidelberg, 2013, pp. 196–212.

[29] B. Lucia and B. Ransford, "A Simpler, Safer Programming and Execution Model for Intermittent Systems," *ACM SIGPLAN Notices*, vol. 50, no. 6, pp. 575–585, Jun. 2015.

[30] D. Balsamo, A. S. Weddell, G. V. Merrett, B. M. Al-Hashimi, D. Brunelli, and L. Benini, "Hibernus: Sustaining Computation During Intermittent Supply for Energy-Harvesting Systems," *IEEE Embedded Systems Letters*, vol. 7, no. 1, pp. 15–18, Mar. 2015.

[31] H. Jayakumar, A. Raha, W. S. Lee, and V. Raghunathan, "QuickRecall: A HW/SW Approach for Computing across Power Cycles in Transiently Powered Computers," *ACM Journal on Emerging Technologies in Computing Systems*, vol. 12, no. 1, pp. 1–19, Aug. 2015.

[32] D. Balsamo, A. Das, A. Weddell, D. Brunelli, B. Al-Hashimi, G. Merrett, and L. Benini, "Graceful Performance Modulation for Power-Neutral Transient Computing Systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. PP, no. 99, pp. 1–1, 2016.

[33] N. Sakimura, Y. Tsuji, R. Nebashi, H. Honjo, A. Morioka, K. Ishihara, K. Kinoshita, S. Fukami, S. Miura, N. Kasai, T. Endoh, H. Ohno, T. Hanyu, and T. Sugibayashi, "10.5 a 90nm 20mhz fully nonvolatile microcontroller for standby-power-critical applications," in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, Feb. 2014, pp. 184–185.

[34] N. Onizawa, A. Mochizuki, A. Tamakoshi, and T. Hanyu, "A sudden power-outage resilient nonvolatile microprocessor for immediate system recovery," in *Proceedings of the 2015 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH15)*, Jul. 2015, pp. 39–44.

[35] K. Ma, Y. Zheng, S. Li, K. Swaminathan, X. Li, Y. Liu, J. Sampson, Y. Xie, and V. Narayanan, "Architecture Exploration for Ambient Energy Harvesting Nonvolatile Processors," in *2015 IEEE 21st International Symposium on High Performance Computer Architecture (HPCA)*, IEEE, Feb. 2015, pp. 526–537.

[36] K. Palem and A. Lingamneni, "Ten years of building broken chips: The physics and engineering of inexact computing," *ACM Trans. Embed. Comput. Syst.*, vol. 12, no. 2s, 87:1–87:23, May 2013.

[37] Q. Xu, T. Mytkowicz, and N. S. Kim, "Approximate computing: A survey," *IEEE Design Test*, vol. 33, no. 1, pp. 8–22, Feb. 2016.

[38] S. Mittal, "A survey of techniques for approximate computing," *ACM Comput. Surv.*, vol. 48, no. 4, 62:1–62:33, Mar. 2016.

[39] F. Samie, L. Bauer, and J. Henkel, "An approximate compressor for wearable biomedical healthcare monitoring systems," in *Proceedings of the 10th International Conference on Hardware/Software Codesign and System Synthesis*, ser. CODES '15, Amsterdam, The Netherlands: IEEE Press, 2015, pp. 133–142.

[40] B. Martinez, X. Vilajosana, I. Vilajosana, and M. Dohler, "Lean sensing: Exploiting contextual information for most energy-efficient sensing," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 5, pp. 1156–1165, Oct. 2015.

[41] X. Lu, P. Wang, D. Niyato, and Z. Han, "Resource allocation in wireless networks with rf energy harvesting and transfer," *IEEE Network*, vol. 29, no. 6, pp. 68–75, Nov. 2015.

[42] R. Hof, *As AI moves to the chip, mobile devices are about to get much smarter*, http://siliconangle.com/blog/2016/05/05/as-ai-moves-to-the-chip-mobile-devices-are-about-to-get-much-smarter/, Visited on 2016-08-19, 2016.