Contribution à la conception de systèmes en virgule fixe

Daniel Ménard

HABILITATION À DIRIGER DES RECHERCHES

présentée devant

L'Université de Rennes 1
Spécialité : traitement du signal et télécommunications

par

Daniel MÉNARD

Contribution à la conception de systèmes en virgule fixe

soutenue le 29 novembre 2011 devant le jury composé de :

Michel PAINDAVOINE	Professeur des Universités, Univ. Dijon.	Président
Maurice BELLANGER	Professeur émérite, CNAM Paris	Rapporteur
Jean-Michel MULLER	Directeur de recherche, CNRS	Rapporteur
Eric GOUBAULT	Directeur de recherche, CEA.	Membre
Michel JEZEQUEL	Professeur, Télécom Bretagne	Membre
Olivier SENTIEYS	Professeur des Universités, Univ. Rennes I	Membre

et au vu des rapports de :

Maurice BELLANGER	Professeur émérite, CNAM
Wayne BURLESON	Professeur, Univ. of Massachusetts Amherst
Jean-Michel MULLER	Directeur de recherche, CNRS

Remerciements

Tout d'abord, je souhaite témoigner ma gratitude à l'ensemble des membres du jury pour le temps consacré à juger mon travail de recherche. J'adresse mes sincères remerciements à messieurs Wayne Burelson, *professeur à l'Université de Massachusetts Amherst, Jean Michel* Muller *directeur de recherche au CNRS et Maurice* Bellanger *professeur au CNAM Paris pour avoir accepté de juger mon travail en tant que rapporteur.*

Je tiens à remercier particulièrement monsieur Michel Paindavoine, *professeur à l'Université de Dijon pour m'avoir fait l'honneur d'être le président du jury de cette HDR. Je remercie également messieurs Olivier* Sentieys, *professeur à l'Enssat et Éric* Goubault, *directeur de recherche au CEA pour leur participation à ce jury.*

Je remercie vivement Olivier Sentieys, *responsable de l'équipe Cairn. Il m'a remis le pied à l'étrier de la recherche en 1999, en me permettant de faire une thèse sous sa direction au sein de l'équipe Signal Architecture du LASTI. Depuis, notre collaboration s'est poursuivi à travers le co-encadrement de doctorants et la participation à différents projets de recherche. Ses conseils sont toujours aussi pertinents.*

Je remercie Arnaud Tisserand *pour sa lecture attentive de ce manuscrit, ses conseils et les corrections proposées. De même, que mon frère Gérard, soit assuré de ma gratitude pour sa relecture finale.*

C'est avec plaisir que j'ai présenté, ce 29 novembre 2011, la synthèse de mes travaux de recherche effectués au cours de ces neuf dernières années. Cette synthèse s'appuie en grande partie sur un travail d'équipe. En premier lieu, je souhaite remercier vivement les doctorants co-encadrés depuis 2003, Nicolas Hervé, *Romuald* Rocher, *Shafqat* Khan, *Hai-Nam* Nguyen, *Karthick* Parashar, *Andreï* Bancïu, *Mahtab* Alam, *Jean-Charles* Naud. *Chaque encadrement a été une expérience humaine riche.*

Ces travaux de recherche se sont accompagnés du développement d'une infrastructure logicielle. Je remercie les ingénieurs Mohammed Diab, *Loic* Cloâtre, *Jeremy* Guillot, *Quentin* Meunier, *Nicolas* Simon *et les différents stagiaires ayant travaillés sur ces développements.*

Je souhaite remercier les collègues avec qui j'ai directement collaboré sur différents travaux de recherche : Olivier Berder, *Emmanuel* Casseau, *Romuald* Rocher *Pascal* Scalart, *Thibault* Hilaire, *Quentin* Meunier, *Steven* Derrien, *Arnaud* Tisserand, *Daniel* Chillet *et Sébastien* Pillement.

Ces différentes collaborations, ces échanges de connaissances, ces réflexions et ces discussions riches contribuent à rendre cette activité de recherche passionnante.

Je remercie tous les membres de l'équipe CAIRN. Ce fameux Cairn spirit *régnant au sein de l'équipe contribue à rendre le travail plus agréable. Vivement le prochain séminaire au vert !*

Je remercie le personnel technique et administratif de l'ENSSAT, de l'Université de Rennes I et de l'IRISA/INRIA pour avoir contribué, de près ou de loin, à ce travail.

Enfin, je n'oublie pas de remercier chaleureusement tous les membres de ma famille et amis qui m'ont soutenu, encouragé et aidé tout au long de ces différentes années. J'adresse un grand merci à ma petite famille, pour qui, ces derniers mois n'ont pas été faciles. Vous avez du subir mon manque de disponibilité et ma patiente limitée. Alors, merci de m'avoir soutenu et permis de finir ce travail dans les temps.

À Élio, Lisa et Marie,

Avant propos

Ce document présente la synthèse de mes travaux de recherche réalisés depuis ma nomination aux fonctions de maître de conférences à l'Enssat [1] en 2003. Ces activités ont été réalisées au sein du laboratoire UMR 6074 IRISA [2] et du centre de recherche INRIA [3] Rennes Bretagne Atlantique et plus particulièrement dans l'équipe projet commune IRISA/INRIA CAIRN [4]. Cette équipe projet fait suite à l'équipe projet IRISA R2D2 [5]. Cette équipe intègre des personnels de l'Université de Rennes 1, du CNRS [6], de l'INRIA et de l'ENS [7] Cachan–Bretagne sur les sites de Rennes et Lannion. Au sein de cette équipe, les différentes compétences en architecture des circuits intégrés, en traitement du signal et en informatique permettent de développer trois axes de recherche. Le premier axe vise à définir des plate-formes hétérogènes reconfigurables dynamiquement. Le second concerne la définition et la mise en œuvre d'outils de compilation et de synthèse pour ces architectures. Le troisième s'intéresse à l'interaction entre algorithmes et architectures.

Mes activités de recherche se situent dans le domaine de l'implantation efficace d'applications de traitement du signal et de l'image au sein de systèmes embarqués. Ces activités concernent plus particulièrement les aspects arithmétiques de l'implantation. L'objectif de mes travaux de recherche est de proposer une méthodologie efficace de conversion automatique en virgule fixe et de développer les outils associés. De plus, je travaille sur la mise en œuvre de techniques permettant d'optimiser l'implantation d'applications au sein de systèmes embarqués. Plus particulièrement, les applications de communication numérique, les aspects énergétiques et la représentation des données en virgule fixe sont considérés. Ces activités s'intègrent dans les trois axes de recherche de l'équipe CAIRN. L'aspect architecture a été abordé à travers la conception d'opérateurs reconfigurables flexibles. Les travaux sur les méthodes de conversion automatique en virgule fixe s'intègrent dans le second axe traitant des outils d'implantation. Les travaux sur l'évaluation des effets de la précision finie sur les performances d'une application et ceux sur la réduction d'énergie au sein des systèmes de communication sont réalisés dans le cadre du troisième axe concernant l'interaction algorithme architecture.

Ce document est composé de deux parties. Dans la première partie, les activités réalisées dans le cadre de mes fonctions de maître de conférences sont résumées. Dans la seconde partie, la synthèse de mes travaux de recherche est présentée. Après avoir présenté dans le chapitre 2, le contexte de mes travaux de recherche, nos contributions dans le domaine de l'évaluation des performances et de la précision des systèmes en virgule fixe sont détaillées dans le chapitre 3. Dans le chapitre 4, les travaux réalisés sur la conversion automatique en virgule fixe sont présentés. Dans le chapitre 5, nos travaux sur l'adéquation application système sont présentés. Plus particulièrement, nos activités sur la réduction d'énergie au sein des systèmes de communication sont détaillées. Finalement, le bilan scientifique de ces travaux et les perspectives de recherche sont présentés dans le chapitre 6.

1. École nationale supérieure des sciences appliquées et de technologie.
2. Institut de recherche en informatique et systèmes aléatoires.
3. Institut national de recherche en informatique et en automatique.
4. *Energy efficient computing architectures with embedded reconfigurable resources.*
5. Reconfigurable and retargetable digital devices.
6. Centre national de la recherche scientifique
7. École normale supérieure

Table des matières

Chapitre 1

Résumé des activités

1.1 Curriculum Vitae

Informations personnelles

Daniel MÉNARD
17 lotissement Convenant Dillec
22300 ROSPEZ
Tél. : 02 96 38 00 89
37 ans.
Marié, 2 enfants.

Coordonnées professionnelles

Enssat
6 Rue de Kérampont
22300 LANNION
Tél. : 02 96 46 27 48
Fax : 02 96 46 66 75
Daniel.Menard@enssat.fr

Parcours professionnel

Depuis sept. 2011	**Délégation INRIA**
Depuis sept. 2003	**Maître de conférences** en 61^e section à l'Enssat (Lannion - 22), Université de Rennes I
Septembre 2001 à août 2003	**Attaché Temporaire d'Enseignement et de Recherche** en 61^e section à l'Enssat, Université de Rennes I, (mi-temps).
Octobre 2000 à décembre 2002	**Préparation du doctorat de l'Université de Rennes 1**, au sein du groupe Signal-Architecture du Laboratoire d'Analyse des Systèmes de Traitement de l'Information (LASTI).
Octobre 1996 à septembre 2000	**Ingénieur de recherche (ITARF) en électronique** à l'Enssat. Soutien à la recherche et à la pédagogie. – Mise en œuvre de plates-formes pour les TPs et projets de la spécialité EII et au sein du laboratoire LASTI
Février à juillet 1996	**Stage de DEA** au sein de la société DASSAULT SERCEL NP (Carquefou - 44) et en collaboration avec le laboratoire SEI (CNRS - Université de Nantes). *Détection et estimation des multi-trajets dans le système GPS.*

Formation et titres universitaires

1999–2002 **Doctorat de l'Université de Rennes I en *traitement du signal***, mention : Très Honorable.

Thèse soutenue le 12 décembre 2002 devant le jury composé de E. Martin (UBS), D. Demigny (ENSEA), T. Risset (INRIA Rhône-Alpes), F. Charot (INRIA Rennes) P. Le Guernic (INRIA Rennes) et O. Sentieys (Enssat)

Méthodologie de compilation d'algorithmes de traitement du signal pour les processeurs en virgule fixe sous contrainte de précision.

1995–1996 **DEA d'Électronique de l'Université de Nantes,** option Télécommunications et Radar. *Mention : Très Bien.*

1993–1996 **Diplôme d'ingénieur IRESTE (Polytech'Nantes - Université de Nantes)**, option Systèmes Électroniques et Informatique Industrielle.

Activités para-pédagogiques

- 2004–2007 : Responsable de la première année du cycle ingénieur EII (Électronique Informatique Industrielle)
- Depuis 2007 : Responsable de la formation continue et de la VAE à l'Enssat
  - Gestion de la formation continue diplomante (réseau Fontanet)
  - Gestion des stages cours (formation qualifiante)
  - Participation à 4 jurys de VAE
- 2005–2010 : Chargé de la communication vers les IUT au sein de la spécialité EII

Mandats

- 2004–2007 : Membre nommé du conseil des études de l'Enssat
- Depuis 2007 : Membre élu du conseil de direction de l'Enssat

Participation à des groupes de travail

- 2009–2010 : groupe de travail *Développement durable* du centre INRIA Rennes Bretagne Atlantique
- depuis 2011 : groupe de travail *Développement durable* de l'Enssat

1.2 Activités d'enseignement

J'effectue mon activité d'enseignement en tant que maître de conférences depuis 2003 au sein de l'École Nationale Supérieure de Sciences Appliquées et de Technologie (Enssat) située à Lannion. Auparavant, j'ai participé à des actions d'enseignement au sein de ce même établissement en tant que vacataire de 1998 à 2001 et en tant qu'ATER, pendant deux ans, de 2001 à 2003.

L'Enssat recrute les étudiants au niveau *Bac + 2* et forme des ingénieurs dans les quatre cycles d'étude suivants : Électronique et Informatique Industrielle (EII), Logiciel et Système Informatique (LSI) et Optronique (OPT), Informatique Multimédia et Réseaux (IMR). Mes enseignements sont dispensés au sein de la spécialité Électronique et Informatique Industrielle et concernent les domaines du traitement numérique du signal, de l'électronique numérique, des systèmes embarqués et des systèmes temps réel.

De plus, j'enseigne le module *Logiciels embarqués pour le signal* au sein du Master SISEA depuis 2008 et je réalise une intervention à l'Institut Polytechnique de Bordeaux depuis 2009 sur les méthodologies de conversion en virgule fixe. Par ailleurs, j'ai réalisé une formation de 3 jours pour les industriels sur l'implantation d'algorithmes au sein de DSP virgule fixe et j'ai participé à l'enseignement d'une partie du module *Architecture des systèmes informatiques* dispensé au sein du centre régional de Brest du Conservatoire National des Arts et Métiers (CNAM).

Le détail des différents modules enseignés est présenté ci-dessous. Les modules dont je suis responsable sont en gras et les enseignements (CM, TD, TP, Projet) que j'ai entièrement créés ou renouvelés sont en gras :

- *Cycle d'ingénieur Enssat 1ère année (niveau L3) :*
  - **Logique combinatoire et séquentielle**, OPT1, (**CM : 10h, TD : 8h, TP : 8h**)
  - **Systèmes à microprocesseurs**, LSI1, (**CM : 16h, TD : 12h, TP : 8h, Projet : 36h**)
  - **Systèmes numériques, tronc commun** EII1-LSI1-OPT1, (**CM : 10h, TD : 10h, Projet : 18h**)
  - **Architecture des microprocesseurs**, IMR1, (**CM : 12h, TD : 10h, Projet : 18h**)
  - VHDL (TP : 16h).

- *Cycle d'ingénieur Enssat 2ème année (niveau M1) :*
  - **Processeurs de traitement du signal**, EII2, (**CM : 4h, TD : 6h, TP : 12h, Projet : 20h**)
  - **Chaîne de Traitement Numérique du signal**, OPT2, (**CM : 4h, Projet : 20h**)
  - Traitement numérique du signal, EII2, (CM : 4h, TD : 16h, TP : 8h),
  - Systèmes embarqués, IMR2, (**CM : 4h, TD : 2h, TP : 12h**),
  - Interface USB, EII2, LSI2 (TP : 12h),
  - Méthodologie SART - Systèmes temps réel (TD : 20h, Projet : 32h).

- *Cycle d'ingénieur Enssat 3ème année (niveau M2) :*
  - Optimisation de code, EII3, (CM : 4h, TD : 2h, **TP : 4h**),
  - Projet d'intégration : simulation système, EII3, (**TP : 8h**).

- *Master 2 SISEA de l'Université de Rennes I :*
  - **Logiciels embarqués pour le signal** (**CM 12h, TP : 4h**)

- *Cycle d'ingénieur Institut Polytechnique de Bordeaux, 3ème année (niveau M2) :*
  - Conversion en virgule fixe pour le TNS (**CM : 4h**).

- *Formation continue Enssat, stage cours de 3 jours :*
  - **Conception de systèmes en virgule fixe** (**CM : 6h, TP : 15h**).

- *Cnam :*
  - Architecture des systèmes informatiques (**CM : 5h**).

Le nombre d'heures réalisées chaque année en équivalent TD est présenté ci-dessous :

- 1998–1999 : 45 h
- 1999–2000 : 32 h
- 2000–2001 : 76 h
- 2001–2002 : 99 h
- 2002–2003 : 105 h
- 2003–2004 : 218 h
- 2004–2005 : 226 h
- 2005–2006 : 206 h
- 2006–2007 : 217 h
- 2007–2008 : 208 h
- 2008–2009 : 207 h
- 2009–2010 : 243 h
- 2010–2011 : 240 h

Pour le module Processeurs de Traitement du Signal, j'ai rédigé un polycopié de 75 pages. Pour le module *Logique Combinatoire et Séquentielle* un polycopié de cours (80 pages) a été co-rédigé avec Hélène DUBOIS. Pour les différents enseignements autour des systèmes à microprocesseurs (*Systèmes à microprocesseurs* (LSI1), *Systèmes numériques*, tronc commun (EII1-LSI1-OPT1), *Architecture des microprocesseurs* (IMR)), j'ai rédigé un polycopié regroupant la partie cours et les différents TD, TP et projets proposés ses dernières années (183 pages dont 93 pages pour la partie cours).

Pour les modules *Systèmes à microprocesseurs* et *Systèmes numériques*, les différents enseignements de TD, TP et projet se focalisent sur le développement d'une même application afin de pouvoir réaliser un système complet de taille raisonnable. Le cahier des charges et la décomposition fonctionnelle de cette application sont détaillés au cours du CM, puis, le système est développé au fur et à mesure des enseignements de TD, TP/projet. Cette approche permet d'aborder les problématiques associées à la mise en œuvre d'un système numérique complet, au fonctionnement en temps réel et au cadencement des tâches. De plus, les étudiants sont nettement plus motivés par ce type d'approche que par la résolution d'une suite d'exercices sans lien apparent entre eux. Cette pratique permet aux étudiants de prendre peu à peu de l'autonomie en étant de moins en moins guidés dans le travail à réaliser.

Pour les enseignements destinés aux étudiants en alternance (IMR), cette approche est développée encore plus en ayant une pédagogie centrée sur un projet de développement d'un système numérique. Le module débute avec la présentation du cahier des charges et la décomposition fonctionnelle. Ensuite, les modules composant le système sont développés progressivement. Les concepts théoriques sont présentés au fur et à mesure, lorsqu'ils sont nécessaires au développement de chaque fonctionnalité. Ceci permet aux étudiants d'appliquer aussitôt les concepts enseignés. Ainsi, la notion de séances de CM, TD, TP disparaît au profit d'alternances de présentation des concepts puis de mise en pratique par les étudiants. Cependant, cette approche trouve ses limites lorsque la taille des groupes est trop importante et nécessite de dédoubler les groupes et lorsque le niveau des étudiants au sein du groupe n'est pas homogène.

1.3 Activités de recherche

Domaines de recherche

- Au niveau de mes activités de recherche, mes différents centres d'intérêt sont :
  - Implantation d'applications de traitement du signal et de l'image au sein de systèmes embarqués
  - Arithmétique pour les systèmes embarqués, arithmétique virgule fixe
  - Architectures : DSP, ASIP, architectures reconfigurables, FPGA, ASIC
  - Applications : traitement du signal, communications numériques

Activités concernant la recherche

- Co-éditeur associé pour la revue *Journal of Advances in Signal Processing* de l'édition spéciale *Quantization of VLSI Digital Signal Processing Systems*, 2011 [Caffarena 12].
- Relecteur pour les conférences et revues suivantes : *IEEE Transactions on Industrial Electronics*, *Signal Processing*, *IET Circuits Devices & Systems*, *Journal of Embedded Systems*, *NewCas*, *DATE*, *DAC*.
- Bénéficiaire de la Prime d'Encadrement Doctoral et de Recherche (PEDR), 2007-2011.
- Présentations invitées :
  - Écoles d'été : ARCHI'03 [Ménard 03a], ARCHI'09 [Ménard 09b], RAIM 2009 [Ménard 09a].
  - Séminaires : LRTS [Ménard 05].
- Membre des GDR ISIS, ASR, SOC-SIP.
- Membre IEEE, Eurasip, club EEA.

Membre de jurys de thèse

- Rapporteur extérieur pour la mention européenne de la thèse de Luis Esteban Hernandez (*Universidad Politécnica de Madrid*), *High-precision FPGA-based Phase Meters for Infrared Interferometers Fusion Diagnostics*.
- Examinateur pour la thèse suivante : T. Saidi [74].
- Co-encadrant pour les thèses suivantes : R. Rocher [71], N. Hervé [44], S. Khan [51].

Obtention de financement

- Obtention d'un Bonus Qualité Recherche de l'Université de Rennes 1 en 2004 et en 2005 d'un montant global de 24000€ pour la mise en œuvre d'une plate-forme de radiocommunications pour les systèmes de transports intelligents.

Collaborations

Collaborations nationales

- Action Spécifique CNRS *Arithmétique des ordinateurs* (2002–2003), LIP, LIP6, LORIA, LIAFA, LIRMM, MANO Univ. Perpignan.
- Action Spécifique CNRS *Validation numérique pour le calcul embarqué* (2003–2004), Arénaire, LIP6, MANO Univ. Perpignan, LE2I Univ. de Dijon.
- PEPS CNRS *FiltrOptim* avec le LIP (Lyon) (2008) : Optimisation de la synthèse de filtres en virgule fixe et en virgule flottante.
- Projet avec le GDR RO (depuis 2011) : Recherche opérationnelle pour la CAO micro-électronique.

Collaborations internationales

- Université de Laval à Québec (Canada). Séjour en juin 2005 (1 semaine). Conception optimisée de récepteurs MIMO [Ménard 05].
- IMEC (Belgique), depuis 2007. Séjour en novembre 2008 (1 semaine). Raffinement de la spécification virgule fixe d'un décodeur sphérique (SSFE) pour récepteur MIMO [Parashar 10b, Ménard 10].
- EPFL (Suisse), depuis 2010. Exploitation du parallélisme au niveau donnée et évaluation du coût d'implantation.
- *Universidad Politécnica de Madrid* (Espagne), depuis 2009. Méthodologie de conversion en virgule fixe, évaluation de la précision [Caffarena 12].
- *University of Massachusetts of Amherst* VLSI CAD Lab (USA) : Prise en compte de la précision finie au sein des TED (*Taylor Expansion Diagram*).

Participation à des projets nationaux et internationaux

- **Projet UE FP7-ICT (STREP) ALMA** : *Architecture oriented paraLlelization for high performance embedded Multicore systems using scilAb* en collaboration avec Karlsruhe IT (D), Recore (NL), Univ. Peleponnese (GR), TMES (GR), Intracom (GR), Fraunhofer IO (D). Durée 36 mois, (2011-2014). L'objectif de ce projet de développer une infrastructure de compilation pour MPSoC avec une description Scilab de l'application. Plus particulièrement, le parallélisme à grain fin et moyen des architectures MPSoC. Ce projet permet de financer, pour notre activité, 2 p.an de post-doc, 2 p.an d'ingénieur et une bourse de thèse.

- **Projet ANR "Ingénierie Numérique et Sécurité" DEFIS** : *DEsign of FIxed-point Systems* en collaboration avec LIP6, CEA, LIRMM, Thales, InPixal. Durée 36 mois, (2011-2014). L'objectif de ce projet est de développer une infrastructure de conversion en virgule fixe permettant de traiter une application complète. Plus particulièrement des optimisations au niveau système et algorithmique seront proposées. Ce projet permet de financer, pour notre activité, 2 p.an de post-doc et 2 p.an d'ingénieur.

- **Projet R&D Nano 2012 S2S4HLS - ST Microelectronics** : *Source-to-Source Transformations for High-Level Synthesis*. Durée 48 mois, (2009-2012). L'objectif de ce projet est de développer une infrastructure logicielle de conversion en virgule fixe permettant de traiter un sous-système d'une application. De plus, une approche d'optimisation au niveau système est définie. Ce projet permet de financer, pour notre activité, 3 p.an d'ingénieur et une bourse de thèse (Jean-Charles Naud).

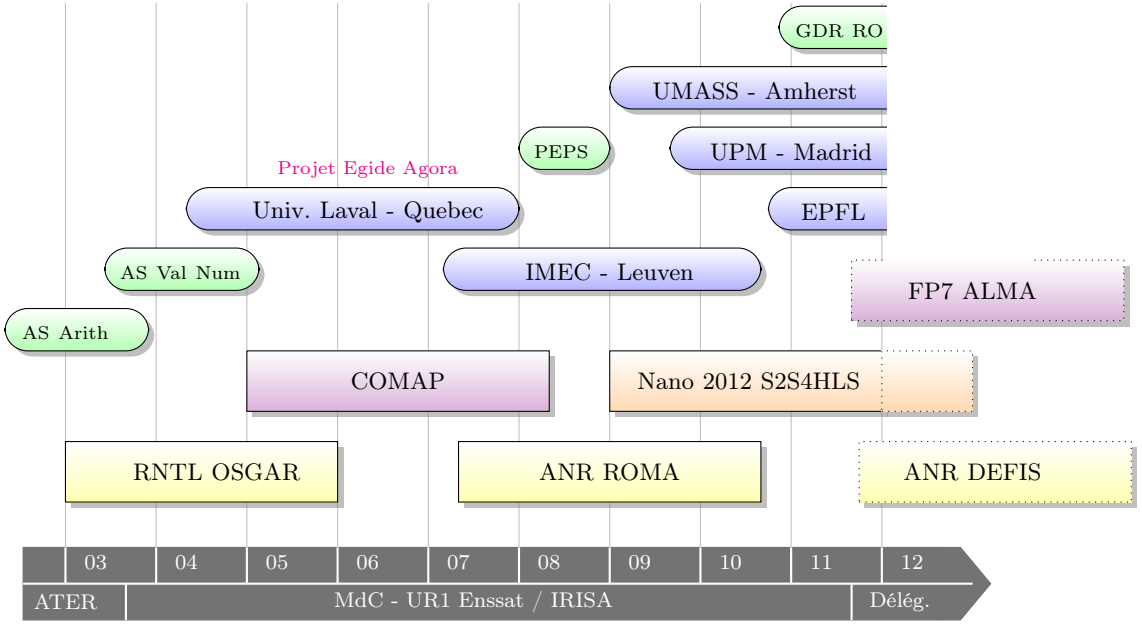
- **Projet ANR "Architecture du futur" ROMA** : *Reconfigurable Operators for Multimedia Applications*. Projet en collaboration avec le LIRMM, le CEA et Thomson. Durée 36 mois (2007-2009). L'objectif du projet est de développer une architecture reconfigurable au niveau opérateur destinée aux applications multimédia et les outils associés. Ce projet a permis de financer, pour notre activité 1, p.an d'ingénieur et une bourse de thèse (Shafqat Khan).

- **Projet COMAP** *Co-Design of Massively Parallel Embedded Processor Architectures* (Programmes de recherche en réseaux franco-allemand) [Hannig 05]. Projet en collaboration avec l'Université d'Erlangen-Nuremberg, l'Université Technologique de Dresde, l'ENST Bretagne et l'Université de Bretagne Occidentale (2005-2008). L'objectif de ce projet est la définition simultanée d'architectures massivement parallèle et des outils d'implantation associés. Je suis intervenu au sein de ce projet sur l'exploitation du parallélisme au niveau données.

- **Projet RNTL OSGAR** *Outils de Synthèse Générique pour Architectures Reconfigurables*. Projet

en collaboration avec le CEA, TNI-Valiosys et l'Université de Bretagne Occidentale. Durée 30 mois (2003-2005). L'objectif du projet était d'étudier et de développer des outils de synthèse de haut niveau permettant, à partir de code C, ou de spécifications semi-formelles, de générer automatiquement les configurations pour plusieurs circuits reconfigurables. Ce projet a permis de financer, pour notre activité, 1 p.an d'ingénieur et une bourse de thèse (Nicolas Hervé).

La répartition dans le temps de ces différents projets et collaborations est présentée dans le planning suivant :



Expertises auprès d'entreprises

– Réalisation de 6 expertises dans le cadre du programme JESSICA entre 1999 et 2005. Ces expertises ont concernées la définition de plate-formes et l'accompagnement des sociétés pour l'implantation d'applications dans le domaine des communications numériques ou du traitement du signal

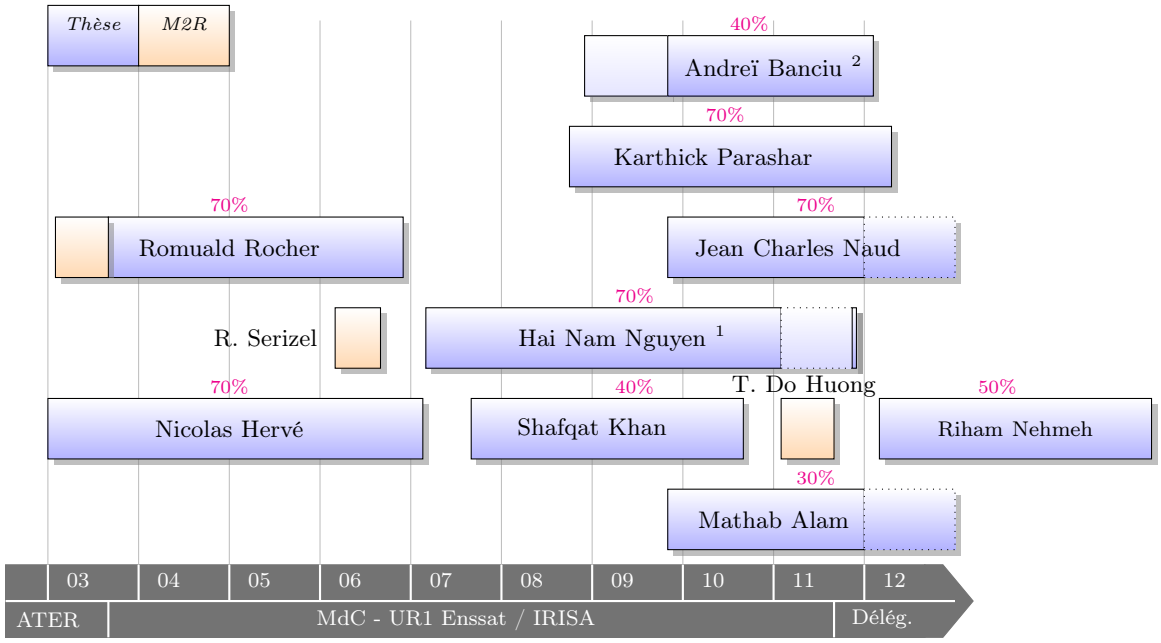
Encadrements de thèses et de master

Encadrement de Master recherche et de stages de DEA

Diplomé	Période	Intitulé du stage
Thao Do Huong	Mars 2011 - Septembre 2011	Évaluation des effets des débordements sur les performances
Romain Serizel	Mars 2006 - Septembre 2006	Implantation en virgule fixe d'un codeur audio
Romuald Rocher	Mars 2003 - Septembre 2003	Évaluation de la précision dans les filtres adaptatifs

Co-encadrement de thèses

Les différents doctorants co-encadrés sont présentés dans le tableau présenté à la page suivante. La répartition dans le temps de ces différents encadrements est présentée dans le planning suivant :



[1] H.N. Nguyen avait une activité professionnelle à plein temps en 2011
[2] Ma participation à l'encadrement a débuté 10 mois après le début de la thèse

Synthèse des co-encadrements de thèses

Diplômé	Date de soutenance	T_{enc}	Directeur de thèse	Financement	Intitulé de la thèse	Situation actuelle
Khan Shafqat	29/10/2010	40%	E. Casseau	Contrat ANR	Développement d'architectures matérielles hautes performances pour des applications multimédia	Ingénieur de recherche AERO[1]
Hervé Nicolas	19/03/2007	70%	O. Sentieys	Contrat RNTL	Contributions à la synthèse d'architecture virgule fixe à largeurs multiples	Ingénieur de Recherche LSITEC[2]
Rocher Romuald	07/12/2006	70%	O. Sentieys	Bourse MESR	Evaluation analytique de la précision des systèmes en virgule fixe	Maître de Conférence Université de Rennes 1
Nguyen Hai-Nam	16/12/2011	70%	O. Sentieys	Bourse CG22	Optimisations de la précision des calculs pour réduire la consommation d'énergie des systèmes embarqués	Ingénieur OWS (Paris)
Bancïu Andreï	28/2/2012	40%	E. Casseau	CIFRE ST	Stochastic approach for the range estimation (CIFRE ST Microelectronnics)	
Parashar Karthick	à soutenir 5/2012	70%	O. Sentieys	INRIA Cordi	System level approach for fixed-point conversion	
Alam Mahtab	à soutenir 12/2012	30%	O. Sentieys	Contrat FP7	Power Aware Signal Processing for Reconfigurable Radios in the context of Wireless Sensor Networks	
Naud Jean-Charles	à soutenir 11/2012	70%	O. Sentieys	Contrat S2S4HLS	Transformations source à source pour la conversion en virgule fixe	
Nehmeh Riham	à soutenir 3/2015	50%	**D. Ménard**	CIFRE ST	Évaluation de performances et optimisation de systèmes en virgule fixe	

T_{enc} : Taux d'encadrement.
[1] AREO : Advanced Engineering Research Organization (Pakistan).
[2] LSI-TEC : Integrating Systems Technological Laboratory (Brésil).

1.4 Description des travaux de recherche

Mes activités de recherche se situent dans le domaine de l'implantation efficace d'applications de traitement du signal et de l'image au sein de systèmes embarqués. Ces activités concernent plus particulièrement les aspects arithmétiques de l'implantation. De très nombreux systèmes embarqués intègrent des traitements mathématiques. Pour satisfaire les contraintes d'implémentation inhérentes aux systèmes embarqués, l'arithmétique virgule fixe est largement utilisée. Les applications sont conçues et simulées en virgule flottante, mais au final, elles sont implantées au sein d'architectures utilisant l'arithmétique virgule fixe. Le nombre limité de bits, pour coder les données, nécessite une analyse détaillée de la dynamique des données et de la précision des calculs. Le codage en virgule fixe est une tâche fastidieuse, longue et source d'erreurs qui ralentit l'implantation des applications embarquées. Ainsi, la réduction du temps de mise sur le marché couplée à l'augmentation de la complexité des applications nécessite des outils de développement de haut niveau permettant d'automatiser la conversion en virgule fixe.

Ainsi, l'objectif de mes travaux de recherche est de définir une méthodologie de conversion automatique en virgule fixe, de développer les outils associés et de mettre en œuvre des techniques permettant d'optimiser l'implantation d'un point de vue énergétique à travers des aspects arithmétiques.

Méthodologie de conversion en virgule fixe

La conversion en virgule fixe d'une application consiste à déterminer, pour chaque donnée, le nombre de bits alloués à la partie entière et à la partie fractionnaire. Ainsi, le processus de conversion en virgule fixe est composé de deux étapes principales correspondant à la détermination de la position de la virgule et à l'optimisation de la largeur des données.

Évaluation de la dynamique des données

La première étape de la conversion en virgule fixe correspond à la détermination du nombre de bits pour la partie entière de chaque donnée. Celle-ci nécessite de connaître la dynamique (domaine de définition) de chaque donnée. Les méthodes classiques de détermination de la dynamique basées sur l'arithmétique d'intervalle ou l'arithmétique affine permettent de déterminer les bornes du domaine de définition et ainsi de garantir l'absence de débordement (dépassement de capacité). Cependant, ces méthodes surestiment les bornes de l'intervalle et conduisent ainsi, à la présence de bits non utilisés au niveau de la partie entière. Dans le cadre de la thèse d'Andrei Banciu (2009-2011, CIFRE avec ST Microelectronics), nous étudions des méthodes de détermination de la dynamique des données permettant de fixer la probabilité de débordement. En effet, pour de nombreux systèmes pour lesquels un compromis est réalisé entre le coût de l'implantation et les performances (qualité) de l'application, la présence de débordements peut être acceptée si la dégradation des performances de l'application reste limitée. Dans ce cas, la problématique est de déterminer la fonction de densité de probabilité de chaque variable afin d'en déduire la dynamique pour une probabilité de débordement donnée. Les méthodes basées sur une modélisation stochastique ont été utilisées [Banciu 10]. En particulier, pour les systèmes linéaires invariant dans le temps (LIT), une décomposition de Karhunen-Loève [58] est utilisée pour modéliser les signaux en entrée du système puis ces paramètres sont propagés au sein du système afin d'obtenir ceux de la sortie. Pour les systèmes non-linéaires, des polynômes de chaos sont utilisés. Ces deux approches permettent d'approcher finement la fonction de densité de probabilité.

Optimisation de la largeur des données

La seconde étape du processus de conversion en virgule fixe consiste à optimiser la largeur des données. L'objectif est de minimiser le coût de l'implantation tant que les performances de l'application sont satisfaites. Ce processus d'optimisation combinatoire nécessite d'évaluer la fonction de coût correspondant au coût de l'implantation et d'évaluer la fonction de contrainte correspondant aux performances de l'application ou à la précision des calculs. L'évaluation de la fonction de contrainte est le point crucial du processus de

conversion en virgule fixe et nos travaux sur ces aspects sont présentés dans la partie 1.4.

Différents algorithmes ont été proposés dans la littérature pour optimiser la largeur des données. Les techniques proposées se basent sur des approches déterministes comme la recherche locale ou avec tabous, les techniques de séparation et évaluation et la programmation linéaire en nombre entiers ou sur des approches aléatoires telles que le recuit simulé ou les algorithmes génétiques. Dans le cadre de la thèse d'Hai-Nam Nguyen (2007-2011), des techniques d'optimisation de la largeur basées sur les algorithmes génétiques ont été améliorées afin d'obtenir une meilleure qualité de la solution. Les algorithmes génétiques conduisent à des temps d'optimisation assez longs mais ils permettent d'obtenir directement la frontière d'efficacité de Pareto du coût de l'implantation en fonction de la précision des calculs. Un algorithme de recherche locale stochastique basé sur GRASP [36] a été proposé pour l'optimisation de la largeur des données [Nguyen 11]. Cette approche combine une phase de construction aléatoire d'une solution de départ et une phase de raffinement de cette solution à l'aide d'une approche de recherche locale basée sur un algorithme glouton. Pour les différentes expérimentations, cette approche permet d'obtenir une solution meilleure que celles obtenues par les techniques proposées dans la littérature. De plus, nous avons proposé des techniques d'optimisation dans le cas des architectures à grain moyen en termes de largeurs supportées [Ménard 06, Ménard 11].

Dans le cadre d'une implantation matérielle, l'objectif est de synthétiser une architecture optimisée par rapport aux contraintes fournies par l'utilisateur. Le nombre d'unités fonctionnelles et les caractéristiques de celles-ci en termes de largeur étant choisis lors de la phase de synthèse, ce processus offre de nombreux degrés de liberté pour l'optimisation de la largeur des données. L'utilisation de largeurs spécifiques pour chaque opérateur permet d'obtenir, pour une même contrainte de précision des calculs, une solution dont le coût est plus faible que celui obtenu pour des opérateurs possédant tous la même largeur. Dans le cadre de la thèse de Nicolas Hervé (2003-2007) et du projet RNTL OSGAR, une méthodologie de synthèse d'architecture à largeurs multiples a été définie. La problématique est d'optimiser la largeur des opérateurs afin de minimiser le coût de l'implantation pour une contrainte de précision donnée. Ceci nécessite de connaître l'affectation des opérations aux opérateurs obtenue uniquement après la phase de synthèse d'architecture. Cependant, la synthèse d'architecture ne peut être réalisée qu'après l'optimisation de la largeur des données car elle nécessite la connaissance des largeurs des données pour en déduire le coût de chaque opérateur. Pour résoudre ce dilemme, une heuristique basée sur une approche itérative a été proposée [Herve 05, Hervé 07]. Chaque itération réalise le regroupement des données, l'optimisation de la largeur des groupes puis la synthèse d'architecture. Le regroupement des données est dirigé par les résultats de la synthèse précédente. Les résultats obtenus permettent de réduire significativement le coût de l'implantation par rapport à une approche mono-largeur.

Approche au niveau système

La conversion en virgule fixe d'une application complète nécessite de déterminer la largeur de plusieurs centaines de variables. La détermination de la spécification virgule fixe d'un système complet se traduit par un problème d'optimisation avec un nombre de variables à optimiser important. Dans le cadre de la thèse de Karthick Parashar (2009-2012), une approche hiérarchique permettant d'optimiser la spécification virgule fixe au niveau système a été définie [Parashar 10c]. Cette approche hiérarchique est utilisée pour découper le problème d'optimisation en plusieurs niveaux et ainsi restreindre le nombre de variables à optimiser à chaque niveau. Au niveau système, une variable d'optimisation est affectée à chaque bloc et correspond à la précision des calculs (puissance du bruit de quantification) au sein de ce bloc. Cette approche consiste, ainsi, à budgéter au sein de chaque bloc, la dégradation de précision permettant de réduire le coût global de l'implantation et de maintenir la précision des calculs. Pour cette approche système, le principal verrou à lever pour obtenir une approche efficace concerne l'évaluation des performances du système complet. Le comportement en précision finie de chaque bloc est modélisé par une source de bruit unique (voir section 1.4) présente en sortie du bloc et l'approche mixte analytique/simulation présentée dans la partie 1.4 est utilisée. Cette approche est en cours de validation sur un système complet correspondant à un récepteur de communication numérique OFDM MIMO.

Évaluation des performances des systèmes en virgule fixe

L'utilisation d'une arithmétique en précision finie entraîne la présence d'une erreur entre la valeur réelle et celle codée. Cette erreur modifie le comportement de l'application et conduit à une dégradation des performances de celle-ci. L'évaluation des performances est une phase cruciale du processus de conversion en virgule fixe, car celle-ci est répétée de nombreuses fois lors du processus d'optimisation de la spécification en virgule fixe. Ainsi, le challenge est de proposer une approche conduisant à une estimation précise et efficace en termes de temps d'exécution. De nombreuses méthodes sont basées sur la simulation de l'application en virgule fixe. Ces méthodes permettent de traiter tous les types d'applications mais conduisent à des temps de simulation trop élevés pour pouvoir explorer l'espace de conception lors du processus d'optimisation. Nos recherches se sont orientées sur des approches analytiques. L'objectif est de déterminer l'expression analytique d'une métrique estimant la précision des calculs. Dans le domaine des applications de traitement du signal et de l'image, la métrique la plus utilisée est la puissance du bruit de quantification en sortie du système (moment d'ordre deux du bruit). Les techniques proposées se basent sur la théorie de la perturbation et considèrent l'erreur de quantification comme une perturbation nettement plus faible que le signal.

Évaluation analytique de la puissance du bruit de quantification

Dans le cadre de ma thèse, une approche pour traiter les systèmes linéaires invariant dans le temps a été proposée. Les travaux menés ces dernières années ont eu pour objectif d'étendre la classe des applications pouvant être supportées.

Dans le cadre de la thèse de Romuald Rocher (2003-2006), une approche analytique d'évaluation de la précision des systèmes en virgule fixe a été définie pour les systèmes à base d'opérations arithmétiques [Ménard 04a, Ménard 08a, Rocher 07a]. Cette méthode permet d'obtenir l'expression analytique de la puissance du bruit en sortie de systèmes à base d'opérations pour lesquelles une relation linéaire (pouvant varier dans le temps) entre le bruit en entrée et en sortie peut être utilisée. Pour linéariser certains opérateurs comme la division, un développement en série de Taylor du premier ordre est utilisé. Cette approche permet de déterminer les gains entre chaque source de bruit et la sortie. Pour réduire le temps de calcul de ces gains une approche basée sur la prédiction linéaire est utilisée. La qualité de l'approche en termes de temps d'exécution et de précision de l'estimation a été vérifiée à travers un outil développé sous Matlab.

L'utilisation d'un graphe flot de signal pour représenter l'application nécessite d'éliminer les structures de contrôle. En particulier, les structures répétitives dont le nombre d'itérations est connu statiquement, sont complètement déroulées. Dans le cadre de la thèse de Jean Charles Naud (2010-2012), notre approche d'évaluation analytique de la précision a été étendue afin de pouvoir traiter les structures conditionnelles. Dans ce cas, les sources de bruit présentes avant la structure conditionnelle peuvent emprunter différents chemins en fonction des alternatives des structures conditionnelles. L'expression de la puissance du bruit de quantification en sortie de l'application a été adaptée afin de prendre en compte les différents chemins parcourus par chaque source de bruit. Cette approche nécessite de connaître la probabilité d'exécuter chaque chemin. Celle-ci est obtenue statiquement ou à l'aide d'une phase de *profiling* sur un jeu de tests représentatif en entrée de l'application.

Évaluation des performances

Modèle de source de bruit unique

Le concepteur d'applications connaît les spécifications de son application et les performances attendues pour celle-ci, cependant, il n'est pas aisé de définir directement la contrainte de précision des calculs. Ainsi, dans le cadre d'une méthodologie de conversion automatique en virgule fixe, il est nécessaire de faire un lien entre les métriques de performance définies par le concepteur et la contrainte de précision nécessaire à notre approche de conversion en virgule fixe. Dans le cadre du stage de master recherche de Romain Serizel (2006) [Ménard 07, Ménard 08b], le modèle de source de bruit unique et une approche permettant de faire le lien entre les métriques de performance et la contrainte de précision ont été proposés et testés. L'objectif

de ce modèle est de remplacer l'ensemble des traitements d'un système en virgule fixe par une unique source de bruit présente en sortie du système. La problématique est de définir les caractéristiques statistiques de cette source afin de modéliser le plus finement possible le bruit réellement présent en sortie du système. L'obtention des caractéristiques spectrales a été proposée dans [Parashar 10a] et la densité de probabilité dans [Parashar 10e]. Ce modèle de source de bruit unique a été utilisé dans le cadre de l'optimisation de la spécification virgule fixe au niveau système.

Approche mixte analytique/simulation

L'approche proposée pour l'évaluation de la précision et basée sur la théorie de la perturbation permet de traiter les systèmes composés d'opérations pour lesquelles une approximation par un modèle de propagation linéaire du bruit, pouvant varier dans le temps, peut être utilisée. Cependant cette approche n'est plus valide dans le cas des opérations de décision car les hypothèses de la théorie de la perturbation ne sont plus satisfaites. En effet, le bruit de quantification en entrée de l'opération peut modifier la prise de décision par rapport à la précision infinie et, en conséquence, l'erreur entre la précision finie et infinie commise en sortie possède une amplitude de l'ordre de grandeur de la valeur en précision infinie et celle-ci ne peut plus être assimilée à une perturbation de faible amplitude. Dans le cadre de la thèse de Karthick Parashar et d'une collaboration avec l'IMEC, les erreurs de décision ont été prises en compte pour l'évaluation des performances d'un système.

L'expression analytique de la densité de probabilité de cette erreur a été proposée dans [Parashar 10d]. Cependant, la propagation des erreurs de décision au sein d'un système est complexe. Ainsi, une approche mixte utilisant la simulation et les résultats analytiques a été proposée [Parashar 10b]. Cette technique utilise le modèle de source de bruit unique, présenté dans la partie 1.4, pour modéliser le bruit en entrée de l'opération de décision. En l'absence d'erreur de décision, les hypothèses associées à la théorie de la perturbation sont valides et les résultats de l'approche d'évaluation analytique de la précision peuvent être utilisés. Lorsqu'une erreur de décision survient, la partie de l'application impactée par cette erreur est simulée afin d'obtenir la valeur en précision finie en sortie de l'application. Par nature, les occurrences d'erreurs de décision doivent être faibles afin d'avoir un système fonctionnel. Ainsi, l'application est simulée très peu souvent. Par rapport à une approche classique basée sur la simulation, le temps nécessaire pour l'évaluation des performances est inférieur de trois à quatre ordres de grandeur.

Outils de conversion en virgule fixe

ID.Fix : Outil de conversion en virgule fixe

Ces différents travaux de recherche sur la conversion automatique en virgule fixe sont accompagnés du développement d'une infrastructure logicielle ID.Fix. Celle-ci est développée dans le cadre du projet R&D Nano 2012 avec la société ST Microelectronics. Cet outil se base sur l'infrastructure de compilation GECOS développée au sein de l'équipe CAIRN. L'infrastructure ID.Fix réalise des transformations de code source à source et permet de transformer un code C intégrant des types flottants en un code C avec une spécification virgule fixe optimisée définie à travers des classes C++. L'évaluation de la dynamique est réalisée à l'aide d'une approche mixant l'arithmétique d'intervalle pour les systèmes non-récursifs (absence de cycle au sein du graphe de l'application) et de la norme L1 pour les systèmes linéaires invariants dans le temps et récursifs. L'évaluation de la précision est réalisée à l'aide de l'approche analytique détaillée dans la partie 1.4. L'optimisation de la spécification virgule fixe est réalisée à l'aide de l'algorithme présenté dans [Ménard 11].

L'effort de développement en termes de ressources humaines contractuelles est de 60 hommes.mois (h.m) dont 38 h.m d'ingénieur (ingénieur jeune diplômé, ingénieur expert contrat S2S4HLS), le reste étant réalisé dans le cadre de projets et de stages de fin d'études d'ingénieur.

Synthèse de blocs paramétrables

L'expérience acquise dans le domaine de l'arithmétique virgule fixe a permis de proposer de nouveaux types de générateurs de composants virtuels configurables (IP) [Rocher 06a, Rocher 06b]. Ces générateurs d'IPs fournissent pour une application particulière de traitement du signal une architecture (code VHDL) optimisée en fonction d'une contrainte de précision des calculs. Des générateurs ont été réalisés pour des filtres linéaires (FIR, IIR) et pour des filtres adaptatifs (LMS [Rocher 04], NLMS [Rocher 10], DLMS, APA [Rocher 05a]). Ce concept de composants dédiés a été étendu dans le cadre du post-doctorat effectué par T. Hilaire pour les systèmes de contrôle-commande. Un environnement logiciel développé sous Matlab permettant d'optimiser d'un point de vue arithmétique un système de contrôle-commande a été proposé [Hilaire 07, Hilaire 08]. Celui-ci permet d'optimiser sous contrainte la structure utilisée (exploration algorithmique) et la largeur des coefficients et des signaux (exploration architecturale).

Adéquation Application-Arithmétique

L'implantation efficace d'applications orientées traitement de données au sein de plateformes embarquées nécessite d'optimiser le dimensionnement des données afin de minimiser le coût de l'implantation et de fournir une précision des calculs suffisante pour garantir des performances de l'application minimales. Ce dimensionnement est réalisé lors de la phase d'implantation de l'application mais, celui-ci peut évoluer au cours du temps afin de s'adapter aux modifications des contraintes de l'application.

Optimisation de l'implantation d'applications

Pour tester et valider nos travaux de recherche, les applications utilisées sont issues du domaine des systèmes communicants. Plus particulièrement, les systèmes de téléphonie mobile de troisième génération basés sur la technologie WCDMA [Ménard 03c, Nguyen 09a], ceux de quatrième génération basés sur la technologie MIMO-OFDM et les réseaux de capteurs [Alam 11a] sont étudiés. Dans le cadre de la thèse de Taofik Saïdi (2004-2008), l'implantation temps-réel d'un récepteur MIMO a été étudiée. Les aspects arithmétiques ont été particulièrement pris en compte pour optimiser l'architecture.

Adaptation dynamique de la précision

Dans le cadre de la thèse d'Hai-Nam Nguyen (2007-2011), le concept d'adaptation dynamique de la précision a été défini [Nguyen 08, Nguyen 09b]. L'objectif est d'optimiser la consommation d'énergie à travers les aspects arithmétiques. L'intérêt de techniques adaptant la spécification virgule fixe (largeur des opérations) au cours du temps afin de réduire la consommation d'énergie a été montré sur différents exemples. L'adaptation de la spécification virgule fixe est liée à l'évolution au cours du temps de la contrainte de précision. Cette dernière dépend des conditions externes au système. Par exemple, pour un récepteur de communication numérique embarqué, les paramètres, modifiant la contrainte de précision, peuvent être le niveau de bruit en entrée du récepteur, la qualité de service désirée (taux d'erreurs binaires). Ces travaux se poursuivent dans le cadre de la thèse de Mahtab Alam (2009-2012) dont l'objectif plus général est de mettre en œuvre des techniques [Alam 11a] d'adaptation dynamique des traitements et des paramètres du système afin d'optimiser la consommation d'énergie des objets communicants présents dans les réseaux de capteurs.

L'adaptation dynamique de la précision se base sur des opérateurs reconfigurables fournissant une flexibilité plus importante en termes de largeur de données supportée. La conception de ces opérateurs a été réalisée dans le cadre du projet ANR ROMA et de la thèse de Shafkat Khan (2008-2010). La flexibilité des opérateurs en termes de largeur est obtenue en utilisant le parallélisme au niveau des données (SWP : Sub-Word Parallelism). Un opérateur SWP de largeur N peut traiter en parallèle, k opérations de largeurs N/k. Nous avons conçu des opérateurs [Khan 09b, Khan 10, Ménard 09c] permettant de traiter plus de largeurs différentes que ceux proposés dans la littérature. Dans ce cadre, différentes représentations des nombres et structures d'opérateurs ont été testées.

Première partie

Synthèse des travaux de recherche

Chapitre 2

Introduction

Systèmes embarqués

Les évolutions technologiques rapides dans le domaine de la micro-électronique ont permis de de développer le secteur des systèmes embarqués. Plus de 10 milliards de processeurs sont vendus chaque année et 98% sont destinés au marché des systèmes embarqués [4]. Ce marché représentait 92 Md$ en 2008 avec une croissance annuelle de 5,6 % [48]. Pour le grand public, le nombre d'appareils électroniques utilisés dans la vie quotidienne personnelle et professionnelle n'a cessé de s'accroître au cours de ces dernières années et les services offerts par ces appareils sont toujours plus nombreux. Ces appareils sont de plus en plus connectés à un réseau afin de pouvoir échanger de l'information. Cette tendance va se poursuivre à travers le concept d'informatique ubiquitaire.

Nombre de ces systèmes embarqués intègre des applications contenant des traitements mathématiques de données. Ces applications sont issues, par exemple, des domaines du traitement du signal et de l'image (TDSI) ou du contrôle/commande. Ces applications sont présentes dans de nombreux secteurs comme les télécommunications, les transports, l'aérospatiale, la robotique, l'électronique médicale ou l'électronique grand public. Par exemple, les nouveaux systèmes de téléphonie mobile tels que les *smartphones* intègrent de très nombreuses fonctionnalités basées sur des applications de TDSI pour la connectivité, l'interface avec l'utilisateur ou pour fournir de nouveaux services. La connectivité est assurée selon différents standards tels que les systèmes de communication cellulaire de seconde génération (GSM [1]/EDGE [2]) et de troisième génération (UMTS [3]), les systèmes d'accès aux réseaux locaux sans fil (WiFi [4]) ou de communication à courte distance (*bluetooth*). Un système de positionnement tel que le GPS [5] peut être présent au sein de l'appareil. De nombreuses interfaces avec l'utilisateur (écran, caméra, micro, haut-parleurs) sont disponibles et ainsi, l'appareil possède des modules de compression/décompression du son, de la parole, de l'image et de la vidéo selon différents standards.

En conséquence, les contraintes majeures des systèmes embarqués sont le coût, la consommation d'énergie, le temps de développement et la fiabilité du système. Les systèmes embarqués évoluant dans un contexte temps réel, les temps d'exécution des applications doivent être minimisés ou au moins maitrisés.

Le coût est une contrainte forte des systèmes embarqués et plus particulièrement lorsque ces produits sont destinés au marché du grand public. Le coût de la partie matérielle est principalement lié à la surface du circuit et à la taille de la mémoire. En fonction du marché visé, ces aspects ont un impact différent sur la conception du système. Pour les systèmes à fort volume de production, le coût est directement proportionnel à la surface du circuit et ainsi le concepteur cherche à minimiser celle-ci. Pour les systèmes à faible volume de production, le temps de développement est la contrainte forte.

1. *Global System for Mobile communications.*
2. *Enhanced Data Rates for GSM Evolution.*
3. *Universal Mobile Telecommunications System.*
4. *Wireless Fidelity.*
5. *Global Positioning System.*

Pour de nombreux systèmes embarqués et en particulier ceux destinés au marché du grand public, le temps de mise sur le marché est un critère crucial pour la réussite de la commercialisation du produit et ainsi pour sa rentabilité. Ces éléments soulignent la nécessité d'outils permettant d'automatiser la conception de systèmes électroniques. De plus, l'objectif de ces outils est d'explorer l'espace de conception afin de sélectionner une solution optimisée au niveau du coût d'implantation.

La consommation d'énergie est devenue une contrainte majeure des systèmes embarqués. Dans la majorité des cas, ces systèmes étant autonomes en énergie ou alimentés par batterie, il est nécessaire de minimiser la consommation d'énergie afin de maximiser le temps d'autonomie ou réduire le coût du système d'alimentation en énergie. De plus, la prolifération des systèmes embarqués se traduit par un impact qui n'est plus négligeable sur la consommation totale d'énergie électrique et ainsi, pose des problèmes d'un point de vue économique et environnemental.

La consommation d'énergie est composée d'une partie statique et d'une partie dynamique. La consommation statique est liée aux courants de fuite au sein des transistors. Ainsi, elle dépend des caractéristiques technologiques des transistors et de leur nombre au sein du circuit. Pour réduire la consommation statique, le nombre de transistors doit être minimisé, les caractéristiques des transistors doivent être optimisées et des techniques de *power gating* peuvent être utilisées pour ne pas alimenter les parties du circuit ne travaillant pas. La consommation dynamique dépend de l'activité du circuit, de sa capacité équivalente, de sa fréquence de fonctionnement et de sa tension d'alimentation. L'activité du circuit est liée à la taille des opérateurs, des bus et de la mémoire. La réduction de la tension d'alimentation permet de diminuer la consommation d'énergie mais elle se traduit par une augmentation de la latence des opérateurs. Ainsi, avec la technique DVFS [6] [69], la tension d'alimentation et la fréquence de fonctionnement sont ajustées en fonction de la charge de calculs à réaliser.

Finalement, la fiabilité des systèmes embarqués est une contrainte forte. En particulier, il est nécessaire de garantir l'absence de défaut de fonctionnement lié aux calculs réalisés au sein du système. Ainsi, le comportement numérique de l'application doit être soigneusement estimé, contrôlé et validé. Des dépassements de capacité d'une variable, comme dans le cas du premier vol de la fusée Ariane V [57], ou un manque de précision des calculs, comme dans le cas des missiles Patriot [10], peuvent avoir des conséquences parfois dramatiques. La réduction du nombre de transistors au sein d'un circuit permet de rendre le système plus fiable.

Représentation des nombres

Différents types de représentation des données peuvent être utilisés pour implanter les calculs au sein d'un système numérique. L'arithmétique virgule fixe est largement utilisée dans les systèmes embarqués. Elle permet de représenter des données réelles avec un nombre de bits fixe pour la partie entière et pour la partie fractionnaire. Ce nombre restreint de bits conduit à une dynamique des données et à une précision des calculs limitées. Une alternative est l'arithmétique virgule flottante qui est présente dans les processeurs généralistes pour les ordinateurs personnels ou les serveurs. La représentation en virgule flottante est composée d'un exposant et d'une mantisse. Cette représentation permet de s'adapter à la valeur à coder en utilisant un facteur d'échelle explicite à travers l'exposant. Les types de donnée disponibles au sein de la norme IEEE-754 offrent une dynamique des données et une précision assez élevées pour implanter rapidement une application au sein d'un système embarqué. Cependant, le coût d'implantation est significativement plus élevé par rapport au coût obtenu avec une représentation en virgule fixe.

La représentation en virgule fixe conduit à des architectures plus rapides, nécessitant une surface plus faible et consommant moins d'énergie par rapport à celles obtenues avec la représentation en virgule flottante. Avec la représentation en virgule fixe, la largeur des bus et des mémoires au sein des architectures en virgule fixe étant plus faible, le prix et la consommation d'énergie de ces architectures sont moins importants. L'arithmétique en virgule flottante basée sur la norme IEEE-754 utilise majoritairement des données sur 32 (simple précision) ou 64 bits (double précision). A titre d'exemple, la majorité des processeurs de traitement

6. *Dynamic Voltage and Frequency Scaling.*

numérique du signal programmables (DSP) virgule fixe possède une largeur naturelle nettement plus faible (16 bits). De plus, pour la représentation en virgule flottante certains traitements comme l'alignement de la position de la virgule sont pris en charge par le matériel. Par rapport à la représentation en virgule fixe, ceci décharge le développeur de cette tâche, mais augmente la complexité et la latence de l'opérateur en incluant, en particulier, le module de normalisation.

À titre d'exemple, Texas Instrument propose au sein de la famille de DSP C6000 des processeurs ayant la même architecture et disponibles avec la représentation en virgule fixe ou en virgule flottante. Un rapport de 9, au profit de la version en virgule fixe, est obtenu en termes de temps d'exécution entre les deux versions pour des noyaux représentatifs du domaine du traitement du signal [8]. De même, un rapport de 6 est obtenu en termes d'efficacité au niveau du coût (temps d'exécution/prix) et un ratio de 5 en termes d'efficacité énergétique (temps d'exécution/mW).

Avec la représentation en virgule fixe, le codage des données n'est pas normalisé, ainsi, celui-ci peut être adapté en fonction des exigences de l'application. De nombreuses applications dans le domaine des télécommunications, du contrôle/commande, de la vidéo ou du traitement d'image peuvent tolérer une précision faible ou moyenne. Ces applications sont souvent interfacées avec le monde physique à travers des capteurs ou des actionneurs. Les données sont issues de convertisseurs analogique-numérique ou alimentent des convertisseurs numérique-analogique pour lesquels le nombre de bits est limité. Ainsi, la précision des données aux interfaces de l'application est limitée et en conséquence la précision des calculs sera dans les mêmes ordres de grandeur. Les contraintes de précision de ces applications sont compatibles avec l'utilisation d'une représentation en virgule fixe sur un nombre de bits raisonnable. Ainsi, pour satisfaire les différentes contraintes des systèmes embarqués, présentées auparavant, la représentation en virgule fixe est préférée pour de nombreuses applications [33, 35, 78, 34].

Outils pour automatiser le développement des systèmes embarqués

Face à la complexité grandissante des applications implantées au sein des systèmes embarqués, et face à la nécessité de réduire les temps de mise sur le marché, des outils sont nécessaires pour automatiser le processus d'implantation de ces applications sur des plateformes embarquées. Progressivement, des outils permettant d'automatiser certaines phases de l'implantation en élevant le niveau d'abstraction de la description de l'application sont apparus. Pour les implantations logicielles au sein d'un processeur, les compilateurs permettent depuis de nombreuses années de pouvoir décrire l'application à l'aide d'un langage de haut niveau comme le C. Même pour les architectures spécialisées comme les DSP ou les ASIP [7], depuis une dizaine d'années, les compilateurs C sont généralement assez efficaces pour limiter l'écriture de code en assembleur. Pour les implantations matérielles au sein d'un ASIC [8] ou d'un FPGA [9], les outils de synthèse logique ont permis dans les années 90 de pouvoir décrire l'architecture au niveau fonctionnel puis, plus récemment, l'avènement des outils de synthèse de haut niveau permet de décrire l'application au niveau comportemental [37].

Du point de vue de l'application, la conception et l'évaluation des performances des algorithmes sont réalisées à l'aide d'outils de simulation tels que Matlab/Simulink [62], CoCentric Studio (Synopsys) [50], SPW (CoWare [10]) [25, 26], Ptolemy [32], Scicos/Scilab. Les applications embarquées orientées traitement de données sont conçues et simulées en virgule flottante mais il est nécessaire de convertir l'application en virgule fixe afin de pouvoir réaliser l'implantation au sein de l'architecture.

La conversion en virgule fixe vise à fixer le nombre de bits pour la partie entière et la partie fractionnaire de chaque donnée. La dynamique limitée du codage nécessite de connaître le domaine de définition de chaque donnée pour déterminer le nombre de bits nécessaires pour représenter la partie entière de la donnée en garantissant l'absence de débordement. Les données doivent être alignées avant de réaliser les opérations d'addition et de soustraction afin d'obtenir un résultat correct. La précision limitée conduit à une erreur non

7. *Application-Specific Instruction-set Processor.*
8. *Application-Specific Integrated Circuit.*
9. *Field Programmable Gate Array.*
10. La société Synopsys a fait l'acquisition de la société CoWare en 2010.

désirée entre la valeur codée (précision finie) et la valeur exacte (précision infinie). Cette erreur doit être évaluée afin de garantir que la précision numérique est suffisante pour limiter la dégradation des performances de l'application liée aux traitements en précision finie. De plus, dans le cadre d'une implantation matérielle, la largeur des opérateurs doit être déterminée. Ainsi, le processus de conversion en virgule fixe doit explorer l'espace de conception pour trouver la combinaison des largeurs des données permettant de minimiser le coût de l'implantation et maintenir une précision des calculs suffisante.

La phase de conversion en virgule fixe se situe entre les phases de conception et d'implantation de l'application. Elle nécessite des connaissances sur l'application afin de bien maîtriser les critères de performance associés à celle-ci et des connaissances sur l'architecture afin de tirer profit des types supportés par celle-ci. Cette étape est réalisée par les personnes en charge de la conception de l'application ou ceux devant réaliser l'implantation. La conversion en virgule fixe est une tâche longue, fastidieuse et source d'erreurs. Dans une enquête réalisée en 2006 [47], la conversion en virgule fixe était identifiée comme l'une des tâches les plus difficiles de l'implantation d'une application au sein d'un FPGA. Certaines expérimentations [18] ont montré que la conversion manuelle d'une application peut représenter entre 25% et 50% du temps total de développement. Dans [9], l'analyste J. Bier (BDTI Inc.), spécialiste de l'implantation d'applications de TNS [11] au sein de systèmes embarqués, estime que dans la décennie à venir, malgré leur surcoût, les architectures en virgule flottante pourrait être de plus en plus présentes dans les applications embarquées en raison de la difficulté à convertir en virgule fixe manuellement des applications de complexité croissante. Ainsi, l'augmentation de la complexité des applications et la réduction du temps de mise sur le marché nécessitent des outils de haut niveau permettant d'automatiser ce processus de conversion en virgule fixe. Au niveau industriel, la demande pour ce type d'outils est forte. Les outils commerciaux existants sont inefficaces pour obtenir une spécification virgule fixe optimisée. Ces outils utilisent des approches basées sur la simulation pour évaluer les effets de la précision finie et conduisent à des temps d'optimisation longs ne permettant pas d'explorer l'espace de conception.

Travaux de recherche et plan du document

L'objectif de mes travaux de recherche est de proposer une méthodologie efficace de conversion automatique en virgule fixe et de développer les outils associés. L'objectif est de réduire le temps de développement de systèmes en virgule fixe en automatisant le processus de conversion et d'optimiser le coût de l'implantation en explorant l'espace de conception en virgule fixe. De plus, la mise en œuvre de techniques permettant d'optimiser l'implantation d'applications au sein de systèmes embarqués a été étudiée. Plus particulièrement, les applications de communication numérique, les aspects énergétiques et la représentation des données en virgule fixe sont considérés.

Ces travaux de recherche sont illustrés à travers le synoptique présenté à la figure 2.1 et décrivant une partie des étapes du cycle de développement des applications embarquées. Tout d'abord, la phase de conception de l'application permet, à partir des spécifications, de définir les différents algorithmes composant le système. Au cours de cette phase, les performances de l'application (qualité du service fourni) sont évaluées à travers un ensemble de simulations. Lorsque l'architecture ciblée utilise l'arithmétique en virgule fixe, une conversion de l'application en virgule fixe est réalisée. Ensuite, les algorithmes sont implantées au sein de l'architecture à travers une phase de synthèse d'architecture pour une implantation matérielle, ou de compilation pour une implantation logicielle.

Dans le processus de conversion en virgule fixe, l'évaluation des effets de la précision finie sur les performances de l'application est l'un des problèmes majeurs. Dans le chapitre 3, nos différentes contributions pour l'évaluation des performances et de la précision des calculs sont détaillées. Dans la première partie, notre approche d'évaluation analytique de la précision est présentée. La démarche suivie pour obtenir l'expression analytique de la puissance du bruit est détaillée. Les limites de ce type d'approche sont fournies et notre outil permettant d'implanter cette approche est présenté. Dans la seconde partie, notre approche

11. Traitement Numérique du Signal.

mixte analytique/simulation permettant de gérer les opérations, dont le modèle de propagation du bruit de quantification n'est pas linéaire, est proposée. La qualité et l'efficacité des différentes approches proposées sont évaluées à travers un ensemble d'expérimentations.

Nos différentes contributions à l'automatisation du processus de conversion en virgule fixe sont présentées dans le chapitre 4. L'objectif est d'optimiser la spécification en virgule fixe en vue de minimiser le coût de l'implantation. Dans la première partie, nous présentons nos études en cours sur l'évaluation de la dynamique à base d'approches stochastiques. Dans la seconde partie, nous traitons du problème d'optimisation des largeurs à travers deux aspects. Le premier concerne la définition d'un algorithme d'optimisation adapté au type de l'architecture ciblée. Le second aspect concerne l'optimisation de la largeur des opérateurs dans le cas de la synthèse d'architecture. Dans la troisième partie, l'approche hiérarchique proposée pour pouvoir optimiser la largeur d'un système complexe est détaillée. Dans la quatrième partie, l'infrastructure logicielle développée pour réaliser la conversion en virgule fixe est présentée.

Dans le chapitre 5, nous présentons nos travaux sur l'optimisation de l'implantation d'applications de TDSI au sein de systèmes embarqués. Dans une première partie, nous présentons les travaux réalisés sur l'implantation d'applications issues du domaine des communications numériques et sur la génération de blocs matériels dédiés. Dans la seconde partie, le concept d'adaptation dynamique de la précision (ADP) est présenté puis l'architecture développée dans le cadre du projet ROMA et supportant l'ADP est détaillée.

Dans ce document, deux modes de citations sont utilisés afin de différencier mes publications personnelles des autres. Pour mes publications, le mode de citation utilisant le nom de l'auteur et l'année ([Ménard 11]) est utilisé. Les autres publications sont référencées par numéro.

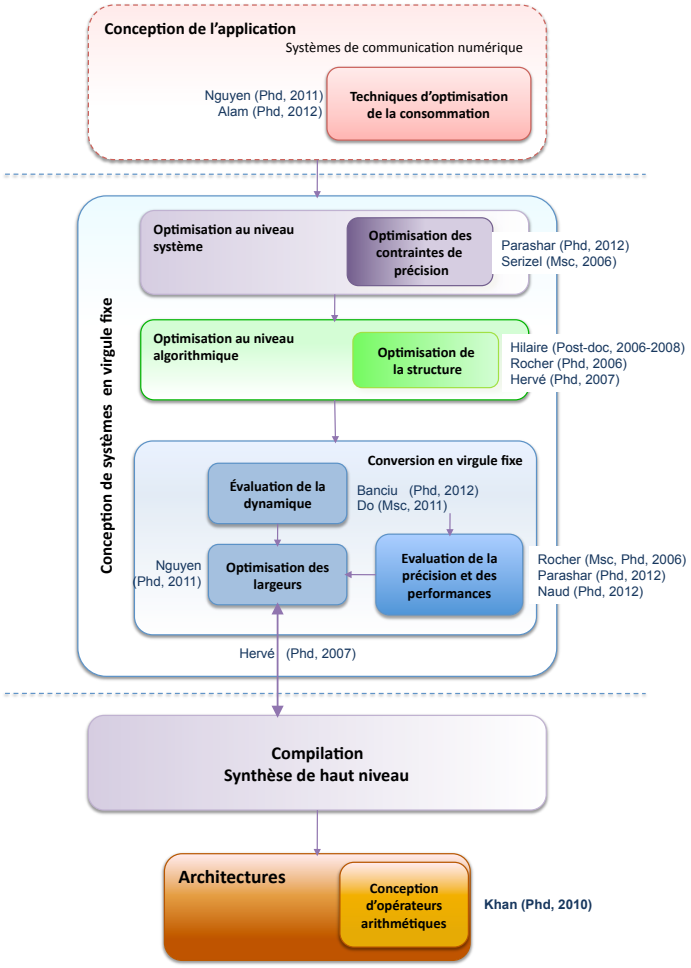


FIGURE 2.1 – Synoptique des travaux de recherche réalisés. Les étudiants encadrés durant leur doctorat (Phd) ou leur Master (Msc) sont reportés.

Chapitre 3

Évaluation des performances

3.1 Introduction

L'utilisation de l'arithmétique en virgule fixe conduit à une erreur entre la valeur codée et la valeur exacte dont l'amplitude dépend du nombre de bits utilisés pour coder les données. Cette erreur modifie le comportement de l'application et entraîne une dégradation des performances de l'application. Ces performances correspondent aux critères de qualité associés à l'application et définis au sein des spécifications opératoires. Le processus de conversion en virgule fixe va chercher à optimiser la largeur des données en minimisant le coût de l'implantation sous contrainte de performance de l'application. L'évaluation des performances en précision finie est l'un des problèmes majeurs de la conversion en virgule fixe. Cette évaluation doit être de qualité mais surtout efficace en termes de temps d'exécution. En effet, elle est intégrée dans le processus d'optimisation des largeurs et ainsi sollicitée plusieurs fois à chaque itération de ce processus.

Deux types d'approche peuvent être utilisés pour évaluer les performances d'une application en fonction de la spécification virgule fixe utilisée. Les méthodes par simulation permettent de traiter tous les types de systèmes, mais comme nous le verrons par la suite, les simulations en virgule fixe conduisent à des temps d'exécution élevés. Par exemple, les performances d'un récepteur de communication numérique sont évaluées à travers le taux d'erreur binaire (TEB) mesurant la probabilité que les symboles émis et reçus ne soient pas identiques. Pour un TEB de 10^{-x}, 10^{x+2} échantillons sont nécessaires afin d'avoir une estimation statistique précise [6].

Dans le cas d'une approche analytique, l'expression mathématique du critère de performance, en fonction de la largeur des données, est déterminée. Chaque type d'application ayant ses propres critères de performance, il n'est pas envisageable d'avoir un outil permettant d'automatiser la génération de l'expression analytique des critères de performance. En conséquence, une métrique de précision des calculs intermédiaire est utilisée. Le terme précision des calculs signifie l'écart entre la valeur exacte et la valeur codée et correspond au terme anglais *accuracy*. Différentes normes, conduisant à différentes métriques, peuvent être utilisées pour analyser cet écart. Dans le reste du document, le terme précision est employé seul et correspond à la précision des calculs. A l'aide de cette métrique de précision des calculs, l'analyse de la dégradation des performances est scindée en deux étapes. La première permet de définir la valeur minimale de la métrique de précision en fonction de la dégradation souhaitée des performances. La seconde étape réalise l'optimisation des largeurs de données sous contrainte de précision des calculs minimale.

Dans la première partie de ce chapitre, notre approche d'évaluation analytique de la précision est présentée. Cette approche vise à estimer la puissance du bruit de quantification en sortie d'un système et permet de générer automatiquement l'expression de cette métrique de précision. Cette approche basée sur la théorie de la perturbation permet de traiter tous les systèmes à base d'opérations dont la sortie est une fonction dérivable de ses entrées. Ainsi, cette approche ne peut pas traiter les opérations de décision. Dans la seconde partie de ce chapitre, nous nous intéressons à l'évaluation des performances. Tout d'abord, nous présentons le concept de source de bruit unique permettant de modéliser un sous-système par une seule source de bruit.

Ensuite, nous montrons l'intérêt de cette source de bruit unique pour déterminer la contrainte de précision au sein du processus de conversion en virgule fixe. Finalement, nous présentons notre approche mixte analytique/simulation permettant de gérer les opérations de décision. L'approche analytique est utilisée pour toutes les parties du système composées d'opérations ayant un modèle de bruit linéaire et la simulation est utilisée pour évaluer les performances lorsqu'une erreur de décision survient sur une des opérations de décision.

3.2 Évaluation de la précision des calculs

Encadrement :	Romuald Rocher, doctorat en 2006 [71] Jean-Charles Naud, doctorant depuis 2009
Conférences :	ICASSP 2004 [Rocher 04] ICASSP 2005 [Rocher 05a], Eusipco 2004 [Ménard 04a], Eusipco 2007 [Rocher 07a], Eusipco 2010 [Ménard 10], Eusipco 2011 [Naud 11b] Gretsi 2007 [Rocher 05b], Gretsi 2007 [Rocher 07b], Sympa 2011 [Naud 11a]
Revues :	IEEE Trans. Circuits & Systems I 2008 [Ménard 08a], 2012 [Rocher 12] Digital Signal Processing (Elsevier) 2010 [Rocher 10]
Collaboration :	Programme R&D nano 2012 (ST Microelectronics), Univ. Poly. Madrid (Espagne)
Logiciel :	ID.Fix-AccEval (Évaluation analytique de la précision (partie 3.2.3))

Dans cette partie, après avoir fait un état de l'art des méthodes existantes pour évaluer la précision des calculs et plus particulièrement la puissance du bruit de quantification, les concepts théoriques de notre approche d'évaluation analytique de la précision sont présentés, puis, l'outil associé est décrit. A travers un ensemble d'expérimentations, notre approche a été analysée et comparée aux approches existantes en termes de qualité et d'efficacité de l'estimation. La qualité est mesurée à travers la précision de l'estimation par rapport à une valeur de référence et l'efficacité est mesurée à travers le temps d'exécution de l'outil.

3.2.1 État de l'art

3.2.1.1 Métriques de précision

Différentes métriques peuvent être utilisées pour évaluer la précision des calculs en virgule fixe. Cette précision peut être évaluée à travers les bornes de l'erreur [29, 2], le nombre de bits significatifs [17] ou la puissance de l'erreur de quantification [Ménard 02f], [76, 16]. La densité spectrale de puissance de l'erreur de quantification est utilisée comme métrique dans [20] pour l'implémentation de filtres linéaires. Dans [15], une métrique plus complexe capable de supporter plusieurs modèles d'erreur est proposée.

Soit e_q, l'erreur entre la valeur $\widehat{x}$ en précision finie et la valeur x en précision infinie. Pour la métrique correspondant aux bornes de l'erreur, l'intervalle de l'erreur e_q doit être déterminé. Cette métrique est utilisée pour les systèmes embarqués critiques. Ces systèmes dont une défaillance peut mettre en danger des vies humaines sont présents dans le domaine du transport ou du nucléaire par exemple. Pour ces systèmes, le surcoût lié à l'utilisation d'une arithmétique plus complexe n'étant pas un critère prépondérant, l'arithmétique en virgule flottante est privilégiée afin d'avoir une dynamique des valeurs représentables et une précision nettement plus élevée. Cette métrique est utilisée au sein de systèmes critiques pour lesquels, l'erreur doit être bornée afin de garantir la sûreté de fonctionnement du système. Des outils tels que Fluctuat [30] basé sur l'interprétation abstraite ou Gappa [27] ont été proposés.

Pour la métrique correspondant à la puissance de l'erreur de quantification, l'erreur e_q est considérée être une variable aléatoire (bruit) et le moment statistique d'ordre deux est calculé. Cette métrique analyse la dispersion des valeurs en précision finie par rapport à la précision infinie et le comportement moyen de l'erreur. La métrique de puissance du bruit de quantification est utilisée lorsqu'une dégradation relativement faible des performances de l'application par rapport à la précision infinie est acceptable. Dans ce cas, l'implantation d'une application au sein d'un système embarqué est un compromis entre les performances de

l'application (qualité de service) et le coût de l'implantation (prix, surface, consommation d'énergie, temps d'exécution). Ce type d'approche est utilisé dans de nombreux domaines tels que l'électronique grand public ou les télécommunications.

3.2.1.2 Approches basées sur la simulation

Les techniques basées sur la simulation ont été largement utilisées pour évaluer la précision des calculs mais aussi pour évaluer directement les performances de l'application. L'application est simulée en virgule fixe et en virgule flottante et les résultats des simulations sont comparés afin d'en déduire la valeur de la métrique de précision ou la dégradation des performances liée à l'arithmétique virgule fixe. L'arithmétique en virgule flottante (double précision) est considérée fournir les valeurs de référence. Cette approximation est correcte lorsque la valeur des largeurs des données en virgule fixe n'est pas trop importante. Dans ce cas, les erreurs de calculs associées à l'arithmétique en virgule flottante sont nettement plus faibles que celles associées à l'arithmétique en virgule fixe et ainsi, peuvent être négligées. Si la précision des calculs en virgule flottante peut poser des problèmes, des bibliothèques de calcul en multi-précisions tel que MPFR [39] peuvent être utilisées. Cette approche pragmatique, basée sur la simulation en virgule fixe, est largement utilisée dans l'industrie.

De nombreuses infrastructures de simulation telles que Matlab/Simulink [62], CoCentric Studio [50], SPW [25, 26], Ptolemy [32] offrent la possibilité de modéliser et simuler une application en virgule fixe. L'avantage de ces méthodes est de pouvoir supporter tous les types de systèmes. Cependant, le défaut majeur réside dans la durée de l'évaluation de la précision [24]. L'émulation de l'arithmétique virgule fixe sur une machine en virgule flottante et le nombre important d'échantillons nécessaires pour obtenir des statistiques précises, conduisent à des temps de simulation élevés. Des techniques ont été proposées pour diminuer le temps de simulation [52, 63, 1, 72, 7, 53, 49, 24, 55, 22, 23], mais celui-ci reste malgré tout trop important pour aboutir à une conversion en virgule fixe efficace. En effet, dans le cadre du processus d'optimisation, la précision des calculs est évaluée de très nombreuses fois (plusieurs fois à chaque itération). En conséquence, ce type de technique n'est viable que pour un nombre de variables dans le processus d'optimisation faible ou si l'espace de conception en virgule fixe (espace de recherche du problème d'optimisation) est fortement limité conduisant, ainsi, à une solution sous-optimale.

3.2.1.3 Approches analytiques

Dans le cas des approches analytiques, une expression mathématique de l'estimation de la métrique de précision est déterminée. Le temps nécessaire pour déterminer cette métrique est plus ou moins important mais ce processus n'est réalisé qu'une seule fois avant le processus d'optimisation. Ensuite, l'évaluation de la métrique de précision pour une combinaison des largeurs de données est rapide et correspond à l'évaluation d'une expression mathématique. Les techniques utilisées dépendent de la métrique de précision choisie. Uniquement les techniques permettant d'évaluer la puissance du bruit de quantification sont présentées ci-dessous.

Pour calculer l'expression analytique de la puissance du bruit de quantification, les approches existantes sont basées sur la théorie de la perturbation. Les valeurs en précision finie correspondent aux valeurs en précision infinie (signal) entachées d'une perturbation dont l'amplitude est très faible par rapport à l'amplitude des valeurs en précision infinie. Cette perturbation est dénommée erreur de quantification. Une erreur de quantification e_q est générée lorsque des bits sont éliminés lors du processus de quantification (changement de format virgule fixe). Cette erreur est assimilée à un bruit additif b_i se propageant à l'intérieur du système. Cette source de bruit b_i contribue à la présence d'un bruit de quantification b_y en sortie du système. Chaque source de bruit traverse un système S_i. L'objectif de cette approche est de définir l'expression de la puissance du bruit en sortie en fonction des paramètres statistiques des sources du bruit b_i et des gains associés au système S_i traversé.

Le modèle (PQN [1]) couramment utilisé pour la quantification d'un signal x d'amplitude continue, a été proposé dans [81] et raffiné dans [77]. Le bruit additif b_i est un bruit blanc [2] uniformément réparti, non corrélé avec le signal x et les autres bruits de quantification. Ce modèle est valable lorsque la dynamique du signal x est suffisamment grande par rapport au pas de quantification et si la bande passante du signal est assez grande [77, 82]. Ce modèle a été étendu pour modéliser le bruit de quantification généré lorsque des bits sont éliminés lors d'un changement de format [5]. Dans [19], un modèle basé sur une densité de probabilité (DDP) discrète est proposé et les moments du premier et du second ordre sont donnés en fonction du nombre k, de bits éliminés.

Chaque source de bruit b_i se propage au sein du système et contribue au bruit global b_y en sortie du système. Cette propagation doit être modélisée pour obtenir l'expression du bruit de quantification en sortie. Le modèle de propagation du bruit repose sur l'hypothèse que le bruit de quantification est suffisamment faible par rapport au signal. Les valeurs en précision finie correspondent aux valeurs en précision infinie auxquelles s'ajoute une petite perturbation. Ces modèles, basés sur la théorie de la perturbation, ne sont valables que pour les opérations dont la sortie est une fonction dérivable de ses entrées.

Le bruit de sortie b_y est la somme de toutes les contributions des N_e sources de bruit. Le moment du second ordre de b_y peut être exprimé comme un somme pondérée des paramètres statistiques des sources de bruit :

$$E\left[b_y^2\right] = \sum_{i=1}^{N_e} K_i \cdot \sigma_{b_i}^2 + \sum_{i=1}^{N_e}\sum_{j=1}^{N_e} L_{ij} \cdot \mu_{b_i}\mu_{b_j}. \tag{3.1}$$

où $E[\ ]$ représente l'espérance mathématique. Les termes μ_{b_i} et $\sigma_{b_i}^2$ sont respectivement la moyenne et la variance de la source de bruit b_i. Les termes K_i et L_{ij} sont des constantes et dépendent du système situé entre b_i et la sortie. Ainsi, ces termes sont déterminés une seule fois pour l'obtention de l'expression analytique de la précision.

Dans [12], les coefficients K_i et L_{ij} sont calculés avec une technique utilisant une simulation basée sur l'arithmétique affine. La méthode a été proposée pour les systèmes linéaires invariants dans le temps (LTI [66]) dans [60] et pour les systèmes non LTI dans [12]. Les valeurs des coefficients de K_i et L_{ij} sont extraites de la forme affine du bruit de sortie. Dans le cas des systèmes récursifs [3], un nombre suffisant d'itérations pour la simulation basée sur l'arithmétique affine doit être utilisé pour converger vers des valeurs stables. Ce temps de convergence dépend de la longueur des réponses impulsionnelles entre les sources de bruit et la sortie et peut devenir important pour des systèmes dont la réponse impulsionnelle [4] est longue.

Différentes techniques hybrides [76, 21, 38] ont été proposées pour estimer les coefficients K_i et L_{ij} de l'équation 3.1 à partir d'un ensemble de simulations. Dans [76], les $N_e(N_e+1)$ coefficients sont obtenus en résolvant un système d'équations linéaires dans lequel K_i et L_{ij} sont des variables. Les autres éléments de l'équation 3.1 sont déterminés à travers la réalisation de simulations en virgule fixe et en modifiant un par un les formats de chaque donnée. Cette approche nécessite au moins $N_e(N_e+1)$ simulations en virgule fixe. Le temps d'obtention de l'expression analytique peut devenir important lorsque le nombre de sources de bruit à considérer est élevé.

3.2.2 Évaluation analytique de la puissance du bruit de quantification

Dans cette partie, le modèle proposé pour évaluer analytiquement la précision des calculs est présenté. L'expression de la puissance du bruit de quantification est fournie. Ce modèle est valide pour tous les types

1. *Pseudo-Quantization Noise.*
2. Les échantillons ne sont pas corrélés dans le temps $E[b_i(n)b_i(n-\tau)] = \sigma_{b_i}^2\,\delta(\tau) + \mu_{b_i}^2$.
3. Au sens traitement du signal du terme. La sortie y à l'instant n d'un système récursif dépend des échantillons précédents $y(n-j)$.
4. La réponse impulsionnelle d'un système, dont l'entrée est x et la sortie y, correspond à la sortie $y(n)$ obtenue lorsque l'entrée est une impulsion de Dirac $x(n) = \delta(n)$ avec $\delta(n) = 1$ si $n = 0$ et $\delta(n) = 0$ sinon.

de système composé d'opérations dont le modèle de bruit est linéaire. Ces travaux ont été réalisés dans le cadre de la thèse de Romuald Rocher [71] et de Jean-Charles Naud (2009–2012).

3.2.2.1 Systèmes composés d'opérations à modèle de bruit linéaire

Modélisation du bruit de quantification

Modèle de source du bruit Le modèle PQN est utilisé pour modéliser le processus de quantification $\mathcal{Q}$ d'un signal x. Les travaux présentés dans [19] ont été étendu, dans [Ménard 10], afin de proposer les expressions des moments d'ordre 1 et 2 du bruit de quantification b_i pour les différents modes de quantification (troncature, arrondi, arrondi convergent). Ce modèle considère une DDP discrète de la source de bruit.

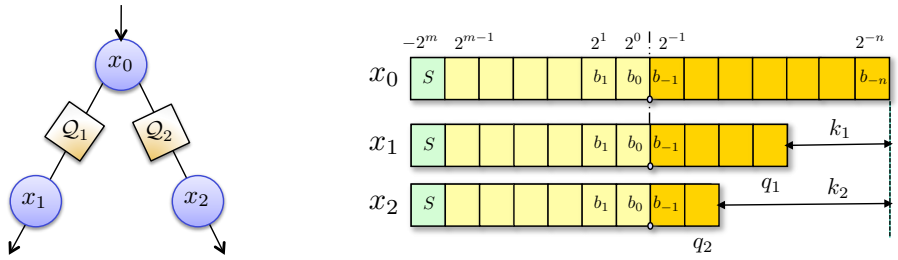


FIGURE 3.1 – Processus de quantification double de la variable x_0 conduisant à deux variables x_1 et x_2.

A travers le modèle PQN, les bruits sont considérés non corrélés entre eux. Cette hypothèse est valide uniquement lorsque les sources de bruit ne sont pas issues de la quantification d'un même signal. Ainsi, dans le cas de la thèse de Jean Charles Naud, l'expression de l'intercorrélation entre deux sources de bruit a été proposée afin d'intégrer celle-ci dans l'expression globale de la puissance du bruit de quantification en sortie du système. Considérons deux processus de quantification $\mathcal{Q}_1$ et $\mathcal{Q}_2$ d'une même donnée x_0 et représentés à la figure 3.1. Soient q_1 et q_2 le pas de quantification de la donnée après respectivement les processus de quantification $\mathcal{Q}_1$ et $\mathcal{Q}_2$. Soient k_1 et k_2 le nombre de bits éliminés au sein des processus $\mathcal{Q}_1$ et $\mathcal{Q}_2$. Dans le cas de la troncature, l'expression de l'intercorrélation entre les deux sources de bruit b_1 et b_2 est la suivante :

$$E\left[b_1.b_2\right] = \frac{q_2^2}{12}\left(1 + 2^{-2k_2+1} - 3\cdot 2^{-k2}\right) - \frac{q_1^2\cdot 2^{-k_1}}{4} + \frac{q_1 q_2}{4}. \tag{3.2}$$

Modèle de propagation du bruit Les techniques d'évaluation analytique de la puissance du bruit, basées sur la théorie de la perturbation, s'appuient sur un modèle de propagation du bruit au sein des opérations présentes dans l'application. Une opération composée de deux entrées x et y et d'une sortie z est considérée. Les entrées $\widehat{x}$ et $\widehat{y}$ et la sortie $\widehat{z}$ en virgule fixe sont respectivement la somme des valeurs exactes (précision infinie) x, y et z et des bruits de quantification associés b_x, b_y et b_z. Le modèle de propagation est défini tel que le bruit de sortie b_z soit une combinaison linéaire des deux bruits d'entrée b_x et b_y : $b_z = \nu_1 b_x + \nu_2 b_y$.

Ce modèle permet de ne pas avoir de produits croisés entre les termes de bruit pour obtenir une expression plus simple de la puissance du bruit de quantification en sortie du système. Les termes ν_1 et ν_2 sont obtenus à partir d'un développement en série de Taylor à l'ordre 1 de la sortie de l'opération de fonction f dérivable sur son intervalle de définition :

$$z = f(\widehat{x},\widehat{y}) = f(x,y) + \frac{\partial f}{\partial \widehat{x}}(x,y)(\widehat{x}-x)\frac{\partial f}{\partial \widehat{y}}(x,y)(\widehat{y}-y). \tag{3.3}$$

Ainsi les expressions des termes ν_1 et ν_2 sont les suivantes :

$$\nu_1 = \frac{\partial f}{\partial \widehat{x}}(x,y) \qquad \text{et} \qquad \nu_2 = \frac{\partial f}{\partial \widehat{y}}(x,y). \tag{3.4}$$

Les valeurs des termes ν_1 et ν_2 peuvent varier au cours du temps et leur expression est fournie dans [Rocher 07a]. Dans le cadre de la thèse de Romuald Rocher [71], une notation matricielle est utilisée afin d'obtenir des expressions plus compactes par rapport à une notation scalaire.

Modélisation du système Le modèle de propagation du bruit permet de ne pas avoir de produits croisés entre les termes de bruit. Ainsi, le bruit en sortie b_y, à l'instant n, correspond à la somme des contributions b'_i de chaque source de bruit b_i comme représenté à la figure 3.2 et à l'équation 3.5. Soit N_e, le nombre de sources de bruit au sein du système.

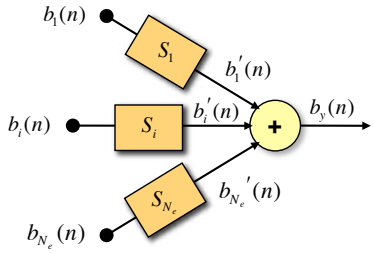


FIGURE 3.2 – Modélisation au niveau bruit du système pour N_e sources de bruit b_i.

$$b_y(n) = \sum_{i=1}^{N_e} b'_i(n). \tag{3.5}$$

Chaque contribution $b'_i(n)$ est obtenue à travers la propagation de chaque source de bruit $b_i(n)$ à travers le système S_i. La contribution $b'_i(n)$ de chaque source de bruit $b_i(n)$ dépend des échantillons précédents $b_i(n-k)$ de cette source de bruit avec $k \in 1, \cdots, N_i$. Si le système S_i est récursif au sens traitement du signal du terme, le bruit $b'_i(n)$ dépend de ses échantillons précédents $b'_i(n-m)$ avec $m \in [1 : D_i]$. Ainsi, la contribution du bruit $b'_i(n)$ en sortie du système peut être définie à l'aide de l'équation récurrente suivante :

$$b'_i(n) = \sum_{k=0}^{N_i} g_i(k) b_i(n-k) + \sum_{m=1}^{D_i} f_i(m) b'_i(n-m), \tag{3.6}$$

avec $g_i(k)$ la contribution en sortie du système de la source de bruit b_i à l'instant $(n-k)$ et $f_i(m)$ la contribution de b'_i à l'instant $(n-m)$. Ces termes f_i et g_i peuvent varier au cours du temps et dépendent du système traversé. Pour les systèmes linéaires invariants dans le temps (LTI), ces termes sont constants.

Notre approche d'évaluation analytique de la précision repose sur la notion de réponse impulsionnelle. L'objectif est d'exprimer la contribution b'_i uniquement en fonction de la source de bruit b_i. En déroulant l'équation récurrente 3.6, la contribution du bruit $b'_i(n)$ peut être exprimée à l'aide de l'expression suivante :

$$b'_i(n) = \sum_{k=0}^{n} h_i(k,n) b_i(n-k). \tag{3.7}$$

avec $h_i(k,n)$ la réponse impulsionnelle du système S_i à l'instant n. Le terme $h_i(k,n)$ représente la contribution de la source de bruit b_i à l'instant $n-k$ pour générer le bruit b'_i à l'instant n. Pour les systèmes non LTI, cette réponse impulsionnelle varie au cours du temps.

Pour simplifier les notations, la sortie du système S_i est toujours considérée à l'instant n, ainsi, la réponse impulsionnelle $h_i(k,n)$ est notée $h_i(k)$. Ce terme est calculé à partir des termes f_i et g_i à l'aide de l'équation récurrente suivante :

$$h_i(k) = \sum_{j=1}^{D_i} f_i(j) h_i(k-j) + g_i(k). \tag{3.8}$$

Puissance du bruit de quantification La puissance de bruit de quantification P_b en sortie du système est obtenue à partir du moment du second ordre du bruit de quantification b_y en sortie du système :

$$P_b = E[b_y^2] = \sum_{i=1}^{Ne}\sum_{j=1}^{Ne}\sum_{k=0}^{n}\sum_{m=0}^{n} E\big[h_i(k)h_j(m)b_i(n-k)b_j(n-m)\big]. \tag{3.9}$$

Les termes b_i et b_j représentent des sources de bruit de quantification. Ainsi, d'après la modélisation de Widrow [82], ces termes sont non corrélés avec les termes représentant les valeurs exactes (signal). L'expression de la puissance du bruit est la suivante :

$$P_b = \sum_{i=1}^{Ne}\sum_{j=1}^{Ne}\sum_{k=0}^{n}\sum_{m=0}^{n} E\big[h_i(k)h_j(m)b_i(n-k)b_j(n-m)\big], \tag{3.10}$$

$$P_b = \sum_{i=1}^{Ne} K_i\left(\sigma_{b_i}^2 + \mu_{b_i}^2\right) + 2\sum_{i=1}^{Ne-1}\sum_{j=i+1}^{Ne} L_{ij}\cdot E[b_i b_j]. \tag{3.11}$$

avec $\sigma_{b_i}^2$ la variance de chaque source de bruit b_i et $E[b_i b_j]$ l'intercorrélation entre les sources de bruit b_i et b_j. Les termes L_i et K_{ij} sont définis de la manière suivante :

$$K_i = \sum_{k=0}^{n} E\big[h_i^2(k)\big], \qquad L_{ij} = \sum_{k=0}^{n}\sum_{m=0}^{n} E\big[h_i(k)h_j(m)\big]. \tag{3.12}$$

Le terme K_i représente le gain sur la variance associé à la source de bruit b_i. Le terme L_{ij} représente le gain sur la moyenne pour le couple de sources de bruit b_i et b_j. Au sein de l'expression de la puissance du bruit, ces termes correspondent à des constantes et intègrent uniquement des termes représentant les valeurs exactes (signal). Les valeurs exactes des variables sont approximées par les valeurs obtenues lors d'une simulation utilisant l'arithmétique en virgule flottante.

Les paramètres statistiques μ_i, σ_i et $E[b_i b_j]$ de chaque source de bruit dépendent des largeurs des données testées. Ces largeurs correspondent aux variables de l'expression analytique de la puissance du bruit en sortie du système.

Réduction de la complexité de calcul des coefficients K_i et L_{ij} Les expressions des termes K_i et L_{ij} sont composées de sommes allant de 0 jusqu'à n. Ainsi, il est nécessaire de fixer le nombre d'éléments N_{sum} intégrés dans le calcul de la somme. Le choix de la valeur de N_{sum} est un compromis entre la qualité de l'estimation et le temps nécessaire à calculer cette somme. Pour diminuer la complexité du calcul de la somme sur N_{sum} éléments, une approche basée sur la prédiction linéaire est utilisée. Les termes de la réponse impulsionnelle h_i sont liés entre eux à travers une équation récurrente correspondant à l'expression 3.8.

La réponse impulsionnelle variant dans le temps, la relation entre les éléments de la réponse impulsionnelle est non linéaire. L'objectif est de linéariser cette expression avec les coefficients de prédiction λ_i minimisant l'erreur quadratique moyenne entre la réponse impulsionnelle $\tilde{h}_i$ estimée avec les coefficients de prédiction et la réponse impulsionnelle réelle. Le calcul des coefficients λ_i est présenté dans [71]. A l'aide des coefficients λ_i, la somme est réécrite de la manière suivante :

$$\sum_{k=0}^{n\to\infty} h_i(k) = \sum_{j=0}^{D_i-1} \cdot \frac{\left(\sum_{k=1}^{D_i-j-1}\lambda_{i_k} - 1\right)}{\sum_{m=1}^{D_i}\lambda_{i_m} - 1} h_i(n - N_i + 1 - j) + \sum_{k=0}^{N_i-2} h_i(n-k). \tag{3.13}$$

Dans le cas d'un système LTI ayant une réponse impulsionnelle infinie, les coefficients de prédiction sont égaux aux coefficients du dénominateur de la fonction de transfert du système.

Conclusion L'approche présentée dans cette partie permet de déterminer l'expression analytique de la puissance du bruit de quantification en sortie de tous types de systèmes composés d'opérations dont la sortie est une fonction dérivable de ses entrées. L'utilisation de la notion de réponse impulsionnelle permet de traiter les systèmes récursifs. Dans [Rocher 12] [71], l'expression analytique de la puissance du bruit de sortie et la démarche suivie sont présentées pour le cas du filtre adaptatif LMS [5], et différentes applications non linéaires telles que les filtres de Volterra, ou la normalisation de vecteurs. De même, l'intérêt d'une modélisation matricielle a été montré à travers une FFT [6] permettant ainsi, d'avoir des expressions compactes et indépendantes de la taille de la FFT.

3.2.2.2 Structures conditionnelles

Dans le cadre de la thèse de Jean-Charles Naud, l'évaluation de la précision dans le cas de structures conditionnelles a été étudiée. Ces structures conditionnelles sont issues des expressions *if-else* ou *switch* présentes au sein du code C décrivant l'application. Pour ce type de structure de contrôle, l'alternative sélectionnée, c.-à-d. la trace d'exécution du code, dépend de la valeur de la condition. Ces structures génèrent deux types d'erreur. Le premier type concerne les bruits de quantification affectés par un traitement alternatif. La propagation du bruit de quantification à travers une structure conditionnelle dépend de l'alternative sélectionnée. Ce cas est traité dans la suite de cette partie. Le second type concerne les bruits de quantification affectant la condition de la structure conditionnelle. La présence d'un bruit de quantification au niveau de la condition peut engendrer une différence entre la trace d'exécution en précision infinie et en précision finie. Les traitements étant différents au sein de chaque alternative, l'erreur entre la précision finie et la précision infinie peut être grande. En conséquence, les hypothèses associées à la théorie de la perturbation ne sont plus respectées et cette erreur ne peut être traitées comme un bruit de quantification. Ces aspects sont traités dans la partie 3.3.2.

Modélisation du nœud φ Les nœuds φ permettent de représenter la convergence de variables y_j issues de différentes alternatives d'une structure conditionnelle. Les nœuds sont introduits lors du passage en représentation SSA [7] du graphe représentant l'application. Ils permettent de représenter l'assignation d'un variable y_j, calculée au sein de l'alternative j de la structure conditionnelle, à la variable y, située à la sortie de la structure conditionnelle [Naud 11a].

Les nœuds φ présents au sein du graphe flot de signal représentant l'application permettent de modéliser la convergence des bruits issus de différentes alternatives des structures conditionnelles. Au niveau des variables aléatoires, un nœud φ correspond à un mélangeur de population utilisant une probabilité α_i associée à chaque alternative. Celle-ci dépend de la condition associée à la structure conditionnelle.

Soit un nœud φ à N_p entrées y_j et une sortie y. La sortie y prend la valeur de y_j si $c \in E_j$. Soit α_j, la probabilité que la sortie y prenne la valeur de y_j c.-à-d. $c \in E_j$. L'obtention de la moyenne et de la variance de la variable aléatoire y en sortie du nœud φ est réalisée à l'aide de la fonction de répartition d'un mélangeur de population décrite à l'équation 3.14.

$$\int_{-\infty}^{y} f_Y(t)dt = \sum_{j=1}^{N_p} \alpha_j \int_{-\infty}^{y} f_{Y_j}(t)dt, \tag{3.14}$$

avec f_Y et f_{Y_j}, les DDP de y et y_j.

Le modèle générique de propagation du bruit au sein d'un système intégrant des structures conditionnelles est présenté à la figure 3.3. Ce modèle est composé de différentes parties correspondant à la génération de la source de bruit, à la propagation de celle-ci au sein des différentes alternatives et à la combinaison des différentes sources de bruit.

5. *Least Mean Square.*
6. *Fast Fourier Transform.*
7. *Static Single Assignment.*

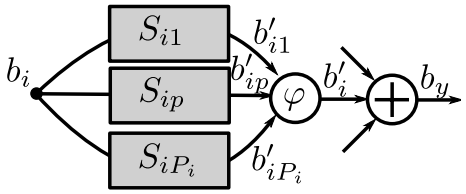


FIGURE 3.3 – Modèle générique de propagation du bruit au sein d'un système intégrant des structures conditionnelles.

La première partie du modèle concerne la présence de la source de bruit au sein d'une alternative d'une ou plusieurs structures conditionnelles imbriquées. Ainsi, le processus de quantification entraînant la génération de la source de bruit n'est pas présent en permanence car l'alternative concernée est n'exécutée que si la condition associée est vraie. La source de bruit b_i contribue à la présence d'un bruit en sortie du système avec une probabilité d'occurrence égale à α'_i.

La seconde partie du modèle concerne la transmission du bruit b_i au sein des différentes alternatives puis la combinaison en sortie de ces différentes contributions. Ce modèle nécessite de déterminer auparavant tous les chemins (traces d'exécution) pouvant être empruntés par la source de bruit b_i. Ce processus est réalisé à travers un ensemble de transformations de graphes détaillé dans [Naud 11a] et permettant d'avoir pour chaque source de bruit un unique nœud φ situé en sortie du système. L'expression du bruit b'_{ip}, correspondant à la contribution en sortie de la source de bruit b_i lorsque le chemin p est emprunté est présentée à l'équation 3.15. Elle est égale au produit de convolution entre b_i et $h_{ip}(n)$, la réponse impulsionnelle du système S_{ip} associée au chemin p.

$$b'_{ip}(n) = b_i(n) * h_{ip}(n). \tag{3.15}$$

Soit α_{ip} la probabilité que le bruit b_i suive le chemin p. La probabilité d'occurrence de la source b_i étant égale à α'_i, la relation suivante est obtenue pour les P_i chemins pouvant être empruntés par la source b_i :

$$\sum_{p=1}^{P_i} \alpha_{ip} = \alpha'_i. \tag{3.16}$$

En présence de structures conditionnelles, l'expression de la puissance du bruit en sortie est la suivante :

$$P_{b_y} = \sum_{i=1}^{N_e} \left(\sigma_{b_i}^2 + \mu_{b_i}^2\right) \cdot K_i + 2 \sum_{i=1}^{N_e-1} \sum_{j=i+1}^{N_e} L_{ij} \cdot E[b_i b_j], \tag{3.17}$$

avec

$$K_i = \sum_{p=1}^{P_i} \alpha_{ip} \sum_{k=0}^{n} E\left[h_{ip}^2(k)\right] \qquad \text{et} \qquad L_{ij} = \sum_{p=1}^{P_i} \sum_{q=1}^{P_j} \alpha_{ijpq} \sum_{k=0}^{n} \sum_{m=0}^{n} E\left[h_{ip}(k) h_{jq}(m)\right]. \tag{3.18}$$

Les termes K_i et L_{ij} sont constants et sont déterminés exclusivement à partir de la réponse impulsionnelle des systèmes h_{ip} et des probabilités α_{ip} et α_{ijpq}. Le terme α_{ijpq} correspond à la probabilité que le bruit b_i suive le chemin p et que le bruit b_j suive le chemin q. Par rapport à l'équation 3.10, les termes représentant les probabilités associées à chaque chemin sont introduits. Ces termes sont obtenus par simulation à partir d'une phase de *profiling* sur un jeu de test représentatif.

3.2.2.3 Qualité de l'estimateur

Différentes expérimentations ont été conduites pour évaluer la qualité de notre estimateur de la puissance du bruit en sortie d'un système composé d'opérations dont le modèle de bruit est linéaire. La référence $P_{b_y}^{sim}$ est obtenue à partir d'une approche basée sur la simulation. L'application est simulée d'un côté avec des types en virgule fixe dont les paramètres dépendent de la spécification virgule fixe testée et de l'autre côté avec des types en virgule flottante. La virgule flottante correspond à une approximation de la valeur exacte. L'erreur relative E_r de notre estimation P_{b_y} par rapport à la valeur de référence $P_{b_y}^{sim}$ est mesurée. Pour une application donnée, différentes erreurs relatives correspondant à différentes spécifications en virgule fixe sont mesurées. Nous avons reporté dans le tableau 3.1, l'erreur relative moyenne (E_r^{mean}) et maximale (E_r^{max}) obtenues pour les différentes applications [66] (filtre FIR [8], filtres Polyphases, DCT [9], FFT, filtre IIR [10], polynôme, filtre de Volterra, corrélateur). Les résultats sont fournis pour différentes applications LTI non récursives et récursives et non LTI non récursives. L'erreur relative entre notre estimation et la référence est faible. L'erreur relative moyenne est inférieure à 10% pour les différentes applications considérées. Même lorsque le nombre de sources de bruit est important comme dans le cas d'une FFT sur 256 points, l'erreur relative reste faible. L'ordre de grandeur des erreurs relatives obtenues est suffisant pour la conception de systèmes en virgule fixe. En effet, un écart d'un bit entre deux processus de quantification se traduit par un rapport de quatre entre les puissances des bruits de quantification associés à chaque processus.

Applications	Caractéristiques	E_r^{mean} (%)	E_r^{max} (%)
Filtre FIR 32 (Arrondi)	Non Réc., LTI	2.5	7.9
Filtre FIR 32 (Troncature)	Non Réc., LTI	1.1	2.4
Filtres Polyphases	Non Réc., LTI	5.2	20.5
DCT 8	Non Réc., LTI	8.4	24.8
FFT 128	Non Réc., LTI	3.6	7.3
FFT 256	Non Réc., LTI	4.5	7.8
IIR 8	Réc., LTI	1.6	3.3
Polynôme (2^d degré)	Non Réc., Non LTI	0.6	0.8
Filtre de Volterra	Non Réc., Non LTI	1.7	3.2
Corrélateur	Non Réc., Non LTI	1.3	5.7

TABLE 3.1 – Erreurs relatives moyennes et maximales obtenues pour différentes applications.

Les résultats obtenus dans le cas de différents types de filtres adaptatifs sont présentés dans le tableau 3.2. L'algorithme NLMS (Normalized Least Mean Square) réalise une normalisation du vecteur d'entrée en amont de la structure LMS (Least Mean Square). L'algorithme APA (Affine Projection Algorithms) utilise en entrée une matrice regroupant les dernières observations du signal au lieu d'un vecteur dans le cas des algorithmes LMS et NLMS. Au niveau du bruit, ces trois applications conduisent à des systèmes non LTI et récursifs. Ainsi, les expressions des termes K_i et L_{ij} (3.12) contiennent des sommes infinies. Pour l'approche de calcul direct des sommes (CDS), celles-ci sont tronquées et calculées sur N_{sum} éléments. L'alternative proposée pour le calcul de cette somme correspond à la méthode de prédiction linéaire (PL). Les résultats montrent l'influence du nombre d'éléments N_{sum} utilisés pour approcher les sommes infinies. L'utilisation de 1000 éléments pour l'approximation des sommes permet d'obtenir une erreur relative relativement faible. La détermination du nombre de termes N_{sum} résulte d'un compromis entre la précision de l'estimation et le temps d'exécution de l'estimation. La méthode de prédiction linéaire permet l'approximation du calcul des sommes et ainsi de réduire le temps de calcul. En contrepartie l'erreur relative est un peu plus élevée et se situe en moyenne entre 20% et 25% mais reste tout a fait acceptable.

8. *Finite Impulse Response.*
9. *Discrete Cosine Transform.*
10. *Infinite Impulse Response.*

Nous avons comparé ces résultats avec les méthodes d'évaluation de la précision proposées récemment dans [60], [12]. Les erreurs relatives mesurées sont relativement proches. Cependant, pour l'application LMS, la méthode proposée dans [12] permet d'obtenir une estimation de très bonne qualité mais au détriment du temps d'exécution qui devient très élevé (voir partie 3.2.3.6). Pour les différentes techniques hybrides [76, 21, 38], la qualité de l'estimateur n'a pas été réellement mesurée.

Applications	CDS (%) avec $N_{sum} = 500$	CDS (%) avec $N_{sum} = 1000$	PL (%)	Nombre de sources de bruit
LMS 32	14.4	3.7	22.6	66
NLMS 32	14.2	4.2	20.6	67
APA 32x12	17.6	6.5	25.4	812

TABLE 3.2 – Erreur relative obtenue avec notre approche directe ou par prédiction linéaire pour différents filtres adaptatifs.

3.2.3 Outil d'évaluation de la précision : ID.Fix-AccEval

Une infrastructure logicielle a été développée pour évaluer analytiquement la précision des systèmes en virgule fixe. La majeure partie des développements associés à cet outil, a été réalisée dans le cadre du projet S2S4HLS présenté dans la partie 4.5. Cet outil s'appuie sur les techniques présentées dans la partie 3.2.2. Il permet de traiter les systèmes à base d'opérations dont le modèle de bruit est linéaire. Les structures conditionnelles sont supportées en prenant en compte la probabilité de passage dans chaque alternative de celles-ci. De plus, la corrélation entre les sources de bruit issues de la quantification d'un même signal est prise en compte.

Le synoptique de cet outil est présenté à la figure 3.4. Le cœur de l'outil, dénommé ID.Fix-AccEval, a été développé en C++ pour des raisons historiques mais aussi de performance. Ainsi, ce module est indépendant de l'infrastructure de conversion en virgule fixe ID.Fix présentée dans la partie 4.5. L'interface entre les deux outils est réalisée à l'aide de fichiers XML [11]. L'entrée du module ID.Fix-AccEval est le graphe flot de signal (SFG) associé à l'application. L'obtention de ce SFG à partir de la représentation intermédiaire utilisée dans l'outil de conversion en virgule fixe et correspondant à un graphe flot de données et de contrôle (CDFG) est réalisée à l'aide du module *Génération SFG*.

Notre modèle d'évaluation de la précision nécessite de connaître, pour la gestion des structures conditionnelles, les probabilités associées à chaque trace d'exécution et pour les systèmes non LTI, les valeurs des variables prises au cours du temps. Ces informations sont obtenues à travers une simulation de l'application sur un jeu de test représentatif. Ces simulations sont réalisées à partir d'un code C utilisant des types flottants. Ce code est instrumenté afin de pouvoir récupérer les informations nécessaires. Le module *Génération Simulateur* génère le code C instrumenté, compile celui-ci et le module *Simulation* exécute ce code C et récupère les informations nécessaires.

L'objectif de cet outil d'évaluation de la précision est de générer le code source de la fonction permettant de déterminer la puissance du bruit P_b en fonction de la largeur des données et des modes de quantification. Plus particulièrement, les entrées de cette fonction P_{b_y} sont le vecteur $\mathbf{w_{FP}}$ de taille N_o correspondant à la largeur de la partie fractionnaire des opérandes des N_o opérations présentes dans le SFG et le vecteur $\mathbf{q}$ de taille N_o représentant les modes de quantification utilisés.

L'expression analytique de la puissance du bruit est obtenue à partir de trois transformations majeures du SFG présentées dans la suite du document. Tout d'abord, l'application est représentée au niveau bruit. Ensuite, les pseudo-fonctions de transfert régissant l'application sont déterminées. Finalement, l'expression de la puissance du bruit présentée à l'équation 3.10 est générée sous la forme d'un code C après avoir calculé

11. *Extensible Markup Language.*

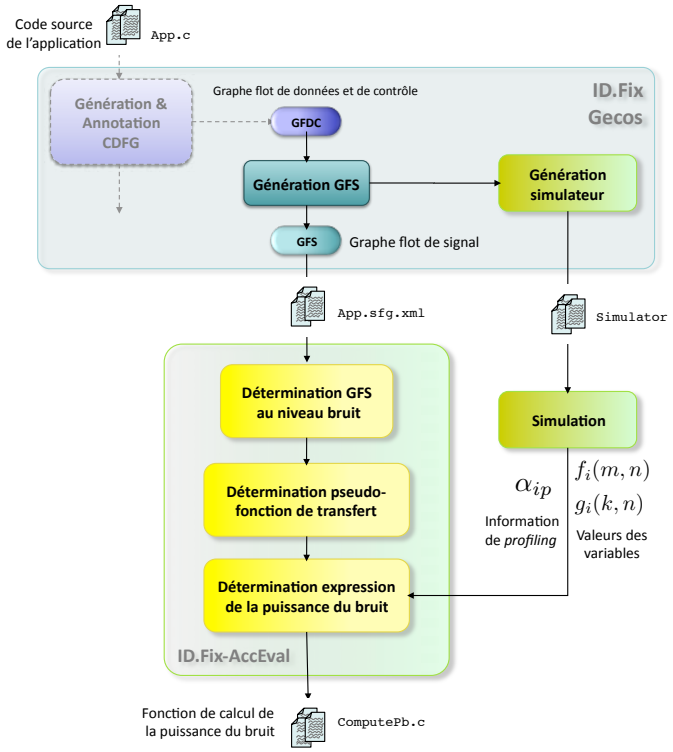


FIGURE 3.4 – Synoptique de l'outil d'évaluation de la précision.

les gains sur la moyenne et la variance dont les expressions sont fournies à l'équation 3.12. La technique utilisée pour traiter les systèmes LTI est décrite dans [Ménard 08a].

3.2.3.1 Génération du graphe flot de signal

La méthode d'évaluation de la précision présentée dans la partie 3.2.2 nécessite actuellement d'avoir un unique graphe représentant l'ensemble des traitements réalisés dans l'application pour déterminer les expressions de la puissance du bruit des sorties de cette application. Notre approche repose sur la détermination des expressions du système, obtenues par un parcours du graphe. En conséquence, les différentes structures de contrôle présentes dans la représentation intermédiaire de type CDFG doivent être éliminées. La représentation utilisée par l'outil est un graphe flot de signal (SFG) G_s.

Les structures répétitives de type `for`, `while`, ou `do ... while` sont déroulées afin d'obtenir un graphe unique associé à la structure répétitive. Le cœur de la structure répétitive est dupliqué pour chaque itération. Ce déroulage complet des structures répétitives nécessite que les itérations réalisées par cette structure puissent être déterminées statiquement. En conséquence, les blocs de contrôle associés à la structure répétitive (gestion de l'indice de boucle et test de sortie de boucle) ne doivent faire intervenir que des variables de contrôle évaluables statiquement.

Les structures conditionnelles de type *if...else* et *switch* sont mises à plat en juxtaposant dans le SFG les graphes associés à chaque alternative. Chaque alternative est représentée par un sous-graphe au sein du SFG. Des nœuds φ sont insérés afin de réaliser la convergence des variables affectées dans les différentes alternatives. Le bloc associé à la condition n'est pas conservé dans le SFG car son contenu n'est pas utile pour l'évaluation du bruit. Les informations nécessaires pour la prise en compte des structures conditionnelles correspondent juste aux probabilités associées à chaque trace d'exécution. Cette information est obtenue par la phase de *profiling*.

Le SFG permet de spécifier les relations temporelles entre les données à travers la présence d'opérations de retard. Une opération de retard (symbole z^{-1}) est présente entre les variables x_0 et x_1 ($x_1 = z^{-1}(x_0)$) indique que la valeur de la variable x_1 à l'instant n correspond à la valeur de x_0 à l'instant $n-1$, c.-à-d. à la période d'échantillonnage précédente. Ces relations temporelles entre les données sont implantées en utilisant une structure de type FIFO [12] au sein de la mémoire. Les opérations de retard sont présentes lors du *vieillissement* des données au sein de la FIFO. Pour faciliter la détection de ces opérations de retard, un pragma (DELAY) est disponible pour spécifier cette fonctionnalité dans le code C de l'application.

3.2.3.2 Génération et simulation du code C instrumenté

L'objectif des modules *Génération Simulateur* et *Simulation* est de fournir la probabilité associée à chaque trace d'exécution du programme et les valeurs prises par chaque variable du programme. Ces deux éléments sont obtenus à travers une simulation de l'application sur un jeu de test représentatif. Cette simulation est réalisée à partir d'un code C instrumenté qui est généré par l'outil puis compilé et exécuté.

Le code C généré correspond à celui représentant les traitements réalisés dans le SFG afin d'avoir une concordance entre chaque sous-graphe du SFG et chaque bloc de contrôle du code C. En conséquence, les structures répétitives sont déroulées et les structures conditionnelles sont conservées. La trace d'exécution est implantée à travers une liste d'entier. Chaque élément de cette liste correspond au numéro du bloc composite exécuté. Au début de chaque bloc composite, un code est ajouté permettant d'insérer dans la trace le numéro du bloc exécuté.

3.2.3.3 Génération du SFG au niveau bruit

L'objectif de la première transformation (T_1) est d'obtenir le graphe G_{sn} représentant l'application au niveau bruit de quantification à partir du graphe flot de signal G_s. Ce graphe G_{sn} regroupe l'ensemble des sources de bruit de quantification et décrit la propagation des termes correspondant au signal et ceux correspondant au bruit de quantification. Cette transformation nécessite d'insérer toutes les sources de bruit potentielles et le modèle de bruit des opérations.

T_{11} : Insertion des sources de bruit Afin de prendre en compte toutes les combinaisons de largeur des opérations, une source de bruit potentiel est insérée en entrée et en sortie de chaque opération. Afin de limiter les traitements réalisés par la suite au sein de l'outil, le nombre de sources de bruit présentes au sein de G_{sn} est réduit en regroupant les sources de bruit. Ces sources sont propagées à travers les opérations d'addition et de soustraction en direction des sorties. Pour chaque source de bruit, les expressions des paramètres statistiques associés sont générées, sous la forme d'un code C, au sein du fichier de sortie. Ces expressions font intervenir la largeur de la partie fractionnaire des opérandes ($\mathbf{w_{FP}}$) et les modes de quantification utilisés ($\mathbf{q}$).

T_{12} : Insertion du modèle de bruit des opérations Dans cette transformation, chaque nœud du graphe représentant une opération est remplacé par son modèle de bruit issu du développement en série de Taylor à l'ordre 1 (équation 3.3).

3.2.3.4 Génération du graphe de pseudo-fonctions de transfert

L'objectif de cette transformation est de déterminer le graphe G_H des pseudo-fonctions de transfert spécifiant le système. Le graphe $G_H = (V_H, E_H)$ est un graphe acyclique orienté. Chaque arc est annoté par une pseudo-fonction de transfert. L'arc dirigé $(b'_i, b_i, H_{b'_i b_i}(z))$ du nœud b_i vers le nœud b'_i est annoté avec la pseudo-fonction de transfert $H_{b'_i b_i}(z)$ entre les variables $B'_i(z)$ et $B_i(z)$ respectivement associées avec l'extrémité b'_i et l'origine b_i de cet arc. La notion de pseudo-fonction de transfert (PFT) reprend la notion de fonction transfert pour pouvoir spécifier des équations récurrentes dont les coefficients varient au cours du temps. Les outils mathématiques tels que la transformée en $\mathcal{Z}$ ne peuvent pas être utilisés pour les PFT. En

12. *First In First Out.*

repartant de l'équation récurrente de b'_i présentée à l'équation 3.6, la pseudo-fonction de transfert $H_{b'_i b_i}(z)$ est définie de la manière suivante :

$$H_{b'_i b_i}(z) = \frac{\sum_{k=0}^{N_i} g_i(k) z^{-k}}{1 - \sum_{m=1}^{D_i} f_i(m) z^{-m}}. \tag{3.19}$$

T_{21} **: Détection et démantèlement des circuits** Le but de cette première étape est de transformer le graphe $G_{sn} = (V_{sn}, E_{sn})$ en plusieurs graphes acycliques G_k si G_{sn} contient des circuits. Ceci est réalisé en énumérant puis en démantelant les différents circuits présents au sein du graphe.

T_{22} **: Détermination du graphe des équations récurrentes** Le graphe $G_{eq} = (N_{eq}, E_{eq})$ est un graphe orienté et annoté spécifiant l'application par un ensemble d'équations récurrentes. Les nœuds de ce graphe correspondent à certaines variables de l'application. L'arc (b_i, y, f_{yb_i}) annoté par f_{yb_i} et orienté de b_i vers y signifie que la variable y est définie à partir de la variable b_i à l'aide de l'équation récurrente f_{yb_i}. Pour chaque DAG, les équations récurrentes associées à celui-ci sont obtenues par un parcours en profondeur de ce graphe.

T_{23} **: Détermination du graphe des pseudo-fonctions de transfert partielles** Le graphe $G_{Hi} = (N_{G_{Hi}}, E_{G_{Hi}})$ est un graphe orienté et annoté spécifiant l'algorithme par un ensemble de pseudo-fonctions de transfert intermédiaires. Les nœuds du graphe G_{Hi} appartiennent à l'ensemble N_{eq}. L'arc (b_i, y, f_{yb_i}) orienté de b_i vers y est annoté par la pseudo-fonction de transfert H_{yb_i} entre les variables associées aux nœuds y et b_i.

La transformation du graphe G_{eq} en un graphe G_{Hi} nécessite d'obtenir un graphe d'équations récurrentes avec uniquement des circuits unitaires. Un circuit de longueur unitaire associé à une variable $y(n)$ signifie que la variable est définie à partir de ses versions précédentes $y(n-k)$ avec $k > 0$. Cette transformation implique de détecter et de démanteler au sein du graphe G_{eq}, tous les circuits d'une longueur supérieure à l'unité. Les circuits sont démantelés en réalisant un ensemble de substitutions de variables au sein des équations récurrentes associées aux circuits. Des règles ont été définies afin de déterminer les nœuds sur lesquels débuter la substitution de variables.

Le modèle utilisé pour supporter les structures conditionnelles et présenté à la figure 3.3, nécessite la présence d'un unique nœud φ juste en amont du nœud de sortie. Lorsque plusieurs nœuds φ sont présents entre la source de bruit et la sortie, différentes transformations de graphes sont appliquées afin de déplacer vers la sortie et fusionner ces nœuds φ et ainsi obtenir le modèle présenté à la figure 3.3. Ceci permet d'exhiber toutes les traces d'exécutions (chemins) entre la source de bruit et la sortie.

T_{24} **: Détermination du graphe des pseudo-fonctions de transfert globales** Le graphe $G_H = (N_{G_H}, E_{G_H})$ est un graphe annoté spécifiant le système avec un ensemble de fonctions de transfert globales entre les sorties et chacune des sources de bruit. Chaque arc orienté est annoté par la pseudo-fonction de transfert entre la sortie et la source de bruit considérée. L'objectif de cette transformation T_{24} est de calculer les pseudo-fonctions de transfert globales entre la sortie et chaque source en éliminant les nœuds représentant les variables intermédiaires. Cette transformation est réalisée actuellement à l'aide de l'outil Matlab. Les opérations d'addition, de soustraction et de multiplication de PFT ont été redéfinies pour traiter les systèmes non LTI.

3.2.3.5 Génération de la fonction d'évaluation de la puissance du bruit.

Pour chaque pseudo-fonction de transfert globale, la réponse impulsionnelle associée est déterminée de manière récursive à l'aide des expressions présentées dans [71]. La méthode de prédiction linéaire est utilisée pour les systèmes non LTI récursifs. Ensuite, les gains K_i et L_{ij} sont calculés à partir de la réponse

impulsionnelle et des probabilités de chaque trace d'exécution dans le cas de structures conditionnelles. Ces valeurs sont incluses dans le fichier de sortie et le reste du code C nécessaire pour calculer la puissance du bruit de quantification est inséré dans ce fichier.

3.2.3.6 Temps d'exécution de l'outil.

Le temps t_{obt} d'obtention du code décrivant l'expression analytique de la puissance du bruit de quantification a été mesuré sur différents benchmarks. Les résultats sont présentés dans le tableau 3.3. Pour une même complexité de graphe, le temps d'exécution des systèmes récursifs est plus important. Pour les systèmes récursifs une part importante du temps est consacrée à la transformation T_{21} correspondant à la gestion des circuits au sein du graphe et à la transformation T_3 correspondant au calcul des réponses impulsionnelles entre la sortie et chaque source de bruit. Pour la FFT, le temps t_{obt} est élevé, car l'expression de la puissance du bruit est évaluée pour chaque sortie.

Applications	Caractéristiques	t_{obt}
FIR 64	Non Réc., LTI	1,7
DCT 8	Non Réc., LTI	0,2
FFT 32	Non Réc., LTI	120,1
IIR 8	Réc., LTI	17,2
Filtre de Volterra	Non Réc., Non LTI	2,8

TABLE 3.3 – Temps d'obtention de l'expression analytique t_{obt} (s) obtenus pour différentes applications.

Notre approche d'évaluation de la précision est comparée aux méthodes existantes au sein de la littérature. Les techniques hybrides [76, 21, 38] déterminent les gains sur la moyenne (K_i) et la variance (L_{ij}) à partir d'un ensemble de simulations. Aucun résultat sur les temps d'exécution n'est fourni mais, il est possible d'estimer celui-ci. Pour une application avec N_e sources de bruit, $N_e(N_e + 1)$ simulations en virgule fixe sont nécessaires. Pour l'exemple d'un filtre adaptatif LMS composé de 32 cellules, présenté dans [76], 66 sources de bruit sont considérées. Ceci nécessite 4422 simulations en virgule fixe. Le temps nécessaire pour réaliser l'ensemble de ces simulations en virgule fixe représente environ 20 fois le temps t_{obt} obtenu avec notre approche basée sur le calcul direct des sommes (CDS) et 200 fois le temps t_{obt} obtenu avec notre approche basée sur la prédiction linéaire (PL). Les méthodes hybrides nécessitent des temps d'obtention de l'expression analytique non négligeables lorsque le nombre de sources de bruit devient élevé.

Notre approche est comparée avec celle proposée dans [12] et utilisant une simulation basée sur l'arithmétique affine pour déterminer les coefficients K_i et L_{ij}. Les résultats sont présentés dans le tableau 3.4. Le temps d'obtention t_{obt} mais aussi la qualité de l'estimation (erreur relative E_r) sont fournis. Notre approche d'évaluation de la précision conduit à des temps d'obtention de l'expression analytique faibles mais légèrement plus élevés que ceux obtenus dans [12] pour les systèmes non récursifs. Pour les systèmes récursifs,

Applications	Méthode CDS		Méthode PL		Méthode [12]	
	t_{obt}	E_r (%)	t_{obt}	E_r (%)	t_{obt}	E_r (%)
IIR 2	0.15	1.6	0.08	0.5	0.88	0.7
IDCT 8	0.04	0.6	0.04	0.6	0.01	0.9
LMS 5	0.13	2.8	0.04	10.6	1646	1.1

TABLE 3.4 – Comparaison de nos approches avec celle proposée dans [12] en termes de temps d'obtention de l'expression analytique t_{obt} (s) et d'erreur relative E_r sur l'estimation de la puissance du bruit. Pour la détermination des coefficients K_i et L_{ij}, la méthode CDS réalise le calcul direct des sommes et la méthode PL utilise la prédiction linéaire.

la méthode proposée par [12] conduit à des temps t_{obt} très élevés. En effet, pour la simulation basée sur l'arithmétique affine, de très nombreuses itérations peuvent être nécessaires pour converger vers des valeurs stables. Ce temps de convergence dépend de la longueur des réponses impulsionnelles entre les sources de bruit et la sortie et peut devenir important pour des systèmes dont la réponse impulsionnelle est longue. Notre approche basée sur le calcul direct des sommes (CDS) conduit une qualité proche de celle proposée dans [12] mais avec des temps t_{obt} nettement plus faibles. Notre approche basée sur la prédiction linéaire (PL) conduit à des estimations moins précises mais permet de réduire un peu plus le temps t_{obt}.

3.3 Évaluation des performances

Encadrement :	Romain Serizel, Master 2006 [75]
	Karthick Parashar doctorant depuis 2008
Conférences :	DASIP 2007 [Ménard 07], Asilomar 2010 [Parashar 10e], Eusipco 2010 [Parashar 10a]
	ICASSP 2010 [Parashar 10d], ICCAD 2010 [Parashar 10b],
Revues :	Journal of Embedded Systems 2008 [Ménard 08b]
Collaboration :	IMEC, programme R&D nano 2012 (ST Microelectronics)

Dans cette partie, la technique d'évaluation des performances basée sur une approche mixte combinant simulation et résultats analytiques est présentée dans la partie 3.3.2. Cette approche mixte se base sur le modèle de source de bruit unique présenté dans la partie 3.3.1.

3.3.1 Modèle de source de bruit unique

Ces travaux de recherche sur le modèle de source de bruit unique (SBU) ont été initiés dans le cadre du stage de Master de Romain Serizel et poursuivis dans le cadre de la thèse de Karthick Parashar. L'objectif est de modéliser le comportement du système $\widehat{S}$ en virgule fixe par le système S en précision infinie et une unique source de bruit b_{u_s} située en sortie du système S. Dans un premier temps, ce modèle SBU a été défini pour pouvoir déterminer par simulation la contrainte de précision permettant de fournir un certain niveau de performance [Ménard 08b, Ménard 07]. L'objectif est de pouvoir prédire les performances d'un système pour un niveau de bruit de quantification donné. Pour déterminer la contrainte de précision, la puissance de la source de bruit est augmentée progressivement tant que les performances de l'application sont respectées. Cette approche est détaillée dans [Ménard 08b].

Dans un second temps, ce modèle SBU a été utilisé pour évaluer les performances d'un système composé de plusieurs blocs. Ce modèle s'intègre à notre approche de conception au niveau système présentée dans la partie 4.4. Ce modèle est un des éléments de base de l'approche mixte [Parashar 10b] présentée dans la partie 3.3.2 et permettant de supporter les opérations de décision. Dans ce cas, la largeur des différentes données est connue et ainsi les paramètres de chaque source de bruit de quantification b_{g_i} sont disponibles pour calculer les caractéristiques du bruit en sortie.

La source de bruit b_{u_s} ajoutée en sortie du système S doit avoir, d'un point de vue statistique, un comportement similaire au bruit b_{y_s} présent en sortie du système $\widehat{S}$ en virgule fixe. Cette source b_{u_s} doit avoir une densité de probabilité et un fonction d'autocorrélation correspondant à celles de b_{y_s}.

3.3.1.1 Densité de probabilité

Nous avons tout d'abord proposé, dans [Ménard 08b, Ménard 07], de modéliser la source de bruit b_{u_s} par la somme pondérée d'une source de bruit gaussienne b_{norm} et d'une source de bruit uniforme b_{uni}. La source de bruit b_{uni} permet de modéliser la présence au sein du système d'une source de bruit prépondérante par rapport aux autres sources. La source de bruit b_{norm} permet de modéliser la somme de nombreuses sources de bruit ayant des variances similaires. Le théorème de la limite centrale permet de montrer que la somme de ces variables aléatoires de même loi va tendre vers une variable suivant une loi normale.

Ce modèle a été testé sur différentes applications de TNS et pour différentes configurations virgule fixe conduisant à des bruits différents en sortie. Un test du χ^2 est utilisé pour mesurer l'adéquation entre les densités de probabilité du bruit réel b_y et du bruit modélisé b_u. Pour les différentes applications, le test est passé avec succès pour quasiment toutes les configurations virgule fixe testées. Ce modèle est valide pour les systèmes LTI. Pour les systèmes non LTI, le bruit en sortie peut être modélisé si son coefficient d'aplatissement (kurtosis) est inférieur à 3. Un coefficient d'aplatissement supérieur à 3 peut être obtenu, dans les systèmes non LTI, lors de la multiplication d'une source de bruit de quantification par un signal variant dans le temps. Ainsi, pour avoir un modèle plus général nous travaillons actuellement sur l'utilisation d'une variable aléatoire b_{ng} suivant une loi normale généralisée. Cette loi possède un paramètre permettant de fixer le coefficient d'aplatissement.

3.3.1.2 Densité spectrale de puissance

Le comportement temporel du bruit en sortie du système, et plus particulièrement la dépendance entre les échantillons au cours du temps, est analysée à travers la fonction d'autocorrélation $\varphi_{b_y b_y}(\tau)$. Ce comportement peut être étudié dans le domaine fréquentiel en utilisant la densité spectrale de puissance $\Phi_{b_y b_y}(e^{j\omega})$ correspondant à la transformée de Fourier de $\varphi_{b_y b_y}$. L'objectif est de concevoir un filtre linéaire H_G (filtre formeur) permettant de modifier la densité spectrale de puissance du bruit b_u afin qu'elle soit la plus proche possible de celle du bruit réel b_{y_s} en sortie du système B. Dans [Parashar 10a], l'expression de la densité spectrale de puissance $\Phi_{b_y b_y}(e^{j\omega})$ est calculée dans le cas des systèmes composés d'opérations ayant un modèle de bruit linéaire. Cette expression nécessite de calculer la fonction d'autocorrélation des signaux présents au sein du système. Ceci est réalisé à partir des valeurs des signaux obtenus au cours d'une unique simulation en virgule flottante. Dans le cas des systèmes LTI, nous retrouvons la présence dans l'expression de la réponse fréquentielle du système. L'application de cette technique nécessite que les signaux soient stationnaires. Si le signal n'est pas stationnaire, alors, celui-ci est décomposé en portions pour lesquelles, le signal peut être considéré stationnaire. Ainsi, un filtre linéaire H_G est calculé pour chaque portion stationnaire.

3.3.1.3 Évaluation des performances à l'aide du modèle SBU

La capacité du modèle SBU pour évaluer les performances d'une application a été évaluée sur un codeur/décodeur MP3 [13] dans le cadre du stage de Master de Romain Serizel. Les résultats des expérimentations sont détaillés dans [Ménard 08b] et brièvement présentés dans cette partie. Pour ce type d'application, la métrique de précision correspondant à la puissance du bruit de quantification n'est pas représentative pour évaluer la qualité de la compression. La dégradation de la compression audio est mesurée à l'aide de la métrique ODG (*Objective Degradation Grade*). Cette métrique varie entre 0 (absence de dégradation) et -4 (inaudible). Le niveau -1 correspond au seuil en dessous duquel les dégradations deviennent gênantes.

Dans un premier temps, le modèle SBU a été utilisé pour évaluer les performances (ODG) en considérant uniquement la conversion en virgule fixe du sous-système correspondant aux filtres polyphases et dans un second temps en considérant la conversion en virgule fixe des sous-systèmes correspondant aux filtres polyphases et la MDCT [14]. Dans le premier cas, l'erreur relative entre l'estimation de l'ODG à l'aide du modèle SBU et sa valeur réelle est en moyenne de 2.6% pour différentes puissances de bruit testées. Dans le second cas, l'erreur relative est de 11.8%. Ces résultats montrent la capacité de notre modèle à prédire correctement les performances de ce type d'applications.

3.3.2 Approche mixte analytique-simulation

3.3.2.1 Opérations à modèle de bruit non linéaire

La méthode d'évaluation de la précision présentée dans la partie 3.2.2 et basée sur la théorie de la perturbation utilise un modèle de propagation du bruit au sein des opérations, défini à l'équation 3.3.

13. *Moving Picture Experts Group-1/2 Audio Layer 3.*
14. *Modified Discrete Cosine Transform.*

Ce modèle est basé sur un développement en série de Taylor et ainsi nécessite que la fonction associée à l'opération soit dérivable sur son domaine de définition.

Lorsque la fonction associée à l'opération est continue mais pas dérivable comme par exemple la fonction valeur absolue, le modèle de propagation, défini à l'équation 3.3, n'est pas applicable. Cependant, l'amplitude de l'erreur en sortie de l'opération liée aux bruits en entrée est du même ordre de grandeur que celle des bruits d'entrée. Ainsi, il est possible d'obtenir un modèle de propagation bornant la propagation du bruit et la technique basée sur la théorie de la perturbation, présentée dans la partie 3.2.2, peut être utilisée.

Lorsque la fonction associée à l'opération n'est pas continue, alors l'amplitude de l'erreur en sortie de l'opération liée aux bruits en entrée n'est plus, dans la majorité des cas, du même ordre de grandeur que celle des bruits d'entrée. Ainsi, la technique basée sur la théorie de la perturbation ne peut plus être utilisée. Dans la suite de cette partie, nous étudions plus en détails ce cas de figure.

Considérons l'opération conditionnelle O_j similaire à celle définie dans la partie 3.2.2.2. La sortie y_j prend la valeur de $y_{j,k}$ si l'entrée $c \in E_{j,k}$ avec $k \in [1, N_{dec}]$. Le terme N_{dec} représente le nombre d'ensembles $E_{j,k}$ différents associés à cette opération conditionnelle. Soient c_j et y_j, les valeurs exactes respectivement de l'entrée et de la sortie en précision infinie. Soient e_{c_j} et e_{y_j}, l'erreur de quantification respectivement en entrée et en sortie de l'opération. Cette erreur correspond à la différence entre la valeur en précision finie ($\widehat{y_j}$ ou $\widehat{c_j}$) et la valeur en précision infinie (y_j ou c_j). L'expression de l'erreur en sortie est liée à la concordance des décisions prises en précision finie et en précision infinie :

$$e_{y_j} = \begin{cases} \widehat{y}_{j,k} - y_{j,k} & \text{si} \quad \widehat{c_j} \in E_k \quad \text{et} \quad c_j \in E_k \\ \widehat{y}_{j,l} - y_{j,k} & \text{si} \quad \widehat{c_j} \in E_l \quad \text{et} \quad c_j \in E_k, \quad \forall k \neq j \end{cases} \tag{3.20}$$

Dans le premier cas de l'équation 3.20, les décisions prises en précision finie et en précision infinie sont identiques. Si l'erreur e_{y_j} est issue de traitements à base d'opérations à modèle de bruit linéaire, alors, cette erreur peut être assimilée à un bruit de quantification et traitée à l'aide de la méthode présentée dans la partie 3.2.2.2. Si les termes $y_{j,k}$ sont des valeurs constantes non entachées d'erreur de quantification comme pour la fonction signe, alors, l'erreur e_{y_j} est nulle.

Dans le second cas de l'équation 3.20, les décisions prises en précision finie et en précision infinie sont différentes. Par la suite le terme *erreur de décision* est utilisé pour dénommer cette situation. L'amplitude de l'erreur e_{y_j} peut être élevée et sa propagation dans le reste du système ne peut pas être traitée avec l'approche présentée dans la partie 3.2.2. Par exemple, dans le cas de l'opération réalisant la fonction signe ($y_j = \text{sgn}(c_j)$), les valeurs prises par l'erreur e_{y_j} en sortie de l'opération sont -2, -1, 0, 1 et 2. Ces valeurs sont du même ordre de grandeur que l'entrée c_j de l'opération. L'occurrence de l'une de ces quatre valeurs aura des conséquences importantes sur les traitements réalisés après cette opération. Ainsi, la probabilité d'occurrence de valeurs non nulles pour e_{y_j} doit être très faible pour que le système possède un fonctionnement proche de celui désiré.

Dans [Parashar 10d], un modèle analytique pour les opérations de décision est proposé. Celui-ci permet de déterminer la densité de probabilité (DDP) de l'erreur e_{y_j}. Dans le cas des opérations de décision, la puissance du bruit de quantification en sortie de l'opération n'est plus une métrique pertinente pour évaluer la précision des calculs. Il est nécessaire de connaître les valeurs prises par l'erreur e_{y_j} et les probabilités d'occurrence associées. L'expression de la DDP dépend de la distribution de l'entrée c de l'opération de décision et de la distribution du bruit de quantification en entrée e_{c_j}. L'expression de la DDP a été développée dans le cas ou l'entrée c_j et le bruit e_{c_j} sont gaussiens. Le modèle a été testé sur des opérations de *slicing*. Ces opérations de décision sont situées en sortie des modules de décodage des récepteurs de communications numériques. Ils permettent de décider à partir du signal démodulé, les valeurs des symboles transmis. Des expérimentations ont été réalisées pour des modulations BPSK [15] ($N_{dec} = 2$) et QAM [16]-16 ($N_{dec} = 16$). L'écart moyen entre les valeurs de la DDP, obtenues avec notre modèle et par simulation, est d'environ 1%. Cette approche permet d'évaluer finement les performances d'une application dont la sortie est composée d'une opération

15. *Binary Phase-Shift Keying.*
16. *Quadrature Amplitude Modulation.*

de décision, mais lorsque cette opération de décision est située au milieu de l'application, la propagation de l'erreur e_{y_j} au sein de l'application et la prise en compte de la corrélation entre les différentes erreurs e_{y_j} est une tâche complexe. Ainsi, nous nous sommes orientés vers une approche mixte intégrant la simulation.

3.3.2.2 Description de l'approche mixte analytique-simulation

Les techniques d'évaluation des performances basées sur la simulation permettent de traiter tous les types de systèmes mais conduisent à des temps d'évaluation élevés. Les techniques analytiques permettent de réduire fortement ce temps d'évaluation mais sont limitées au niveau des systèmes supportés. Dans [Parashar 10b], nous avons proposé une approche mixte combinant les approches analytiques et par simulation afin d'utiliser les points forts de chacune. L'objectif est d'utiliser les techniques basées sur la simulation uniquement lorsque des erreurs de décision surviennent et d'utiliser les résultats analytiques dans les autres cas. Cette approche peut être vue comme une technique permettant d'accélérer significativement l'évaluation, par simulation, des performances d'une application en utilisant des modèles analytiques. Ce travail a été réalisé dans le cadre de la thèse de Karthick Parashar et en collaboration avec l'IMEC [17].

Modélisation du système Cette approche se base sur le modèle de source de bruit unique permettant de modéliser par une seule source de bruit b_{u_i} l'ensemble des bruits de quantification d'un sous-système B_i composé d'opérations dont le modèle de bruit est linéaire.

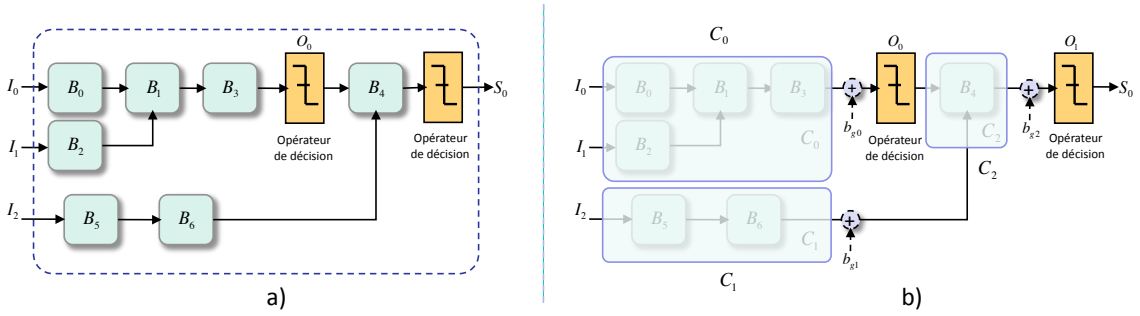


FIGURE 3.5 – a) Synoptique du système $\mathbb{B}$. b) Synoptique du système après la phase de clusterisation et d'ajout des sources de bruit.

Considérons le système $\mathbb{B}$ présenté à la figure 3.5.a. Ce système est composé de N_o opérations de décision O_j et N_b sous-systèmes B_k, intégrant chacun uniquement des opérations dont le modèle de bruit est linéaire. Les sous-systèmes B_k sont regroupés entre eux s'il ne sont pas séparés par des opérations de décision pour former les *clusters* C_i. L'objectif est de regrouper au sein d'un même *cluster* le maximum de blocs pouvant être traités par l'approche analytique. Le comportement en virgule fixe de chaque C_i est modélisé par une unique source de bruit b_{u_i}. Les paramètres statistiques (moments d'ordre 1, 2 et 4, densité spectrale de puissance) de cette source de bruit b_{u_i} sont calculés à partir du modèle analytique associé au *cluster* C_i. Les valeurs de ces paramètres sont fonction de la combinaison, des largeurs des opérations, testée. Soit $\left[\underline{b_{u_i}}\,\overline{b_{u_i}}\right]$ le domaine de définition de cette source de bruit.

Le système obtenu après la phase de clusterisation forme un graphe $G_{sys}(V_{sys}, E_{sys})$ dont l'ensemble V_{sys} des nœuds contient les *clusters* C_i et les opérations O_j et les arcs représentent les signaux connectant les opérations et clusters. Pour la présentation de la méthode, nous considérons que ce graphe G_{sys} est acyclique.

Dans l'exemple considéré, les sous-systèmes B_0 à B_3 sont regroupés pour former le *cluster* C_0. Une source de bruit b_{u0} est introduite à la sortie du cluster. De même, les sous-systèmes B_5 et B_6 sont regroupés au sein du *cluster* C_1 et la source de bruit b_{u1} est associée à ce cluster. Le sous-système B_4 forme le *cluster* C_2 et n'est pas regroupé avec le *cluster* C_1 afin de ne pas avoir à simuler le *cluster* C_1 si une erreur de décision est

17. *Interuniversity Microelectronics Centre, Leuven, Belgium.*

présente en sortie de l'opération de décision O_0. Le synoptique du système après la phase de clusterisation et d'ajout des sources de bruit est présenté à la figure 3.5.b.

Stratégie d'évaluation Pour appliquer notre approche mixte, il est nécessaire de posséder les modèles analytiques associés à chaque cluster. De plus, une simulation de référence est réalisée. Les valeurs des entrées c_j de chaque opération de décision O_j sont stockées afin de pouvoir les réutiliser par la suite.

L'objectif de l'approche mixte est d'obtenir, par simulation, les valeurs de sortie du système pour les N_p échantillons de chaque entrée $I_m : [I_m(0), \ldots, I_m(n), \ldots I_m(N_p - 1)]$. Pour chaque échantillon $I_m(n)$ des entrées du système, le graphe est parcouru des sources vers les puits. Le traitement de chaque opération de décision O_j est réalisé en deux étapes présentées ci-dessous.

Pour chaque opération de décision O_j, la possibilité d'une erreur de décision est analysée en fonction de la valeur considérée $c_j(n)$ de l'entrée c_j et des bornes du bruit b_{u_i} associé à cette entrée. En effet, si la valeur $c_j(n)$ est assez éloignée des frontières de décision au regard des bornes du bruit alors nous pouvons conclure à l'absence d'erreur de décision. Cette condition, exprimée à l'équation 3.21 est vérifiée si le domaine de définition de la somme du bruit b_{u_i} et de la valeur considérée $c_j(n)$ est inclus dans l'ensemble $E_{j,k}$, sachant que $c_j(n) \in E_{j,k}$. Si la condition exprimée à l'équation 3.21 est vérifiée, alors le parcours du graphe se poursuit en utilisant la valeur $y_{j,k}$ pour la donnée y_j :

$$\left[\underline{b_{u_i}} + c_j(n),\ \overline{b_{u_i}} + c_j(n)\right] \subset E_{j,k} \quad \text{et} \quad c_j(n) \in E_{j,k}. \tag{3.21}$$

Si la condition, exprimée à l'équation 3.21 n'est pas vérifiée, alors une valeur aléatoire $b_{u_i}(n)$ est générée pour la source de bruit b_{u_i}. La possibilité d'une erreur de décision est analysée à l'aide de la condition exprimée à l'équation 3.22.

$$b_{u_i}(n) + c_j(n) \in E_{j,k} \quad \text{et} \quad c_j(n) \in E_{j,k}. \tag{3.22}$$

Si la condition, exprimée à l'équation 3.22 est vérifiée, la valeur $b_{u_i}(n)$ du bruit n'ayant pas entraînée d'erreur de décision, le parcours du graphe se poursuit en utilisant la valeur $y_{j,k}$ pour la donnée y_j. Si la condition, exprimée à l'équation 3.22 n'est pas vérifiée, alors une erreur de décision est apparue et les conséquences de celles-ci sur les performances de l'application doivent être analysées par simulation. Tous les nœuds utilisant un résultat issu de la sortie de l'opération de décision O_j doivent être traités par simulation. La simulation est réalisée en virgule flottante et le comportement en virgule fixe des nœuds représentant les *clusters* C_l impliqués dans la simulation, est toujours modélisé par la source de bruit b_{u_l} associée. Ceci permet ainsi d'éviter de réaliser des simulations en virgule fixe, toujours plus coûteuses en termes de temps d'exécution. Plus l'erreur de décision se situe proche des sources du graphe G_{sys}, plus le nombre de nœuds (*clusters* et opérations de décision) devant être traités par simulation est élevé. Le temps de simulation dépend de la localisation de cette erreur au sein du graphe G_{sys}. Les différents *clusters* dont les entrées n'exploitent pas de résultats issus d'une opération de décision ne seront jamais simulés. Ainsi, dans le système $\mathbb{B}$ présenté à la figure 3.5.b, les nœuds représentant les *clusters* C_1 et C_2 ne sont jamais traités par simulation.

Afin d'avoir une estimation statistique réaliste des performances, le nombre d'échantillons utilisés pour réaliser la simulation est défini afin d'avoir assez d'erreurs de décision (au moins 100) liées à la virgule fixe.

Systèmes contenant des retards Dans le cas où les *clusters* contiennent des opérations de retard, l'erreur de décision e_{y_j} générée par l'opération O_j aura des conséquences sur les échantillons futurs (d'un point de vue temporel). L'objectif est de simuler le système tant que les conséquences de l'erreur de décision e_{y_j} sont perceptibles au niveau de la sortie globale du système.

Soit S_{zy_j} le sous-système ayant pour entrée y_j (sortie de l'opération de décision O_j) et pour sortie z (sortie du système global). Soit N_{ret}, le retard maximal au sein de ce sous système. Si le sous-système S_{zy_j} n'est pas récursif (absence de circuit au sein du graphe représentant le système), alors les conséquences de l'erreur de décision e_{y_j} peuvent être perceptibles, en sortie du système global, pendant N_{ret} échantillons.

En conséquence, les N_{ret} échantillons suivants : $c_i(n+1), \ldots, c_i(n+N_{ret})$ de c_i doivent être traités par simulation. Si le système est récursif, les erreurs présentes en sortie sont réinjectées dans le système. Les conséquences de e_{y_j} peuvent être perceptibles, en sortie du système global, pendant un intervalle de temps théoriquement non borné. Ainsi, en pratique, il est nécessaire de déterminer l'intervalle de temps requis pour que les conséquences de e_{y_j} ne soient plus perceptibles en sortie. La nécessité de simuler une suite d'échantillons pour une erreur de décision donnée, va augmenter les temps de simulation au sein de cette approche mixte et limiter l'accélération par rapport aux approches classiques basées sur la simulation en virgule fixe.

3.3.2.3 Expérimentations

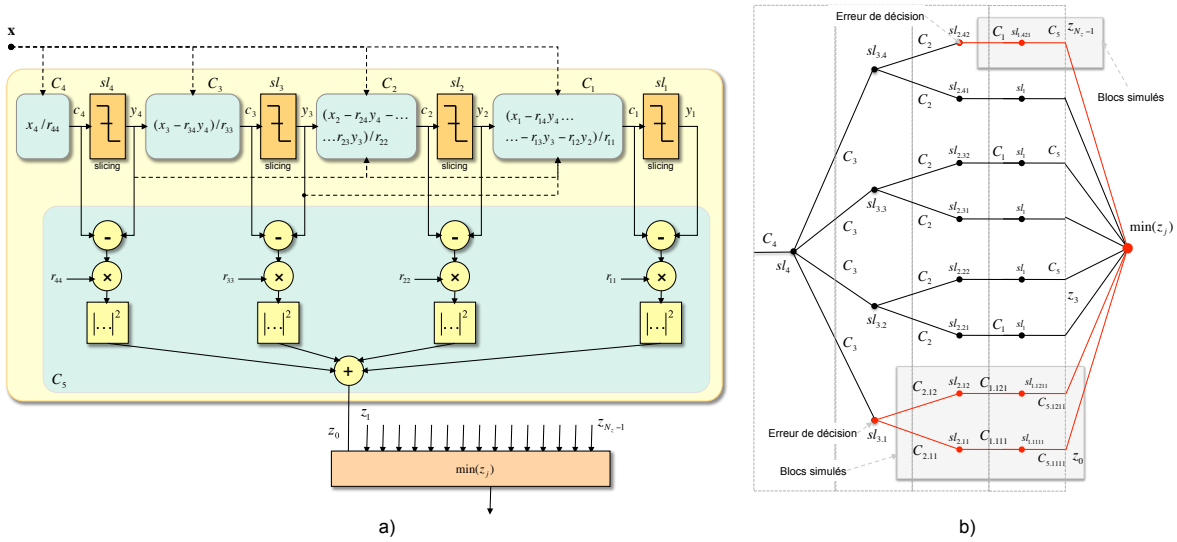


FIGURE 3.6 – a) Traitement associé à un chemin du SSFE pour 4 antennes. b) Topologie des traitements réalisés pour un SSFE $[4, 2, 1, 1]$.

Notre approche mixte d'évaluation des performances a été testée, dans le cadre de notre collaboration avec l'IMEC, sur une application de décodage sphérique [43]. Cette application est utilisée au sein des récepteurs de communication numérique MIMO [18]. Le décodage MIMO réalise le décodage des symboles transmis au niveau de chaque antenne du récepteur. N_r antennes sont considérées en réception. Le synoptique de ce type de récepteur est présenté à la figure 3.6.a. L'entrée du décodeur correspond au vecteur $\mathbf{x} = [x_1, \ldots, x_i, \ldots x_{N_r}]$ avec x_i le signal associé à l'antenne i. Ce vecteur $\mathbf{x}$ est décodé séquentiellement en traitant les antennes par ordre décroissant de puissance. A chaque étape, les contributions des symboles $y_{N_r}, \ldots, y_{i+1}$ ayant déjà été décodés sont soustraites du signal à traiter x_i. Ce principe a été formalisé à travers l'algorithme BLAST [19] [83]. Chaque symbole y_i est décodé à partir du signal c_i en prenant le symbole dont la distance est la plus proche. Le bruit de transmission perturbe la décision et peut amener à des erreurs de décision sur les symboles. La technique la plus robuste pour réduire l'influence du bruit du récepteur est de tester toutes les combinaisons possibles des symboles et de retenir la plus probable. Cette technique est connue sous le nom de maximum de vraisemblance (MV). La complexité de cette approche est exponentielle par rapport au nombre d'antennes. L'intermédiaire entre ces deux types de décodeur est le décodage sphérique qui teste pour chaque symbole y_i, uniquement m_i possibilités. Cette approche permet d'obtenir des performances proches du décodeur MV en réduisant significativement sa complexité. Nous avons travaillé sur l'algorithme de décodage sphérique

18. *Multiple Input Multiple Output.*
19. *Bell Laboratories Layered Space-Time.*

SSFE [20] dont la particularité réside dans le choix des symboles y_i à tester pour une valeur donnée de d_i. Les paramètres de ce décodeur $[m_{N_r}, \ldots, m_i \ldots, m_1]$ sont le nombre m_i de symboles testés pour chaque antenne i. La figure 3.6.b. montre la topologie des traitements réalisés pour un SSFE $[4, 2, 1, 1]$. Le traitement réalisé au sein de chaque chemin est présenté à la figure 3.6.a. Les différents signaux x_i, c_i, y_i, r_{lm} correspondent à des valeurs complexes.

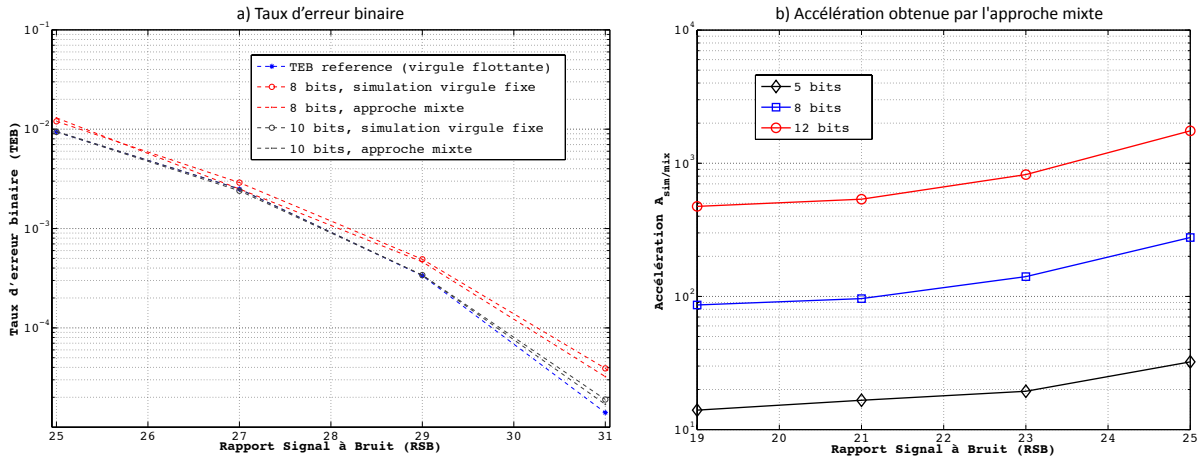


FIGURE 3.7 – a) Comparaison des TEB obtenus avec l'approche mixte proposée et avec l'approche basée sur la simulation en virgule fixe. b) Accélération obtenue par l'approche mixte en fonction du RSB et pour trois spécifications virgule fixe.

Pour le cas d'un système à quatre antennes, le traitement réalisé par chaque chemin (figure 3.6.a) est composé de cinq *clusters* C_1 à C_5 et de quatre opérations de décision sl_k dénommées *slicing*. Cette opération de décision permet de déterminer le symbole complexe s_i le plus proche de d_i. Ceci revient à quantifier d_i avec un nombre de bits faible. La dernière opération de décision correspond au calcul de la valeur minimale des sorties z_i de chaque chemin. Les modèles analytiques des *clusters* sont détaillés dans [Parashar 10b]. Pour cette application, les performances de l'application sont mesurées à travers le taux d'erreur binaire (TEB). Au sein de la figure 3.6.b, un exemple avec deux erreurs de décision est proposé et les blocs devant être simulés sont définis. La présence d'une erreur de décision au niveau de l'opération de décision $sl_{3.1}$ nécessite de simuler tous les nœuds utilisant un résultat issu de $sl_{3.1}$, c'est à dire : $C_{2.11}$, $sl_{2.11}$, $C_{1.111}$, $sl_{1.1111}$, $C_{5.1111}$ $C_{2.12}$, $sl_{2.12}$, $C_{1.121}$, $sl_{1.1211}$, $C_{5.1211}$ et l'opération min.

Comme pour les estimateurs de précision présentés dans la partie 3.2.2, la qualité de l'estimation est analysée et le temps d'évaluation de notre approche (t_{mix}) est comparé avec celui obtenu pour une méthode classique basée sur la simulation en virgule fixe (t_{sim}). Pour cela nous avons calculé le facteur d'accélération $A_{sim/mix}$ de notre approche correspondant au rapport entre t_{sim} et t_{mix}.

Pour analyser la qualité de cette approche mixte, le TEB de l'application obtenu avec notre approche (TEB_{mix}) et celui obtenu avec une simulation en virgule fixe (TEB_{sim}) sont comparés pour différentes configurations virgule fixe et différents rapports signal à bruit de transmission (RSB) en entrée du récepteur. Les résultats sont présentés à la figure 3.7.a. Les résultats montrent que les valeurs obtenues pour TEB_{mix} et TEB_{sim} sont très proches. Les expérimentations montrent que l'erreur relative entre TEB_{mix} et TEB_{sim} est inférieure à 10%, même pour des RSB élevés. Les nombres d'erreurs de décision liées au traitement en précision finie obtenus avec l'approche mixte et avec l'approche basée sur la simulation en virgule fixe, sont très proches.

L'accélération en termes de temps d'exécution, obtenue par l'approche mixte par rapport à l'approche basée sur la simulation en virgule fixe est présentée à la figure 3.7.b en fonction du RSB et pour trois

20. *Selective Spanning with Fast Enumeration.*

RSB	$m = [1,2,2,4]$			$m = [1,1,2,4]$			$m = [1,1,1,4]$		
	t_{sim}	t_{mix}	$A_{sim/mix}$	t_{sim}	t_{mix}	$A_{sim/mix}$	t_{sim}	t_{mix}	$A_{sim/mix}$
19 dB	128	110e-3	**1.1e+3**	113	72e-3	**1.5e+3**	98	54e-3	**1.8e+3**
21 dB	128	96e-3	**1.3e+3**	113	72e-3	**1.5e+3**	98	39.e-3	**2.5e+3**
23 dB	129	98e-3	**1.3e+3**	113	67e-3	**1.6e+3**	102	31e-3	**3.2e+3**
25 dB	128	97e-3	**1.3e+3**	113	67e-3	**1.6e+3**	98	32e-3	**3.0e+3**

TABLE 3.5 – Temps d'évaluation (s) obtenus pour l'approche mixte t_{mix} et pour l'approche basée sur la simulation t_{sim} et accélération en termes de temps d'exécution $A_{sim/mix}$. Les résultats sont présentés pour différents niveaux de RSB et trois configurations du SSFE.

spécifications virgule fixe différentes. L'accélération obtenue par l'approche mixte est significative elle se situe entre 10 et 2000. L'accélération augmente lorsque le RSB augmente. En effet, lorsque le RSB est élevé, le niveau du bruit de transmission est plus faible et ainsi, en virgule fixe, l'entrée des opérations de décision est moins souvent proche des frontières de décision. En conséquence, moins d'erreurs de décision potentielles sont présentes. L'accélération augmente lorsque le niveau du bruit de quantification généré au sein des *clusters* diminue. En effet, dans ce cas, moins d'erreurs de décision liées au bruit de quantification sont présentes. Dans le tableau 3.5, les temps d'évaluation obtenus pour l'approche mixte t_{mix} et pour l'approche basée sur la simulation t_{sim}, ainsi que l'accélération $A_{sim/mix}$ sont présentés pour trois configurations du SSFE. Le temps d'évaluation t_{mix} de l'approche mixte est faible. Celui-ci est environ de $0.1s$. Les résultats montrent que l'accélération diminue légèrement lorsque le nombre de chemins augmente. En effet, plus d'opérations de décision sont présentes et ainsi, la probabilité qu'une erreur de décision apparaisse est plus importante.

3.4 Conclusion

Au cours de ces différentes années de recherche, nous avons proposé différentes approches complémentaires d'évaluation de la précision et des performances permettant d'étendre progressivement la classe des systèmes supportés. La méthode d'évaluation de la précision proposée au cours de ma thèse pour les systèmes LTI a été ensuite automatisée et outillée. L'outil ID.Fix-AccEval permet de générer automatiquement, à partir d'un graphe flot de signal, le code C décrivant l'expression analytique de la puissance du bruit de quantification. Dans la thèse de Romuald Rocher [71], une méthode d'évaluation de la précision des calculs a été proposée pour les systèmes composés d'opérations dont le modèle de bruit est linéaire. Ce modèle est étendu dans le cadre de la thèse de Jean-Charles Naud afin de prendre en compte les traitements alternatifs du bruit de quantification dans le cas de structures conditionnelles. Ces différents aspects ont été intégrés à l'outil ID.Fix-AccEval. Dans le cas des systèmes composés d'opérations dont le modèle de bruit n'est pas linéaire, la métrique de précision ne permet pas de bien quantifier les effets des calculs en précision finie. Ainsi, dans le cadre de la thèse de Karthick Parashar, nous avons proposé une approche mixte combinant la simulation et les résultats des modèles analytiques pour évaluer les performances d'une application.

Au sein de la figure 3.8, nous resituons et comparons notre travail par rapport aux approches existantes. Pour chaque méthode, les systèmes supportés sont fournis. La classification des systèmes repose sur la linéarité du modèle de bruit des opérations composant le système, la linéarité et l'invariance dans le temps (LTI) du système et la récursivité du système (Rec.). Les méthodes sont comparées de manière qualitative en termes de temps d'évaluation de la précision ou des performances. Pour l'évaluation de la précision, les méthodes basées sur la théorie de la perturbation conduisent à la même expression de la puissance du bruit, ainsi, le temps d'évaluation de cette expression est le même pour toutes ces approches mais il faut ajouter le temps d'obtention de l'expression analytique. Les techniques hybrides [76] (2004), [21] (2006), [38] (2008) déterminent les gains sur la moyenne (K_i) et la variance (L_{ij}) à partir d'un ensemble de simulations. Ainsi, le temps d'obtention dépend du nombre de sources de bruit considérées et peut être relativement élevé. Les

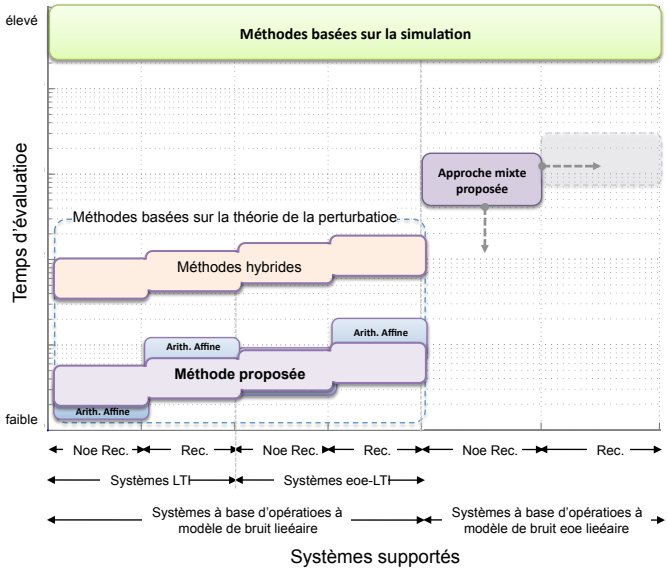


FIGURE 3.8 – Comparaison des approches d'évaluation de la précision et des performances.

méthodes présentées dans [60] (2008, systèmes LTI), et [12] (2010, systèmes non LTI) utilisent la simulation basée sur l'arithmétique affine pour déterminer les coefficients K_i et L_{ij}. Ces deux méthodes fournissent rapidement l'expression analytique pour les systèmes non récursifs mais nécessitent beaucoup plus de temps pour les systèmes récursifs. Notre approche d'évaluation de la précision conduit à des temps d'obtention de l'expression analytique faibles mais légèrement plus élevés que ceux obtenus dans [12] pour les systèmes non récursifs. Pour les systèmes récursifs, le temps d'obtention augmente peu, mais il est nettement plus faible que celui obtenu dans [12]. Pour les systèmes intégrant des opérations de décision, notre approche mixte permet de réduire fortement les temps d'évaluation des performances par rapport aux méthodes existantes basées uniquement sur la simulation. L'objectif est maintenant d'étudier et de valider à travers des expérimentations la possibilité d'étendre cette approche mixte aux systèmes récursifs (présence d'une opération de décision au sein d'une boucle de retour) et d'utiliser les techniques de simulation d'évènements rares pour réduire les temps d'évaluation des performances.

Chapitre 4

Conversion en virgule fixe

4.1 Introduction

Les contraintes inhérentes aux systèmes embarqués nécessitent une implantation efficace en termes de surface, de consommation d'énergie ou de temps d'exécution. Ainsi, de nombreux systèmes embarqués utilisent une représentation en virgule fixe pour implanter les applications contenant des traitements mathématiques de données. En effet, de nombreuses applications destinées aux systèmes embarqués et issues notamment du domaine du traitement du signal et de l'image (TDSI), peuvent tolérer une dynamique des données et une précision limitées.

Les applications de TDSI sont conçues et simulées en utilisant une représentation en virgule flottante mais au final elles sont souvent implantées au sein de systèmes utilisant la représentation en virgule fixe. Ainsi, il est nécessaire de réaliser une conversion en virgule fixe des applications ciblées. Cette conversion consiste à déterminer, pour chaque donnée, le nombre de bits alloués à la partie entière et à la partie fractionnaire. Ainsi, le processus de conversion en virgule fixe est composé de deux étapes principales correspondant à la détermination de la position de la virgule et à l'optimisation de la largeur des données.

La première étape, correspondant à la détermination de la position de la virgule définit le nombre de bits pour la partie entière. L'objectif est de minimiser ce nombre de bits en se prémunissant des débordements dont l'apparition fait chuter rapidement les performances de l'application. Cette première étape du processus de conversion nécessite de déterminer la dynamique des données. Des méthodes classiques permettent de garantir l'absence de débordement mais conduisent à une sur-estimation parfois importante de la dynamique. Nous travaillons sur des approches stochastiques permettant de déterminer la dynamique des données pour une probabilité de débordement fixée. Ce travail est présenté dans la partie 4.2.

La seconde étape correspond à la détermination du nombre de bits pour la partie fractionnaire. La largeur des différentes données de l'application est optimisée. L'objectif est de minimiser le coût de l'implantation tant que la précision des calculs est suffisante pour maintenir les performances de l'application. Dans notre travail de recherche, deux aspects de l'optimisation des largeurs ont été abordés. Le premier concerne la définition d'un algorithme d'optimisation adapté au problème. Celui-ci fait l'objet de la partie 4.3.1. Le second aspect concerne l'optimisation de la largeur des opérateurs dans le cas de la synthèse d'une architecture pour une implantation au sein d'un ASIC ou d'un FPGA. Ces travaux, présentés dans la partie 4.3.2, prennent en compte la problématique du couplage des processus de synthèse d'architecture et d'optimisation de la largeur des opérations.

La conversion en virgule fixe d'une application complète nécessite de déterminer la largeur de plusieurs centaines de variables et conduit à un problème d'optimisation avec un nombre de variables à optimiser important. De plus, l'obtention d'un modèle analytique pour évaluer la précision des calculs du système complet se heurte au problème de traitement de graphes ayant un nombre de noeuds élevé. Une approche hiérarchique permettant d'optimiser la spécification virgule fixe au niveau système a été définie. Cette approche

hiérarchique est utilisée pour découper le problème d'optimisation en plusieurs niveaux et ainsi restreindre le nombre de variables à optimiser à chaque niveau.

Ces différents travaux de recherche sont accompagnés du développement d'une infrastructure logicielle permettant d'automatiser la conversion en virgule fixe. Cette infrastructure est détaillée dans la partie 4.5.

4.2 Évaluation de la dynamique

Encadrement :	Andreï Banciu, doctorant depuis 2008
	Huong Thao Do, Master 2011
Conférences :	DASIP 2010 [Banciu 10], SIPS 2011 [Banciu 11]
Collaboration :	Programme R&D nano 2012 (ST Microelectronics)

La première étape de la conversion en virgule fixe correspond à la détermination du nombre de bits w_{IP} pour la partie entière de chaque donnée. Cette étape nécessite de connaître la dynamique (domaine de définition) de chaque donnée. Les méthodes classiques de détermination de la dynamique surestiment la dynamique et ainsi conduisent à un coût d'implantation plus important par rapport à la solution optimale. Dans le cadre de la thèse d'Andrei Banciu (2009–2011, CIFRE avec ST Microelectronics), nous étudions les méthodes de détermination de la dynamique des données plus efficaces et basées sur une approche stochastique.

4.2.1 Motivations

L'arithmétique d'intervalle (AI) [2] détermine les bornes des variables dans le pire cas. Ainsi, cette approche permet de garantir l'absence de débordement. Soit w_{IP}^{IA}, le nombre de bits nécessaires pour coder la partie entière en utilisant une estimation de la dynamique basée sur l'AI. La version de base de l'AI conduit à une estimation pessimiste car elle ne prend pas en compte la corrélation spatiale (opérations sur des données issues d'une même variable) et la corrélation temporelle (p. ex. des opérations sur des données issues d'une même variable mais à des instants différents). L'arithmétique affine [29] permet de prendre en compte la corrélation spatiale mais pas la corrélation temporelle. Cette corrélation temporelle est présente dans de nombreuses applications de TNS au sein desquelles des opérations de retard sont présentes. Ce type d'approche surestime les bornes de l'intervalle et conduit ainsi, à la présence de bits non utilisés au niveau de la partie entière entraînant une augmentation du coût de l'implantation.

Dans le cadre de la thèse d'Andrei Banciu (CIFRE avec ST Microelectronics), nous travaillons sur la définition de méthodes d'évaluation de la dynamique moins pessimistes. En effet, lors de la conception de ses modems haut-débit basés sur la technologie OFDM [1], la société ST Microelectronics est confrontée à ce problème de surestimation de la dynamique. En particulier, les modules IFFT [2] et FFT présents au sein de l'émetteur et du récepteur OFDM sont sensibles à ces surestimations de la dynamique. L'utilisation de l'AI pour estimer la dynamique conduit à l'ajout d'un bit supplémentaire à chaque étage de la FFT ou de l'IFFT pour éviter les débordements. Cette surestimation donne lieu à des opérateurs ayant des largeurs plus élevées et donc une surface et une latence plus importante. Pour la technologie utilisée, cette augmentation de la latence ne permet plus de respecter les contraintes temps réel de l'application. Ainsi, des méthodes d'estimation de la dynamique moins pessimistes sont nécessaires.

L'utilisation d'une largeur w_{IP} plus faible que w_{IP}^{IA} permet de réduire le coût de l'implantation mais va entraîner la présence de débordements (dépassement de capacité) et ainsi modifier le comportement de l'application. Les débordements conduisent à des écarts importants entre les valeurs en précision infinie (absence

1. *Orthogonal Frequency-Division Multiplexing.*
2. *Inverse Fast Fourier Transform.*

de débordement) et en précision finie (présence de débordement). Ainsi, ces débordements perturbent fortement l'application et font chuter rapidement les performances lorsque leur occurrence est non négligeable. Cependant, certaines applications peuvent tolérer la présence de débordement si leur probabilité d'occurrence P_{ov} est inférieure à un seuil P_{ov}^{max}. La détermination du nombre de bits pour la partie entière peut être vue comme un problème d'optimisation de la largeur w_{IP} de la partie entière de chaque donnée. Soit $\mathbf{w_{IP}}$ le vecteur contenant les largeurs w_{IP} de l'ensemble des N_d données de l'application. L'objectif est de minimiser le coût C de l'implantation tant que les contraintes λ sur les performances de l'application sont supérieures à un seuil λ_{obj} :

$$\min\left(C\left(\mathbf{w}\right)\right) \quad \text{tel que} \quad \lambda\left(\mathbf{w_{IP}}\right) \geq \lambda_{obj}. \tag{4.1}$$

Le coût C dépend de la largeur totale $\mathbf{w}$ des données et pas uniquement de la largeur de la partie entière $\mathbf{w_{IP}}$. Ainsi, il est nécessaire de fixer les valeurs des largeurs de la partie fractionnaire pour évaluer le coût.

Dans un premier temps, le problème d'optimisation a été modifié afin d'utiliser une métrique intermédiaire pour quantifier les débordements. La probabilité de débordement $\mathbf{P_{ov}}(i)$ associée à chaque variable i de l'application est utilisée. Ainsi, le problème d'optimisation est reformulé de la manière suivante :

$$\min\left(C\left(\mathbf{w}\right)\right) \quad \text{tel que} \quad \mathbf{P_{ov}}(i) \leq \mathbf{P_{ov}^{max}}(i) \quad \forall i \in 1, N_d. \tag{4.2}$$

avec $\mathbf{P_{ov}^{max}}$ la probabilité de débordement maximale associée à chaque donnée.

La problématique des travaux de recherche réalisés par Andreï Banciu est de déterminer la dynamique des données pour une probabilité maximale de débordement P_{ov}^{max}. Ceci nécessite de déterminer la densité de probabilité (DDP) des variables pour lesquelles des débordements sont autorisés.

4.2.2 Approches stochastiques pour l'évaluation de la dynamique

La méthodologie générale permettant de déterminer la dynamique des données pour une probabilité de débordement fixée est présentée à la figure 4.1. Cette méthodologie se base sur une modélisation stochastique des signaux en vue de déterminer la DDP de la sortie. La première étape consiste à modéliser les différents signaux d'entrées x_i en les décomposant sur une base déterminée. Soit Υ_{x_i}, les différents paramètres stochastiques résultant de la décomposition de x_i. La seconde étape calcule les paramètres stochastiques Υ_y de la sortie y du système. Différentes approches peuvent être utilisées pour déterminer ces paramètres. A partir de ces paramètres Υ_y, la DDP de la variable y est déterminée, puis la dynamique de la donnée peut être obtenue à partir de la probabilité maximale de débordement P_{ov}^{max}.

4.2.2.1 Description de la méthodologie

Dans cette partie, uniquement la modélisation basée sur la décomposition de Karhumen Loeve est présentée, actuellement, pour étendre la classe des systèmes supportés, l'utilisation de polynômes de chaos est étudiée.

Modélisation des signaux d'entrée. La décomposition de Karhumen Loeve (KLE) [58] permet l'approximation d'un processus aléatoire $x(n)$ par une combinaison linéaire de fonctions déterministes f_i multipliées avec une variable aléatoire η_i :

$$x(n) = \mu_x + \sum_{i=0}^{\infty} \sqrt{\lambda_i} f_i(n) \eta_i. \tag{4.3}$$

où μ_x est la moyenne de x, $f_i(n)$ sont les fonctions propres, λ_i les valeurs propres de la matrice de covariance C_{xx}, et η_i des variables aléatoires de variance unitaire, de moyenne nulle et non-corrélées entre elles.

Le nombre d'éléments de la somme étant infini, il est nécessaire de tronquer celle-ci et ainsi ne conserver que M composantes au sein de la décomposition KLE. La valeur de M va dépendre de la rapidité de

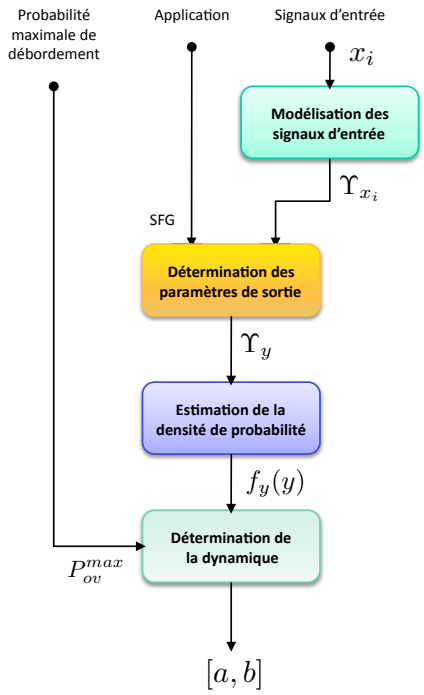


FIGURE 4.1 – Synoptique de l'approche stochastique pour déterminer la dynamique des données $[a, b]$ pour une probabilité de débordement P_{ov}^{max}.

décroissance des valeurs propres λ_i. Lorsque x est fortement corrélé, la décroissance est rapide et peu de termes sont nécessaires. Lorsque x est un bruit blanc (les échantillons de x ne sont pas corrélés entre eux), la décroissance est faible et de nombreux termes sont nécessaires pour que l'erreur liée à la troncature ne soit pas trop importante.

Propagation des paramètres de la modélisation stochastique. La méthode KLE est utilisée dans le cas des systèmes linéaires invariants dans le temps, car elle exploite les propriétés de linéarité du système pour obtenir les paramètres de la sortie y. Dans [84], le calcul des paramètres modélisant la sortie est réalisé par simulation. Pour chaque composante i de la décomposition KLE, une simulation est réalisée pour déterminer le gain sur cette composante. Ainsi, M simulations du système doivent être réalisées. Une autre approche consiste à propager les paramètres associés aux entrées au sein du graphe représentant l'application. Pour chaque opération, les paramètres de la sortie de l'opération sont déterminés en fonction de ceux en entrée. Disposant de la possibilité de déterminer la réponse impulsionnelle h_{yx} entre deux variables x et y à l'aide de l'outil ID.Fix-AccEval, nous utilisons celle-ci pour obtenir la décomposition de la sortie [Banciu 11]. L'expression de la sortie est obtenue à l'aide de l'expression suivante :

$$y(n) = \mu_x * h_{yx}(n) + \sum_{i=0}^{M} \sqrt{\lambda_i} f_i(n)\eta_i * h_{yx}(n) = \mu_y + \sum_{i=0}^{M} y_i(n)\eta_i. \tag{4.4}$$

Cette technique a été implantée dans l'outil d'évaluation de la dynamique ID.Fix-DynEval utilisant la même infrastructure logicielle que ID.Fix-AccEval présentée dans la partie 3.2.3.

Estimation de la DDP. Différentes méthodes peuvent être utilisées pour déterminer la DDP de la variable y à partir de ses paramètres stochastiques. La méthode la plus simple est de déterminer l'histogramme de y en générant les différentes variables aléatoires η_i. Ceci nécessite que la DDP des variables aléatoires η_i

puisse être facilement générée comme dans le cas d'une loi normale ou uniforme. La seconde approche utilise l'expansion de Edgeworth [11] permettant de traiter les lois proches d'une gaussienne. La loi est déterminée à partir des moments statistiques d'ordre un à quatre de y et de la loi normale. La troisième méthode permet l'approximation de la DDP de y en utilisant une estimation par noyau. Cette dernière méthode est a été retenue et est présentée succinctement dans [Banciu 11].

Détermination de la dynamique. Les valeurs des bornes a et b permettant d'obtenir une probabilité de débordement inférieure à P_{ov}^{max} sont obtenues en résolvant l'inégalité suivante :

$$\int_a^b f_y(y)dy \leq 1 - P_{ov}^{max}, \tag{4.5}$$

avec $f_y(y)$ la DDP de y.

4.2.3 Expérimentations

L'approche d'estimation stochastique utilisant la décomposition KLE est comparée aux résultats obtenus par simulation et avec l'arithmétique d'intervalle (AI). La méthode proposée est utilisée pour déterminer l'intervalle $[a, b]$ permettant d'obtenir une probabilité de débordement inférieure à P_{ov}^{max}. Ensuite, la probabilité de débordement P_{ov}^{sim} obtenue par simulation pour le domaine de définition correspondant à l'intervalle $[a, b]$ est mesurée et comparée à la probabilité de débordement ciblée P_{ov}^{max}. Les résultats sont présentés, dans le tableau 4.1, pour trois applications correspondant à des filtres FIR, IIR et une IFFT. Pour montrer l'écart obtenu par rapport à l'arithmétique d'intervalle (AI), le domaine de définition obtenu avec cette technique est fourni. Les résultats montrent que les probabilités de débordement ciblées P_{ov}^{max} et celles obtenues par simulation P_{ov}^{sim} sont relativement proches. Ces résultats montrent la surestimation de la dynamique des données obtenue avec l'AI. Cette surestimation conduit à la présence de 3 à 4 bits supplémentaires en sortie de la IFFT pour l'AI par rapport à la méthode KLE. La version de base de l'AI est utilisée, ainsi elle ne prend pas en compte la corrélation spatiale entre les données. Cependant, pour la IFFT, chaque variable n'intervenant qu'une seule fois dans le calcul d'un élément du vecteur de sortie, ce problème de corrélation spatiale n'est pas présent.

Application	P_{ov}^{max}	P_{ov}^{sim}	$[a, b]$ avec KLE	$[a, b]$ avec AI
FIR	10^{-3}	$0.94.10^{-3}$	[-1.66 ;1.54]	$[-8.04; 8.04]$
	10^{-4}	$1.14.10^{-4}$	[-1.88 ;1.76]	
	10^{-5}	$0.74.10^{-5}$	[-1.99 ;2.04]	
IIR	10^{-3}	$0.96.10^{-3}$	[-4.49 ;4.73]	[-14.89 ;14.89]
	10^{-4}	$0.97.10^{-4}$	[-5.18 ; 5.42]	
	10^{-5}	$1.98.10^{-5}$	[-5.56 ; 5.80]	
IFFT	10^{-3}	$0.97.10^{-3}$	[-4.35 ; 4.14]	[-60.01 ;60.28]
	10^{-4}	$1.08.10^{-4}$	[-5.35 ; 5.14]	
	10^{-5}	$1.13.10^{-5}$	[-6.24 ;6.03]	

TABLE 4.1 – Comparaison de la probabilité de débordement ciblée P_{ov}^{max} et celle obtenue P_{ov}^{sim}. L'intervalle $[a, b]$ est obtenu avec l'approche KLE pour une probabilité de débordement ciblée P_{ov}^{max}. La probabilité de débordement P_{ov}^{sim} est obtenue par simulation pour le domaine de définition $[a, b]$.

Pour l'application correspondant à la IFFT, le nombre de bits w_{IP} nécessaires pour coder la partie entière est déterminé à partir des estimations de la dynamique obtenues par l'approche KLE, l'approche basée sur la théorie des valeurs extrêmes (TVE) [68], la simulation et l'arithmétique d'intervalle (AI). Les valeurs de w_{IP}

sont présentées dans le tableau 4.2 en fonction de la probabilité de débordement ciblée P_{ov}^{max}. L'approche KLE conduit à une estimation, du nombre de bits, identique à celle obtenue par la simulation. Cette simulation peut être considérée comme la référence si le nombre d'échantillons utilisés est assez important. La théorie des valeurs extrêmes sous-estime les probabilités de débordement. L'arithmétique d'intervalle surestime de 3 à 4 bits la dynamique en fonction de la probabilité P_{ov}^{max} ciblée.

P_{ov}^{max}	Largeur w_{IP}		
	10^{-3}	10^{-4}	10^{-5}
Simulation	2	3	3
KLE	2	3	3
TVE	2	2	3
AI	6	6	6

TABLE 4.2 – Nombre de bits pour la partie entière de la sortie de l'IFFT.

4.2.4 Conclusion

Les approches stochastiques permettent de déterminer la DDP des variables en sortie d'un système et d'en déduire la dynamique permettant de limiter la probabilité de débordement à une valeur fixée. Ce type d'approche permet d'obtenir une estimation moins pessimiste que celle obtenue avec l'arithmétique d'intervalle. Cette amélioration de l'estimation de la dynamique est obtenue par la prise en compte des caractéristiques du signal d'entrée. En conséquence, cette estimation est fortement liée à la nature du signal d'entrée et n'est plus valide pour des signaux ayant des caractéristiques différentes. Ainsi, lors de la détermination de la dynamique avec ce type d'approche la question de la pertinence des signaux utilisés en entrée doit être résolue. Notre objectif en termes d'outil est de fournir différentes approches d'évaluation de la dynamique. Ensuite, l'utilisateur choisit la méthode la plus adaptée en fonction de ses contraintes. La décomposition KLE étant valable uniquement sur les systèmes LIT, nous travaillons actuellement sur l'utilisation de polynômes de chaos pour modéliser des opérations non-LIT. Les approches stochastiques peuvent aussi être utilisées pour déterminer la DDP du bruit de quantification en sortie d'un système [Banciu 11].

La prochaine étape pour pouvoir résoudre le problème d'optimisation du nombre de bits pour la partie entière (équation 4.1) est de proposer une approche efficace permettant de déterminer les effets des débordements sur les performances de l'application.

4.3 Optimisation de la largeur des données

L'objectif de la seconde étape du processus de conversion en virgule fixe est de déterminer le nombre de bits pour la partie fractionnaire de chaque donnée. Ceci consiste à optimiser la largeur des données afin de minimiser le coût de l'implantation et de garantir des performances données. Soit $\mathbf{w}$ le vecteur regroupant la largeur des opérandes de toutes les opérations d'une application donnée. Soit $\mathcal{C}$ la fonction de coût de l'application et λ la fonction définissant les performances de l'application en fonction de la largeur des données $\mathbf{w}$. L'objectif de cette étape de détermination du nombre de bits pour la partie fractionnaire est de minimiser la fonction de coût sous contrainte que les performances restent supérieures à un seuil minimal λ_{obj}. Ainsi, le processus d'optimisation peut être modélisé à l'aide de l'expression suivante :

$$\min\left(\mathcal{C}(\mathbf{w})\right) \quad \text{tel que} \quad \lambda(\mathbf{w}) \geq \lambda_{obj}. \tag{4.6}$$

Comme nous le montrons dans la partie 3.1, l'évaluation directe des performances d'un système en virgule fixe n'étant pas une tâche aisée, le problème d'optimisation est souvent modifié afin d'utiliser une métrique intermédiaire correspondant à la précision des calculs. Plus précisément, la métrique utilisée dans notre cas

correspond à la puissance du bruit de quantification P_b en sortie du système. Ainsi, le problème d'optimisation présenté à l'équation 4.6 est reformulé pour obtenir le problème présenté à l'équation 4.7.

$$\min\left(\mathcal{C}(\mathbf{w})\right) \quad \text{tel que} \quad P_b(\mathbf{w}) \leq P_{b_{max}}. \tag{4.7}$$

La résolution de ce problème d'optimisation nécessite la mise en œuvre de trois éléments principaux correspondant à l'évaluation de la fonction de coût, à l'évaluation de la fonction de contrainte et au choix de l'algorithme d'optimisation. L'évaluation des performances ou de la précision en fonction de la largeur des données a été présentée dans le chapitre 3. La définition d'un algorithme d'optimisation fait l'objet de la partie 4.3.1. L'évaluation de la fonction de coût et la problématique du partage des ressources sont abordées dans la partie 4.3.2.

4.3.1 Algorithme d'optimisation

Encadrement :	Hai-Nam Nguyen, doctorat à soutenir en décembre 2011 [64]
Conférences :	ICASSP 2011 [Ménard 11], SCOPES 2004 [Ménard 04b]
	Eusipco 2011 [Nguyen 11], ReCoSoc 2005 [Hannig 05],
Revues :	Journal on Advances in Signal Processing 2006 [Ménard 06]
Collaboration :	projet ANR ROMA

La définition d'un algorithme d'optimisation pour résoudre le problème présenté à l'équation 4.7 nécessite tout d'abord de classifier les architectures afin de connaître l'ensemble des valeurs prises par les variables du problème d'optimisation. Ainsi, dans la partie 4.3.1.1, la notion de granularité des largeurs supportées est présentée et les architectures sont classées en fonction de ce critère. Dans la partie 4.3.1.2, des algorithmes d'optimisation des largeurs des données sont proposés pour des architectures ayant une granularité fine en termes de largeurs supportées. Dans la partie 4.3.1.3, des algorithmes sont proposés pour des architectures ayant une granularité moyenne.

4.3.1.1 Classification des architectures

Différentes plateformes peuvent être considérées pour l'implantation d'applications de traitement du signal au sein de systèmes embarqués. Ces différentes plateformes fournissent différentes caractéristiques en termes de granularité de largeurs de données supportées. Trois catégories de granularité de largeurs peuvent être distinguées comme présenté à la figure 4.2. Cette classification dépend du niveau auquel l'architecture est programmée ou configurée.

Pour les architectures à granularité de largeur élevée, uniquement une largeur de donnée est supportée pour chaque type d'opérateur. Dans ce cas, l'architecture ne fournit aucun mécanisme supportant une autre largeur de donnée même pour des opérations en multi-précision.

Pour les architectures à granularité de largeur fine, toutes les largeurs ou une majorité des largeurs comprises dans un intervalle donné, sont supportées. Classiquement, Δ_w, le pas entre deux largeurs consécutives supportées varie de un à trois bits. Cette granularité est obtenue lorsque l'architecture peut être configurée ou conçue au niveau bit comme pour les ASIC ou pour les FPGA lorsque les éléments logiques sont utilisés.

Pour les architectures à granularité de largeur moyenne, pour chaque opérateur, plusieurs largeurs sont supportées. Ce niveau de granularité est présent dans de nombreux processeurs ou dans les architectures reconfigurables au niveau opérateur (CGRA, *Coarse Grained Reconfigurable Architectures*) lorsque que celles-ci ciblent des applications de traitement du signal et dans les FPGA à travers les ressources dédiées. Le support de largeurs multiples est obtenu en utilisant les techniques de parallélisme au niveau données (SWP [3] : *Sub-Word Parallelism*) ou de multi-précision (MP).

3. Le terme SIMD (*Single Instruction Multiple data*) est aussi employé pour dénommer cette technique.

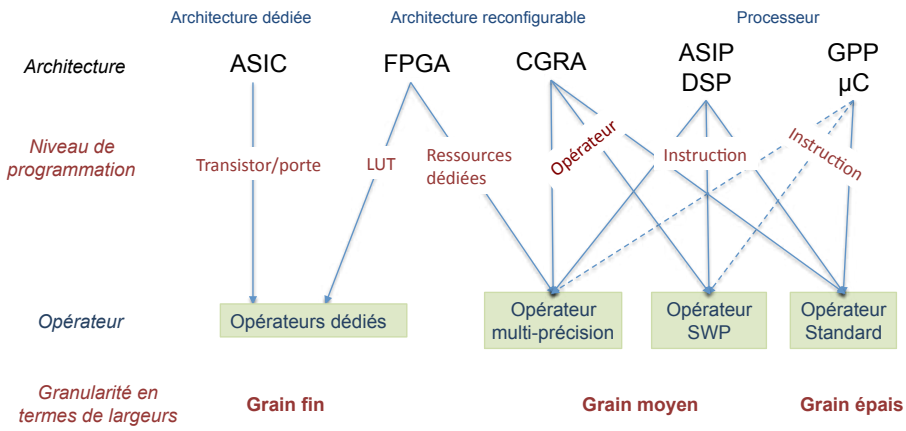


FIGURE 4.2 – Granularité en termes de largeurs supportées pour différents types d'architectures.

Parallélisme de données L'idée principale de l'exploitation du parallélisme de données est d'utiliser un opérateur pour exécuter, en parallèle sur celui-ci, des opérations sur des données de taille plus faible. Cette technique SWP divise un opérateur manipulant des données de largeur w_{SP} bits afin de pouvoir exécuter en parallèle k opérations sur des sous-mots de largeur w_{SP}/k [40] [3].

Le traitement des données en parallèle au sein d'un même opérateur engendre des contraintes au niveau du placement des données en mémoire. Ces données doivent être alignées correctement les unes par rapport aux autres. Les données doivent être rangées en mémoire en fonction de l'ordre de traitement de celles-ci.

Les architectures supportant cette technique doivent fournir des capacités de concaténation et d'extraction de fractions de mot afin de pouvoir modifier la largeur des données. L'efficacité d'une solution utilisant les opérations SWP réside dans la limitation du surcoût lié aux instructions de concaténation ou d'extraction. En effet, ce surcoût ne doit pas annihiler le gain obtenu par la réalisation d'opérations en parallèle.

Multi-précision L'idée principale des opérations en multi-précison (MP) est d'utiliser un ensemble d'opérateurs manipulant des données en simple précision (SP) sur w_{SP} bits pour effectuer des opérations sur des données ayant une largeur supérieure à w_{SP} bits. L'addition de deux données sur w_{MP} bits nécessite N_{MP}^{add} additions SP sur w_{SP}^{add} bits. Le terme N_{MP}^{add} est défini à l'équation 4.8. La multiplication MP de w_{MP1} bits par w_{MP2} bits nécessite N_{MP}^{mult} multiplications SP de w_{SP}^{mult} bits. Le terme N_{MP}^{mult} est défini à l'équation 4.8.

$$N_{MP}^{add} = \left\lceil \frac{w_{MP}}{w_{SP}^{add}} \right\rceil \qquad N_{MP}^{mult} = \left\lceil \frac{w_{MP1}}{w_{SP}^{mult}} \right\rceil \cdot \left\lceil \frac{w_{MP2}}{w_{SP}^{mult}} \right\rceil . \tag{4.8}$$

Comme indiqué dans l'équation 4.8, les calculs en MP augmentent significativement le nombre d'opérations à effectuer et en particulier pour la multiplication. Cette technique permet de traiter efficacement des opérations dont la largeur des opérandes d'entrée est asymétrique. Ainsi, si les largeurs des opérandes d'une opération sont optimisées indépendamment, le coût de l'implantation peut être réduit par rapport au cas où les largeurs des deux entrées sont identiques.

Deux types d'implantation des techniques multi-précision peuvent être distingués. Une implantation spatiale d'une opération MP peut être considérée. Dans ce cas, plusieurs opérateurs SP sont utilisés en parallèle. Ces opérateurs sont inter-connectés entre eux afin de composer un macro-opérateur permettant de traiter des données ayant une largeur plus importante. Un pipeline peut être mis en œuvre pour augmenter la cadence des traitements. Cette technique est très utilisée dans le cas d'architectures reconfigurables et plus particulièrement pour les FPGA à travers l'utilisation des ressources dédiées.

Une implantation temporelle d'une opération MP peut être considérée lorsque le nombre de ressources disponibles au sein de l'architecture est limité. Ainsi, un partage de l'opérateur au cours du temps est effectué pour pouvoir exécuter la suite d'opérations. L'opération MP est décomposée en une suite d'opérations SP

exécutées séquentiellement. Un microcode est associé à chaque opération MP. Plusieurs cycles sont nécessaires pour calculer une opération MP. Cette technique est utilisée dans le cas des processeurs lorsque le nombre d'opérateurs disponibles est limité.

Opérateurs dédiés L'implantation d'applications au sein d'un ASIC ou d'un FPGA nécessite de concevoir et développer des opérateurs spécifiques de largeurs arbitraires. Étant donné que les ASIC sont définis au niveau porte logique et les FPGA au niveau élément logique, ces opérateurs permettent d'obtenir une granularité fine en termes de largeurs égale à un bit. Cependant, pour les FPGA cette granularité fine est obtenue au détriment des performances de l'opérateur en terme de surface et de latence. La réduction d'un bit pour un opérateur doit permettre de réduire le coût de celui-ci. En conséquence, ce type d'architecture permet d'obtenir de nombreux compromis entre le coût de l'opérateur et la précision (largeur des données) fournie par celui-ci. Ainsi, la frontière d'efficacité de Pareto du coût de l'implantation en fonction de la précision des calculs conduit à une diversité de points.

4.3.1.2 Architectures à grain fin en termes de largeurs supportées

Dans le cas d'architectures à grain fin en termes de largeurs supportées (ASIC, FPGA avec LUT [4]), la largeur des opérateurs est à définir par le concepteur. Ainsi, le domaine de définition des variables d'optimisation correspond potentiellement à l'ensemble $\mathbb{N}^*$. Cependant celui-ci est restreint à des valeurs raisonnables, p. ex. de 6 à 40 bits.

Dans cette partie, les travaux réalisés dans le cadre de la thèse d'Hai-Nam Nguyen (2007–2011) sont présentés. La nouvelle approche proposée est basée sur les algorithmes de recherche locale stochastiques. Ces algorithmes combinent les approches heuristiques déterministes et stochastiques. L'algorithme de recherche locale déterministe utilisé est basé sur une recherche avec tabous.

Algorithme glouton et de recherche avec tabous Différents algorithmes gloutons ont été proposés pour l'optimisation de la largeur des données. Les algorithmes utilisant une stratégie *meilleure descente*, dénommés *max–1 bit*, débutent avec une solution respectant la contrainte de précision. Par exemple, toutes les variables sont fixées à leur valeur maximale. Ensuite, l'algorithme itère tant que la contrainte est satisfaite. A chaque itération, la valeur (largeur de l'opération) d'une variable est réduite. Les algorithmes utilisant une stratégie *ascension modérée*, dénommés *min+1 bit*, débutent avec la combinaison des largeurs minimales $\mathbf{w}_{clm}$. La largeur minimale $\mathbf{w}_{clm}(i)$ d'une variable i correspond à la valeur minimale de cette variable permettant de respecter la contrainte de précision lorsque toutes les autres variables sont fixées à leur valeur maximale. La solution $\mathbf{w}_{clm}$ ne respecte pas la contrainte de précision. L'algorithme itère tant que la contrainte de précision n'est pas satisfaite. A chaque itération, la valeur (largeur de l'opération) d'une variable est augmentée.

Différents métriques ∇_k peuvent être utilisées pour déterminer le choix de la direction de déplacement au sein de l'espace de recherche. Soit le vecteur $\mathbf{w}_{\Delta,k,d}$ correspondant à la position suivante pour la $k^{ème}$ variable et défini par rapport à la position courante $\mathbf{w}$ de la manière suivante :

$$\mathbf{w}_{\Delta,k,d}(i) = \begin{cases} \mathbf{w}(k) + (-1)^d \cdot \Delta_w & \text{si} \quad i = k \\ \mathbf{w}(i) & \text{si} \quad i \neq k \end{cases} \tag{4.9}$$

où d représente le sens de déplacement et est égal à 0 pour l'algorithme *min+1 bit* et est égal à 1 pour l'algorithme *max–1 bit*. Δ_w représente la différence entre deux largeurs consécutives supportées.

La métrique proposée dans [14] correspond au gradient sur la précision :

$$\nabla_{k/P_b} = \frac{P_b(\mathbf{w}) - P_b(\mathbf{w}_{\Delta,k,d})}{|\mathbf{w}_{\Delta,k,d} - \mathbf{w}|_1}. \tag{4.10}$$

4. *Look-Up Table.*

Cette métrique permet de sélectionner la direction fournissant la meilleure amélioration de la précision. Cependant, l'augmentation de coût n'est pas prise en compte. Afin de choisir le meilleur compromis en termes de coût et de précision, la métrique ∇_k, utilisée dans [44], est définie par la relation suivante :

$$\nabla_k = \nabla_{k/P_b C} = \frac{\nabla_{k/P_b}}{\nabla_{k/C}} = \frac{P_b(\mathbf{w}) - P_b(\mathbf{w}_{\Delta,k,d})}{C(\mathbf{w}_{\Delta,k,d}) - C(\mathbf{w})}. \tag{4.11}$$

L'algorithme proposé pour améliorer les algorithmes gloutons combine la recherche avec tabous et les algorithmes gloutons *min+1 bit* et *max–1 bit*. Cet algorithme est détaillé dans [Nguyen 11]. La métrique définie à l'équation 4.11 est utilisée pour sélectionner la direction de déplacement. L'objectif de l'algorithme glouton *max–1 bit* est de descendre jusqu'à la limite $P_{b_{max}}$ puis de s'arrêter. Dans notre cas, l'exploration est poursuivie en espérant trouver une meilleure solution. Lorsque la limite $P_{b_{max}}$ est dépassée, la variable considérée est ajoutée à la liste des tabous puis le sens de recherche d est inversé. Ensuite, l'algorithme poursuit tant que la limite n'est pas dépassée, puis ajoute cette variable à la liste des tabous et inverse le sens et réitère pour chaque variable. Le processus s'arrête lorsque toutes les variables appartiennent à la liste des tabous. Si une variable atteint sa valeur maximale dans le sens ascendant ($d = 0$) ou sa valeur minimale dans le sens descendant ($d = -1$), elle est ajoutée à la liste des tabous. Pour chaque itération, la variable ayant la valeur de ∇_k la plus élevée est choisie si le sens est ascendant, ou avec la plus petite valeur de ∇_k dans le cas contraire. Cela permet d'avoir la meilleure efficacité précision/coût lorsque l'objectif est d'augmenter la précision. Dans la direction descendante, l'objectif est de diminuer le coût tout en conservant une diminution de précision la plus faible, ainsi la variable ayant une valeur de ∇_k minimale est sélectionnée. Il est possible de montrer [64] que l'algorithme de recherche avec tabous conduit à une solution égale ou meilleure que celle obtenue par l'algorithme glouton utilisé.

Algorithme de recherche locale stochastique Les algorithmes simples de recherche locale comme le glouton ou la recherche avec tabous permettent de trouver rapidement un optimum local mais la solution est plus ou moins éloignée de la solution optimale. De plus, les algorithmes étant déterministes, une nouvelle exécution de l'algorithme d'optimisation conduit à la même solution et ne permet pas d'obtenir une meilleure solution. Afin de contourner ces problèmes, plusieurs études ont été menées sur les algorithmes stochastiques de recherche locale. Dans ce cadre, l'algorithme GRASP (*Greedy Randomized Adaptive Search Procedure*) [36] a été utilisé pour optimiser la largeur des données. Cet algorithme est détaillé dans [Nguyen 11].

L'algorithme GRASP est itératif et chaque itération est composée de deux étapes correspondant à une phase de construction et à une phase de raffinement. Chaque itération fournit une solution et la meilleure solution obtenue pour les N_{GRASP} itérations est retenue. Le nombre d'itérations est défini à priori et résulte d'un compromis entre la qualité de la solution obtenue et le temps d'optimisation.

Dans la phase de construction, une solution est obtenue à l'aide d'un algorithme glouton aléatoire. Pour chaque itération, le choix de la direction de déplacement est réalisé aléatoirement. A chaque itération de l'algorithme glouton aléatoire, la métrique de choix de direction de déplacement ∇_k (équation 4.10 ou 4.11) est calculée pour chaque variable $\mathbf{w}(k)$. Les candidats ayant les valeurs les plus élevées de ∇_k sont placés dans la liste RCL (*Restricted Candidat List*) de taille N_{RCL}. La direction de déplacement est choisie aléatoirement parmi les N_{RCL} de la liste RCL et la valeur de la variable associée est augmentée. Cet algorithme glouton itère tant que la contrainte de précision n'est pas respectée.

Dans la phase de raffinement, une recherche locale est utilisée pour améliorer la solution. La solution issue de la phase de construction est utilisée comme point de départ pour cette recherche locale et l'algorithme glouton avec tabous présenté précédemment est utilisé pour améliorer cette solution.

Comparaison des algorithmes L'analyse et la comparaison des méthodes d'optimisation se basent sur deux critères que sont la qualité de la solution obtenue et la rapidité d'obtention de cette solution. L'objectif principal de ce problème d'optimisation est d'obtenir la solution de meilleure qualité en un temps raisonnable. Pour ces expérimentations, les applications utilisées sont un filtre à réponse impulsionnelle infinie (IIR) d'ordre 8 composé de 4 cellules d'ordre 2, une FFT sur 64 points et un filtre adaptatif NLMS sur 128 points. Le nombre de variables à optimiser est de 14, 18 et 36 pour le filtre IIR, 8, 12, 20, 28 pour la FFT et 7,

10, 13, 18, 25, 49 pour le filtre NLMS. Ces différentes valeurs sont obtenues en modifiant l'affectation des opérations aux opérateurs. Les expérimentations ont été réalisées à l'aide de l'outil MATLAB.

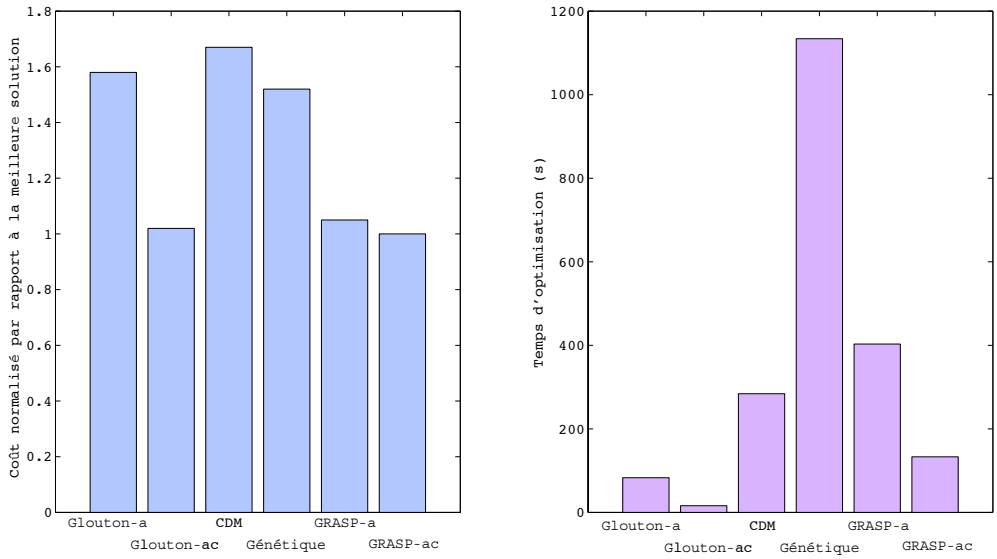


FIGURE 4.3 – a) Coût d'implantation normalisé obtenu pour les différentes méthodes testées. Le coût est normalisé par rapport à la meilleure solution. b)Temps d'exécution (s) des différentes méthodes d'optimisation.

Dans un second temps, les algorithmes GRASP ont été comparés à ceux proposés dans la littérature. L'algorithme GRASP-a utilise la métrique présentée à l'équation 4.10 pour le choix de la direction de déplacement et GRASP-ac utilise la métrique présentée à l'équation 4.11. Les algorithmes utilisés sont un glouton utilisant la métrique présentée à l'équation 4.10 (glouton-a), un glouton utilisant la métrique présentée à l'équation 4.11 (glouton-ac), un algorithme de recherche locale proposé par Han dans [42] (CDM [5]), et l'algorithme génétique proposé par Han dans [41]. La nature aléatoire des algorithmes génétiques et GRASP, nécessite d'exécuter plusieurs fois l'algorithme (20 fois), puis le résultat moyen est utilisé pour l'analyse. La qualité de la solution obtenue dans le cas du filtre IIR pour 36 variables est présentée à la figure 4.3.a. Le coût normalisé par rapport à la meilleure solution est fourni pour les différents algorithmes testés. Le surcoût des approches génétique et CDM est respectivement de 50% et 63%. L'approche glouton-a conduit aussi à un surcoût élevé de 58%. Pour cette configuration, les algorithmes glouton-ac et GRASP-ac conduisent à des résultats proches. L'algorithme GRASP-a fournit une solution légèrement moins bonne que GRASP-ac.

A travers les différentes applications considérées, pour les 224 cas testés, la meilleure solution est toujours obtenue avec l'un des deux algorithmes GRASP. La meilleure solution est obtenue dans 70% des cas avec l'algorithme GRASP-ac et dans 30% des cas avec l'algorithme GRASP-a. Les algorithmes glouton-ac et GRASP-ac ont été comparés plus en détails. Pour les différents cas testés, l'algorithme GRASP-ac conduit toujours à une solution meilleure que celle obtenue par l'algorithme glouton-ac. Sur les 224 cas testés, le surcoût de la solution glouton-ac par rapport à GRASP-ac est en moyenne de $4,5\%$ et au maximum de 20%. L'algorithme GRASP-a permet d'obtenir une solution de qualité nettement supérieure à celle obtenue avec l'algorithme glouton-a. Le surcoût de l'algorithme glouton-a par rapport à GRASP-a est en moyenne de 83%

Pour les différentes méthodes testées, le temps d'exécution obtenu pour les filtres IIR avec 36 variables est fourni à la figure 4.3.b. Le meilleur temps d'optimisation est obtenu avec l'algorithme glouton-ac. L'algorithme

5. *Complexity-and-distortion measure.*

génétique conduit à des temps nettement plus élevés que les autres méthodes. Le temps d'exécution des algorithmes GRASP est significativement plus élevé que celui de l'algorithme glouton utilisant la même métrique de recherche de direction. Mais, le temps d'exécution de l'algorithme GRASP-ac est du même ordre de grandeur que celui du glouton-a. Pour cette configuration, les algorithmes GRASP et glouton utilisant la métrique ac (équation 4.11) conduisent à des temps nettement plus faibles que leur homologue utilisant la métrique a. Ce phénomène est lié au fait que ces algorithmes utilisent moins d'itérations pour atteindre la solution.

4.3.1.3 Architectures à grain moyen en termes de largeurs supportées

Pour les architectures à grain moyen en termes de largeurs supportées, chaque variable du problème d'optimisation ne peut prendre que quelques valeurs. Le domaine de définition très restreint des variables permet de modéliser ce problème de minimisation sous la forme de la recherche d'un chemin au sein d'un arbre. Un algorithme de séparation et évaluation progressive (SEP) peut être utilisé pour explorer efficacement cet arbre. Dans [Ménard 06], nous avons utilisé cet algorithme et présenté différentes techniques pour restreindre l'espace de recherche. Pour des problèmes d'optimisation avec un nombre de variables élevé et des architectures pouvant supporter un nombre de largeurs relativement important, il est nécessaire de restreindre encore plus fortement l'espace de recherche afin d'obtenir des temps d'exécution raisonnables. Ainsi, dans le cadre du projet ROMA présenté dans la partie 5.2.2, nous avons été amené à proposer une nouvelle approche d'optimisation des largeurs pour les architectures à grain moyen en termes de largeurs supportées. En effet, l'architecture développée dans ce projet permet de supporter pour les différentes opérations 5 largeurs différentes.

L'approche proposée combine un algorithme glouton afin d'obtenir rapidement une solution de bonne qualité puis utilise un algorithme SEP pour raffiner cette solution. La solution issue de l'algorithme glouton est raffinée car cet algorithme ne garantit pas de conduire à la solution optimale. Cette approche permet de restreindre l'espace de recherche aux solutions entourant la solution initiale. Sans dégrader la qualité de la solution, cette approche permet de réduire significativement le temps d'optimisation par rapport à une approche SEP seule. Cette approche est détaillée dans [Ménard 11].

4.3.1.4 Conclusion

Dans cette partie, les algorithmes proposés pour l'optimisation de la largeur des données ont été présentés. Dans le cas d'architectures à grain fin en termes de largeurs supportées, l'algorithme d'optimisation des largeurs proposé est basé sur une recherche locale stochastique GRASP. Cet algorithme combine un algorithme glouton aléatoire, permettant de trouver une solution initiale, et un algorithme de recherche locale basé sur la recherche avec tabous pour raffiner la solution initiale. Les comparaisons avec les autres méthodes montrent que la meilleure solution est toujours obtenue avec la solution GRASP mais pas toujours avec la même métrique de recherche de la meilleure direction ∇_k (équations 4.10 et 4.11).

Les temps d'optimisation obtenus avec l'algorithme GRASP, sont plus élevés que pour les algorithmes gloutons. Pour contrecarrer ce problème, nous pouvons proposer une approche fournissant une solution s'améliorant au cours du temps. La taille N_{RCL} de la liste RCL peut être ajustée au cours des itérations. En débutant avec N_{RCL} égal à 1, la solution correspondant à l'algorithme glouton est obtenue, puis ensuite la taille N_{RCL} est augmentée progressivement afin d'avoir plus de diversité et ainsi améliorer la probabilité d'obtenir une meilleure solution. Ceci permet à l'utilisateur d'obtenir, en fonction du temps dont il dispose pour l'optimisation, différents compromis entre la qualité de la solution et le temps d'optimisation. Une seconde piste est de combiner au cours des différentes itérations les deux métriques $\nabla_{k/Pb}$ et $\nabla_{k/PbC}$.

4.3.2 Synthèse d'architecture à largeurs multiples

Encadrement :	Nicolas Hervé, doctorat en 2007 [44]
Conférences :	SIPS 2005 [Herve 05], ARC 2007 [Hervé 07], Sympa'05 [Hervé 05a], GRETSI 2005 [Hervé 05b]
Revues :	Journal of Electrical and Computer Engineering, 2012 [Ménard 12]
Collaboration :	projet RNTL OSGAR [Blanc 06]

Dans le cadre d'une implantation matérielle, l'objectif est de synthétiser une architecture optimisée par rapport aux contraintes fournies par l'utilisateur. Le nombre d'unités fonctionnelles et les caractéristiques de celles-ci en termes de largeur étant choisis lors de la phase de synthèse, ce processus offre de nombreux degrés de liberté pour l'optimisation de la largeur des données. Dans cette section, les travaux réalisés dans le cadre de la thèse de Nicolas Hervé [44] et du projet RNTL [6] OSGAR sont présentés. Une méthodologie de synthèse d'architecture à largeurs multiples a été définie. L'objectif est de générer l'architecture de coût minimal pour une contrainte de précision et une contrainte de latence donnée.

Le projet OSGAR (Outils de Synthèses Génériques pour Architectures Reconfigurables) a été financé dans le cadre du programme RNTL (2003–2005). Les partenaires du projet OSGAR étaient l'IRISA (Lannion), le CEA LIST (Saclay), l'Université de Bretagne Occidentale (Brest) et TNI [7] (Brest). L'objectif du projet était d'étudier et de développer des outils de synthèse de haut niveau permettant, à partir de code C, ou de spécifications semi-formelles, de générer automatiquement les configurations pour plusieurs circuits reconfigurables. Dans le cadre de ce projet, nous avons adapté notre outil de synthèse de haut niveau en prenant en compte l'optimisation de la largeur des opérateurs, modélisé les architectures reconfigurables et validé l'infrastructure logicielle globale sur deux applications du domaine du traitement d'image et des communications numériques.

4.3.2.1 Approche itérative pour le couplage de la synthèse et de l'optimisation des largeurs

Lors de la conception d'un système en virgule fixe, différentes stratégies de répartition des largeurs au sein de l'architecture sont possibles. La solution la plus simple, dénommée Sl_{u-arch}, consiste à choisir une largeur unique pour l'ensemble des données de l'architecture. Une solution un peu plus évoluée, dénommée Sl_{u-opr}, consiste à choisir une largeur propre à chaque type d'opérateurs. Ces deux solutions ont l'avantage de réduire l'espace de recherche lors de l'optimisation de la largeur des données. Ces deux approches sont efficaces dans le cas d'une implantation temporelle de l'algorithme ou un seul opérateur par type d'opération est utilisé. Mais dès que du parallélisme est nécessaire pour satisfaire les contraintes temporelles, contraindre les différents opérateurs à avoir la même largeur conduit à une solution sous optimale. L'approche opposée correspondant à une implémentation spatiale consiste à utiliser une ressource dédiée à chaque opération. Dans ce cas, aucun partage de ressources n'est effectué et chaque opérateur peut posséder sa propre largeur.

Dans la majorité des cas, le processus de synthèse d'architecture conduit à une solution intermédiaire intégrant du parallélisme et du partage de ressources. Comme les opérateurs sont partagés dans le temps entre plusieurs opérations, les opérations effectuées sur le même opérateur doivent être connues afin d'intégrer cette information dans le processus d'optimisation des largeurs. Cette information de répartition des opérations sur les opérateurs ne peut-être connue uniquement après le processus de synthèse d'architecture, plus exactement après l'étape d'assignation. Or, la synthèse nécessite de connaître les largeurs des opérandes afin de sélectionner les largeurs adéquates pour les opérateurs. De plus, l'assignation dépend de l'ordonnancement, ce dernier est lui-même dépendant des temps d'exécution des opérateurs et ces temps d'exécution sont fonction de la largeur des opérateurs. Ceci conduit donc à un paradoxe où chacun des deux processus (la synthèse d'architecture et l'optimisation des largeurs) nécessite des informations issues de l'autre processus.

6. Réseau National en Technologies Logicielles.
7. La société TNI devenue GenSys a été rachetée par Dassault System.

Dans le cadre de nos travaux, une approche de synthèse d'architecture à largeurs multiples a été proposée. L'objectif de notre approche est de trouver le groupement des opérations et la combinaison des largeurs pour ce groupement permettant au processus de synthèse d'obtenir l'architecture avec le plus faible coût et respectant les contraintes de temps et de précision.

L'intérêt de la constitution de ces groupes d'opérations est d'anticiper la phase de synthèse en prédisant, pour l'optimisation des largeurs, la répartition des opérations sur les opérateurs. Ainsi, les opérations censées s'exécuter sur un même opérateur seront optimisées avec la même largeur. Lors du processus de synthèse, les opérations d'un même groupe ne seront pas obligatoirement exécutées sur un même opérateur. En effet, le processus d'optimisation peut considérer plus opportun de placer des opérations sur d'autres opérateurs de largeur plus importante. En conséquence, le nombre d'opérateurs après synthèse peut être différent du nombre de groupes considérés avant la synthèse.

Ainsi, pour converger vers une solution optimisée pour laquelle le nombre de groupes estimé et le nombre d'opérateurs sont identiques, un processus itératif est employé. Une itération de ce processus est décomposée en quatre étapes, telles que présentées figure 4.4. Les résultats de la synthèse de l'itération i sont utilisés pour déterminer puis constituer les groupes lors de l'itération $i+1$. Les différentes étapes sont présentées ci-dessous. Le processus global se termine lorsque l'allocation effectuée par la synthèse est en accord avec le groupement soumis ou lorsque le nombre d'itérations atteint une valeur maximale prédéfinie.

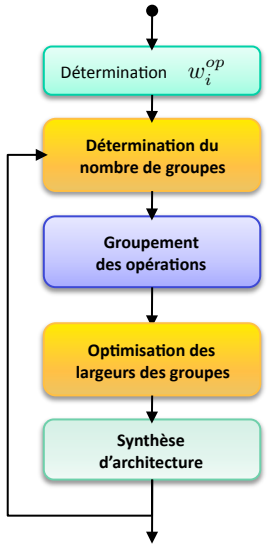


FIGURE 4.4 – Processus global d'optimisation. Les résultats de la synthèse de l'itération i sont utilisés pour déterminer puis constituer les groupes lors de l'itération $i+1$.

Étape 1 : détermination du nombre de groupes. La première étape permet de fixer le nombre de groupes d'opérations utilisés dans la suite du processus. Initialement, pour la première itération, un seul groupe, donc une même largeur par type d'opération, est considéré. Pour les itérations suivantes, le nombre de groupes défini pour chaque type d'opération, correspond au nombre d'opérateurs utilisés par la synthèse d'architecture réalisée à l'itération précédente.

Étape 2 : groupement. Dans la seconde étape, un algorithme de groupement est appliqué afin d'affecter chaque opération à un groupe en fonction de son type, des résultats de synthèse précédents et des résultats d'optimisation des largeurs. Deux algorithmes de groupement ont été proposés. Le premier algorithme réalise un groupement dirigé par la synthèse. Pour cette approche, le groupement des données est directement déterminé à partir des résultats de la synthèse précédente. Toutes les opérations assignées à un opérateur sont

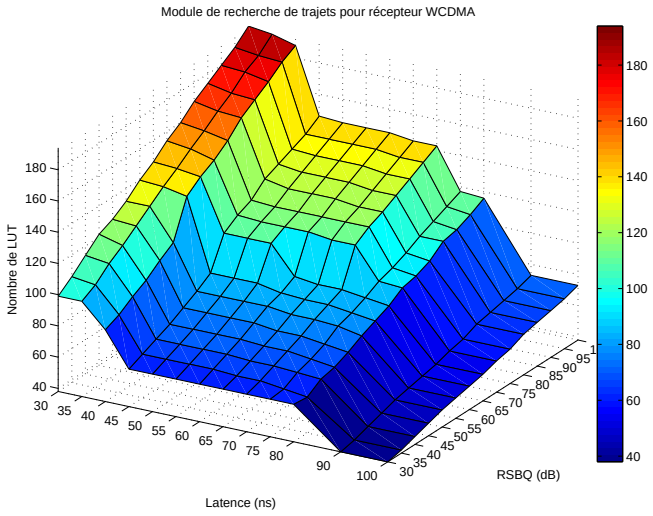


FIGURE 4.5 – Coût en LUT en fonction des contraintes de précision et de latence pour un module de recherche de trajets d'un récepteur WCDMA.

réunies au sein d'un même groupe lors de l'itération suivante. Le second algorithme utilise une technique avancée de groupement des données et est présenté dans [Herve 05].

Étape 3 : optimisation des largeurs de groupe. Une fois un groupement déterminé, la largeur associée à chaque groupe est optimisée afin de minimiser le coût de l'implantation sous contrainte de précision. Pour ce processus d'optimisation, les différents algorithmes présentés dans la partie 4.3.1 peuvent être utilisés.

Étape 4 : synthèse d'architecture. La dernière étape correspond à la synthèse d'architecture pour le graphe flot de données dont la largeur des opérations a été optimisée à l'étape précédente. Ce processus de synthèse doit supporter une spécification à largeurs multiples et doit pouvoir affecter des opérations à des opérateurs de largeur supérieure.

4.3.2.2 Expérimentations

L'approche de synthèse d'architecture à largeurs multiples proposée ci-dessus a été implantée en utilisant l'outil BSS [67] pour la synthèse d'architecture et une première version de notre outil de conversion en virgule fixe. La technique de regroupement dirigée par la synthèse a été implantée dans l'outil de conversion en virgule fixe.

Frontière d'efficacité de Pareto Notre approche de synthèse d'architecture à largeurs multiples permet d'obtenir, pour une contrainte donnée de latence et de précision, une architecture dont le coût a été optimisé. En testant différentes valeurs de contrainte de latence et de précision, il est possible d'obtenir la frontière d'efficacité de Pareto du coût en fonction de la latence et de la précision et de visualiser les différents compromis possibles entre le coût, la latence et la précision des calculs.

Les résultats obtenus dans le cadre du module de recherche de trajets d'un récepteur WCDMA sont présentés à la figure 4.5. L'architecture ciblée est un FPGA et pour cette expérimentation, uniquement les LUT sont utilisées. La précision est évaluée à travers le Rapport Signal à Bruit de Quantification (RSBQ) exprimé en dB et correspondant au rapport entre la puissance du signal et la puissance du bruit de quantification. Ces résultats montrent une évolution en palier du coût en fonction de la latence et de la précision des calculs. Pour la latence, les ruptures sont liées à la nécessité d'ajouter un ou plusieurs opérateurs travaillant

Applications	Gain sur le coût	
	Moyen	Maximal
FIR	18 %	35 %
FFT	28 %	50 %
IIR	22 %	47 %
Searcher	10 %	20 %

TABLE 4.3 – Gain moyen et maximal obtenu par rapport à la solution Sl_{u-opr} pour différentes contraintes de précision et de latence.

en parallèle pour réduire la latence de l'application.

Pour la précision des calculs, l'évolution est linéaire par morceaux. L'évolution linéaire est liée à la nécessité d'augmenter progressivement la largeur des opérateurs pour augmenter la précision et à la présence importante d'opérations d'addition et de soustraction au sein de l'application. En effet, ces opérations possèdent une évolution linéaire du coût en fonction de la largeur des opérandes. Comme pour la latence, les ruptures sont liées à la nécessité d'ajouter un ou plusieurs opérateurs pour satisfaire les contraintes. L'augmentation de la précision nécessite des opérateurs de largeur plus élevée et en conséquence de latence plus importante. Lorsque cette augmentation de latence ne permet plus de satisfaire la contrainte de latence globale, des opérateurs supplémentaires travaillant en parallèle sont nécessaires. La localisation de ces ruptures dans la frontière d'efficacité de Pareto est fortement liée à la période de l'horloge. La discrétisation de la latence des opérateurs en nombre de cycles entraîne ces effets de paliers.

Algorithme de groupement dirigé par la synthèse Dans cette partie, le groupement des données est réalisé directement à partir des résultats de synthèse. L'évolution du coût pour les différentes itérations est présentée à la figure 4.6 dans le cas d'une FFT pour trois couples de contraintes (précision, latence). La méthode proposée permet de réduire le coût de l'implantation par rapport à la solution classique Sl_{u-opr}. Cette solution classique correspond à la solution de départ pour laquelle une largeur unique est utilisée par type d'opération. Pour illustrer cette réduction du coût, le gain par rapport à la solution Sl_{u-opr} a été mesuré pour quatre applications de traitement du signal. Ces expérimentations ont été réalisées pour différentes contraintes de précision et de latence et les gains moyens et maximaux sont reportés dans le tableau 4.3. Pour ces quatre applications le gain est significatif, il est en moyenne autour de 20% et peut atteindre jusqu'à 50%. Comme montré à la figure 4.6, le processus itératif permet de réduire progressivement le coût de l'implantation. Cependant, les résultats montrent que l'évolution du coût n'est pas monotone et que celui-ci peut augmenter au cours des différentes itérations. En conséquence, la solution retenue correspond à celle ayant donnée les meilleurs résultats au cours des N_i itérations réalisées. L'analyse en détails de la non monotonicité n'a pas pu être réalisée par manque de temps.

Pour pouvoir comparer notre méthode aux méthodes existantes, nous avons analysé le gain obtenu par rapport à la solution Sl_{u-arch} (une même largeur pour tous les opérateurs). Pour l'application FFT, le gain obtenu par rapport à la solution Sl_{u-arch} est compris entre 33% et 78%. Dans [80], une approche séquentielle est utilisée. La largeur des données est d'abord optimisée et la synthèse d'architecture est réalisée par la suite. Cette approche conduit à des gains plus faibles. Ceux-ci sont au maximum de 40%. Ces résultats montrent l'intérêt de coupler les processus d'optimisation des largeurs et de synthèse d'architecture. Dans [13], l'approche proposée combine l'optimisation de la largeur des données et la synthèse d'architecture au sein d'un processus d'optimisation utilisant le recuit simulé. Les gains obtenus par rapport à l'approche à largeur uniforme sont compris entre 22% et 77%. Ceux-ci sont proches de ceux obtenus avec notre méthode. Cependant, notre méthode permet d'obtenir une solution avec un nombre d'itérations beaucoup plus faible (10 itérations) et ainsi, conduit à des temps d'optimisation plus faibles. De plus, notre méthode permet de combiner l'optimisation de la largeur des données et la synthèse d'architecture sans modifier le processus de synthèse d'architecture. Ainsi, des outils commerciaux de synthèse de haut niveau peuvent être utilisés.

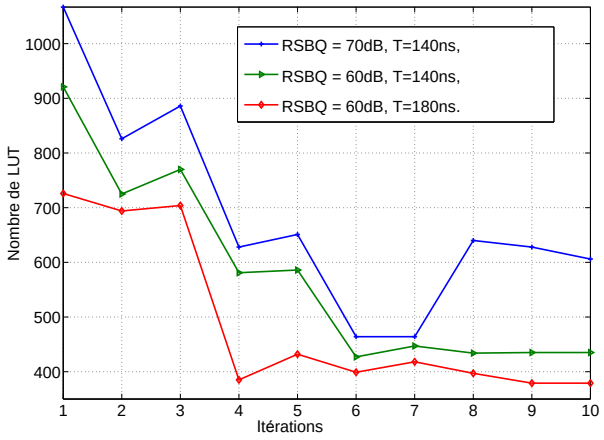


FIGURE 4.6 – Évolution du coût au cours des itérations dans le cas d'une FFT pour trois couples de contraintes (précision : RSBQ (dB), latence : T (s)).

4.3.2.3 Conclusion

Dans cette partie, l'approche proposée pour la synthèse d'architecture à largeurs multiples a été présentée. Elle permet de générer l'architecture dont le coût a été minimisé pour des contraintes de précision et de latence données. Par rapport à de nombreuses approches existantes, les processus d'optimisation de la largeur des données et de synthèse d'architecture ne sont pas traités de manière séquentielle. Un processus itératif est mis en œuvre pour combiner les deux processus. Chaque itération réalise le regroupement des données, l'optimisation de la largeur des groupes puis la synthèse d'architecture. Les résultats d'expérimentation montrent des gains significatifs par rapport à des solutions ne supportant pas des largeurs multiples.

Par rapport aux approches existantes combinant les deux processus, le processus de synthèse d'architecture n'est pas modifié et ainsi, des outils de synthèse de haut niveau commerciaux peuvent être utilisés. De plus, le nombre d'itérations est beaucoup plus faible par rapport à l'approche proposée dans [13] et utilisant le recuit simulé pour combiner les deux processus.

Dans l'approche proposée, l'hétérogénéité des ressources de calcul disponibles au sein d'un FPGA n'a pas été prise en compte pour l'évaluation du coût de l'implantation. Une fonction de coût plus élaborée doit être définie pour intégrer l'utilisation des ressources dédiées et des éléments logiques.

4.4 Approche au niveau système

ENCADREMENT : Karthick Parashar, doctorant depuis 2008
CONFÉRENCES : VLSI Design 2010, [Parashar 10c]
COLLABORATION : Programme R&D nano 2012 (ST Microelectronics)

Les concepteurs et développeurs de systèmes embarqués doivent faire face au challenge d'implanter des applications de plus en complexes et intégrant de nombreuses fonctionnalités. L'objectif est d'optimiser globalement le système afin de minimiser le coût total de l'implantation. Réaliser l'optimisation de toutes les largeurs des données du système complet au sein d'un même processus n'est pas réaliste. Le nombre de variables à prendre en compte au sein du processus d'optimisation est beaucoup trop important. Ceci conduit à un nombre d'itérations élevé pour l'optimisation. Dans le cadre de la thèse de Karthick Parashar (2008–2011), une approche hiérarchique est proposée pour pouvoir traiter des systèmes complexes.

4.4.1 Description de l'approche hiérarchique

L'objectif de la conversion en virgule fixe est d'optimiser la largeur des opérations présentes au sein de l'application. Ces largeurs sont regroupées au sein du vecteur $\mathbf{w}$. Le coût d'implémentation $C(\mathbf{w})$ est minimisé tout en garantissant que les performances de l'application $\lambda(\mathbf{w})$ sont supérieures à un seuil minimal λ_{obj}.

$$\min\left(C\left(\mathbf{w}\right)\right) \quad \text{tel que} \quad \lambda\left(\mathbf{w}\right) \geq \lambda_{obj}. \tag{4.12}$$

Ce processus de détermination des largeurs est itératif et nécessite l'évaluation du coût d'implantation et des performances de l'application à chaque itération. Par conséquent, le challenge principal de ce problème d'optimisation consiste en une évaluation, rapide et précise, des performances λ en fonction de la largeur $\mathbf{w}$ des opérations. Pour un système complet, le nombre de variables de ce problème d'optimisation peut devenir rapidement élevé et chercher à optimiser toutes les variables en même temps conduit à une impasse. Comme dans de nombreuses méthodologies de conception de systèmes, nous adoptons une approche hiérarchique pour optimiser la largeur des données.

Considérons le système $\mathbb{B}$ présenté à la figure 4.7. Ce système est composé de N_o opérations de décision O_j et N_b sous-systèmes B_i, intégrant chacun uniquement des opérations dont le modèle de bruit est linéaire. La distinction entre ces deux types d'opérations est liée aux méthodes utilisées pour évaluer les performances globales. Les sous-systèmes B_i pourront être traités par des méthodes analytiques pour évaluer la précision des calculs. La méthode utilisée et l'outil associé sont présentés respectivement dans les parties 3.2.2 et 3.2.3. Les opérations de décision ne peuvent pas être traitées avec le même type d'approche et nécessitent l'utilisation de techniques basées sur la simulation. Le découpage du système en sous-systèmes est à l'imitative du développeur. Mais, celui-ci est aussi contraint par la présence d'opérations de décision imposant les frontières entre les sous-systèmes. Les sous-systèmes B_i peuvent être redécomposés une nouvelle fois avec des sous-systèmes B_{ij} plus petits. Soit $\mathbf{w_{B_i}}$, le vecteur regroupant les largeurs de toutes les opérations du sous-système B_i.

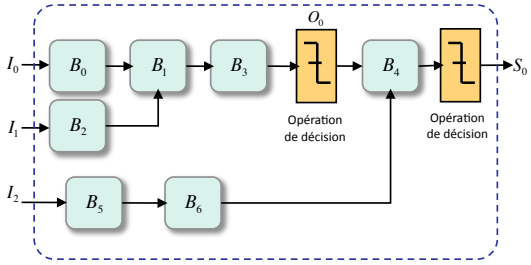


FIGURE 4.7 – Description flot de données d'un système $\mathbb{B}$ sous forme hiérarchique.

Notre approche hiérarchique et la transformation du problème d'optimisation présenté à l'équation 4.12 repose sur le modèle de source de bruit unique présenté dans la partie 3.3.1. Ce modèle permet de modéliser le comportement du système en virgule fixe $\widehat{B_i}$ par le système en précision infinie B_i et une unique source de bruit b_{u_i} située en sortie du système B_i. Ainsi, avec ce modèle utilisé pour l'évaluation des performances, il est possible de réduire toutes les variables $\mathbf{w_{B_i}}$ correspondant aux largeurs des opérations du sous-système B_i en une seule variable P_{b_i} correspondant à la puissance du bruit de quantification en sortie de B_i. Soit $\mathbf{p} = [P_{b_1}, \ldots, P_{b_i}, \ldots, P_{b_{N_b}}]$ le vecteur regroupant les puissances des sources b_{u_i}. Le problème d'optimisation présenté à l'équation 4.12 peut être redéfini de la manière suivante :

$$\min\left(C\left(\mathbf{p}\right)\right) \quad \text{tel que} \quad \lambda\left(\mathbf{p}\right) \geq \lambda_{obj}. \tag{4.13}$$

En d'autres termes, le problème d'optimisation peut être vu comme une répartition *(budgeting)* optimisée de la puissance des sources de bruit vers chaque sous-système, de façon à ce que le critère de performance minimale équivalent en sortie de B soit respecté et que le coût global soit minimisé. La solution à ce problème demande l'évaluation de la fonction de coût C et des performances λ conjointement à un algorithme d'optimisation efficace pour trouver une bonne solution avec un nombre minimum d'itérations.

4.4.1.1 Évaluation du coût

Au niveau du système, l'évaluation du coût global d'implémentation C est obtenue à partir des coûts $C_i(\mathbf{p}(i))$ de chaque sous-système B_i et dépendant de la puissance de bruit associée $\mathbf{p}(i)$. Dans un premier temps, la métrique de coût utilisée correspond à la consommation d'énergie, car un modèle simple du coût global peut être défini. Le coût global C correspond à la somme du coût $C_i(\mathbf{p}(i))$ de chaque sous-système B_i. Le même type de modèle de coût peut être utilisé dans le cas d'une implantation logicielle sur un processeur mono-cœur pour une métrique de coût correspondant au temps d'exécution global de l'application. La modélisation du coût est plus complexe dans le cas général. Dans le cas d'une implantation matérielle, la métrique de coût correspondant à la surface doit prendre en compte le partage des ressources.

Le coût d'implémentation C_i de chaque sous-système B_i pour une valeur donnée de $\mathbf{p}(i)$ doit être évalué. Ce coût doit être minimisé pour une contrainte de puissance du bruit $\mathbf{p}(i)$. Cet objectif se rapporte à l'optimisation de la largeur des opérations et correspond au problème classique présent au sein du processus de conversion en virgule fixe. Pour le sous-système B_i, le coût C_i est minimisé sous la contrainte de précision $\mathbf{p}(i)$. Ce coût dépend de $\mathbf{w_{B_i}}$, la largeur des opérations présentes au sein du sous système B_i. Ce problème d'optimisation peut être énoncé comme suit :

$$\min(C_i(\mathbf{w_{B_i}})) \quad \text{tel que} \quad P_{b_i}(\mathbf{w_{B_i}}) < \mathbf{p}(i), \tag{4.14}$$

avec $P_{b_i}(\mathbf{w_{B_i}})$ la puissance du bruit de quantification du sous-système B_i. L'expression analytique peut être obtenue à partir de la méthode présentée dans la partie 3.2.2 et l'outil associé présenté dans la partie 3.2.3. L'utilisation dans le processus de conversion en virgule fixe d'une approche analytique pour évaluer la précision des calculs permet d'obtenir des temps d'optimisation raisonnables.

Afin de ne pas réaliser une conversion en virgule fixe à chaque évaluation de C_i pour une valeur donnée de $\mathbf{p}(i)$, la frontière d'efficacité de Pareto est déterminée pour chaque sous-système puis elle est utilisée pour chaque évaluation de C_i. Une approximation de la frontière d'efficacité de Pareto est obtenue à partir d'un algorithme glouton modifié.

4.4.1.2 Évaluation des performances

La présence d'opérations de décision au sein de l'application ne permet plus d'utiliser la puissance du bruit de quantification comme métrique pertinente pour évaluer les effets de la précision finie. Ainsi, les performances de l'application sont mesurées directement. Au niveau du système, les performances sont évaluées par l'approche mixte combinant la simulation et les résultats de l'approche analytique présentée dans la partie 3.3.2.

L'évaluation des performances λ pour une valeur donnée de $\mathbf{p}$ est réalisée après l'évaluation du coût C, ainsi pour chaque sous-système B_i, une optimisation de la largeur des opérations $\mathbf{w_{B_i}}$ présentes au sein de B_i a été réalisée pour la contrainte de précision $\mathbf{p}(i)$. En conséquence, la valeur de la largeur des opérations, $\mathbf{w_{B_i}}$, est connue et peut être utilisée pour calculer les paramètres de la source de bruit unique b_{u_i}.

4.4.1.3 Algorithme d'optimisation *Max -δ_P dB*

L'algorithme d'optimisation utilisé est un algorithme glouton s'inspirant des algorithmes *min+1 bit* et *max–1 bit*. Les variables du problème sont les puissances de bruit $\mathbf{p}(i)$ issues du modèle de source de bruit unique. Une phase d'initialisation est tout d'abord exécutée pour trouver les valeurs initiales des variables. Pour cela, les valeurs maximales de la puissance des sources b_{u_i} permettant d'atteindre les performances de l'application sont recherchées :

$$\mathbf{p}^{\max}(i) = \max(\mathbf{p}(i)) \quad \text{tel que} \quad \begin{cases} \mathbf{p}(j) = 0 \quad \forall j \neq i \\ \lambda(\mathbf{p}) \geq \lambda_{obj} \end{cases} \tag{4.15}$$

Pour chaque variable $\mathbf{p}(i)$, la puissance du bruit est augmentée tant que la contrainte sur les performances n'est pas satisfaite, tout en gardant le reste des variables $\mathbf{p}(j)$ à une valeur nulle. Ensuite, les variables $\mathbf{p}(i)$

sont initialisées à leur valeur maximale associée : $\mathbf{p} = \mathbf{p}^{\max}$. Dans ce cas, les performances souhaitées ne sont plus atteintes : $\lambda(\mathbf{p}^{\max}) < \lambda_{obj}$.

Soit $\mathbf{p}_k$ le vecteur des variables obtenu à l'itération k. Pour trouver la meilleure direction pour converger vers une distribution optimisée de la puissance du bruit, le gain apporté par la diminution de la puissance du bruit de chaque variable $\mathbf{p}(i)$ par une valeur de δ_P est exploré. La valeur de δ_P est fixée à 3 dB, soit l'équivalent d'une augmentation de 1 bit. Soit $\delta_{\mathbf{P_i}}$ un vecteur ayant toutes ses éléments nuls, sauf l'élément i, égal à δ_P. Soit $\Delta\lambda_i(\mathbf{p}_k)$ la dérivée partielle de la fonction des performances par rapport à la variable $\mathbf{p}(i)$ évaluée au point $\mathbf{p}_k$ et $\Delta C_i(\mathbf{p}_k)$ la dérivée partielle de la fonction de coût par rapport à la variable $\mathbf{p}(i)$ évaluée au point $\mathbf{p}_k$. Une métrique d est utilisée pour analyser la meilleure direction. Cette métrique est calculée pour chaque variable d'optimisation $\mathbf{p}_k(i)$. L'objectif de cette métrique est de trouver la direction permettant d'augmenter au maximum les performances λ sans augmenter trop le coût global. La métrique $\mathbf{d}_k(i)$ est définie comme le rapport entre $\Delta\lambda_i$ et ΔC_i :

$$\mathbf{d}_k(i) = \frac{\Delta\lambda_i(\mathbf{p}_k)}{\Delta C_i(\mathbf{p}_k)} = \frac{\lambda\left(\mathbf{p}_k - \delta_{\mathbf{P_i}}\right) - \lambda\left(\mathbf{p}_k\right)}{C\left(\mathbf{p}_k - \delta_{\mathbf{P_i}}\right) - C\left(\mathbf{p}_k\right)}. \tag{4.16}$$

La direction l conduisant à la valeur de $\mathbf{d}_k(l)$ la plus élevée est conservée et la variable associée modifiée.

$$\begin{cases} \mathbf{p}_{k+1}(l) & = & \mathbf{p}_k(l) - \delta_P & \text{avec} \quad l = \arg\max(\mathbf{d}_k) \\ \mathbf{p}_{k+1}(m) & = & \mathbf{p}_k(m) & \forall m \neq l \end{cases} \tag{4.17}$$

L'algorithme s'arrête lorsque les performances souhaitées λ_{obj} sont atteintes.

4.4.2 Expérimentations

L'approche hiérarchique a été testée sur le module de décodage sphérique SSFE dont le synoptique est présenté à la figure 3.6 (section 3.3.2.3 à la page 41). Cette application est composée de cinq sous-systèmes délimités par les opérations de décision. En conséquence, au niveau système, le problème d'optimisation est composé de $N_b = 5$ variables $\mathbf{p}(i)$.

Le temps global d'optimisation t_h dépend du nombre de variables N_b au sein du problème d'optimisation, du nombre d'itérations N_{it-h} pour converger vers la solution, et des temps nécessaires pour obtenir la frontière de Pareto de chaque sous système (t_{c-h}) et pour évaluer les performances (t_{mix}). Les performances sont évaluées par l'approche mixte en raison de la présence d'opérations de décision. A chaque itération, il est nécessaire d'évaluer le coût et les performances globales. Ainsi, l'expression du temps global d'optimisation pour l'approche hiérarchique est la suivante :

$$t_h = N_{it-h} \cdot N_b \cdot t_{mix} + t_{c-h} \tag{4.18}$$

Dans le cas de l'approche hiérarchique, le nombre d'itérations est de 11. Le temps t_{c-h} est égal à 186 s et le temps t_{mix} est égal à 10 s. L'utilisation d'une approche analytique pour obtenir la frontière d'efficacité de Pareto permet d'obtenir un temps t_{c-h} raisonnable.

Pour comparer avec l'approche hiérarchique, les largeurs de toutes les opérations ont été optimisées au sein d'un même processus d'optimisation. Dans ce cas, le processus d'optimisation est composé de $N_v = 30$ variables. L'application contenant des opérations de décision, il est nécessaire d'utiliser l'approche mixte pour évaluer les performances du système. Le temps d'évaluation du coût t_{c-h} est très faible et peut être négligé. Dans le cas de l'approche d'optimisation à plat, l'expression du temps global d'optimisation est la suivante :

$$t_f = N_{it-f} \cdot N_v \cdot (t_{c-f} + t_{mix}) \tag{4.19}$$

Le nombre d'itérations est de $N_{it-f} = 279$. Avec l'approche d'optimisation à plat, les performances sont évaluées avec l'approche mixte 8378 fois alors que les performances ne sont évaluées avec l'approche

hiérarchique que 40 fois. Au final, l'approche hiérarchique permet de diviser le temps d'optimisation d'un facteur 144 pour cette application.

Dans [Parashar 10c], le cas d'un récepteur WCDMA a été traité. Actuellement, des expérimentations sur un récepteur MIMO-OFDM sont réalisées. Ce système intègre une FFT, une décomposition QR basée sur le CORDIC et le module de décodage sphérique basé sur le SSFE.

4.4.3 Conclusions

Pour traiter des systèmes complexes, une approche hiérarchique permettant d'optimiser la largeur des données a été proposée. Le système complet est découpé en sous-systèmes en fonction de la complexité des traitements et de la présence d'opérations de décision. Une variable d'optimisation est affectée à chaque sous-système et correspond à la puissance du bruit de quantification en sortie de celui-ci. Ainsi, le problème d'optimisation revient à budgéter au sein de chaque sous-système, la dégradation de précision permettant de minimiser le coût global de l'implantation et de maintenir la précision des calculs. L'évaluation des performances du système repose sur l'approche mixte analytique/simulation et le modèle de source de bruit unique.

Cette approche a été testée sur un récepteur WCDMA et sur un décodeur sphérique. Les résultats des expérimentations montrent la capacité de cette approche à optimiser un système relativement complexe en un temps raisonnable et de réduire fortement ce temps par rapport à une approche non hiérarchique.

4.5 Outil de conversion en virgule fixe

Collaborations : Programme R&D nano 2012 (ST Microelectronics), projet ANR ROMA
Logiciel : ID.Fix (Infrastructure de conversion en virgule fixe)

Nos différents travaux de recherche sur la conversion automatique en virgule fixe sont accompagnés du développement d'une infrastructure logicielle ID.Fix. Celle-ci est développée principalement dans le cadre du projet R&D Nano 2012 avec la société ST Microelectronics. L'objectif de ce projet est de réaliser des transformations du code source afin d'améliorer la synthèse d'architecture réalisée avec des outils de HLS commerciaux tels que Catapult C. Ce projet est composé de trois sous-projets (SP). Le premier SP se concentre sur les transformations de boucles, le second sur la conversion en virgule fixe et le troisième sur la synthèse d'architecture multi-modes. Dans le cadre du second SP, l'objectif est de développer une infrastructure de conversion automatique en virgule fixe. Le synoptique de l'infrastructure ID.Fix est présenté à la figure 4.8.

4.5.1 Spécification de l'outil

4.5.1.1 Entrées de l'outil

Les éléments en entrée de l'outil nécessaires pour réaliser la conversion en virgule fixe d'une application sont les suivants :

- `App.c` : code source d'entrée décrivant en langage C l'application avec des types en virgule flottante. Uniquement un sous-ensemble du langage C est supporté pour décrire l'application. Actuellement, les pointeurs et les structures ne sont pas supportés.
- $P_{b_{max}}$: valeur de la contrainte de précision utilisée pour le processus d'optimisation de la largeur des opérations. Cette contrainte de précision correspond à la puissance du bruit de quantification ciblée en sortie de l'application. Cette métrique d'évaluation de la précision est détaillée dans la partie 3.2. Sa valeur est exprimée en dB.

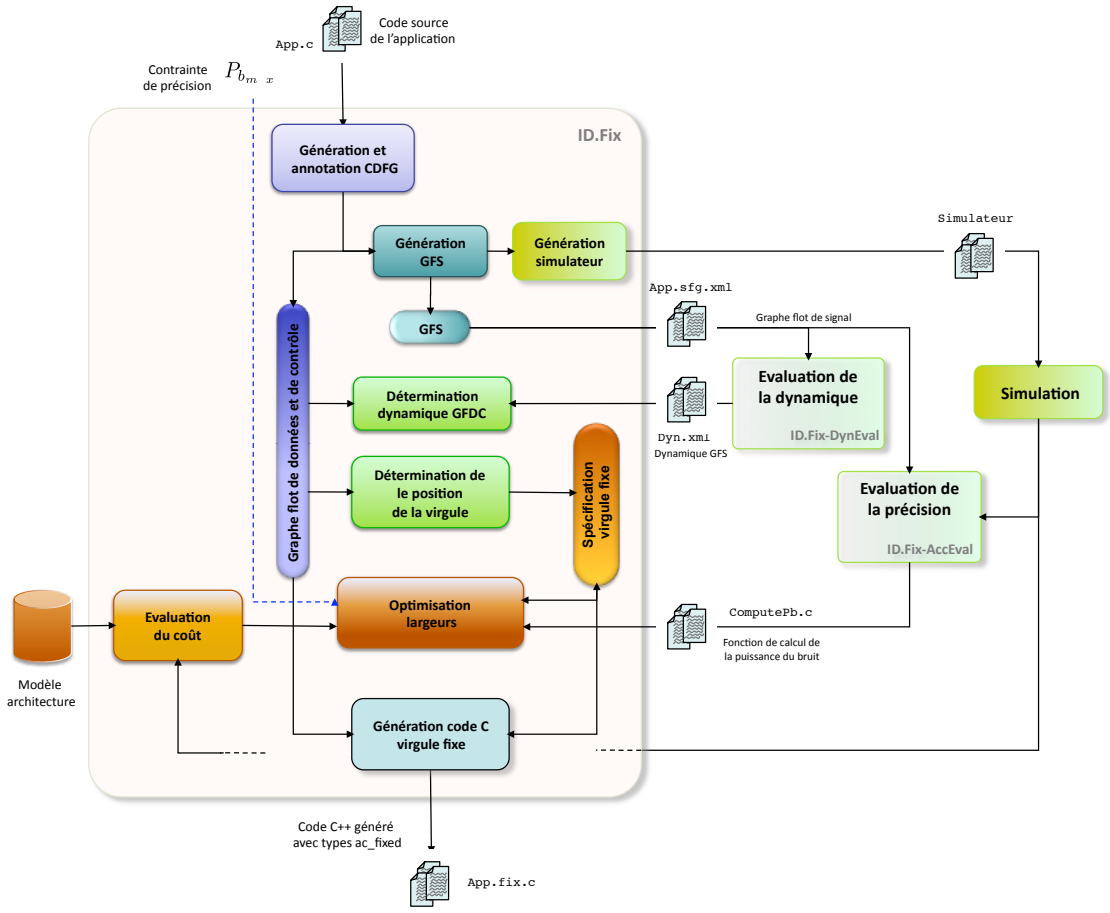


FIGURE 4.8 – Synoptique de l'infrastructure de conversion en virgule fixe ID.Fix.

– `App.Arch.xml` : fichier au format XML définissant le modèle de l'architecture pour évaluer le coût de l'implantation lors du processus d'optimisation de la largeur des opérations. Ce modèle regroupe l'ensemble des largeurs supportées par l'architecture et le coût d'implantation associé.

Différents "pragmas" sont disponibles pour annoter le code C source en vue de spécifier certaines fonctionnalités nécessaires pour la conversion en virgule fixe. Ils permettent de définir la fonction principale à traiter, la dynamique des données, la sortie du système.

4.5.1.2 Sortie de l'outil

La sortie de l'outil correspond au fichier `App.fix.cc` représentant le code source décrivant l'application avec des types virgule fixe. La spécification virgule fixe a été optimisée en fonction des types supportés par l'architecture et de la contrainte de précision. Actuellement, les types en virgule fixe sont spécifiés par le type `ac_fixed` proposé par Mentor Graphic. Une donnée `var` est déclarée de la manière suivante :

$$\texttt{ac\_fixed<}\ w_D, w_{IP}, S, Q, O\ \texttt{>}\ \texttt{var}$$

avec w_D le nombre de bits total, w_{IP} le nombre de bits pour la partie entière, S un booléen indiquant si la donnée est signée, Q le mode de quantification et O le mode de débordement. L'avantage du type `ac_fixed` réside dans la possibilité de spécifier tous les paramètres d'une donnée en virgule fixe, dans le support des

opérations de changement de format (*cast*) et dans l'alignement des données. L'alignement des positions de la virgule avant une opération d'addition est pris en charge par le type et n'est pas à la charge du développeur (ou de l'outil de conversion en virgule fixe). Ainsi, le code généré ne contient pas d'opérations de décalage. L'inconvénient de ce type réside dans le fait que les paramètres virgule fixe d'une donnée sont statiques et ne peuvent pas évoluer au cours du temps. Ceci est un problème par exemple pour la gestion des fonctions. Une même fonction, ayant pour des appels différents, des dynamiques d'entrée différentes ne pourra pas être représentée par le même code source.

Dans un second temps, nous souhaitons pouvoir utiliser les types entiers, disponibles au sein du langage C, afin de cibler des compilateurs pour processeurs et le type entier `ac_int` permettant de modifier la position de la virgule au cours du temps. Dans ce cas, l'outil doit prendre en charge la gestion des opérations de décalage.

4.5.1.3 Fonctionnalités

L'objectif de cet outil est de convertir en virgule fixe une application, décrite à l'aide d'un code C utilisant les types flottants. Ce processus de conversion définit le format virgule fixe optimisé des opérandes de chaque opération. Le nombre de bits pour la partie entière est déterminé afin de garantir l'absence de débordement ou de limiter la probabilité des débordements. Le nombre de bits pour la partie fractionnaire est déterminé afin de minimiser le coût de l'implantation pour une contrainte de précision $P_{b_{max}}$.

4.5.2 Description de l'outil

L'outil de conversion en virgule fixe ID.Fix-Conv est développé au sein de l'infrastructure de compilation GeCoS [8] (Generic Compiler Suite) développée au sein de l'équipe Cairn depuis 2004. La partie frontale de Gecos permet de générer la représentation intermédiaire sur laquelle les différentes transformations sont réalisées. L'outil développé au sein de Gecos est composé de deux branches. La première contient les différentes transformations nécessaires pour la réalisation de la conversion en virgule fixe et la regénération du code source C utilisant les types en virgule fixe. La seconde branche a pour objectif de générer le graphe unique représentant l'application et correspondant à un SFG. Ce graphe est utilisé par les modules d'évaluation de la dynamique et d'évaluation de la précision. Cette seconde branche est présentée dans les parties 3.2.3.1 et 3.2.3.2. La représentation intermédiaire et les différentes transformations, réalisées pour la conversion en virgule fixe, sont détaillées ci-dessous.

4.5.2.1 Représentation intermédiaire

La représentation intermédiaire utilisée pour la conversion en virgule fixe correspond à un Graphe Flot de Données et de Contrôle (CDFG). Ce CDFG est un graphe orienté dont les nœuds représentent des blocs de contrôle. Un type de bloc spécifique à chaque structure de contrôle du langage C est disponible.

L'évaluation de la précision et de la dynamique sont réalisées sur un unique graphe représentant l'ensemble des traitements de l'application. Ce graphe correspond à un Graphe Flot de Signal (SFG). Lors de la création du SFG, obtenu par la mise à plat du CDFG, chaque opération o_i du CDFG conduit à différentes opérations o'_j dans le SFG. Soit $\mathcal{T}_i$, l'ensemble regroupant les indices de toutes les opérations o'_j du SFG correspondant à la duplication de l'opération o_i lors de la mise à plat du CDFG.

4.5.2.2 Transformations

Annotation du CDFG Les opérations o_i et les données d_i du CDFG contribuant au calcul des sorties de l'application sont annotées par un indice i unique. Cet indice est utilisé pour accéder à la spécification virgule fixe associée à cette opération o_i ou cette donnée d_i.

8. `http://gecos.gforge.inria.fr/`

Évaluation de la dynamique Pour le calcul de la dynamique des différentes données présentes au sein de l'application, trois méthodes sont disponibles. Les méthodes basées sur l'arithmétique d'intervalle et la norme L1 permettent d'obtenir le domaine de définition des données considérées et ainsi garantir l'absence de débordement. La norme L1 utilise la notion de fonction de transfert et ainsi est valide uniquement pour les systèmes linéaires et invariants dans le temps (LIT). L'approche basée sur l'arithmétique d'intervalle réalise la propagation des domaines de définition des entrées vers les sorties de l'application. L'approche développée étant basée sur un parcours de graphe, celui-ci ne doit pas contenir de cycle et ainsi, uniquement les systèmes non récursifs sont supportés. L'arithmétique d'intervalle peut traiter les systèmes récursifs en déroulant les récurrences, mais cette fonctionnalité n'est pas implantée actuellement dans notre outil. La méthode KLE présentée dans la partie 4.2 permet de déterminer la dynamique en sortie d'un système LTI pour une probabilité de débordement fixée. Cette probabilité est spécifiée à travers un pragma associé à la donnée considérée. La méthode KLE et celle basée sur la norme L1 nécessitent de déterminer les réponses impulsionnelles des systèmes. Ainsi, la dynamique est évaluée sur le graphe flot de signal de l'application et l'infrastructure logicielle utilisée pour évaluer la précision des calculs est utilisée. L'outil ID.Fix-DynEval permet de déterminer la dynamique de toutes les données d'_j du SFG. La dynamique d'une donnée d_i du CDFG est obtenue à partir de celle des données associées d'_j du SFG à l'aide des expressions suivantes :

$$\overline{d_i} = \max_{j \in \mathcal{T}_i}(\overline{d'_j}) \qquad \text{et} \qquad \underline{d_i} = \min_{j \in \mathcal{T}_i}(\underline{d'_j}). \tag{4.20}$$

Pour améliorer la conversion en virgule fixe, nous pouvons envisager de faire des transformations algorithmiques pour modifier le code source afin de différencier les données d'_j ayant des dynamiques très différentes. Par exemple, nous pouvons dupliquer le code source d'une fonction lorsque les dynamiques des données d'entrée sont différentes entre les appels.

Détermination de la position de la virgule Dans la version actuelle de l'outil, la position de la virgule d'une donnée est directement obtenue à partir de la dynamique. Les opérations de décalage nécessaires pour aligner et recadrer les données ne sont pas insérées dans le code car cette fonctionnalité est supportée par les types `ac_fixed`. Cependant, la présence de ces différentes opérations de décalage au sein de l'architecture peut entrainer une augmentation non négligeable du coût de l'implantation. Ainsi, l'optimisation du placement des opérations de décalage en vue de minimiser leur influence peut permettre d'améliorer le coût de l'implantation. Ce problème a été étudié dans [Ménard 02a] dans le cas d'une implantation logicielle de l'application. La mise en œuvre de cette technique nécessite de pouvoir déplacer les opérations de décalage au sein du CDFG et ainsi complexifie la représentation intermédiaire.

Optimisation de la largeur des données L'optimisation de la largeur des données correspond à un problème de minimisation du coût de l'implantation sous contrainte de précision. Le choix d'algorithmes pour réaliser l'optimisation a été étudié dans la partie 4.4.1.3. Actuellement, les algorithmes implantés dans l'outil correspondent à l'algorithme glouton (*min+1 bit*) et l'algorithme de séparation et évaluation progressive (SEP). Notre objectif est d'implanter les algorithmes de recherche avec tabous et GRASP afin de pouvoir mesurer les temps d'exécution de ce dernier sur un nombre d'applications plus important.

L'optimisation des largeurs des données nécessite d'évaluer le coût de l'implantation et la précision des calculs. Pour évaluer le coût, nous disposons du code C source, généré par le module ID.Fix-accEval, et permettant de calculer la puissance du bruit de quantification en fonction de la largeur des données et du mode de quantification utilisé. Ce code est compilé et appelé par l'algorithme d'optimisation par l'intermédiaire d'une interface JNI [9].

Le coût global de l'implantation est obtenu à partir du coût associé à chaque opération présent dans la base de données représentant l'architecture et du nombre de fois que l'opération est exécutée. Cette information est déterminée par simulation (*profiling*) sur un jeu représentatif de vecteurs d'entrée. Un modèle de coût

9. *Java Native Interface.*

permettant de calculer la consommation d'énergie est disponible pour les architectures à grain fin en termes de largeurs supportées. La consommation d'énergie globale correspond à la somme de la consommation d'énergie de chaque opération. Dans [Ménard 11, Ménard 06], un modèle de coût pour les architectures à grain moyen en termes de largeurs supportées est proposé. La métrique de coût correspond au temps d'exécution du code.

4.5.2.3 Conclusions sur l'outil ID.Fix

Dans cette partie, l'infrastructure logicielle de conversion en virgule fixe ID.Fix a été présentée. L'objectif de cette infrastructure est de déterminer et d'optimiser la spécification virgule fixe d'une application sous contrainte de précision des calculs. L'application est spécifiée à l'aide d'un code source utilisant le langage C, les types flottants et des "pragmas" pour définir certains paramètres nécessaires à la conversion en virgule fixe. L'infrastructure permet de générer en sortie le code C utilisant les types en virgule fixe (`ac_fixed`) et correspondant à la spécification virgule fixe optimisée.

La conversion en virgule fixe se base sur des approches analytiques pour l'évaluation de la dynamique (ID.Fix-DynEval) et de la précision (ID.Fix-AccEval). Ceci permet d'obtenir des temps de conversion raisonnables.

D'un point de vue des structures de contrôle supportées, l'outil permet actuellement de traiter les fonctions, les structures répétitives dont le nombre d'itérations peut être déterminé par interprétation du code et les structures conditionnelles pour lesquelles la condition n'est pas affectée d'un bruit de quantification.

4.6 Conclusions

Les applications intégrant des traitements mathématiques de données sont de plus en plus complexes et l'automatisation du processus de conversion en virgule fixe devient indispensable pour obtenir des temps de conception des applications raisonnables. Dans ce chapitre, nos différentes contributions au niveau de la conversion en virgule fixe ont été présentées.

Pour l'évaluation de la dynamique, la disponibilité de techniques permettant de déterminer la dynamique pour une probabilité de débordement donnée permet de réduire le coût d'implantation par rapport aux techniques existantes. Nous ne sommes qu'à la moitié du chemin permettant de fournir à l'utilisateur un niveau d'optimisation supplémentaire à travers la minimisation de la largeur de la partie entière. La détermination du nombre de bits pour la partie entière peut être vue comme un processus d'optimisation visant à minimiser le coût de l'implantation sous contrainte de maintien des performances de l'application. La mise en œuvre de ce processus nécessite de posséder une approche permettant d'évaluer les effets des débordements sur les performances de l'application.

Nos travaux sur l'optimisation de la largeur des données nous a permis d'étudier différents algorithmes d'optimisation et d'évaluer pour chaque algorithme le temps d'optimisation et la qualité de la solution obtenue. L'objectif est d'intégrer dans notre infrastructure logicielle, de nouveaux algorithmes ayant des caractéristiques intéressantes. Ainsi, l'utilisateur disposera d'une bibliothèque d'algorithmes d'optimisation et pourra choisir en fonction de ses contraintes (granularité de l'architecture, temps d'optimisation ou qualité) l'algorithme le mieux adapté.

Dans nos travaux sur la synthèse d'architecture à largeurs multiples, nous avons montré la nécessité de prendre en compte le partage des ressources pour optimiser la largeur des données. Ainsi les processus d'optimisation des largeurs et de synthèse d'architecture doivent être couplés. L'intégration de ce type de technique au sein d'un outil de conversion en virgule fixe doit bien prendre en compte les temps d'exécution de ce processus afin d'avoir un temps de conversion raisonnable. Ainsi, notre approche de groupement basé sur la synthèse, pragmatique et simple, peut trouver toute sa place en permettant de fournir une solution optimisée en quelques itérations.

Une partie des méthodes proposées pour l'évaluation de la dynamique et l'optimisation des largeurs des données est implantée au sein de l'infrastructure de conversion en virgule fixe ID.Fix. Celle-ci permet dès à présent de traiter un code C, de complexité moyenne, de réaliser la conversion en virgule fixe et de générer en sortie le code C utilisant des types en virgule fixe. L'infrastructure supporte les structures de contrôle telles que les fonctions, les structures conditionnelles et structures répétitives.

L'approche hiérarchique d'optimisation de la largeur des données fournit une nouvelle dimension à nos méthodes de conversion en virgule fixe. Cette approche, combinée à l'approche mixte pour évaluer les performances, peut permettre de traiter des applications réelles sans restriction trop importante. Les premiers résultats sur des applications de taille moyenne permettent de montrer les accélérations significatives obtenues par rapport à une approche optimisant la largeur de toutes les données. Maintenant, il reste à transformer l'essai en proposant un outil permettant d'implanter cette approche hiérarchique. Ceci nécessite au préalable de définir le formalisme utilisé pour représenter au niveau système une application composée d'un ensemble de sous-systèmes. Le module implantant l'approche mixte doit être développé. Il permettra d'évaluer les performances de l'application. L'infrastructure ID.Fix présentée dans la partie précédente sera utilisée pour chaque sous-système afin d'évaluer le coût d'implantation en réalisant une conversion en virgule fixe pour une contrainte de précision donnée.

Chapitre 5

Adéquation application système

Les contraintes des systèmes embarqués nécessitent d'optimiser l'implantation des applications au sein de ces systèmes. En particulier, le dimensionnement des données peut être optimisé afin de minimiser le coût de l'implantation et de fournir une précision des calculs suffisante pour garantir les performances de l'application. Ce dimensionnement est réalisé lors de la phase d'implantation de l'application mais, celui-ci peut évoluer au cours du temps afin de s'adapter aux modifications des contraintes de l'application.

Dans ce chapitre, l'adéquation application système est principalement vue sous l'angle du dimensionnement des données. Dans la première partie de ce chapitre, nous présentons les travaux réalisés sur l'optimisation de l'implantation d'applications. Dans un premier temps, les travaux sur les systèmes de communication numérique sont présentés puis dans un second temps, les générateurs de blocs dédiés optimisés au niveau de la précision des calculs sont décrits. Dans la seconde partie, le concept d'adaptation dynamique de la précision (ADP) est présenté puis l'architecture développée dans le cadre du projet ROMA et supportant l'ADP est détaillée.

5.1 Optimisation de l'implantation d'applications

Encadrement :	Romuald Rocher, doctorat en 2006 [71] Nicolas Hervé, doctorat en 2007 [44] Mahtab Alam, doctorant depuis 2009
Conférences :	Iscas 2006 [Rocher 06a], Eusipco 2007 [Hilaire 07], CACSD 2008 [Hilaire 08], ARCS 2011 [Alam 11b]
Revues :	Journal on Embedded Systems 2006 [Rocher 06b], 2011 [Alam 11a] IEEE Journal on Emerging and Selected Topics in Circuits and Systems 2012 [Alam 12a]

5.1.1 Systèmes de communication numérique

5.1.1.1 Optimisation de l'implantation de systèmes de téléphonie mobile

Au cours de ma thèse [Ménard 02a], j'ai été amené à travailler sur les systèmes de téléphonie mobile de troisième génération (UMTS). Ce système utilise un accès multiplie à répartition par code (WCDMA [1]). Dans ce cadre, nous avons analysé l'adéquation de différentes plateformes pour l'implantation de cette application nécessitant des capacités de calcul importantes. En particulier, je me suis intéressé à l'implantation de cette application sur un DSP VLIW [2] utilisant la technologie SWP pour optimiser le temps d'exécution du

1. *Wideband Code Division Multiple Access.*
2. *Very Long Instruction Word.*

code [Ménard 03c]. A la suite de ces travaux, dans le cadre de projets de fin d'études à l'Enssat et de stages d'ingénieur, nous avons réalisé l'implantation du récepteur sur un FPGA en vue de réaliser un démonstrateur.

Fort de cette expérience sur la technologie WCDMA, j'ai collaboré avec Taofik Saïdi dans le cadre de sa thèse [74] réalisée en cotutelle avec l'Université Laval au Québec. J'ai participé à la mise en œuvre d'un modèle de simulation sous Matlab et aux aspects dimensionnement des données. L'objectif de la thèse était de définir une architecture matérielle pour des systèmes MIMO basés sur la technologie WCDMA. Cette technologie est présentée succinctement dans la partie 5.2.1.2. Dans un premier temps, une architecture matérielle pour un système mono-antenne WCDMA a été proposée et développée. Les propriétés de l'application ont été exploitées afin de minimiser la complexité de l'implantation. Pour le filtre de réception situé en sortie des convertisseurs analogique numérique, la dynamique des coefficients est prise en compte pour réduire la largeur des opérandes et ainsi optimiser la surface. Pour le module de réception en râteau et celui de recherche de trajets, un opérateur spécifique pour la multiplication avec un code a été défini. Pour le module de réception en râteau, un récepteur multifonctions a été proposé. Pour réduire le coût d'implantation, il est proposé de choisir des formats différents pour chaque branche du récepteur en râteau en fonction des caractéristiques du signal d'entée. Les amplitudes des trajets traités étant différentes, des codages en virgule fixe différents peuvent être utilisés pour chaque branche. De plus, dans le cadre de la thèse d'Hai-Nam Nguyen, les modèles nécessaires pour la détermination de la dynamique et l'optimisation de la largeur des données ont été proposés pour un récepteur WCDMA [Nguyen 09a]. Ces travaux ont été réalisés pour valider notre approche d'adaptation dynamique de la précision détaillée dans la partie 5.2.1.2.

Nous poursuivons l'expérimentation des méthodes proposées sur des applications de communication numérique à travers l'optimisation de récepteurs MIMO-OFDM utilisés pour la quatrième génération des systèmes de téléphonie mobile. Dans ce cadre, nous avons particulièrement travaillé sur le décodage sphérique basé sur l'algorithme SSFE présenté dans la partie 3.3.2.3.

5.1.1.2 Optimisation de la consommation d'énergie au sein des réseaux de capteurs.

La recherche sur les réseaux de capteurs (RdC) a connu un engouement fort ces dix dernières années car les applications sont nombreuses et la technologie permet de réaliser des objets communicants (nœuds du RdC) ayant une faible consommation d'énergie. L'objectif de la thèse de Mathab Alam (2009–2012) est de mettre en œuvre des techniques d'adaptation dynamique des traitements et des paramètres du système afin d'optimiser la consommation d'énergie des objets communicants présents dans les RdC. L'optimisation des différents paramètres au sein d'un RdC en vue d'optimiser l'énergie nécessite de posséder un modèle de consommation d'énergie des nœuds fiable.

Dans [Alam 11a], un modèle de consommation d'énergie hybride combinant un modèle analytique et les résultats de mesures est proposé. Ce modèle permet de prendre en compte les collisions lors des transmissions. Ce modèle est défini pour une plateforme matérielle composée d'un micro-contrôleur, d'un circuit dédié implantant la couche physique et d'un amplificateur de puissance dont le gain peut être ajusté. Les éléments calculés à l'aide des modèles analytiques sont les paramètres associés à la couche liaison de données. Ces paramètres correspondent au nombre moyen de retransmissions, de collisions lors de la transmission des paquets de contrôle et de collisions lors de la transmission des données. La modélisation de l'énergie est basée sur des scénarios permettant de distinguer les différentes phases des processus de traitement et de communication. La consommation d'énergie du nœud est mesurée en vue de caractériser la consommation de chaque phase. Le modèle global combine la consommation d'énergie associée à chaque phase et le nombre d'occurrence de ces phases obtenu à partir du modèle analytique. Ce modèle de consommation d'énergie au sein d'un RdC conduit à une erreur relative par rapport à des mesures réelles inférieure à 10%. Les travaux réalisés actuellement visent à optimiser dynamiquement les paramètres de la couche liaison afin d'optimiser la consommation d'énergie.

5.1.2 Génération de blocs dédiés optimisés

Dans les applications embarquées orientées traitement de données, différents noyaux de traitement sont communément employés. Pour les applications de traitement du signal, ces noyaux correspondent aux filtres

numériques, aux transformées (FFT, DCT, ...). Dans certaines applications, nous pouvons aussi retrouver des noyaux d'algèbre linéaire ou pour l'évaluation de fonctions. Pour des domaines d'application plus ciblés certains traitements sont très souvent utilisés, comme le décodage de canal (Viterbi, LDPC, Turbocodes) pour les systèmes de communication numérique.

La connaissance de l'application considérée permet de mettre en œuvre des optimisations supplémentaires par rapport à une spécification de l'application correspondant à un simple code C. En particulier, des optimisations au niveau algorithmique peuvent être réalisées en recherchant la structure des calculs conduisant au meilleur coût pour les contraintes spécifiées par l'utilisateur. Ces applications sont paramétrables au niveau de la taille (p. ex. l'ordre d'un filtre) et de la valeur des coefficients, mais il est possible de définir pour chaque application une spécification générique.

Pour la conception de systèmes intégrés, la réutilisation de blocs dédiés paramétrables permet d'accélérer le développement du système. L'obtention de ces blocs dédiés (propriété intellectuelle : IP) est réalisée à l'aide d'un générateur fournissant le code du bloc adapté en fonction des paramètres fixés par l'utilisateur. Ce code peut, par exemple, correspondre à du VHDL au niveau RTL. Les générateurs d'IP actuels permettent, dans le meilleur des cas, uniquement de paramétrer la largeur des données au sein du bloc. L'objectif est d'élever le niveau d'abstraction par rapport aux générateurs actuels et de proposer un générateur de blocs dédiés permettant de fournir une architecture optimisée au niveau de la précision des calculs. La méthode recherche la structure de calcul, la largeur des coefficients et la largeur des données permettant de minimiser le coût de l'implantation et de respecter les différentes contraintes fournies par l'utilisateur. Notre approche de conversion automatique en virgule fixe présentée dans le chapitre 4, ne répond pas à toutes ces attentes. De plus, cette approche permet de traiter des applications pas encore supportées par notre outil ID.Fix, mais pour laquelle nous avons un modèle analytique pour l'évaluation de la précision, obtenu manuellement.

5.1.2.1 Description de la méthodologie

Le synoptique de la méthode proposée est présenté à la figure 5.1. La première étape consiste à définir la structure utilisée. Pour cette structure et les coefficients associés, la conversion en virgule fixe est réalisée en différentes étapes. L'objectif de la première partie est de déterminer la largeur des coefficients. Le but de la seconde partie est de déterminer la largeur des données de l'application (entrées, sorties, variables intermédiaires).

Optimisation au niveau algorithmique Pour des applications de traitement du signal telles que les filtres numériques, différentes structures peuvent être utilisées. Ces structures sont strictement équivalentes en précision infinie mais elles conduisent à des résultats différents en précision finie. L'exploration des différentes structures permet d'aboutir à une implantation plus optimisée en sélectionnant la structure la mieux adaptée. Le choix de la meilleure structure dépend des paramètres de l'application. Ainsi, les structures sont testées de manière exhaustive et pour chaque structure une conversion en virgule fixe est réalisée avec les contraintes spécifiées par l'utilisateur.

Détermination de la largeur des coefficients La quantification des coefficients d'un système modifie la fonctionnalité réalisée par celui-ci. Pour les filtres numériques, la quantification des coefficients de la fonction de transfert se traduit par des changements sur la réponse dans les domaines temporel et fréquentiel. Pour les filtres récursifs, le placement des pôles et ainsi la stabilité du filtre dépend du nombre de bits alloués aux coefficients. Ainsi, l'objectif est de minimiser la taille des coefficients sous contrainte de dégradation des réponses temporelles et fréquentielles et du placement des pôles. Ces différentes contraintes sont fixées directement par l'utilisateur.

Détermination de la largeur des données Les différents éléments nécessaires à la détermination de la largeur des données correspondent à la détermination de la dynamique des données, à l'évaluation du coût

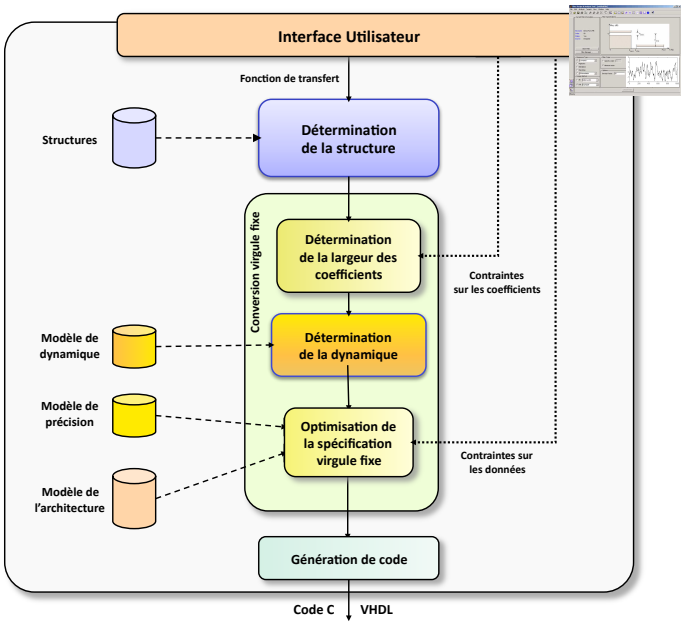


FIGURE 5.1 – Synoptique du générateur de blocs dédiés.

de l'implantation, à l'évaluation de la précision des calculs et à l'optimisation de la largeur des données. Pour pouvoir tester de nombreuses structures différentes, il est nécessaire d'avoir des modèles analytiques permettant de réaliser rapidement les phases d'évaluation. Les modèles analytiques sont développés manuellement avec assez de généricité pour supporter la classe de systèmes visée.

5.1.2.2 Travaux réalisés

Ces travaux sur la génération de blocs dédiés optimisés d'un point de vue de la spécification virgule fixe ont été initiés dans le cadre des thèses de Romuald Rocher et de Nicolas Hervé, travaillant respectivement sur l'évaluation de la précision et l'évaluation du coût d'implantation. Nous avons montré l'intérêt de ce type de générateur sur deux applications correspondant aux filtres à réponse impulsionnelle infinie (RII) et aux filtres adaptatifs basés sur l'algorithme LMS. Les différents modèles et les résultats obtenus sont présentés dans [Rocher 06a, Rocher 06b]. Pour les filtres adaptatifs, l'application étant non LIT, uniquement la phase d'optimisation de la spécification virgule fixe est réalisée. Pour cette application, nous avons utilisé les modèles de bruit obtenus manuellement et proposés dans [Rocher 04, Rocher 10]. A cette date, nous n'avions pas de méthode générale et d'outil permettant de traiter ce type d'application non LTI et récursive. Dans le cadre du filtre IIR, pour les trois types de structure testés, les différents ordres de cellules cascadées et les différentes permutations des cellules conduisent à tester 93 structures différentes. Pour une même contrainte de précision, les écarts de coût d'implantation obtenus entre les différentes structures peuvent être très élevés.

Nous avons poursuivi ces premiers travaux sur les générateurs d'IPs optimisées dans le cadre du post-doc de Thibault Hilaire (2006–2008). Au cours de sa thèse [45], Thibault Hilaire avait proposé un formalisme dénommé *forme implicite spécialisée* (FIS), et basé sur une extension de la représentation d'états pour décrire de manière unifiée les systèmes LIT. De plus, différentes mesures de sensibilité en précision finie avait été proposées afin d'optimiser la largeur des coefficients. Dans le cadre de ce post-doc, nous nous sommes intéressé à l'optimisation de la spécification virgule fixe. Les modèles d'évaluation de la dynamique, du coût d'implantation et de la précision des calculs ont été définis pour le formalisme FIS [Hilaire 07] [Hilaire 08]. Une boîte à outil de Matlab, *FWRTollbox* [46] a été développée par Thibault Hilaire pour implanter l'approche proposée. L'utilisateur définit le système à travers le formalisme FIS, et ensuite, le code C ou VHDL du système en virgule fixe peut être généré. Cette boîte à outil permet de rechercher la structure optimisée

permettant de minimiser le coût de l'implantation sous contraintes.

Dans le cadre du programme PEPS [3] CNRS FiltrOptim nous avons initié une collaboration avec le LIP (ENS Lyon) sur la synthèse de filtres numériques. L'objectif est de définir des techniques permettant d'optimiser la valeur des coefficients en prenant en compte la précision finie. Ces travaux sont complémentaires à ceux présentés auparavant et permettent d'apporter des optimisations supplémentaires pour améliorer la qualité de la solution générée.

5.2 Adaptation dynamique de la précision

Encadrement :	Shafqat Khan, doctorat en 2010 [51]
	Hai-Nam Nguyen, doctorat à soutenir en décembre 2011 [64]
Conférences :	DASIP 2008 [Nguyen 08], ISCAS 2009 [Nguyen 09b], ARC 2009 [Ménard 09c], EUSIPCO 2009 [Nguyen 09a], SETIT 2009 [Khan 09b]
Revues :	Int. Journal on Information Sciences and Computer Engineering 2010 [Khan 10]
Collaboration :	projet ANR ROMA

Dans le cadre de la thèse d'Hai-Nam Nguyen (2007–2011), le concept d'adaptation dynamique de la précision a été défini [Nguyen 08, Nguyen 09b]. L'objectif est d'adapter la spécification virgule fixe (largeur des opérations) au cours du temps afin de réduire la consommation d'énergie. L'adaptation de la spécification virgule fixe est liée à l'évolution au cours du temps de la contrainte de précision. Cette dernière dépend des conditions externes au système. Par exemple, pour un récepteur de communications numériques embarqué, les paramètres, modifiant la contrainte de précision, peuvent être le niveau de bruit en entrée du récepteur ou la qualité de service désirée (taux d'erreurs binaires).

5.2.1 Description du concept d'adaptation dynamique de la précision

Le dimensionnement de la largeur des données d'un système en virgule fixe influence la consommation d'énergie de celui-ci. La puissance consommée au sein d'un circuit VLSI correspond à la somme de la puissance statique et dynamique. La puissance dynamique est liée au taux d'activité du circuit, à la tension d'alimentation, à la fréquence d'horloge et à la capacité équivalente du circuit. Cette dernière dépend de la technologie utilisée. La réduction de la largeur des données permet de réduire la largeur des bus, des unités fonctionnelles et de la mémoire et ainsi, diminuer le taux d'activité du circuit et en conséquence la puissance consommée. La puissance statique est liée aux courants de fuite des transistors et dépend du nombre de transistors présents dans le circuit. La diminution de la largeur des données va permettre de réduire le nombre de transistors présents au sein du circuit et ainsi abaisser la puissance statique.

Cependant, la réduction de la largeur des données se traduit par une diminution des performances de l'application. L'approche traditionnelle de conception d'un système en virgule fixe est basée sur le principe du pire cas. Pour un récepteur de communications numériques, la puissance maximale du bruit de quantification, la dynamique maximale d'entrée et les conditions de canal les plus bruitées sont considérées. Néanmoins, le bruit et le niveau des signaux évoluent au cours du temps. De plus, le débit des données dépend du service (vidéo, image, parole). Les performances requises (taux d'erreurs binaires) sont liées aux services utilisés. Ces différents éléments montrent que la spécification virgule fixe dépend des éléments extérieurs, comme le niveau de bruit, la dynamique du signal d'entrée, la qualité de service souhaitée. Cette spécification peut être adaptée au cours du temps pour réduire la consommation d'énergie moyenne.

Dans [65], différents compromis entre la consommation d'énergie et la précision des calculs sont explorés dans le contexte de la radio logicielle. Les évolutions du standard WLAN [4] (802.11n) permettent d'utiliser

3. Programmes Exploratoires Pluridisciplinaires.
4. *Wireless Local Area Network*

différentes configurations (débit, type de modulation) en fonction de la qualité du canal de transmission. Une spécification virgule fixe propre à chaque configuration est déterminée permettant ainsi d'obtenir des consommations d'énergie différentes en fonction de la configuration choisie. Dans cette approche, l'adaptation de la spécification virgule fixe est uniquement liée à la configuration choisie

Dans [85], une adaptation de la largeur des données a été proposée pour un démodulateur OFDM. La largeur des données est déterminée en temps réel en fonction de l'erreur observée en sortie du système. Des symboles supplémentaires sont insérés dans la trame pour pouvoir mesurer les effets de la quantification des variables. Des gains de 32% et 24% au niveau de la consommation d'énergie sont reportés. Cette approche nécessite d'intégrer une partie matérielle en charge de l'optimisation des largeurs des données. Ainsi, de l'énergie est consommée par ce processus d'optimisation. De plus, cette technique nécessite de modifier la trame des données transmises et limite son application dans le cas de systèmes standardisés.

L'objectif de notre approche d'adaptation dynamique de la précision est d'adapter, au cours du temps, la spécification virgule fixe en fonction des paramètres de l'environnement extérieur, afin de réduire la consommation d'énergie. Lorsque les paramètres de l'environnement extérieur évoluent, le système bascule vers une nouvelle spécification virgule fixe adaptée à la nouvelle valeur de ces paramètres externes. Dans notre approche, différentes spécifications en virgule fixe sont déterminées lors de la phase de conception du système. Ensuite, lors de l'exécution de l'application, une spécification virgule fixe est sélectionnée en fonction des conditions externes.

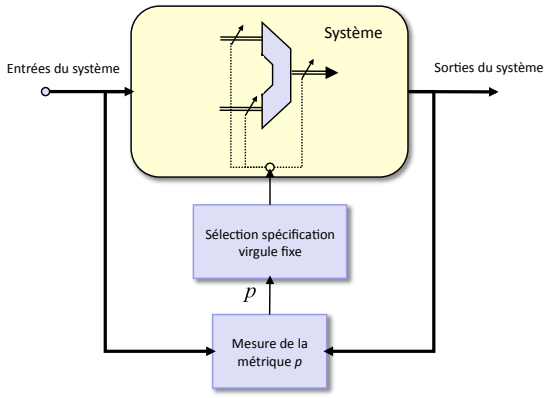


FIGURE 5.2 – Synoptique de l'approche d'adaptation dynamique de la précision.

Le synoptique d'un système exploitant l'adaptation dynamique de la précision est présenté à la figure 5.2. Pour adapter la spécification virgule fixe aux paramètres de l'environnement extérieur, une métrique p permettant de quantifier les conditions externes au système influençant la contrainte de précision des calculs est utilisée. Cette contrainte est fixée de sorte que la dégradation des performances de l'application soit limitée. Le paramètre p est déterminé à l'intérieur du système numérique, par la mesure du signal d'entrée et/ou du signal de sortie. La spécification virgule fixe est sélectionnée en fonction de cette métrique. Les applications retenues pour illustrer notre approche sont issues du domaine des communications numériques, ainsi, le rapport signal sur bruit (RSB) à l'entrée du récepteur est utilisé pour quantifier les conditions externes. Cette valeur est estimée au sein du système numérique.

5.2.1.1 Supports d'exécution possibles pour l'ADP

Caractéristiques nécessaires Le support d'exécution permettant d'implanter l'approche d'adaptation dynamique de la précision (ADP) en vue d'optimiser la consommation d'énergie doit posséder des caractéristiques particulières. L'objectif de notre travail étant d'adapter dynamiquement la précision en vue d'optimiser la consommation d'énergie, il est indispensable que l'architecture considérée soit efficace d'un point de vue énergétique.

L'architecture doit permettre de supporter différentes configurations de spécification virgule fixe. Ces différentes spécifications virgule fixe doivent conduire à une diversité de compromis entre la précision des calculs et la consommation d'énergie. Une configuration particulière est associée à chaque spécification virgule fixe. L'architecture doit permettre de changer de configuration rapidement. Ce changement de configuration ne doit pas entraîner une surconsommation d'énergie annihilant le gain de consommation d'énergie lié au basculement de configuration. Pour les processeurs, chaque configuration peut correspondre à une tâche ou à une fonction. Pour les architectures reconfigurables, chaque spécification virgule fixe correspond à une configuration matérielle donnée. Pour les FPGA, le temps et la consommation d'énergie liés au chargement d'un nouveau *bitstream* ne sont pas négligeables, ainsi le basculement entre les différentes configurations ne doit pas être réalisé trop souvent. Pour les architectures reconfigurables au niveau opérateur, le passage d'une configuration à une autre correspond à la modification des commandes des opérateurs et du réseau d'interconnexion. Ainsi, le temps de changement de configuration et la consommation d'énergie engendrée par celui-ci peuvent être raisonnables.

L'architecture doit fournir un minimum de diversité en termes de types de données supportés afin de pouvoir implanter plusieurs spécifications en virgule fixe conduisant à des précisions de calcul différentes. D'après l'analyse présentée dans la section 4.3.1.1, l'architecture doit posséder une granularité fine ou moyenne en termes de largeurs supportées.

Supports d'exécution utilisés La première solution pour réaliser de l'ADP correspond à un processeur ou une architecture reconfigurable au niveau opérateur. Cette architecture doit posséder des opérateurs fournissant une granularité moyenne en termes de largeurs supportées et doit fournir la possibilité de passer rapidement d'une configuration virgule fixe à une autre en changeant la configuration ou les fonctions utilisées. Pour illustrer ce type d'architecture, nous utilisons par la suite les caractéristiques de l'architecture définie dans le cadre du projet ROMA présenté dans la partie 5.2.2

La seconde solution correspond à un FPGA basse consommation tel que le FPGA Igloo proposé par la société Actel. Cette solution permet d'obtenir une granularité plus fine mais en contrepartie, les changements de configuration ne doivent pas être trop fréquents afin de ne pas être pénalisée par le temps et l'énergie consommée par la reconfiguration.

5.2.1.2 Expérimentations

Ce concept d'adaptation dynamique de la précision a été testé sur un récepteur de communication numérique de troisième génération (UMTS). Ce système utilise un accès multiplie à répartition par code (WCDMA). Les symboles à transmettre sont multipliés par un code, propre à chaque utilisateur. La longueur du code (SF :*Spreading factor*) est égale au rapport entre la fréquence du code et celle des symboles. Afin de pouvoir utiliser différents débits de transmission, la longueur des codes (SF) est variable. Les résultats présentés dans cette partie concernent le module de décodage des symboles présent au sein du récepteur. Ce décodage est réalisé à l'aide d'un récepteur en râteau (*rake receiver*) composé de plusieurs branches. Chaque branche est en charge de traiter un trajet du signal reçu. La sortie du récepteur résulte de la combinaison des sorties de chaque branche. Le traitement réalisé au sein d'une branche est présenté à la figure 5.3.

La conversion en virgule fixe de ce système nécessite de déterminer la dynamique des données, de fixer la contrainte de précision afin que les performances de l'application restent acceptables et optimiser la largeur des données sous contrainte de précision afin de minimiser la consommation d'énergie du système. Les expressions analytiques de la dynamique et de la contrainte de précision sont fournies dans [Nguyen 09a].

Le signal $s(n)$ en entrée du système est normalisé dans l'intervalle $]-1,1[$. Dans un système à étalement de spectre par séquence directe, le rapport signal sur bruit (RSB) au niveau de l'entrée $s(n)$ est particulièrement faible. La dynamique est principalement due au bruit de transmission et aux interférences des autres utilisateurs. Pour retrouver les symboles transmis, le signal d'entrée est corrélé avec le code généré en interne pour amplifier uniquement le signal utile associé aux symboles transmis. En effet, les codes associés aux différents utilisateurs sont orthogonaux entre eux. Ainsi, ce processus amplifie principalement la dynamique

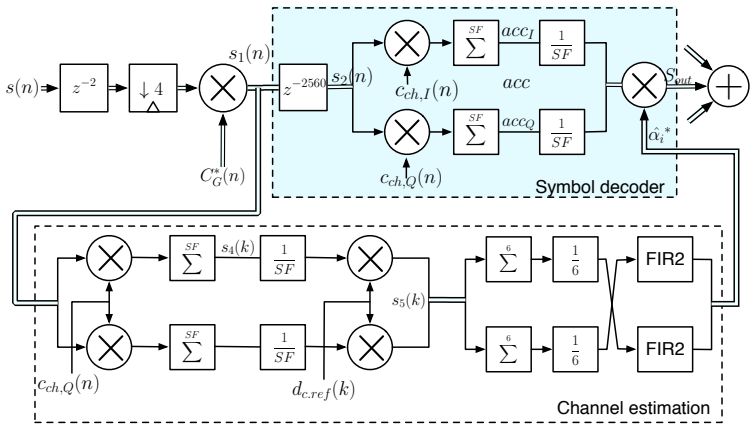
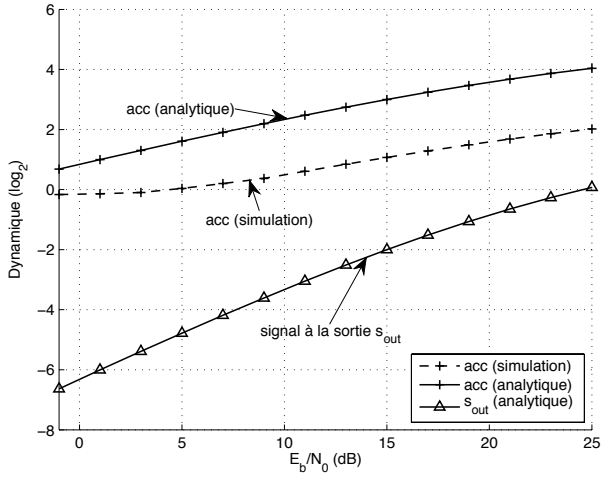


FIGURE 5.3 – Graphe flot de données d'une branche du récepteur en râteau.

a) Évolution de la dynamique



b) Évolution de la puissance

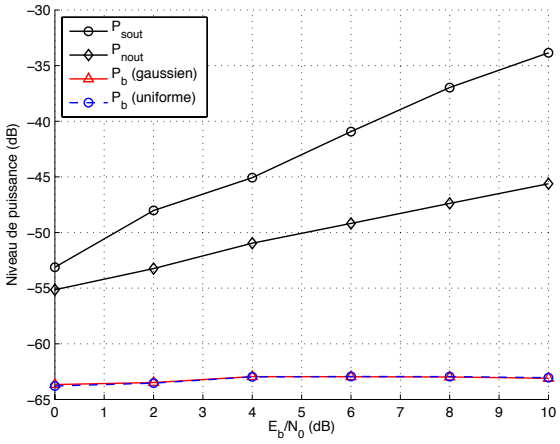
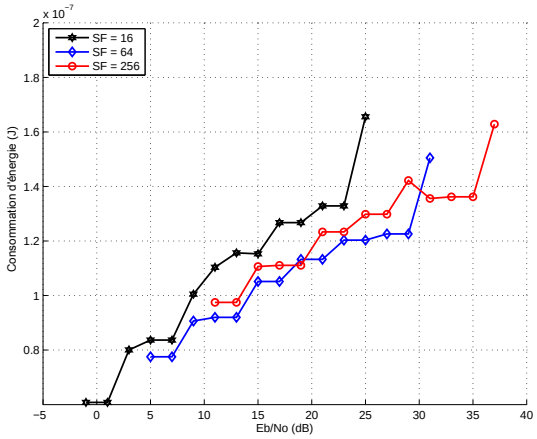


FIGURE 5.4 – a) Évolution de la dynamique des données en fonction du RSB. b) Évolution de la puissance du signal $P_{s_{out}}$, du bruit du récepteur $P_{n_{out}}$ et de la contrainte de précision P_b en fonction du RSB

du signal utile, et non pas celle du bruit et des interférences. A travers cette propriété, une approche est proposée afin de déterminer plus précisément les dynamiques. Avant le processus de corrélation, l'ensemble du signal utile et du bruit est considéré. Après ce processus, seul le signal utile est pris en compte lors du calcul de la dynamique. En conséquence nous obtenons une dynamique des données dépendant du RSB en entrée du récepteur. Les dynamiques en sortie du processus de corrélation (acc) et en sortie d'une branche s_{out} d'un récepteur en râteau sont présentées à la figure 5.4.a en fonction de E_b/N_0. Le terme E_b/N_0 représente le rapport entre l'énergie d'un bit E_b et densité spectrale de bruit N_0 et peut être relié au RSB. Dans le cas d'une transmission à étalement de spectre avec un facteur d'étalement SF, E_b/N_0 est égal à RSB × SF. Les résultats analytiques obtenus à l'aide de l'arithmétique d'intervalle et ceux obtenus à partir d'une approche par simulation sont présentés. Une différence de deux bits est présente entres les résultats obtenus par l'arithmétique d'intervalle et ceux par simulation. Néanmoins, l'évolution de la dynamique en fonction du RSB est identique pour l'estimation analytique et pour le résultat de la simulation. Cela confirme la validité de notre approche d'estimation de la dynamique dans le récepteur WCDMA.

a) Implantation ASIC



b) Implantation opérateur SWP

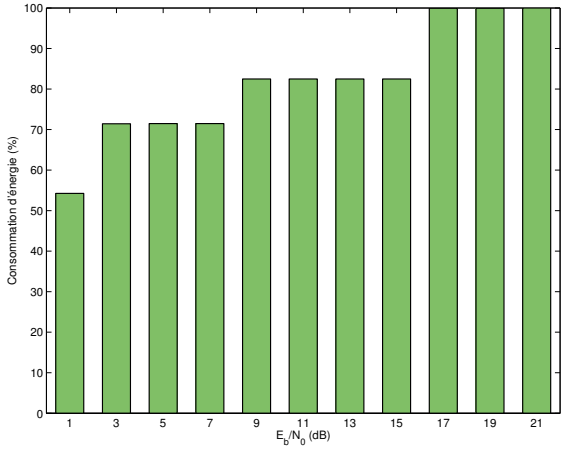


FIGURE 5.5 – a) Énergie consommée par le module de décodage pour différents facteurs d'étalement (SF) dans le cas d'une implantation sur un ASIC. b) Consommation d'énergie normalisée du module de décodage dans le cas d'une implantation avec des opérateurs SWP.

La contrainte de précision est déterminée afin de limiter la dégradation des performances. Les résultats obtenus pour différents RSB sont présentés à la figure 5.4.b. Les courbes $P_{s_{out}}$ et $P_{n_{out}}$ correspondent respectivement à la puissance du signal désiré (symbole) s_{out} et à la puissance du bruit en sortie n_{out}. La différence entre $P_{s_{out}}$ et $P_{n_{out}}$ correspond au RSB à la sortie. La différence entre $P_{s_{out}}$ et P_b correspond au RSB de quantification à la sortie (RSBQ). Les résultats montrent que le RSBQ doit être augmenté afin de maintenir une dégradation du TEB, liée à la précision finie, constante lorsque le RSB croît. Lorsque le RSB est élevé, une plus grande précision est nécessaire pour réduire les erreurs de décision liées à l'arithmétique en virgule fixe.

Pour chaque valeur de RSB considérée, la largeur des données est optimisée. La consommation d'énergie du système est minimisée sous la contrainte de précision P_b présentée à la figure 5.4.b. Dans un premier temps, pour analyser le potentiel de notre approche en terme d'économie d'énergie, une architecture ayant une granularité fine, en termes de largeurs supportées, est considérée. Une bibliothèque d'opérateurs arithmétiques ciblant des ASIC pour une technologie 180 nm est utilisée. L'énergie consommée pour chaque valeur de RSB est présentée à la figure 5.5.a pour des longueurs de code SF de 16, 64 et 256. Les résultats d'optimisation montrent que, pour une longueur de code et un RSB variant entre 0 dB à 20 dB, nous pouvons potentiellement économiser jusqu'à 50% de la consommation d'énergie si la spécification en virgule fixe est adaptée en fonction du RSB. Un même intervalle du RSB a été utilisé. Cependant, le changement de longueur de code entraîne le changement de E_b/N_0. Chaque facteur d'étalement conduisant à des paramètres différents pour le traitement, les configurations de format de données sont différentes pour chaque facteur d'étalement. En particulier, les dynamiques des signaux sont différentes. Les résultats montrent que la consommation d'énergie est différente en fonction du facteur d'étalement. Ainsi, l'approche d'adaptation dynamique de la précision permet de choisir pour chaque valeur de paramètre du système, la configuration virgule fixe adaptée. L'énergie consommée pour chaque valeur de RSB est présentée à la figure 5.5.b pour une longueur de code SF de 16 et le cas d'une architecture à grain moyen en termes de largeurs supportées. Cette architecture est présentée dans la partie suivante. La même tendance que précédemment est observée. Sur le même intervalle de 20 dB, la différence de consommation d'énergie est d'environ 50%.

La même approche a été utilisée pour déterminer la spécification virgule fixe dans le cas d'un autre module du récepteur WCDMA permettant de réaliser la synchronisation de chaque branche du récepteur en

râteau. Sur un intervalle de RSB de 20 dB, l'écart de consommation d'énergie est de 25% dans le cas d'une architecture à grain fin en termes de largeurs supportées et de 36% dans le cas d'une architecture à grain moyen.

Pour montrer l'intérêt de notre approche d'adaptation dynamique de la précision, l'économie obtenue par rapport à une implantation classique (sans ADP) sur un scénario réel a été estimée. Elle est de 36% pour le récepteur en râteau et de 14% pour le module de synchronisation.

5.2.2 Architecture flexible en termes de largeurs supportées

L'adaptation dynamique de la précision se base sur des opérateurs reconfigurables fournissant une flexibilité plus importante en termes de largeurs supportées. La conception de ce type d'opérateurs a été réalisée dans le cadre du projet ANR ROMA et de la thèse de Shafqat Khan [51]. La flexibilité des opérateurs en termes de largeur est obtenue en utilisant le parallélisme au niveau des données (SWP : Sub-Word Parallelism).

Le projet ROMA (Reconfigurable Operators for Multimedia Applications) a été financé dans le cadre du programme ANR (Agence Nationale de la Recherche) Architectures du Futur (ANR-06-ARFU6-004)(2007–2010). Les partenaires du projet ROMA sont : l'IRISA (Rennes et Lannion) le CEA LIST (Saclay), le LIRMM (Montpellier) et Technicolor France (Rennes). Ce projet s'inscrit dans le domaine de l'implantation d'applications de traitement d'images au sein de systèmes embarqués.

Dans le projet ROMA, une architecture reconfigurable au niveau opérateur a été développée. Au contraire des précédentes tentatives de conception de processeurs reconfigurables, ayant conduit à l'utilisation de réseaux d'interconnexions complexes entre opérateurs, le projet ROMA vise à concevoir une architecture pipeline à base d'opérateurs reconfigurables de granularité moyenne. Ainsi, le réseau d'interconnexions entre les opérateurs est simplifié et la complexité est reporté au sein des opérateurs. Deux niveaux de flexibilité sont présents au sein de chaque opérateur complexe à travers le choix de l'opération réalisée et le choix de la largeur des données supportées. La conception de cet opérateur flexible configurable est présentée dans la partie suivante. En parallèle de la définition et de la conception de cette architecture, les outils permettant d'implanter une application au sein de ce processeur reconfigurable ont été développés.

Dans le cadre du projet ROMA, nous avons travaillé, sur l'architecture à travers la conception des opérateurs présentés dans la partie suivante (5.2.2.1) et sur les outils à travers la mise en œuvre d'une approche d'optimisation de la largeur des données pour les architectures ayant une granularité moyenne en termes de largeurs supportées. Cette approche est présentée dans la partie 4.3.1.3 à la page 56.

5.2.2.1 Opérateurs exploitant le parallélisme au niveau données

Comme présenté dans la partie 4.3.1.1 à la page 51, les architectures reconfigurables gros grains peuvent fournir un compromis entre la précision des calculs et le coût d'implantation à travers l'utilisation d'opérateurs arithmétiques ayant une granularité moyenne en termes de largeurs supportées. Cette granularité moyenne peut être obtenue à travers les opérateurs exploitant le parallélisme au niveau données dont le concept est présenté dans la partie 4.3.1.1. Un opérateur SWP de largeur w peut traiter en parallèle, k opérations traitant des sous-mots de largeur w/k. De nombreux opérateurs arithmétiques SWP ont été proposés dans la littérature. Cependant, ces opérateurs opèrent sur des largeurs de sous-mots conventionnelles correspondant à 8, 16 et 32 bits. Les largeurs supportées sont des puissances de deux afin de faciliter le découpage de l'opérateur. De plus, ces largeurs sont multiples de 8 bits et permettent ainsi de faciliter les accès à la mémoire. Pour ces opérateurs SWP conventionnels, la granularité des largeurs supportées est relativement élevée et elle ne permet pas de proposer de nombreux compromis entre le coût et la précision. En traitement du signal et des images, les données en entrée et sortie des applications sont codées classiquement entre 6 et 24 bits. Pour cet intervalle, uniquement deux largeurs différentes sont supportées (8 et 16 bits) par ces opérateurs SWP conventionnels. Ce phénomène aboutit à une sous-utilisation des ressources du processeur lorsque des applications multimédia sont exécutées sur ces opérateurs.

Pour fournir une granularité des largeurs supportées plus fine et pour mieux s'adapter aux largeurs traitées dans les applications multimédia, dans le cadre de la thèse de Shafqat Khan [51], nous avons conçu un opérateur SWP supportant les largeurs de 8, 10, 12 et 16 bits. Pour cet opérateur, dans l'intervalle de 6 à 24 bits, 4 largeurs différentes sont proposées et l'écart entre les largeurs est de 2 et 4 bits. Les largeurs choisies correspondent à celles utilisées classiquement dans les applications multimédia. le plus petit commun multiple des nombres 8, 10, 12 et 16 est égal à 240. Un opérateur de cette taille n'étant pas envisageable, il est nécessaire de trouver une taille d'opérateur raisonnable permettant de minimiser le nombre de bits non utilisés. La taille d'opérateur retenu est de 40 bits. Cette largeur permet de réaliser 5 opérations sur 8 bits, 4 opérations sur 10 bits, 3 opérations sur 12 bits, 2 opérations sur 16 bits ou une opération sur 40 bits. Le placement de ces différents sous-mots au sein de la donnée sur 40 bits est présenté à la figure 5.6. Différents opérateurs SWP permettant d'implanter les opérations de base dans les applications multimédia ont été conçus.

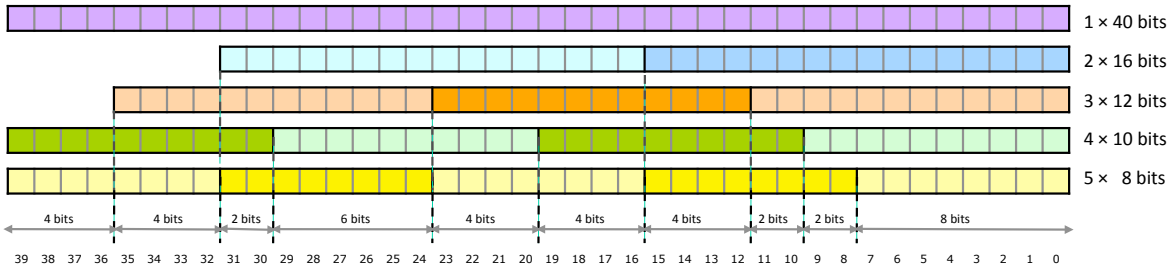


FIGURE 5.6 – Placement des sous-mots au sein d'une donnée sur 40 bits.

Les différents opérateurs ont été synthétisés sur 2 technologies ASIC *standard cell*, 130 nm de ST Microelectronics et 90 nm d'UMC. La synthèse logique pour ASIC a été réalisée à l'aide de l'outil *Design Vision* de Synopsys. Les résultats sont présentés pour la technologie ASIC 90 nm. Le coût de l'implantation est mesuré à travers la latence t_{lat} de l'opérateur et la surface du circuit déterminée à travers le nombre N_{gt} de portes NAND utilisées. Les différentes structures testées vont conduire à un compromis entre ces deux paramètres, ainsi, pour pouvoir comparer plus facilement les structures, la métrique correspondant au produit latence par surface est calculée et cette valeur est normalisée. Chaque opérateur SWP proposé est comparé à un opérateur classique ayant la même taille des opérandes en entrée et en sortie mais n'ayant pas de support SWP. Dans ce cadre, le surcoût lié au support SWP est calculé. L'opérateur SWP d'addition, utilisant une structure à retenue anticipée par bloc, est présenté dans [Khan 09a] et nous présentons ci-dessous uniquement l'opérateur de multiplication.

Opérateur de multiplication SWP La multiplication est une opération essentielle pour les applications de traitement du signal et de l'image. Elle est réalisée en trois étapes. La première correspond à la génération des produits partiels. Cette génération peut être réalisée simplement par un réseau de portes AND. L'utilisation du recodage modifié de Booth (RMB) permet de réduire le nombre de produits partiels. La seconde étape somme les produits partiels à l'aide d'un arbre de réduction. La dernière étape effectue l'assimilation des retenues de la représentation *carry-save* de la sortie de l'arbre de réduction.

Une première approche pour réaliser un multiplieur SWP repose sur l'utilisation des optimisations réalisées par les outils de synthèse logique en vue de minimiser la surface du circuit. Les multiplieurs associés à chaque mode de fonctionnement de l'opérateur SWP sont définis séparément et les sorties sont concaténées afin d'obtenir un mot de 80 bits. Le résultat final du multiplieur SWP est sélectionné en fonction du mode de fonctionnement choisi. Ensuite, la factorisation des termes communs aux différents multiplieurs en vue de réduire la surface du circuit est réalisée par l'outil de synthèse logique. Cette technique a été testée avec et sans RMB.

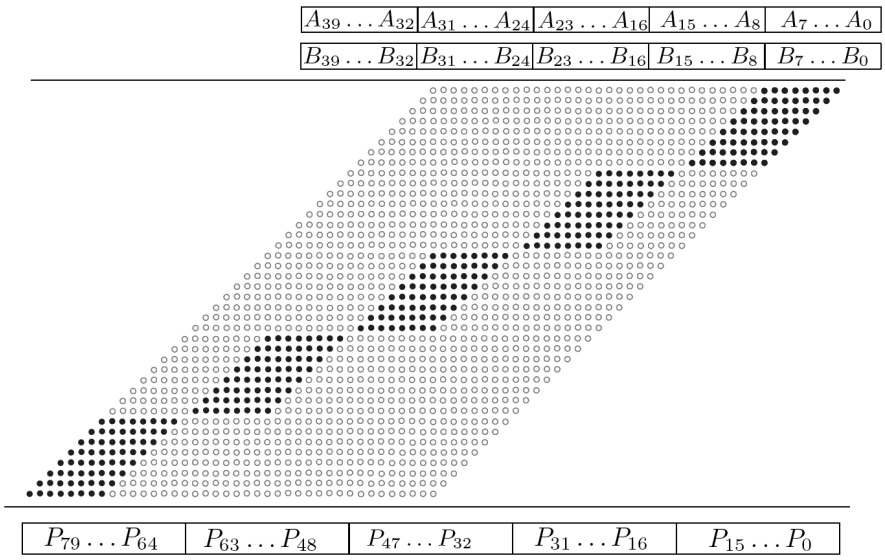


FIGURE 5.7 – Arrangement des produits partiels dans le cas du mode de fonctionnement SWP sur 8 bits.

Dans [54], un opérateur SWP 32 bits supportant les largeurs de 8, 16 et 32 bits est proposé. La génération des produits partiels est basée sur un réseau de portes AND et utilise une structure de Baugh-Wooley pour les nombres signés. En fonction du mode de fonctionnement de l'opérateur SWP, les produits partiels non impliqués pour le résultat final sont forcés à 0. Un exemple est fourni à la figure 5.7 dans le cas de notre opérateur SWP pour le mode 8 bits. Les disques noirs correspondent aux produits partiels utilisés pour le mode de fonctionnement 8 bits et les cercles vides correspondent aux produits partiels forcés à zéro car ils ne doivent pas intervenir pour ce mode. A travers cet exemple, nous pouvons constater que les retenues associées à la multiplication d'un sous-mot ne vont pas perturber la multiplication des autres sous-mots. Ainsi, par nature, il n'est pas nécessaire en fonction du mode de fonctionnement d'inhiber la propagation des retenues au sein de l'arbre de réduction et de l'addition finale. D'autres techniques basées sur le RMB ont été proposées afin de limiter le nombre de produits partiels. Cependant, ce type de recodage entraîne un recouvrement entre les sous-mots et ainsi il est nécessaire en fonction du mode de fonctionnement de détecter et d'inhiber la propagation de la retenue. Le circuit permettant de réaliser cette fonctionnalité devient complexe lorsque le nombre de modes de fonctionnement est plus important comme dans notre cas. Ainsi, la technique proposée dans [54] est la plus efficace et a été étendue pour notre opérateur SWP.

Opérateurs	Mult. 16 bits O_1			Mult. 16 bits + Booth O_2			Mult. [54] O_3		
	N_{gt}	t_{lat} (ns)	$N_{gt} \times t_{lat}$	N_{gt}	t_{lat}	$N_{gt} \times t_{lat}$	N_{gt}	t_{lat}	$N_{gt} \times t_{lat}$
Classique	2500	2,46	1	1748	3,81	1.08			
SWP	4400	2,63	1.88	2950	4,03	1.93	2485	2.83	1.14
Surcoût									
SC_{11} et SC_{22}	76%	7%	88%	69%	6%	79%			
SC_{31}							−1%	15%	14%
SC_{32}							42%	−25%	5%

TABLE 5.1 – Ressources utilisées N_{gt} et latence t_{lat} obtenues pour des multiplieurs 16 bits, classiques et SWP, pour trois approches différentes.

Les résultats obtenus dans le cas d'un opérateur 16 bits permettant de gérer les configurations 4, 8 et 16

bits sont présentés dans le tableau 5.1. Trois techniques ont été testées. Les deux premières utilisent l'outil de synthèse pour réaliser l'optimisation et les tests ont été réalisés avec (O_2, Mult. 16 bits + Booth) et sans (O_1, Mult. 16 bits) RMB. Pour cette technique les surcoûts SC_{11} et SC_{22} sont calculés par rapport à la version classique. La troisième technique correspond à celle proposée dans [54]. Pour cette technique le surcoût est calculé par rapport à l'opérateur classique sans RMB (SC_{31}) et avec RMB (SC_{32}).

Les résultats sur les opérateurs non SWP montrent l'intérêt du RMB pour améliorer la surface de l'opérateur mais au détriment de la latence. La technique basée sur l'optimisation par l'outil de synthèse logique fournit des résultats médiocres en termes de surface et corrects en termes de latence. L'outil de synthèse n'est pas capable de réaliser toutes les factorisations possibles. Pour l'opérateur SWP basé sur la technique issue de [54], le surcoût SC_{31} par rapport à l'opérateur classique sans RMB est très faible aussi bien pour la latence que pour la surface. Par rapport à l'opérateur classique avec RMB le surcoût SC_{32} en termes de surface est important car le nombre de produits partiels est deux fois plus important qu'avec le RMB. Mais l'avantage en termes de latence est préservé et se traduit par un gain de 25%. Ainsi, l'opérateur SWP utilisant la technique proposée dans [54] permet d'obtenir de bonnes performances par rapport à la version classique et ainsi, cette structure a été retenue pour réaliser notre opérateur SWP sur 40 bits. Les coûts d'implantation obtenus pour celui-ci sont présentés dans le tableau 5.2 pour les deux technologies ASIC 90 nm et 130 nm. Par rapport à la version classique le surcoût en surface est faible pour les deux technologies testées. Pour la latence, le surcoût en latence est un peu plus élevé pour la technologie 90 nm.

Opérateurs	Techno 90 nm			Techno 130 nm		
	N_{gt}	t_{lat} (ns)	$N_{gt} \times t_{lat}$	N_{gt}	t_{lat} (ns)	$N_{gt} \times t_{lat}$
Classique	14518	6,07	1	10532	14	1
SWP	15099	7,38	1,26	11081	15	1,13
Surcoût	4%	22%	26%	5%	7%	13%

TABLE 5.2 – Ressources utilisées N_{gt} et latence t_{lat} obtenues pour des multiplieurs 40 bits, classiques et SWP, pour les technologies 90 nm et 130 nm.

Pour améliorer les résultats en termes de latence de l'opérateur, l'arithmétique redondante utilisant la représentation *borrow-save*, utilisant le répertoire de chiffres $\{-1; 0; 1\}$, a été utilisée. L'utilisation de cette arithmétique pour des opérateurs SWP semble naturelle car elle permet de ne pas propager de retenue. Ce type d'arithmétique nécessite d'utiliser des convertisseurs complément à deux vers redondant sur les entrées et des convertisseurs redondant vers complément à deux pour générer la sortie. L'opération arithmétique réalisée en arithmétique redondante permet ainsi d'avoir des latences indépendantes de la largeur des données car la propagation des retenues n'excède pas une position. Pour cet opérateur SWP utilisant l'arithmétique redondante, la même structure que celle utilisée avec l'arithmétique standard est employée. La comparaison entre l'arithmétique standard et redondante et entre les versions SWP et non-SWP est présentée dans le tableau 5.3 pour une technologie de 90 nm. Pour les opérateurs classiques et SWP, l'arithmétique redondante permet de diviser la latence par un facteur proche de 2,4 mais au détriment d'une augmentation de la surface d'un facteur deux. Pour les deux arithmétiques, les surcoûts liés au SWP sont du même ordre de grandeur.

Les opérateurs d'addition et de multiplication ont été utilisés pour concevoir l'opérateur reconfigurable flexible présenté dans le reste de cette partie.

Opérateur reconfigurable flexible De nombreuses applications intègrent des traitements réalisant l'accumulation de N éléments z_i issus d'opérations arithmétiques sur des éléments d'un vecteur. L'objectif est d'obtenir en sortie une seule valeur s à partir de N éléments d'un vecteur traités en parallèle à l'aide des opérateurs SWP. Au final, il est nécessaire de réaliser l'addition des sous-mots entre eux afin d'obtenir le

Opérateurs	Arith. standard			Arith. redondante		
	N_{gt}	t_{lat} (ns)	$N_{gt} \times t_{lat}$	N_{gt}	t_{lat} (ns)	$N_{gt} \times t_{lat}$
Classique	14518	6,07	1	31268	2.5	0,88
SWP	15099	7,38	1,26	31517	3.2	1,14
Surcoût	4%	22%	26%	1%	28%	29, 5%

TABLE 5.3 – Ressources utilisées N_{gt} et latence t_{lat} obtenues pour des multiplieurs 40 bits utilisant l'arithmétique standard et redondante pour une technologie de 130 nm.

résultat final. Pour simplifier l'opération d'accumulation, cette addition des sous-mots est réalisée avant l'accumulation afin d'avoir un opérateur d'accumulation ne nécessitant pas le support SWP. En conséquence, l'accumulation peut être reformulée de la manière suivante :

$$s = \sum_{i=0}^{\lceil \frac{N-1}{k} \rceil} \left(\sum_{j=0}^{k-1} z_{i \cdot k+j} \right) = \sum_{i=0}^{\lceil \frac{N-1}{k} \rceil} y_i. \tag{5.1}$$

avec k le nombre d'éléments traités en parallèle par l'opérateur SWP calculant les z_i.

La structure utilisée correspond à celle présentée à la figure 5.8. Le premier opérateur SWP permet de calculer les k éléments $z_{i \cdot k}$ à $z_{(i+1) \cdot k-1}$, ensuite ces éléments sont additionnés entre eux dans l'opérateur SMA. Finalement, les éléments y_i sont accumulés à l'aide de l'additionneur ACC afin d'obtenir le résultat final sur 40 bits. Cette largeur de 40 bits permet d'avoir un nombre de bits de garde important pour les différents modes SWP. Les bits de garde sont présents au niveau des MSB et permettent d'éviter la présence de débordements liés à l'accroissement de dynamique au cours des accumulations.

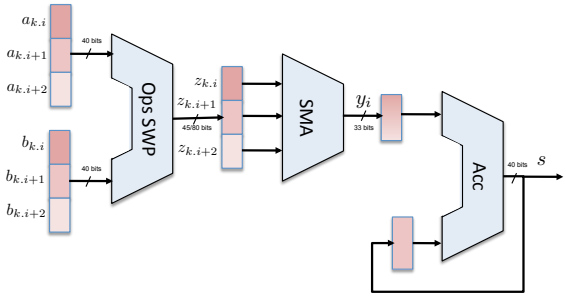


FIGURE 5.8 – Structure pour le calcul de sommes d'opérations SWP.

L'opérateur reconfigurable flexible est présenté à la figure 5.9. Il possède deux entrées a et b et une sortie s sur 40 bits. Cet opérateur est composé de trois tranches de pipeline. La première tranche réalise les opérations SWP d'addition, de soustraction, de multiplication et de valeur absolue de différence. Ces opérateurs ont été présentés dans les parties précédentes. La seconde tranche de pipeline réalise l'addition des sous-mots. La troisième tranche réalise l'accumulation des résultats issus de la seconde tranche. Les différentes opérations pouvant être réalisées par celui-ci sont les suivantes :

- ADD (s/u) : $s_j = a_j + b_j \quad \forall j \in [0; k-1]$ s : *signed*, u : *unsigned*
- SUB (s) : $s_j = a_j - b_j \quad \forall j \in [0; k-1]$
- ADD-ABS (s) : $s_j = |a_j + b_j| \quad \forall j \in [0; k-1]$
- SUB-ABS (s/u) : $s_j = |a_j - b_j| \quad \forall j \in [0; k-1]$

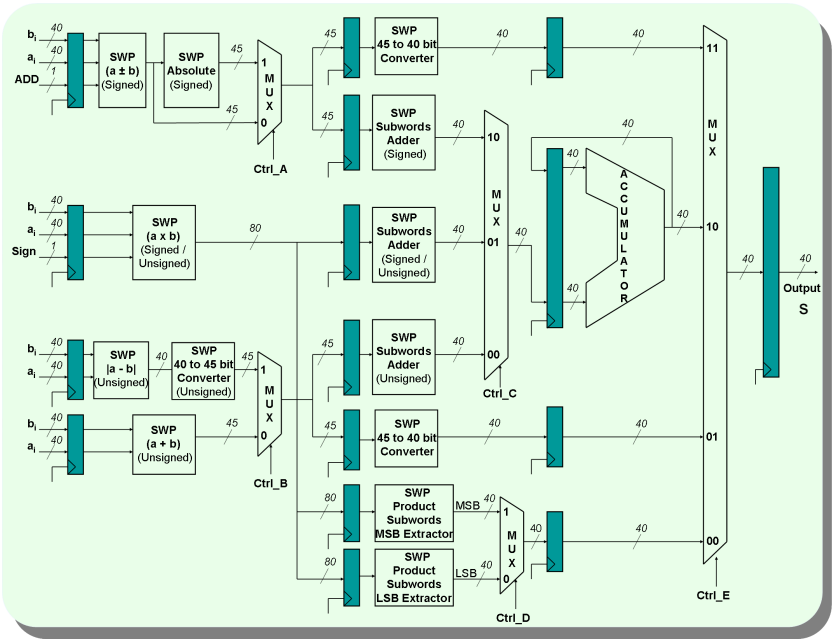


FIGURE 5.9 – Architecture de l'opérateur flexible reconfigurable.

- MULT (s/u) : $s_j = a_j \times b_j \qquad \forall j \in [0; k-1]$,
- SAD (s/u) : $s = \sum_i |a_i - b_i|$
- SAA (s/u) : $s = \sum_i |a_i + b_i|$
- DOTP (s/u) : $s = \sum_i |a_i \times b_i|$
- ADD (s/u) : $s = a + b$

Les cinq premières opérations sont effectuées sur un seul mot de 40 bits et un résultat peut être obtenu à chaque cycle. Pour l'opération de multiplication MULT, la sortie étant sur 80 bits, elle est obtenue en 2 cycles. Les quatre dernières opérations, sont effectuées sur N' mots de 40 bits et un résultat est obtenu tous les N' cycles avec $N' = \lceil \frac{N-1}{k} \rceil$.

Différentes contraintes de temps ont été fournies à l'outil de synthèse logique afin d'obtenir différents compromis entre la surface et la fréquence de fonctionnement du circuit. Pour la technologie ASIC 90 nm, un bon compromis est obtenu pour une période d'horloge de 6 ns conduisant à une surface de l'opérateur de 29980 portes NAND. Pour la technologie ASIC 130 nm, la période d'horloge retenue est égale à 10 ns et la surface obtenue est de 31126 portes NAND.

5.2.3 Conclusions

Dans cette partie, nos travaux sur l'adaptation dynamique de la précision (ADP) et sur la conception d'une architecture flexible en termes de largeurs supportées et permettant d'implanter l'ADP ont été présentés. Le concept d'ADP repose sur l'idée que la contrainte de précision associée au système peut varier au cours du temps. L'ADP cherche à sélectionner la spécification virgule fixe la plus adaptée en fonction des conditions externes au système et des paramètres de celui-ci. Ce concept d'ADP a été testé sur un récepteur WCDMA et les gains potentiels en termes de consommation d'énergie ont été fournis. Les premiers résultats présentés sont encourageants. Cependant, pour valider cette approche, il est nécessaire de faire une implantation complète sur une architecture flexible en termes de largeurs supportées et d'intégrer aussi le coût énergétique du module en charge de la mesure des conditions externes.

Dans le cadre du projet ROMA et de la thèse de Shafqat Khan, un opérateur reconfigurable a été proposé. Celui-ci est flexible en termes de largeurs supportées et d'opérations réalisées. La flexibilité en termes de largeur est fournie à travers la technique SWP. Cet opérateur complexe est réalisé à partir d'opérateurs arithmétiques de base dont la conception a été détaillée pour l'opération de multiplication. L'utilisation de l'arithmétique standard permet d'obtenir un bon compromis entre la surface et la latence. L'arithmétique redondante permet d'améliorer significativement la latence, si une augmentation non négligeable de la surface est tolérée. Les surcoûts de ces opérateurs SWP, par rapport à des opérateurs classiques sont raisonnables aussi bien en termes de surface que de latence.

L'intégration de cet opérateur au sein d'un processeur reconfigurable nécessite de mettre en place un module d'interface au niveau de la mémoire afin de pouvoir accéder à chaque donnée car les largeurs supportées ne sont pas multiples d'une même largeur (8 bits pour les opérateurs SWP conventionnels).

Pour proposer des opérateurs plus flexibles en termes de largeurs supportées, nous avons suivi la voie utilisant la technique SWP mais d'autres pistes peuvent être explorées. Par exemple, la réalisation d'un opérateur utilisant la multi-précision et combinant les implantations spatiales et temporelles en fonction des largeurs supportées peut être envisagée. De même, les techniques SWP et multi-précision peuvent être couplées comme dans les FPGA Altera .

Chapitre 6

Bilan et perspectives de recherche

6.1 Bilan scientifique

Les applications intégrant des traitements mathématiques de données sont de plus en plus complexes et les différentes phases du processus d'implantation doivent être automatisées afin de réduire les temps de développement et d'explorer l'espace de conception en vue d'aboutir à une solution optimisée en termes de coût d'implantation. Le processus de conversion en virgule fixe, situé entre la phase de conception des applications et la phase d'implantation de ces applications, reste un frein à la diminution des temps de développement. La majeure partie de mes travaux de recherche se sont concentrés sur la définition de méthodes pour la conception de systèmes en virgule fixe.

Pour résumer ces travaux de recherche, nous reprenons le synoptique présenté à la figure 6.1 et décrivant une partie du flot de conception et de développement des applications embarquées. Ce flot intègre les phases de conception de l'algorithme, de passage en virgule fixe, et d'implantation au sein de l'architecture ciblée.

Différents travaux ont été réalisés sur l'optimisation de la consommation d'énergie. Plus particulièrement, le concept d'adaptation dynamique de la précision a été proposé. Cette technique permet de réduire la consommation d'énergie en adaptant la spécification virgule fixe en fonction des conditions externes du système. Dans la continuité de ces travaux, des techniques permettant d'adapter les algorithmes utilisés au sein du système ou les paramètres du système sont analysées dans le cadre des réseaux de capteurs.

L'approche hiérarchique proposée pour l'optimisation de la largeur des données fournit une nouvelle dimension à nos méthodes de conversion en virgule fixe. Cette approche, combinée à l'approche mixte pour évaluer les performances de l'application, peut permettre de traiter des applications réelles sans restriction trop importante.

Concernant l'optimisation de la spécification virgule fixe au niveau algorithmique, nous avons travaillé uniquement sur le choix de la structure de l'algorithme dans le cas des systèmes LTI. Ces travaux ont permis de montrer le potentiel de ce type de transformations.

Nous avons débuté assez récemment des travaux sur la première étape de la conversion en virgule fixe correspondant à l'évaluation de la dynamique. Par rapport aux approches existantes, nous abordons ce problème avec l'objectif d'optimiser le nombre de bits pour la partie entière en autorisant des débordements si ceux-ci ne dégradent pas trop les performances de l'application.

Nos contributions sur la conversion en virgule fixe ont surtout porté sur l'optimisation de la largeur des données à travers la proposition d'algorithmes d'optimisation et la mise en œuvre de techniques pour le couplage avec la synthèse d'architectures. Nos travaux sur les algorithmes d'optimisation des largeurs des données ont permis de proposer des algorithmes pour les architectures à grain fin et moyen en termes de largeurs supportées.

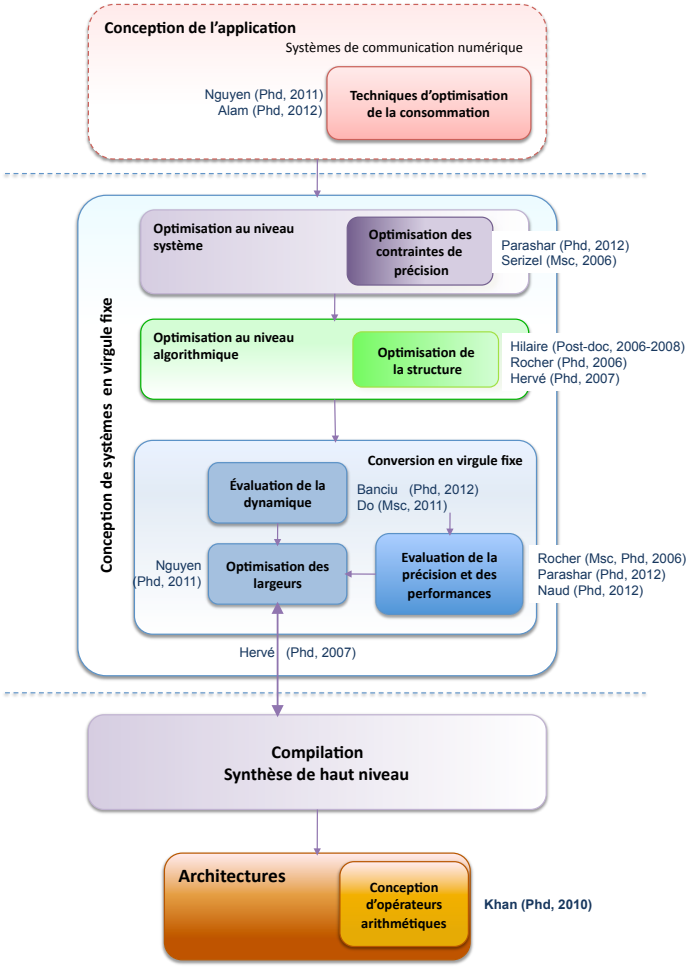


FIGURE 6.1 – Synoptique des travaux de recherche réalisés. Les étudiants encadrés durant leur doctorat (Phd) ou leur Master (Msc) sont reportés.

L'évaluation efficace des effets de la précision finie sur les performances d'une application est probablement le problème le plus difficile de la conversion en virgule fixe. Nous avons consacré une part importante de nos travaux de recherche à cette thématique. Une approche analytique d'évaluation de la précision basée sur la théorie de la perturbation a été proposée. Elle permet de déterminer l'expression de la puissance du bruit de quantification pour les systèmes composés d'opérations dont le modèle de bruit est linéaire. Cette approche prend en compte les structures conditionnelles et la corrélation entre les sources de bruit issues de la quantification d'un même signal. Pour traiter les systèmes intégrant des opérations dont le modèle de bruit n'est pas linéaire, une approche mixte combinant simulation et méthodes analytiques a été proposée. Cette approche permet de traiter les systèmes non supportés par les techniques basées sur la théorie de la perturbation et de réduire les temps d'évaluation des performances de plusieurs ordres de grandeur par rapport aux méthodes basées sur la simulation en virgule fixe. Les différentes méthodes proposées et l'outil développé permettent de bien nous positionner par rapport à l'état de l'art dans le domaine de l'évaluation de la précision et des performances.

La définition de ces méthodes s'est accompagnée du développement d'une infrastructure logicielle. L'infrastructure de conversion en virgule fixe ID.Fix intègre d'ores et déjà certaines méthodes proposées pour

l'évaluation de la dynamique et l'optimisation des largeurs des données. L'infrastructure permet de traiter un code C, de complexité moyenne, et d'obtenir en un temps raisonnable une spécification virgule fixe optimisée. L'outil ID.Fix-AccEval permet de déterminer l'expression de la puissance du bruit de quantification pour les systèmes composés d'opérations dont le modèle de bruit est linéaire. Cet outil génère le code source de cette expression en un temps raisonnable.

Au niveau des architectures, un opérateur reconfigurable de granularité moyenne a été conçu. Cet opérateur complexe fournit deux niveaux de flexibilité à travers le choix de l'opération réalisée et le choix de la largeur des données supportées. Cet opérateur permet de fournir une granularité plus fine en termes de largeurs supportées par rapport aux opérateurs SWP existants.

Les travaux réalisés dans le cadre de ma thèse puis au cours de ces neuf dernières années se sont surtout concentrés sur un domaine de recherche assez ciblé correspondant à la conception de systèmes en virgule fixe. Cependant, cette thématique de recherche nécessite des compétences variées dans les domaines de l'arithmétique, du traitement statistique du signal, de l'architecture et des systèmes embarqués, de l'optimisation, des communications numériques, de la compilation et du génie logiciel. Cette thématique a bénéficié des compétences variées de l'équipe Cairn dans les disciplines de l'électronique, du traitement du signal et de l'informatique.

6.2 Perspectives

Différentes perspectives à ces travaux de recherche sont présentées ci-après. Elles s'inscrivent en partie dans la continuité des travaux sur la conception de systèmes en virgule fixe mais l'objectif est de s'ouvrir à d'autres problématiques de la conception de systèmes embarqués et du domaine des communications numériques.

Les perspectives à très court terme s'inscrivent dans le cadre du projet ANR DEFIS débutant en novembre 2011 et dont le *leadership* est assuré par notre équipe de recherche. L'objectif est de combiner les méthodes d'évaluation de la dynamique et de la précision des différents partenaires afin de proposer une infrastructure de conversion en virgule fixe permettant de traiter une application complète. Cette infrastructure permettra d'optimiser la spécification virgule fixe au niveau système et au niveau algorithmique. Dans le cadre de se projet, nous implanterons les méthodes proposées pour l'optimisation au niveau système et pour l'évaluation de performance basée sur une approche mixte.

6.2.1 Évaluation des performances d'une application

L'évaluation efficace des performances (qualité) d'une application en prenant en compte les imperfections de la plateforme ciblée reste un problème difficile. Ces imperfections peuvent être liées à différents aspects comme la précision finie des calculs, la variabilité des processus de fabrication, la technologie utilisée ou l'approximation des opérations arithmétiques.

Performances en présence de débordements et d'erreurs de décision

Dans le cadre de l'évaluation de la dynamique, l'objectif est de poursuivre les travaux de recherche sur la prise en compte des débordements. Le but est de minimiser le nombre de bits pour la partie entière tant que les performances de l'application sont satisfaites. Dans ce processus d'optimisation, une approche efficace permettant de déterminer les effets des débordements sur les performances de l'application doit être proposée. Les effets fortement non-linéaires des débordements sont difficilement modélisables par les techniques analytiques. Ainsi, une approche par simulation semble plus réaliste. Le challenge est de pouvoir simuler l'application, uniquement lorsque des débordements surviennent. Ces évènements devant être rares afin de maintenir la fonctionnalité des systèmes, le temps nécessaire à la simulation peut être très faible.

La présence d'opérateurs de décision au sein d'un système en virgule fixe conduit à des erreurs de décision lorsque le bruit de quantification en entrée de l'opérateur entraîne une prise de décision différente de celle

obtenue en précision infinie. L'objectif est d'automatiser, dans le cadre du projet ANR DEFIS, l'approche mixte définie dans la thèse de Kathick Parashar. Comme pour les débordements, ces erreurs se caractérisent par une amplitude élevée mais une probabilité d'occurrence faible afin que le système reste fonctionnel. Ainsi, pour ces deux aspects de l'évaluation des performances, nous souhaitons utiliser les techniques de simulation d'évènements rares (SER) [73] pour lesquels de nombreux travaux de recherche, et en particulier à l'INRIA, ont permis de fournir des méthodes efficaces. Dans ce contexte, les simulations sont longues afin de pouvoir observer assez d'évènements rares pour obtenir des statistiques fiables. Ainsi, l'objectif des techniques SER est de réduire les temps de simulation en intégrant des modèles probabilistes.

Performances avec l'arithmétique imprécise ou par estimation

Les opérateurs arithmétiques traitant des entiers sont classiquement conçus pour calculer les bits de la sortie sans introduire d'erreur. Ainsi, la précision du résultat en sortie dépend de la précision des entrées et est de l'ordre de grandeur du poids du bit le moins significatif (LSB). Différentes techniques permettent d'améliorer significativement le coût d'implantation en utilisant des opérateurs arithmétiques dont la précision du résultat est inférieure au poids du LSB, introduisant, en contrepartie, une erreur e_{ops}. Notre objectif est d'étendre nos approches d'évaluation de la précision et des performances à d'autres types d'erreur que le bruit de quantification afin de prendre en compte cette erreur e_{ops}.

Dans le cas de l'arithmétique à estimation [79], cette erreur e_{ops} peut être liée à une simplification de l'opérateur afin de diminuer sa surface, sa latence et sa consommation d'énergie. Pour cela, certains signaux internes à l'opérateur tels que des retenues ne sont pas calculés [79] ou certaines parties de l'opérateur ayant une influence faible sur le résultat sont supprimées [70, 59].

Nous regroupons sous le terme arithmétique imprécise l'ensemble des techniques pour lesquelles la sortie d'un opérateur arithmétique n'est pas toujours identique à celle obtenue avec un opérateur classique. Ces erreurs e_{ops} se caractérisent par une amplitude élevée mais une probabilité d'occurrence faible afin que le système reste fonctionnel. La conception d'un système numérique en prenant en compte pour les différents aspects le pire cas peut conduire à un surcoût d'implantation important. La latence de l'opérateur t_{lat} dépend des données en entrée de celui-ci. Afin de diminuer le coût d'implantation, il est possible de fixer la période d'horloge (prise en compte du résultat en sortie de l'opérateur) à une valeur plus faible que t_{lat} [56]. En conséquence, certaines valeurs calculées seront erronées mais si leur occurrence est assez faible, leur impact sur les performances de l'application peut être tolérable. Ce phénomène est accentué avec l'utilisation du DVS [1] qui permet de diminuer la consommation d'énergie en réduisant la tension d'alimentation mais ceci au détriment des latences des opérateurs.

L'évolution technologique (technologies submicroniques) rendent les systèmes plus vulnérables aux fautes et la tolérance aux fautes est devenu un enjeu majeur des systèmes sur puce. La réduction de la tension d'alimentation des circuits permet de réduire la consommation d'énergie mais entraîne une augmentation de la probabilité de présence d'une faute. Ainsi, un compromis entre la consommation d'énergie et la présence de fautes doit être réalisé [61]. La présence de fautes au sein du système numérique se traduit par une valeur erronée en sortie de l'opérateur.

Un premier objectif est de prendre en compte ces erreurs dans nos modèles de bruit afin d'intégrer celles-ci pour l'évaluation de la précision et des performances. Ceci nécessite une modélisation fine des différents types d'erreur pour analyser leur distribution afin de voir si ces erreurs sont souvent présentes mais avec une amplitude faible ou si elles sont rares mais avec une amplitude élevée. La disponibilité d'une technique basée sur la simulation d'évènements rares permettra de prendre en compte ce second cas de figure. Un second objectif est de concevoir des algorithmes de TDSI plus robustes à ces erreurs liées à l'arithmétique imprécise.

1. Dynamic Voltage Scaling.

6.2.2 Défis des nouvelles architectures

L'évolution actuelle et future des architectures destinées aux systèmes embarqués nécessite d'étendre nos méthodes et outils.

Architectures MPSoC

Pour satisfaire les contraintes de puissance de calcul, liées à l'évolution de la complexité des applications, les architectures intègrent le plus souvent plusieurs cœurs au sein d'une même puce (MPSoC [2]). L'homogénéité des cœurs permet de simplifier l'implantation des applications mais pour satisfaire les contraintes au niveau efficacité énergétique l'hétérogénéité des éléments du MPSoC est souvent nécessaire. Ainsi, des processeurs généralistes, des processeurs spécialisés (DSP, ASIP), des accélérateurs matériels ou des zones reconfigurables sont combinés. Ceci se traduit par la disponibilité, pour implanter l'application, d'architectures ayant des granularités en termes de largeurs supportées différentes. De plus, la largeur des données circulant sur le réseau d'interconnexion (NoC [3]) doit être optimisée afin de minimiser la consommation d'énergie liée à ce transport. Ces différents éléments doivent être intégrés dans une approche d'optimisation de la spécification virgule fixe au niveau système.

Différents niveaux de parallélisme sont présents au sein des architectures multi-cœurs. En particulier, ces architectures offrent de plus en plus de parallélisme au niveau données à travers des capacités SWP. Dans le cadre du projet européen FP7 ALMA [4], ayant débuté en septembre 2011, notre objectif est de combiner au sein de l'outil Gecos l'optimisation de la largeur des données et la parallélisation des données dans le cas d'architectures ayant des capacités SWP. Ceci permettra d'obtenir un flot complet et automatique permettant d'exploiter les opportunités offertes par les opérateurs SWP (p.ex. ARM Neon). Dans ce cadre, nous allons bénéficier de l'expertise de la partie Rennaise de l'équipe CAIRN spécialisée dans la parallélisation de code. De plus, nous souhaitons investiguer l'apport de transformations au niveau algorithmique sur le compromis précision-coût de l'implantation. Plus particulièrement les transformations de boucles seront prises en compte. Comme pour d'autres aspects du développement de systèmes embarqués, l'optimisation au niveau algorithmique est source de gains importants sur le coût de l'implantation.

Convergence virgule flottante et virgule fixe

Face à l'évolution de la complexité des applications nous assistons à une convergence entre l'arithmétique virgule fixe et l'arithmétique virgule flottante au sein des architectures. Les SoC hétérogènes permettent d'intégrer des cœurs de processeurs utilisant l'arithmétique virgule fixe et d'autres l'arithmétique virgule flottante. De même, plusieurs processeurs de traitement du signal récents intègrent des unités de calcul supportant les arithmétiques virgule flottante et virgule fixe. Typiquement, l'ajout d'unités flottantes a pour objectif de pouvoir supporter des traitements sensibles en termes de précision tels que par exemple certains traitements graphiques ou l'inversion de matrices présente pour les systèmes de communications mobiles de quatrième génération. L'arithmétique virgule fixe est présente pour assurer de bonnes performances en termes de temps d'exécution, de consommation d'énergie et l'arithmétique virgule flottante est présente pour assurer une précision suffisante pour les parties de l'application critiques en termes de précision ou de dynamique. Cette approche peut aussi permettre de réduire les temps de mise sur le marché en proposant rapidement une version de l'application fonctionnant avec la représentation en virgule flottante puis, une nouvelle version de l'application, optimisant la représentation des données en utilisant les types en virgule fixe, peut être proposée par la suite. Dans ce contexte, uniquement les parties sensibles en termes de temps d'exécution du code source de l'application peuvent être converties en virgule fixe, car l'optimisation de cette portion de code sera bénéfique pour le coût global de l'implantation.

Pour une implantation matérielle, les types en virgule flottante dédiés (petits flottants) peuvent être utilisés pour réduire la taille des données et des opérateurs par rapport aux types normalisés et ainsi conduire à

2. *Multiprocessor System on Chip.*
3. *Network on Chip.*
4. *Architecture oriented paraLlelization for high performance embedded Multicore systems using scilAb.*

des coûts d'implantation plus faibles [28]. Ces types peuvent fournir des caractéristiques en termes de qualité numérique supérieures à celles proposées par l'arithmétique en virgule fixe. Dans le cas de types en virgule flottante dédiés, les tailles de la mantisse et de l'exposant peuvent être optimisées en fonction des contraintes associées à l'application. Cependant, même pour des applications pouvant tolérer une précision moyenne, la dynamique des données et la précision des calculs doivent être analysées soigneusement.

Dans un premier temps, l'objectif est de comparer les types en virgule flottante (normalisés, petits flottants) et en virgule fixe en termes de compromis coût d'implantation - précision des calculs et de fournir les plages pour lesquelles chaque type possède le meilleur compromis. Dans un second temps, l'objectif des travaux de recherche est de fournir une méthodologie de transformation de code source à source générant un code intégrant des types flottants et fixes. L'objectif est toujours de minimiser le coût de l'implantation pour une contrainte de précision donnée. Le challenge se situe au niveau de l'évaluation de la précision. Notre approche analytique de détermination de l'expression de la puissance du bruit en sortie de l'application doit être étendue afin d'intégrer le bruit lié aux opérations en virgule flottante. Cette extension n'est pas directe car le modèle de bruit utilisé pour la virgule fixe n'est pas valide pour le cas de l'arithmétique virgule flottante. L'intégration de types en virgule flottante concerne les types normalisés au sein de la norme IEEE-754, mais aussi les types flottants customisés. Dans ce dernier cas, les tailles de la mantisse et de l'exposant peuvent être optimisées afin de minimiser le coût de l'implantation dans le cadre du processus de conversion en précision finie.

6.2.3 Adéquation application-système pour les systèmes de communication

Au cours de mes différents travaux de recherche, j'ai été amené à travailler sur des applications issues du domaine des communications numériques. Je souhaite développer cette thématique concernant l'adéquation algorithme architecture dans le domaine des systèmes de communication.

Les systèmes de communication numérique sont confrontés à un challenge majeur correspondant à l'efficacité énergétique des plate-formes utilisées. Les techniques utilisées au sein de la couche physique sont de plus en plus complexes afin d'améliorer le débit ou la robustesse aux dégradations du canal de transmission. Ainsi, la puissance de calcul nécessaire pour implanter ces techniques ne cesse de croître. Cependant, l'autonomie du système devant être préservée, la consommation de l'application doit être maîtrisée. De plus, plusieurs standards de communication doivent être supportés. Dans l'exemple du *smartphone* utilisé dans l'introduction, au moins quatre standards de communication était disponibles et pour chacun plusieurs bandes de fréquence non adjacentes étaient supportées. Ainsi, l'utilisation du concept de radio-logicielle [31] devient indispensable afin de pouvoir supporter différents standards. Ceci nécessite des capacités d'adaptabilité et de supporter la re-configuration.

Je souhaite aborder cette thématique sous deux angles. Le premier concerne les aspects modélisation au niveau système d'un ensemble d'applications de communication numérique à implanter au sein d'une plate-forme. Cet aspect rejoint la mise en œuvre de la méthode d'optimisation des largeurs au niveau système qui nécessite de définir la manière de modéliser l'application. Le deuxième aspect concerne la mise en œuvre de transformations algorithmiques permettant d'optimiser la consommation d'énergie.

Chapitre 7

Bibliographies

7.1 Bibliographie personnelle

RÉFÉRENCES

Revues internationales

[Rocher 12] R. Rocher, **D. Ménard**, P. Scalar & O. Sentieys. *General Model for Analytical Evaluation of Fixed-point systems*. Accepted for publication in IEEE Transactions on Circuits and Systems I, 2012. ⟨pp. 22, 28⟩

[Alam 12a] M. Alam, O. Berder, **D. Ménard**, O. Sentieys TAD-MAC : Traffic-Aware Dynamic MAC Protocol for Wireless Body Area Sensor Networks. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol.PP, 99, 2012, doi :10.1109/JETCAS.2012.2187243 ⟨p. 71⟩

[Ménard 12] **D. Ménard**, N. Hervé, O. Sentieys & H.-N. Nguyen. *High-Level Synthesis under fixed-point accuracy constraint*. Journal of Electrical and Computer Engineering, 2012. ⟨p. 57⟩

[Caffarena 12] G. Caffarena, O. Sentieys, **D. Ménard**, J.A. López, and D. Novo. Quantization of VLSI Digital Signal Processing Systems. *EURASIP Journal on Applied Signal Processing*, 2012, 2012 :32 ⟨pp. 5, 6⟩

[Alam 11a] M. Alam, O. Berder, D. Ménard, T. Anger & O. Sentieys. *A Hybrid Model for Accurate Energy Analysis of WSN nodes*. Eurasip Journal on Embedded Systems, 2011. ⟨pp. 14, 71, 72⟩

[Rocher 10] R. Rocher, D. Ménard, O. Sentieys & P. Scalart. *Accuracy Evaluation of Fixed-Point based LMS Algorithm*. Digital Signal Processing, vol. 20, pages 640–652, May 2010. ⟨pp. 14, 22, 74⟩

[Khan 10] S. Khan, E. Casseau & D. Ménard. *High speed reconfigurable SWP operator for multimedia processing using redundant data representation*. Information Sciences and Computer Engineering, vol. 1, pages 45–52, may 2010. ⟨pp. 14, 75⟩

[Ménard 08a] D. Ménard, R. Rocher & O. Sentieys. *Analytical Fixed-Point Accuracy Evaluation in Linear Time-Invariant Systems*. IEEE Transactions on Circuits and Systems I - Regular Papers, vol. 55, no. 1, pages 3197–3208, november 2008. ⟨pp. 12, 22, 32⟩

[Ménard 08b] D. Ménard, R. Serizel, R. Rocher & O. Sentieys. *Accuracy Constraint Determination in Fixed-Point System Design*. EURASIP Journal on Embedded Systems, vol. 2008, page 12, 2008. ⟨pp. 12, 36, 37⟩

[Ménard 06] D. Ménard, D. Chillet & O. Sentieys. *Floating-to-fixed-point Conversion for Digital Signal Processors*. EURASIP Journal on Applied Signal Processing, vol. 2006, pages 1–19, january 2006. ⟨pp. 11, 51, 56, 69⟩

[Rocher 06b] R. Rocher, D. Ménard, N. Hervé & O. Sentieys. *Fixed-Point Configurable Hardware Components*. EURASIP Journal on Embedded Systems, vol. 2006, pages 1–13, 2006. Article ID 23197. ⟨pp. 14, 71, 74⟩

[Naud 11] J.C. Naud, **D. Ménard**, G. Caffarena, O. Sentieys. *A Discrete Model for Correlation Between Quantization Noises.* Minor revision after submission in IEEE Transactions on Circuits and Systems II, 2011. ⟨ pp. ⟩

Revues nationales

[Ménard 03d] D. Ménard, T. Saidi, D. Chillet & O. Sentieys. *Implantation d'algorithmes spécifiés en virgule flottante dans les DSP virgule fixe.* Technique et Science Informatiques, vol. 22, no. 6, pages 783–810, 2003. ⟨ pp. ⟩

[Naud 11c] J.C. Naud D. Ménard and O. Sentieys. Évaluation de la précision en virgule fixe dans le cas des structures conditionnelles. *Soumis à Technique et Science Informatiques*, 2011. ⟨ pp. ⟩

Conférences internationales

[Ménard 11] D. Ménard, H.N. Nguyen, F. Charot, S. Guyetant, J. Guillot, E. Raffin & E. Casseau. *Exploiting Reconfigurable SWP Operators for Multimedia Applications.* In Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Prague, may 2011. ⟨ pp. 11, 13, 51, 56, 69 ⟩

[Banciu 11] A. Banciu, E. Casseau, D. Ménard & T. Michel. *Stochastic Modeling for Floating-point to Fixed-point Conversion.* In Proc. IEEE International Workshop on Signal Processing Systems, (SIPS), Beirut, october 2011. ⟨ pp. 46, 48, 49, 50 ⟩

[Nguyen 11] H.-N. Nguyen, D. Ménard & O. Sentieys. *Novel Algorithms for Word-length Optimization.* In Proc. European Signal Processing Conference (EUSIPCO), Barcelona, september 2011. ⟨ pp. 11, 51, 54 ⟩

[Alam 11b] M. Alam, O. Berder, D. Ménard & O. Sentieys. *Accurate Energy Consumption Evaluation of Preamble Sampling MAC Protocols for WS.* In Proc. Architecture of Computing Systems (ARCS), Como, february 2011. ⟨ p. 71 ⟩

[Naud 11b] J.C. Naud, Q. Meunier, D. Ménard & O. Sentieys. *Fixed-point Accuracy Evaluation in the Context of Conditional Structures.* In Proc. European Signal Processing Conference (EUSIPCO), Barcelona, september 2011. ⟨ p. 22 ⟩

[Parashar 10a] K. Parashar, D. Ménard, R. Rocher & O. Sentieys. *Estimating Frequency Characteristics of Quantization Noise for Performance Evaluation of Fixed Point Systems.* In Proc. European Signal Processing Conference (EUSIPCO), pages 552–556, Aalborg, August 2010. ⟨ pp. 13, 36, 37 ⟩

[Parashar 10b] K. Parashar, D. Ménard, R. Rocher, O. Sentieys, D. Novo & F. Catthoor. *Fast Performance Evaluation of Fixed-Point Systems with Un-Smooth Operators.* In Proc. IEEE/ACM International Conference on Computer-Aided Design (ICCAD), San Jose, 11 2010. ⟨ pp. 6, 13, 36, 39, 42 ⟩

[Parashar 10c] K. Parashar, R. Rocher, D. Ménard & O. Sentieys. *A Hierarchical Methodology for Word-Length Optimization of Signal Processing Systems.* In Proc. International Conference on VLSI Design, Bangalore, January 2010. ⟨ pp. 11, 61, 65 ⟩

[Parashar 10d] K. Parashar, R. Rocher, D. Ménard & O. Sentieys. *Analytical Approach for Analyzing Quantization Noise Effects on Decision Operators.* In Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pages 1554–1557, Dallas, march 2010. ⟨ pp. 13, 36, 38 ⟩

[Parashar 10e] Karthick Parashar, Daniel Ménard, Romuald Rocher & Olivier Sentieys. *Shaping Probability Density Function of Quantization Noise in Fixed Point Systems.* In Proc. Annual Asilomar Conference on Signals, Systems, and Computers, Monterey, 12 2010. ⟨ pp. 13, 36 ⟩

[Banciu 10] A. Banciu, E. Casseau, D. Ménard & T. Michel. *A Case Study Of The Stochastic Modeling Approach For Range Estimation.* In Proc. Workshop on Design and Architectures for Signal and Image Processing (DASIP), pages 301–308, Edinburgh, october 2010. ⟨ pp. 10, 46 ⟩

[Ménard 10] D. Ménard, D. Novo, R. Rocher, F. Catthoor & O. Sentieys. *Quantization Mode Opportunities in Fixed-Point System Design.* In Proc. European Signal Processing Conference (EUSIPCO), pages 542–546, Aalborg, august 2010. ⟨ pp. 6, 22, 25 ⟩

[Nguyen 09b] H.-N. Nguyen, D. Ménard & O. Sentieys. *Dynamic precision scaling for low power WCDMA receiver* . In Proc. IEEE International Symposium on Circuits and Systems (ISCAS), Taipei, may 2009. ⟨ pp. 14, 75 ⟩

[Nguyen 09a] H.-N. Nguyen, D. Ménard & O. Sentieys. *Design of Optimized Fixed-point WCDMA Receiver*. In Proc. European Signal Processing Conference (EUSIPCO), Glasgow, august 2009. ⟨ pp. 14, 72, 75, 77 ⟩

[Khan 09a] S. Khan, E. Casseau & D. Ménard. *Reconfigurable SWP Operator for Multimedia Processing*. In Proc. IEEE International Conference on Application-Specific Systems, Architectures and Processors (ASAP), pages 199–202, Boston, July 2009. ⟨ p. 81 ⟩

[Ménard 09c] D. Ménard, E. Casseau, S. Khan, O. Sentieys, S. Chevobbe, S. Guyetant & R. David. *Reconfigurable Operator Based Multimedia Embedded Processor*. In Proc. International Workshop on Applied Reconfigurable Computing (ARC), volume 5453, pages 39–49, Karlsruhe, march 2009. Springer Berlin / Heidelberg. ⟨ pp. 14, 75 ⟩

[Khan 09b] S. Khan, E. Casseau & D. Ménard. *SWP for multimedia operator design*. In Proc. Conference, Sciences of Electronic, Technologies of Information and Telecommunications, Hammamet, april 2009. ⟨ pp. 14, 75 ⟩

[Hilaire 08] T. Hilaire, D. Ménard & O. Sentieys. *Bit Accurate Roundoff Noise Analysis of Fixed-point Linear Controllers*. In Proc. IEEE International Conference on Computer-Aided Control Systems (CACSD), pages 607–612, september 2008. ⟨ pp. 14, 71, 74 ⟩

[Nguyen 08] H.-N. Nguyen, D. Ménard, R. Rocher & O. Sentieys. *Energy reduction in wireless system by dynamic adaptation of the ?xed-point speci ?cation*. In Proc. Workshop on Design and Architectures for Signal and Image Processing (DASIP), Bruxelles, november 2008. ⟨ pp. 14, 75 ⟩

[Hervé 07] N. Hervé, D. Ménard & O. Sentieys. *About the Imporance of Operation Grouping Procedures for Multiple Word-Length Architecture Optimizations*. In Proc. International Workshop on Applied Reconfigurable Computing (ARC), Rio de Janeiro, march 2007. Springer Berlin / Heidelberg. ⟨ pp. 11, 57 ⟩

[Hilaire 07] T. Hilaire, D. Ménard & O. Sentieys. *Roundoff noise of finite wordlength realizations with the implicit state-space framework* . In Proc. European Signal Processing Conference (EUSIPCO), Poznan, september 2007. ⟨ pp. 14, 71, 74 ⟩

[Ménard 07] D. Ménard, R. Rocher, O. Sentieys & R. Serizel. *Noise model for Accuracy Constraint Determination in Fixed-point Systems*. In Proc. Workshop on Design and Architectures for Signal and Image Processing (DASIP), grenoble, November 2007. ⟨ pp. 12, 36 ⟩

[Rocher 07a] R. Rocher, D. Ménard, P. Scalart & O. Sentieys. *Analytical accuracy evaluation of Fixed-Point Systems*. In Proc. European Signal Processing Conference (EUSIPCO), Poznan, September 2007. ⟨ pp. 12, 22, 26 ⟩

[Rocher 06a] R. Rocher, N. Herve, D. Ménard & O. Sentieys. *Fixed-point Configurable Hardware Components for Adaptive Filters*. In Proc. IEEE International Symposium on Circuits and Systems (ISCAS), Kos, May 2006. ⟨ pp. 14, 71, 74 ⟩

[Rocher 05a] R. Rocher, D. Ménard, P. Scalart & O. Sentieys. *Accuracy Evaluation of Fixed-point APA Algorithm*. In Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pages 57–60, Philadelphie, March 2005. ⟨ pp. 14, 22 ⟩

[Herve 05] N. Herve, D. Ménard & O. Sentieys. *Data Wordlength Optimization for FPGA Synthesis*. In Proc. IEEE International Workshop on Signal Processing Systems, (SIPS), pages 623–628, Athens, november 2005. ⟨ pp. 11, 57, 59 ⟩

[Hannig 05] F. Hannig, H. Dutta, A. Kupriyanov, J. Teich, R. Schaffer, S. Siegel, R. Merker, R. Keryell, B. Pottier, D. Chillet, D. Ménard & O. Sentieys. *Co-Design of Massively Parallel Embedded Processor Architectures*. In Proc. Workshop on Reconfigurable Communication-Centric SoCs (ReCoSoC 2005), Montpellier, june 2005. ⟨ pp. 6, 51 ⟩

[Rocher 04] R. Rocher, D. Ménard, P. Scalart & O. Sentieys. *Accuracy Evaluation of Fixed-point LMS algorithm.* In Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pages 237–240, Montreal, May 2004. ⟨ pp. 14, 22, 74 ⟩

[Ménard 04a] D. Ménard, R. Rocher, P. Scalart & O. Sentieys. *SQNR determination in non-linear and non-recursive fixed-point systems.* In Proc. European Signal Processing Conference (EUSIPCO), pages 1349–1352, Vienna, september 2004. ⟨ pp. 12, 22 ⟩

[Ménard 04b] D. Ménard & O. Sentieys. *DSP Code Generation with Optimized Data-Word Length Selection .* In Proc. International Workshop on Software and Compilers for Embedded Systems (SCOPES), Amsterdam, september 2004. ⟨ p. 51 ⟩

[Ménard 03c] D. Ménard, M. Guitton, S. Pillement & O. Sentieys. *Design and Implementation of WCDMA Platforms : Challenges and Trade-offs.* In Proc. International Signal Processing Conference (ISPC 2003), Dallas, april 2003. ⟨ pp. 14, 72 ⟩

[Ménard 02b] D. Ménard, D. Chillet, F.Charot & O. Sentieys. *Automatic Floating-point to Fixed-point Conversion for DSP Code Generation.* In Proc. ACM International Conference on Compilers, Architectures and Synthesis for Embedded Systems (CASES), pages 270–276, grenoble, October 2002. ⟨ pp. ⟩

[Ménard 02c] D. Ménard, P. Quemerais & O. Sentieys. *Influence of fixed-point DSP architecture on computation accuracy.* In Proc. European Signal Processing Conference (EUSIPCO), pages 587–590, Toulouse, september 2002. ⟨ pp. ⟩

[Ménard 02e] D. Ménard & O. Sentieys. *A methodology for evaluating the precision of fixed-point systems.* In Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Orlando, may 2002. ⟨ pp. ⟩

[Ménard 02f] D. Ménard & O. Sentieys. *Automatic Evaluation of the Accuracy of Fixed-point Algorithms.* In Proc. Design, Automation and Test in Europe (DATE), Paris, march 2002. ⟨ p. 22 ⟩

Conférences nationales

[Naud 11a] J.C. Naud, D. Ménard, Q. Meunie & O. Sentieys. *Evaluation de la précision en virgule fixe dans le cas des structures conditionnelles.* In Proc. Symposium en Architectures nouvelles de machines (SYMPA), Saint Malo, may 2011. ⟨ pp. 22, 28, 29 ⟩

[Rocher 07b] R. Rocher, D. Ménard, O. Sentieys & P. Scalart. *Evaluation de la précision des systèmes en Virgule Fixe.* In Proc. Colloque sur le traitement du signal et des images (GRETSI), Troyes, september 2007. ⟨ p. 22 ⟩

[Hervé 05a] N. Hervé, D. Ménard & O. Sentieys. *Synthèse d'architecture sur FPGA sous contrainte de précision des calculs.* In Proc. Symposium en Architectures nouvelles de machines (SYMPA), Le Croisic, april 2005. ⟨ p. 57 ⟩

[Hervé 05b] N. Hervé, D. Ménard & O. Sentieys. *Optimisation de la largeur des opérateurs arithmétiques en synthèse de haut-niveau.* In Proc. Colloque sur le traitement du signal et des images (GRETSI), Louvain, september 2005. ⟨ p. 57 ⟩

[Rocher 05b] R. Rocher, D. Ménard, O. Sentieys & P. Scalart. *Evaluation de la précision des Algorithmes de Projection Affine en Virgule Fixe.* In Proc. Colloque sur le traitement du signal et des images (GRETSI), Louvain, september 2005. ⟨ p. 22 ⟩

[Ménard 02d] D. Ménard, T. Saidi, D. Chillet & O. Sentieys. *Implantation d'algorithmes spécifiés en virgule flottante dans les DSP virgule fixe.* In Proc. Symposium en Architectures nouvelles de machines (SYMPA), Hammamet, avril 2002. ⟨ pp. ⟩

[Ménard 01] D. Ménard and O. Sentieys. Influence du modèle de l'architecture des DSPs virgule fixe sur la précision des calculs. In Proc. Colloque sur le traitement du signal et des images (GRETSI), Toulouse, Septembre 2001. ⟨ pp. ⟩

[Ménard 03b] D. Ménard, M. Guitton, R. David, S. Pillement & O. Sentieys. *Évaluation comparative de plates-formes reconfigurables et programmables pour les télécommunications de 3ème génération.* In Proc.

Colloque sur le traitement du signal et des images (GRETSI), France, septembre 2003. ⟨ pp. ⟩

Thèse

[Ménard 02a] D. Ménard. *Methodologie de compilation d'algorithmes de traitement du signal pour les processeurs en virgule fixe, sous contrainte de precision.* PhD thesis, Universite de Rennes I, Lannion, december 2002. ⟨ pp. 68, 71 ⟩

Présentations invitées

[Ménard 09a] D. Ménard. *Arithmétique pour les applications de traitement du signal embarquées.* In 4èmes Rencontres "Arithmétique de l'Informatique Mathématique" (RAIM), Lyon, november 2009. ⟨ p. 5 ⟩

[Ménard 09b] D. Ménard. *Conversion en virgule fixe pour les applications de traitement numérique du signal.* In École thématique CNRS : Architectures des systèmes matériels enfouis et méthodes de conception associées (ARCHI), Pleumeur-Bodou, April 2009. ⟨ p. 5 ⟩

[Ménard 05] D. Ménard. *Techniques de conversion en virgule fixe pour les applications de traitement du signal.* In Séminaire du laboratoire LRTS, Université Laval, Quebec, june 2005. ⟨ pp. 5, 6 ⟩

[Ménard 03a] D. Ménard. *Méthodologies de conversion en virgule fixe.* In École thématique CNRS : Architectures des systèmes matériels enfouis et méthodes de conception associées (ARCHI), Roscoff, April 2003. ⟨ p. 5 ⟩

Rapports

[Blanc 06] F. Blanc, D. Ménard, D. Chillet, O. Sentieys, B. Pottier, L. Lagadec & F. Dupont. *Rapport Final Projet RNTL OSAGR.* Rapport technique, CEA, IRISA, UBO, TNI, january 2006. ⟨ p. 57 ⟩

7.2 Bibliographie générale

Références

[1] AccelChip. Automated Conversion of Floating-point to Fixed-point MATLAB. Technical report, AccelChip Inc., Milpitas, 2004. ⟨ p. 23 ⟩

[2] G. Alefeld and J. Herzberger. *Introduction to Interval Computations*. Academic Press, New York, 1983. ⟨ pp. 22, 46 ⟩

[3] Analog Device. *TigerSHARC Hardware Specification*. Analog Device, december 1999. ⟨ p. 52 ⟩

[4] S. Balacoo, C. Rommel, and J. Weiner. Searching for the Total Size of the Embedded Software Engineering Market, february 2011. ⟨ p. 15 ⟩

[5] C. Barnes, B. N. Tran, and S. Leung. On the Statistics of Fixed-Point Roundoff Error. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 33(3) :595–606, june 1985. ⟨ p. 24 ⟩

[6] J. Barry, E. Lee, and D. Messerschmitt. *Digital Communication*. Springer, third edition, 2003. ⟨ p. 21 ⟩

[7] F. Berens and N. Naser. *Algorithm to System-on-Chip Design Flow that Leverages System Studio and SystemC 2.0.1*. Synopsys Inc., march 2004. ⟨ p. 23 ⟩

[8] J. Bier. BDTI Certified Benchmark Results. http ://www.bdti.com/Resources/BenchmarkResults, 2010. ⟨ p. 17 ⟩

[9] J. Bier. Jeff Bier's Impulse Response-The Floating-Point Future, october 2010. ⟨ p. 18 ⟩

[10] M. Blair, S. Obenski, and P. Bridickas. Patriot Missile Defense : Software Problem Led to System Failure at Dhahran, Saudi Arabia. Technical Report GAO/IMTEC-92-26, United States Department of Defense, General Accounting office, 1992. ⟨ p. 16 ⟩

[11] S. Blinnikov and R. Moessner. Expansions for nearly Gaussian distributions. *Astronomy and astrophysics suplement series*, 130 :193–205, may 1998. ⟨ p. 49 ⟩

[12] G. Caffarena, J.A. López, . Fernandez, and C. Carreras. SQNR Estimation of Fixed-Point DSP Algorithms. *EURASIP Journal on Advance Signal Processing*, volume 2010, 2010. ⟨ pp. 24, 31, 35, 36, 44, 107 ⟩

[13] G. Caffarena, J.A. López, G. Leyva, C. Carreras, and O. Nieto-Taladriz. Architectural Synthesis of Fixed-Point DSP Datapaths Using FPGAs. *International Journal of Reconfigurable Computing*, vol. 2009, volume 2009. ⟨ pp. 60, 61 ⟩

[14] M. Cantin, Y. Savaria, D. Prodanos, and P. Lavoie. An automatic word length determination method. In *IEEE International Symposium on Circuits and Systems*, volume 5, pages 53–56, 2001. ⟨ p. 53 ⟩

[15] M. Cantin, Y. Savaria, D. Prodanos, and P. Lavoie. A Metric for Automatic Word-Length Determination of Hardware Datapaths. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(10) :2228–2231, october 2006. ⟨ p. 22 ⟩

[16] C. Caraiscos and B. Liu. A Roundoff Error Analysis of the LMS Adaptive Algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(1) :34–41, february 1984. ⟨ p. 22 ⟩

[17] J.-M. Chesneaux, L.-S. Didier, and F. Rico. Fixed CADNA library. In *Proc. conference on Real Number Conference (RNC)*, pages 215–221, Lyon, France, september 2003. ⟨ p. 22 ⟩

[18] M. Clark, M. Mulligan, D. Jackson, and D. Linebarger. Accelerating Fixed-Point Design for MB-OFDM UWB Systems. *CommsDesign*, january 2005. ⟨ p. 18 ⟩

[19] G. Constantinides, P. Cheung, and W. Luk. Truncation Noise in Fixed-Point SFGs. *IEE Electronics Letters*, 35(23) :2012–2014, november 1999. ⟨ pp. 24, 25 ⟩

[20] G. Constantinides, P. Cheung, and W. Luk. Roundoff-noise shaping in filter design. In *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, volume 4, pages 57–60, Geneva, may 2000. ⟨ p. 22 ⟩

[21] George A. Constantinides. Word-length optimization for differentiable nonlinear systems. *ACM Transactions on Design Automation of Electronic Systems*, 11(1) :26–43, 2006. ⟨ pp. 24, 31, 35, 43 ⟩

[22] M. Coors, H. Keding, O. Luthje, and H. Meyr. Integer Code Generation For the TI TMS320C62x. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Sate Lake City, may 2001. ⟨ p. 23 ⟩

[23] L. De Coster. *Bit-True Simulation of Digital Signal Processing Applications*. PhD thesis, KU Leuven, 1999. ⟨ p. 23 ⟩

[24] L. De Coster, M. Ade, R. Lauwereins, and J.A. Peperstraete. Code Generation for Compiled Bit-True Simulation of DSP Applications. In *Proc. IEEE International Symposium on System Synthesis (ISSS)*, pages 9–14, Hsinchu, december 1998. ⟨ p. 23 ⟩

[25] Coware. CoWare Signal Processing Designer, Implementing DSP Algorithms for Complex Wireless System Design. Technical report, Coware, 2009. ⟨ pp. 17, 23 ⟩

[26] Coware. Coware SPW. Technical report, Coware, 2010. ⟨ pp. 17, 23 ⟩

[27] M. Daumas and G. Melquiond. Certification of bounds on expressions involving rounded operators. *ACM Transaction on Mathematical Software*, 37 :1–20, january 2010. ⟨ p. 22 ⟩

[28] F. de Dinechin, C. Klein, and B. Pasca. Generating high-performance custom floating-point pipelines. In *Proc. International Conference on Field Programmable Logic and Applications (FPL)*, pages 59 –64, Prague, september 2009. ⟨ p. 92 ⟩

[29] L.H. de Figueiredo and J. Stolfi. Affine arithmetic : Concepts and applications. *Numerical Algorithms*, 37(1) :147–158, 2004. ⟨ pp. 22, 46 ⟩

[30] D. Delmas, E. Goubault, S. Putot, J. Souyris, K. Tekkal, and F. Védrine. Towards an industrial use of FLUCTUAT on safety-critical avionics software. In *Proc. International Workshop on Formal Methods for Industrial Critical Systems (FMICS)*, volume 5825 of *LNCS*, pages 53–69, Eindhoven, november 2009. ⟨ p. 22 ⟩

[31] M. Dillinger, K. Madani, and N. Alonistioti. *Software Defined Radio : Architectures, Systems and Functions*. Wiley, april 2003. ⟨ p. 92 ⟩

[32] J. Eker, J. W. Janneck, E. A. Lee, J. Liu, X. Liu, J. Ludvig, S. Neuendorffer, S. Sachs, and Y. Xiong. Taming Heterogeneity, the Ptolemy Approach. *Proceedings of the IEEE*, 91, 2003. ⟨ pp. 17, 23 ⟩

[33] B. Evans. Modem Design, Implementation, and Testing Using NI's LabVIEW. In *National Instrument Academic Day*, Beirut, june 2005. ⟨ p. 17 ⟩

[34] J. Eyre and J. Bier. DSPs court the consumer. *IEEE Spectrum*, 36(3) :47–53, 1999. ⟨ p. 17 ⟩

[35] J. Eyre and J. Bier. The evolution of DSP processors. *IEEE Signal Processing Magazine*, 17(2) :43–51, march 2000. ⟨ p. 17 ⟩

[36] T. Feo and M. Resende. A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters*, 8(2) :67–71, 1989. ⟨ pp. 11, 54 ⟩

[37] M. Fingerroff. *High-Level Synthesis Blue Book*. Xlibris, 2010. ⟨ p. 17 ⟩

[38] P.D. Fiore. Efficient Approximate Wordlength Optimization. *IEEE Transactions on Computers*, 57(11) :1561 –1570, november 2008. ⟨ pp. 24, 31, 35, 43 ⟩

[39] L. Fousse, G. Hanrot, V. Lefèvre, P. Pélissier, and P. Zimmermann. MPFR : A multiple-precision binary floating-point library with correct rounding. *ACM Transactions on Mathematical Software*, 33, june 2007. ⟨ p. 23 ⟩

[40] J. Fridman. Sub-Word Parallelism in Digital Signal Processing. *IEEE Signal Processing Magazine*, 17(2) :27–35, march 2000. ⟨ p. 52 ⟩

[41] K. Han. *Automating transformations from floating-point to fixed-point for implementing digital signal processing algorithms*. PhD thesis, University of Texas, Austin, august 2006. ⟨ p. 55 ⟩

[42] K. Han and B. Evans. Wordlength optimization with complexity-and-distortion measure and its application to broadband wireless demodulator design. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 37–40, Montreal, may 2004. ⟨ p. 55 ⟩

[43] B. Hassibi and H. Vikalo. On the sphere-decoding algorithm I. Expected complexity. *IEEE Transactions on Signal Processing*, 53(8) :2806–2818, august 2005. ⟨ p. 41 ⟩

[44] N. Hervé. *Contributions à la synthèse d'architecture virgule fixe à largeurs multiples*. PhD thesis, Université de Rennes 1, Lannion, march 2007. ⟨ pp. 5, 54, 57, 71 ⟩

[45] T. Hilaire. *Analyse et synthèse de l'implémentation de lois de contrôle-commande en précision finie*. Phd, Université de Nantes, Nantes, june 2006. ⟨ p. 74 ⟩

[46] T. Hilaire. Finite Wordlength Realizations Toolbox User's Guide. Technical report, 2009. ⟨ p. 74 ⟩

[47] T. Hill. AccelDSP Synthesis Tool Floating-Point to Fixed-Point Conversion of MATLAB Algorithms Targeting FPGAs. White papers, Xilinx, april 2006. ⟨ p. 18 ⟩

[48] A. Joshi. Embedded Systems : Technologies and Markets. Technical report, VDC Research, april 2009. ⟨ p. 15 ⟩

[49] J. Kang and W. Sung. Fixed-Point C Compiler for TMS320C50 Digital Signal Processor. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Munich, april 1997. ⟨ p. 23 ⟩

[50] H. Keding. Pain Killers for the Fixed-Point Design Flow. Technical report, Synopsys Inc., 2010. ⟨ pp. 17, 23 ⟩

[51] S. Khan. *Development of high performance hardware architectures for multimedia applications*. Phd, Université Rennes 1, Lannion, september 2010. ⟨ pp. 5, 75, 80, 81 ⟩

[52] S. Kim, K. Kum, and S. Wonyong. Fixed-Point Optimization Utility for C and C++ Based Digital Signal Processing Programs. *IEEE Transactions on Circuits and Systems II - Analog and Digital Signal Processing*, 45(11) :1455–1464, november 1998. ⟨ p. 23 ⟩

[53] S. Kim and W. Sung. A Floating-point to Fixed-point Assembly program Translator for the TMS 320C25. *IEEE Transactions on Circuits and Systems*, 41(11) :730–739, november 1994. ⟨ p. 23 ⟩

[54] S. Krithivasan, M. J. Schulte, and J. Glossner. A Subword-Parallel Multiplication and Sum-of-Squares Unit. *Proc. IEEE Computer Society Annual Symposium on VLSI*, page 273, 2004. ⟨ pp. 82, 83 ⟩

[55] K. Kum, J.Y. Kang, and W.Y. Sung. AUTOSCALER for C : An optimizing floating-point to integer C program converter for fixed-point digital signal processors. *IEEE Transactions on Circuits and Systems II - Analog and Digital Signal Processing*, 47(9) :840–848, september 2000. ⟨ p. 23 ⟩

[56] M. Lau, K-V Ling, and Y-C Chu. Energy-aware probabilistic multiplier : design and analysis. In *Proc. ACM International Conference on Compilers, Architectures and Synthesis for Embedded Systems (CASES)*, pages 281–290, Grenoble, october 2009. ACM. ⟨ p. 90 ⟩

[57] J.L. Lions, L. Lubeck, J.L. Fauquembergue, G. Kahn, W. Kubbat, S. Levedag, L. Mazzini, D. Merle, and C O'Halloran. ARIANE 5, Flight 501 Failure. Technical report, European Space Agency, july 1996. ⟨ p. 16 ⟩

[58] M. Loève. *Probability Theory*. Springer-Verlag, Berlin, 4 edition, 1977. ⟨ pp. 10, 47 ⟩

[59] T. Lukasiak. Extended-Precision Fixed-Point Arithmetic on the Blackfin Processor Platform. Technical report, Analog Device, 2003. ⟨ p. 90 ⟩

[60] J.A. López, G. Caffarena, C. Carreras, and O. Nieto-Taladriz. Fast and accurate computation of the roundoff noise of linear time-invariant systems. *IET Circuits, Devices and Systems*, 2(4) :393–408, august 2008. ⟨ pp. 24, 31, 44 ⟩

[61] A. Maheshwari, W. Burleson, and R. Tessier. Trading off transient fault tolerance and power consumption in deep submicron (DSM) VLSI circuits. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,*, 12(3) :299 –311, march 2004. ⟨ p. 90 ⟩

[62] Mathworks. *Fixed-Point Blockset User's Guide (ver. 2.0)*, 2001. ⟨ pp. 17, 23 ⟩

[63] Mentor Graphics. *Algorithmic C Data Types*. Mentor Graphics, version 1.3 edition, march 2008. ⟨ p. 23 ⟩

[64] H.N. Nguyen. *Optimisation de la précision de calcul pour la réduction d'énergie des systèmes embarqués*. PhD thesis, Université de Rennes I, A soutenir 16 décembre 2011. ⟨ pp. 51, 54, 75 ⟩

[65] D. Novo, B. Bougard, A. Lambrechts andL. Van der Perre, and F. Catthoor. Scenario-based fixed-point data format refinement to enable energy-scalable software defined radios. In *Proc. IEEE/ACM conference on Design, Automation and Test in Europe (DATE)*, pages 722–727, Munich, march 2008. ⟨ p. 75 ⟩

[66] A. Oppenheim and R.W. Schafer. *Discrete Time Signal Processing*. Prentice All Signal Processing series. Prentice All, Upper Saddle River, 2 edition, 1999. ⟨ pp. 24, 30 ⟩

[67] O.Sentieys, J.P.Diguet, and J.L.Philippe. GAUT : a High Level Synthesis Tool dedicated to real time signal processing application. In *University Booth, IEEE/ACM European Design Automation Conference (EURO-DAC)*, Brighton, september 1995. ⟨ p. 59 ⟩

[68] E. Ozer, A.P. Nisbet, and D. Gregg. Stochastic Bitwidth Approximation Using Extreme Value Theory for Customizable Processors. Technical report, Trinity College, Dublin, october 2003. ⟨ p. 49 ⟩

[69] P.R. Panda, B.V.N. Silpa, A. Shrivastava, and K. Gummidipudi. *Power-efficient System Design*. Springer, 2010. ⟨ p. 16 ⟩

[70] B. Parhami. Gaining Speed and Cost Advantage from Imprecise Computer Arithmetic. In *Seminar of Berkeley Initiative in Soft Computing*, University of California, Berkeley, november 2000. ⟨ p. 90 ⟩

[71] R. Rocher. *Évaluation analytique de la précision des systèmes en virgule fixe*. Phd, Université Rennes 1, Lannion, december 2006. ⟨ pp. 5, 22, 25, 26, 27, 28, 34, 43, 71 ⟩

[72] S. Roy and P. Banerjee. An Algorithm for Trading Off Quantization Error with Hardware Resources for MATLAB-Based FPGA Design. *IEEE Transactions on Computers*, 54 :886–896, july 2005. ⟨ p. 23 ⟩

[73] G. Rubino and B. Tuffin. *Rare Event Simulation using Monte Carlo Methods*. Wiley, march 2009. ⟨ p. 90 ⟩

[74] T. Saidi. *Hardware Architectures for WCDMA technology extended to mulitple-antenna systems*. Phd, Université de Rennes I, Lannion, july 2008. ⟨ pp. 5, 72 ⟩

[75] R. Serizel. Implantation en virgule fixe d'un codeur audio. Master's thesis, Université de Rennes I, Lannion, september 2006. ⟨ p. 36 ⟩

[76] C. Shi and R. Brodersen. A perturbation theory on statistical quantization effects in fixed-point DSP with non-stationary inputs. In *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 373–376, Vancouver, may 2004. ⟨ pp. 22, 24, 31, 35, 43 ⟩

[77] A. Sripad and D. L. Snyder. A Necessary and Sufficient Condition for Quantization Error to be Uniform and White. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 25(5) :442–448, october 1977. ⟨ p. 24 ⟩

[78] W. Strauss. DSP chips take on many forms. DSP-FPGA.com Magazine, march 2006. ⟨ p. 17 ⟩

[79] A. Tisserand. Function approximation based on estimated arithmetic operators. In *Proc. Asilomar Conference on Signals, Systems and Computers*, pages 1798 –1802, nov. 2009. ⟨ p. 90 ⟩

[80] S. Wadekar and A. Parker. Accuracy sensitive word-length selection for algorithm optimization . In *Proc. International Conference on Computer Design (ICCD)*, pages 54 –61, october 1998. ⟨ p. 60 ⟩

[81] B. Widrow. Statistical Analysis of Amplitude Quantized Sampled-Data Systems. *Transaction on AIEE, Part. II : Applications and Industry*, 79 :555–568, 1960. ⟨ p. 24 ⟩

[82] B. Widrow and I. Kollár. *Quantization Noise : Roundoff Error in Digital Computation, Signal Processing, Control, and Communications*. Cambridge University Press, Cambridge, UK, 2008. ⟨ pp. 24, 27 ⟩

[83] P.W. Wolniansky, G.J. Foschini, G.D. Golden, and R.A. Valenzuela. V-BLAST : an architecture for realizing very high data rates over the rich-scattering wireless channel. In *Proc. URSI International Symposium on Signals, Systems, and Electronics (ISSSE)*, Pisa, september 2008. ⟨ p. 41 ⟩

[84] B. Wu, J. Zhu, and F. Najm. An analytical approach for dynamic range estimation. In *Proc. ACM/IEEE Design Automation Conference (DAC)*, pages 472–477, San Diego, june 2004. ⟨ p. 48 ⟩

[85] S. Yoshizawa and Y. Miyanaga. Tunable word length architecture for low power wireless OFDM demodulator. In *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 2789–2792, Kos, may 2006. ⟨ p. 76 ⟩

Glossaire

ALMA	*Architecture oriented paraLlelization for high performance embedded Multicore systems using scilAb.*, 104
ASIC	*Application-Specific Integrated Circuit.*, 20
ASIP	*Application-Specific Instruction-set Processor.*, 20
BLAST	*Bell Laboratories Layered Space-Time.*, 47
BPSK	*Binary Phase-Shift Keying.*, 44
CAIRN	*Energy efficient computing architectures with embedded reconfigurable resources.*, i
CDFG	*Control Data Flow Graph.*, 35
CDM	*Complexity-and-distortion measure.*, 62
CNRS	Centre national de la recherche scientifique, i
DCT	*Discrete Cosine Transform.*, 33
DSP	*Digital Signal Processor.*, 19
DVFS	*Dynamic Voltage and Frequency Scaling.*, 18
DVS	Dynamic Voltage Scaling., 103
EDGE	*Enhanced Data Rates for GSM Evolution.*, 17
ENS	École normale supérieure, i
Enssat	École nationale supérieure des sciences appliquées et de technologie., i
FFT	*Fast Fourier Transform.*, 31
FIFO	*First In First Out.*, 37
FIR	*Finite Impulse Response.*, 33
FPGA	*Field Programmable Gate Array.*, 20
GPS	*Global Positioning System.*, 17
GSM	*Global System for Mobile communications.*, 17
IFFT	*Inverse Fast Fourier Transform.*, 53
IIR	*Infinite Impulse Response.*, 33
IMEC	*Interuniversity Microelectronics Centre, Leuven, Belgium.*, 44
INRIA	Institut national de recherche en informatique et en automatique., i
IRISA	Institut de recherche en informatique et systèmes aléatoires., i
JNI	*Java Native Interface.*, 79

LMS *Least Mean Square.*, 31
LUT *Look-Up Table.*, 60

MDCT *Modified Discrete Cosine Transform.*, 42
MIMO *Multiple Input Multiple Output.*, 47
MP3 *Moving Picture Experts Group-1/2 Audio Layer 3.*, 42
MPSoC *Multiprocessor System on Chip.*, 103

NoC *Network on Chip.*, 103

OFDM *Orthogonal Frequency-Division Multiplexing.*, 53

PEPS Programmes Exploratoires Pluridisciplinaires., 85
PQN *Pseudo-Quantization Noise.*, 26

QAM *Quadrature Amplitude Modulation.*, 44

R2D2 Reconfigurable and retargetable digital devices., i
RNTL Réseau National en Technologies Logicielles., 65

SFG *Signal Flow Graph*, 35
SSA *Static Single Assignment.*, 31
SSFE *Selective Spanning with Fast Enumeration.*, 47

TDSI Traitement du signal et de l'image, 17
TNS Traitement Numérique du Signal., 20

UMTS *Universal Mobile Telecommunications System.*, 17

VLIW *Very Long Instruction Word.*, 82

WCDMA *Wideband Code Division Multiple Access.*, 82
WiFi *Wireless Fidelity.*, 17
WLAN *Wireless Local Area Network* , 86

XML *Extensible Markup Language.*, 35

Table des figures

Liste des tableaux