Stage de DEA Informatique 97-98

Titre : Partitionnement et Ordonnancement sous Contraintes pour la Synthèse Automatique d'Architectures Spécialisées avec Alpha

Equipe d'accueil : API

Encadreur : Tanguy Risset (`risset@irisa.fr`, bureau D174, tél 7460)

Mot-clés : partitionnement, synthèse d'architectures, parallélisation automatique

Sujet :

Le langage Alpha permet de décrire des algorithmes réguliers (algorithmes d'algèbre linaire, traitement du signal et image, ...) sous la forme d'équations récurrentes. Alpha a été proposé pour faciliter la synthèse d'architectures parallèles intégrées pour ces algorithmes afin de les exécuter en temps raisonnable (temps réel pour le traitement du signal). Les transformations du langage Alpha sont implémentées dans l'environnement MMAlpha.

Un des problèmes ouverts est le problème d'ordonnancer les calculs sous certaines contraintes (contraintes de ressources notamment). La résolution de ce problème est très importante parce qu'elle gouverne toute la suite du processus de synthèse. Certaines stratégies sont connues, leur automatisation se révèle un problème délicat.

Le travail du stage consistera, dans un premier temps, en l'étude des quelque techniques de partitionnement, d'ordonnancement et d'allocation sous contraintes de ressources. Puis on choisira une (ou un type de) partitionnement pour intégrer dans l'outils MMAlpha. On s'intéressera particulièrement à la génération du contrôle pour un programme partitionné, sachant que la cible pourra être une architecture dédiée (Asic, FPGA,...) ou une machine parallèle.

Ce stage donnera lieu à de nombreuses perspectives de recherches, on pourra s'intéresser à l'ensemble du processus de synthèse et à la génération pratique de circuits VLSI ou la programmation de circuits FPGA. On pourra aussi se concentrer sur des aspects plus théoriques d'optimisation pour le parallélisme, et mesurer l'impact des différentes stratégies d'ordonnancement et partitionnement sur les architectures synthétisées.

Stage de DEA Informatique 97-98

Titre : Extension du Langage Alpha au Polyèdres à Trous

Equipe d'accueil : API

Encadreur : Sanjay Rajopadhye (`rajopadhye@irisa,fr`, bureau D169, tél 7208, (avec la participation de T. Risset et P. Quinton.

Mot-clés : programmation fonctionnelle, parallélisme à données, modèle polyèdrique

Sujet :

Le langage Alpha [1], développé dans l'équipe peut être vu, à la fois comme un langage spécialisé pour la synthèse d'architectures systoliques, et aussi comme un langage de parallélisme de données à "usage général". Le formalisme qui le sous-tend est appelé le modèle polyèdrique, et se trouve aujourd'hui, à la base de plusieurs méthodes de parallélisation automatique des boucles, et bien sûr, de la synthèse de réseaux systoliques.

Alpha et le modèle polyèdrique sont pour le moment, restreint aux programmes où les domaines (c.a.d. l'éspace d'itération d'une boucle) sont des polyèdres, et les dépendances du programme sont des fonctions affines entières. Il est souhaitable d'étendre le langage pour bien intégrer les "domaines à trous", c.a.d. des $\mathcal{Z}$-polyèdres. Ceci est motivé à la fois par plusieurs applications comme l'algorithme du décodage de Viterbi, les algorithmes multiresolution et fractales dans le traitement de signal et d'images, l'algorithme de SOR (*successive over-relaxation*) en analyse numérique, et aussi par des manipulations que l'on cherche à faire (des transformations non-unimodulaires, des transformations de sous-échantillonage) soit pour la synthèse d'architectures régulieres dédiées, soit pour la parallélisation automatique. Nos premiers resultats dans cette direction [2, 3] sont prometeurs.

Le stage consistera à étudier l'intégration des changements proposés [2, 3] dans le système MMAlpha, un système de transformation des programmes Alpha, developpé conjointement à Rennes et à Provo, Utah USA, et utilise une bibliothèque de domaine publique pour manipuler des polyèdres. Un aspect de stage sera la definition d'interfaces entre les parties du système (le parseur et le(s) pretty-printer(s), les transformations de changement de base et de normalisation, ..., et les extensions de la bibliothèque), ce qui pourra servir à definir un projet d'implémentation. Une deuxiemme partie (th'eorique) du stage consiste à étudier les questions ouvertes, non traité dans [2], notamment l'interaction de cette extension avec la serialisation des réductions, avec les sous-systèmes, ... Des applications comme le filtre de decimation en reseau multi-horloge et (une partie du) decodeur de Viterbi serviront comme support d'étude.

Bibliographie :

- Mauras, C. Alpha : Un langage Équationnel pour la Conception et la Programmation d'Architectures Parallèles Synchrones. Thèse, Université de Rennes-1, 1989.

- Quinton, P., Rajopadhye, S. et Risset, T. "Extension of the Alpha Language to Recurrences on Sparse Periodic Domains", dans ASAP-96: IEEE Conference on Application Specific Arrays Processing, Chicago, IL, Juillet 1996.
- Quinton, P., Rajopadhye, S. et Risset, T. "On Manipulating $\mathcal{Z}$-polyhedra using a Canonical Representation", dans Parallel Processing Letters (à paraitre).

Stage de DEA Informatique 97-98

Titre : Architectures dédiés pour la multiresolution des signaux

Equipe d'accueil : API

Encadreur : Sanjay Rajopadhye (`rajopadhye@irisa,fr`, bureau D169, tél 7208

Mot-clés : synthèse d'architectures regulières, transformée en ondelettes, circuits multi-horloge.

Sujet :

Le traitement des signaux et d'images à souvent besoin d'un volume de calcul enorme. Plusieurs algorithmes de ce domaine exhibent des caractèristiques comme la regularité, la localité, ..., qui rendent très interessante la realisation sur des architectures dédiées. Ceci est aussi le cas pour des algorithmes multigrilles utilisant, par exemple, les décompositions fractales ou les ondelettes, sauf que bien souvent, les accés aux données se font selon des schemas irréguliers, ce qui rend difficile l'optimisation du compromis entre la surface silicium et le temps d'éxecution.

Recamment une méthode a été proposée par Lenders et Rajopadhye [1] pour la concéption, à partir des simples algorithmes multiresolution (comme les filtres de décimation) des architectures réguliers en utilisant des horloges multiples. Le but du stage est d'une part, et d'éxplorer la synthèse de telles architectures pour des algorithmes plus complexes, notamment la transformée en ondelettes, et d'étudier leur mise en oeuvre sur un coprocesseur reprogammable, basés sur des FPGAs (une carte PAM-ette).

Bibliographie :

– Lenders, P. et Rajopadhye, S. "Multirate VLSI Arrays and their Synthesis". IEEE Transactions on Computers, vol 46, no 5, May 1997, pages 515-529.

Stage de DEA Informatique 97-98

Titre : Synthèse de réseaux systoliques à partir de C-stolic

Equipe d'accueil : API

Encadreur : Dominique Lavenier (`lavenier@irisa.fr`) (D166, tél 7217) 7460)

Mot-clés : synthèse d'architecture, réseau systolique, calcul reconfigurable.

Sujet :

C-stolic est un langage développé dans l'équipe pour programmer des réseaux systoliques linéaires sur des machines parallèles. Le compilateur produit deux programmes C : l'un représente le calcul effectué sur le réseau, l'autre les traitements nécessaires au contrôle du réseau (alimentation en données, filtrage des résultats, synchronization vers le monde extérieur, etc). A partir de ces deux programmes, la mise en oeuvre sur une machine parallèle est immédiate : on utilise le compilateur C natif de la machine parallèle et une bibliothèque de quelques fonctions de base adaptées à son réseau d'interconnexion.

L'efficacité de la mise en oeuvre dépend grandement de l'architecture de la machine parallèle et de son adéquation au modèle systolique, notamment en terme de "granularité" des communications. Les communications sur les machines parallèles actuelles sont efficaces si les messages possèdent une "certaine" taille, taille la plupart du temps incompatible avec les besoins des réseaux systoliques qui requièrent des échanges au niveau mot.

L'objet du stage est d'étudier la mise en oeuvre de C-stolic sur une machine reconfigurable. Dans ce type de machine, le matériel n'est pas figé : il se compose de blocs logiques élémentaires que l'on peut, par programmation, connecter et asservir à une fonction logique particulière. Dans notre cas, il s'agit de "configurer" ce matériel en un réseau systolique.

Ce stage est donc centré sur un thème plus général : la synthèse de réseaux systoliques à partir d'une description dans un langage impératif. Il s'intéresse également à une activité de recherche en pleine extension : le calcul reconfigurable.

Bibliographie

- S. Rubini, D. Lavenier. Les Architectures Reconfigurables. Calculateurs Parallèles, vol 9, no 1, 1997.

- F. Raimbault. Architectures Systoliques et Parallélisme de données; l'environnement de programmation ReLaCS. Thèse de l'Université de Rennes-1, 1994.

Stage de DEA Informatique 97-98

Titre : Outils d'aide à la définition de l'architecture d'un processeur spécialisé programmable

Equipe d'accueil : API

Encadreur : François Charot (charot@irisa,fr, bureau D171, tél 7215)

Mot-clés : compilation, définition d'architecture, modélisation d'architecture

Sujet :

La conception d'un système matériel fait de plus en plus souvent appel à des "cœurs" de processeurs qui sont optimisés et spécialisés pour tenir compte des contraintes d'utilisation du système. La compilation pour de tels processeurs appelés ASIP (*Application Specific Instruction Set Processors*), consiste à produire, pour une application particulière, à la fois l'architecture et le compilateur permettant d'atteindre les performances visées par cette application. Cette technique est l'une des clés de la réalisation de systèmes dédiés aux télécommunications.

Il est primordial que les compilateurs atteignent des performances très proches du code produit manuellement. De plus les compilateurs doivent être *paramétrés par l'architecture visée*, parce qu'il est hors de question de redévelopper le compilateur pour chaque changement architectural. Il faut par conséquent pouvoir adapter très rapidement les compilateurs pour ces processeurs, ce qui pose de nombreux problèmes de recherche.

Le travail du stage concerne le niveau définition de l'architecture de l'ASIP. Plutôt que de faire reposer la définition de l'ASIP uniquement sur l'expérience du concepteur, il peut être intéressant d'utiliser des techniques de compilation pour guider cette étape de définition et organiser l'architecture avec pertinence : choix des unités fonctionnelles, des unités de mémorisation, de la structure de registres, des interconnexions, du type de contrôle, etc. Il s'agira donc d'étudier l'apport de techniques de compilation et de synthèse d'architectures pour dégager des caractéristiques jugées intéressantes dans le code de l'application : accès régulier aux données, enchaînements fréquents d'opérations, parallélisme disponible, etc.

Stage de DEA Informatique 97-98

Titre : Modélisation et synthèse de processeurs spécialisés programmables

Equipe d'accueil : API

Encadreur : François Charot (`charot@irisa,fr`, bureau D171, tél 7215)

Mot-clés : modélisation d'architecture, synthèse d'architecture

Sujet :

La conception d'un système matériel fait de plus en plus souvent appel à des "cœurs" de processeurs qui sont optimisés et spécialisés pour tenir compte des contraintes d'utilisation du système. La compilation pour de tels processeurs appelés ASIP (*Application Specific Instruction Set Processors*), consiste à produire, pour une application particulière, à la fois l'architecture et le compilateur permettant d'atteindre les performances visées par cette application. Cette technique est l'une des clés de la réalisation de systèmes dédiés aux télécommunications.

Les compilateurs doivent être *paramétrés par l'architecture visée*, parce qu'il est hors de question de redévelopper les outils pour chaque changement architectural. La modélisation de l'architecture du processeur cible est la base des techniques de compilation reciblable. Le formalisme doit permettre une modélisation rapide de l'architecture, qui soit lisible, compréhensible par un concepteur de processeur et exploitable par les différentes outils.

Le travail du stage concerne le lien entre la modélisation de l'architecture de l'ASIP et les outils de synthèse d'architecture car la définition de l'architecture de l'ASIP ne peut être complètement déconnectée de la phase de conception. Il s'agira d'étudier le passage entre le formalisme de modélisation de processeurs en cours de définition dans le projet et le langage de description de matériel VHDL utilisé pour la synthèse.