N° d'ordre: 3168

THÈSE

Présentée devant

devant l'Université de Rennes 1

pour obtenir

le grade de : DOCTEUR DE L'UNIVERSITÉ DE RENNES 1
Mention INFORMATIQUE

par

Assia DJABELKHIR

Équipe d'accueil : CAPS
École Doctorale : MATISSE
Composante universitaire : IFSIC/IRISA

Titre de la thèse :

Etude de l'exécution dynamique et/ou spéculative et des processeurs enfouis : un cas d'étude de l'architecture découplée

soutenue le 31 Mars 2005 devant la commission d'examen

MM. :	Nathalie	DRACH	Rapporteurs
	Daniel	LITAIZE	
MM. :	Thierry	COLLETTE	Examinateurs
	François	BODIN	
	André	SEZNEC	

Table des matières

Résumé

Dans cette thèse, nous proposons d'utiliser l'architecture découplée pour la conception des processeurs embarqués de haute performance. Cette proposition est motivée par la complexité des nouvelles applications enfouies grand public. L'exécution découplée constitue une solution intermédiaire entre l'exécution dans l'ordre, simple mais dont les performances restent limitées, et l'exécution dans le désordre, qui permet de bonnes performances, mais qui s'avère complexe et coûteuse.

L'architecture découplée atteint de hautes performances si ses sous-processeurs sont complètement découplés en exécutant les flots d'instructions du code. L'apparition de certaines dépendances entre les sous-processeurs cause une perte de découplage, qu'on appelle un évènement LOD (*Loss Of Decoupling*). Les évènements LOD constituent la cause principale de perte de performance sur une architecture découplée. Par conséquent, il est important que le code s'exécutant sur ce type d'architecture soit "découplable", tel est le cas des applications régulières.

Dans cette thèse, nous avons étudié les caractéristiques d'un ensemble de benchmarks enfouis afin de comprendre et prédire leur comportement sur une architecture découplée, et de quantifier le risque d'occurrence des évènements LOD. Cette étude a permis de classifier les applications enfouies en catégories suivant leur régularité et a montré que la quasi-totalité de ces applications peuvent être amenées à de bonnes performances sur une architecture découplée. Pour d'autres applications, nous proposons une politique de placement des données mémoire "critiques" qui a pour but de réduire l'impact, parfois dramatique, des irrégularités causées par l'utilisation de ces données. Nous avons présenté un cas d'étude de la politique de placement des données utilisées comme adresses pour les accès indirects, afin de réduire la latence des dépendances mémoire-mémoire.

Nous avons montré qu'une architecture découplée EDA, constituée de trois sous-processeurs découplés, peut être implémentée pour un jeu d'instructions classique à un fichier de registres unique. Nous avons présenté un mécanisme de partitionnement dynamique du code sur les trois sous-processeurs, qui se base sur le partitionnement dynamique du fichier de registres en sous-ensembles de registres spécifiques pour chaque type de calcul : accès, contrôle, ou traitement. Nous avons présenté une mise en œuvre de cette architecture et décrit les scénarios de communication entre les sous-processeurs découplés ainsi que la validation dans l'ordre des instructions. Ces mécanismes ont été mis en œuvre par DS-sim, un simulateur de l'architecture découplée EDA.

Mots-clés : processeurs embarqués, architecture découplée, benchmarks enfouis, régularité du code, caractéristiques des applications, loss of decoupling (évènements LOD), caches mémoire, simulation de la micorarchitecture, simulateur DS-sim.