République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
***
Université M'hamed Bougara Boumerdès
Département Informatique, Faculté des Sciences
***

Thèse

Pour l'obtention du diplôme

Magistère en Informatique

Option **: Informatique fondamentale**

Soutenu Par : Mr. Mohamed GOUDJIL

Thème

Amélioration de la performance des processeurs généralistes par la réduction du nombre d'instructions exécutées

Jury :

Pr. Mohamed MEZGHICHE	Université de Boumerdès (Algérie)	Président
Pr. Bernard GOOSSENS	Université de Perpignan (France)	Promoteur
MC. Mouloud KOUDIL	INI (Algérie)	Examinateur
MC. David PARELLO	Université de Perpignan (France)	Examinateur

Année universitaire 2006-2007

اَلْمُلَخَّصْ

إن معادلة فعالية المعالج تظهر ثلاثة عوامل أساسية، أولها هو الدور الذي يتعلق بالتكنولوجيا المستخدمة، وثانيها هو الـ(IPC) "عدد الأوامر المطبقة خلال دور واحد" والذي يتعلق بالهندسة الدقيقة للمعالج، وأخيرا عدد الأوامر المطبقة إجمالا والذي يتعلق بالتصميم العام للمعالج. فيما عدا التطورات الحاصلة في ميدان التكنولوجيا، فإن التحسينات في فعالية المعالجات العامة خلال السنوات الأخيرة كانت ترتكز بالأساس على الهندسة الدقيقة للمعالج, ومنذ ظهور المعالجات ذات التصميم (RISC) فإن مجموعة الأوامر الداخلية للمعالجات ما فتأت تدور حول الأوامر البسيطة.

وكنتيجة حتمية لما سبق فإن بلوغ مستوى أعلى من الـ(IPC) يتطلب تطبيق عدد أكبر من الأوامر الأساسية خلال دور واحد, ومن بين التوجهات المطروحة في هذا المجال هي محاولة خفض عدد الأوامر المطبقة إجمالا عن طريق إستغلال التكرار الملاحظ في الأوامر الخاصة بالولوج إلى الذاكرة, إن مما يجدر الإشارة إليه هو أن مجموعة الأوامر في المعالجات ذات التصميم (RISC) تتميز بكون أوامر التحميل والتخزين من وإلى الذاكرة هي أوامر مركبة من شقين أساسيين يتعلق أحدهما بالآخر، وهما عملية الوصول إلى المعلومة(في الذاكرة) المتعلقة بعملية حساب عنوان هذه المعلومة، مما يعني أن خفض عدد الأوامر المنفذة من هذا النوع يحقق مكسبا معتبرا.

في هذا العمل الذي بين أيدينا حاولنا تقديم توجه جديد يهدف إلى تحسين فعالية المعالجات عن طريق خفض عدد الأوامر اللازم تنفيذها. وبهذا الصدد إقترحنا تصميما جديدا لحاوي السجلات (banc de registres). وقد أثبتت قياساتنا أن التصميم الجديد يسمح بخفض عدد أوامر الإدخال إلى الذاكرة بنسبة15% ، وعدد أوامر الإخراج من الذاكرة بنسبة10%.

الكلمات المفتاحية:
الهندسة الدقيقة للمعالجات، المعالجات العامة، فعالية المعالجات، أوامر التحميل من الذاكرة، أوامر التخزين في الذاكرة، محاكات المعالجات.

Abstract

The performance equation of a processor reveals three terms: the cycle which depends on the technology, the IPC (Instructions Per Cycle) which depends on the microarchitecture and finally, the number of instructions executed which depends on the architecture. Apart from the technological advances, the improvements made to the performance of the general purposes processors these last years especially related to the microarchitecture. Since the RISC revolution, the internal instruction set of the processors is fixed around simple instructions.

A consequence is that to reach a high degree of the IPC, it is necessary to issue multiple elementary instructions in the same cycle. An alternative is to reduce the number of instruction to execute while taking advantage from the redundancy which exists in the Load and Store instructions. It is to be noticed that in a RISC instruction set, the instructions of Load and Store are complex instructions combining the effects of two elementary instructions. The access itself depends on the effective address calculation. What it means that reducing the number of such type of instructions has a significant contribution in the processor performance.

In this work we presented a new approach to improve the performance by reducing the number of executed instructions. We proposed a new architecture for the register file. Our measurements show that new architecture would make it possible to reduce by 15% the rate of Load instructions, and 10% that of Store instructions.

Key words: microarchitecture, general purpose processors, processor performances, superscalar, RISC, Load, Store, processor simulation, SimpleScalar.

Résumé

L'équation de performance d'un processeur fait apparaître trois termes : le cycle qui dépend de la technologie, l'IPC (Instructions Par Cycle) qui dépend de la microarchitecture et enfin, le nombre d'instructions exécutées qui dépend de l'architecture. En dehors des avancées technologiques, les améliorations apportées à la performance des processeurs généralistes ces dernières années ont surtout concernées la microarchitecture. Depuis la révolution RISC, le jeu d'instructions interne des processeurs s'est figé autour d'instructions simples.

Une conséquence est que pour atteindre un degré élevé de l'IPC, il faut exécuter de nombreuses instructions élémentaires dans le même cycle. Une alternative est de réduire le nombre d'instruction à exécutée en profitant de la redondance qui existe dans les instruction d'accès mémoire. Il est à remarquer que dans un jeu d'instructions RISC, les instructions de chargement et de rangement sont des instructions complexes combinant les effets de deux instructions élémentaires. L'accès proprement dit dépend du calcul d'adresse. Ce qu'il fait que la réduction du nombre de tel type d'instruction apport un gain important.

Dans ce travail nous avons présenté une nouvelle approche pour améliorer la performance en réduisant le nombre d'instructions à exécuter. Nous avons proposé une nouvelle architecture pour le banc de registres. Nos mesures montrent que la nouvelle architecture permettrait de réduire de 15% le taux d'instructions de chargement, et de 10% celui des instructions de rangements.

Mots clés : microarchitecture, processeur généraliste, performances de processeur, superscalair, RISC, chargement, rangement, simulation de processeur, SimpleScalar.

Table des matières

Résumé……I
Introduction…..……..II

I - Architecture et microarchitecture……..……...6
1. Architecture logicielle……..…….....6
1.1 jeu d'instructions……...7
1.2 La classification des jeux d'instructions (modèle d'exécution)…….7
1.3 Format des instructions……..10
1.4 Instructions arithmétiques et logiques……..10
1.5 Instructions d'accès mémoire……11
1.5.1 Accès entiers……11
1.5.2 Accès flottants……...12
1.5.3 L'adresse effective :……..12
1.6 Instruction de comparaison……13
1.7 Instructions de saut et de branchement……13

2. La microarchitecture……..14
2.1 L'unité centrale……...14
2.2 Principe du pipeline……15
2.3 Le pipeline classique pour l'exécution des instructions……..16
2.3.1 Tâche élémentaire……16
2.3.2 Pipeline pour les instructions UAL……17
2.3.3 Pipeline et accès mémoire……..…..18
2.4 Les contraintes du pipeline…….....18
2.4.1 Ressources matérielles……...…….…….19
Un opérateur par opération……....…….19
Accès multiples au banc de registres……...19
Cache instructions et données séparés…….....19
2.4.2 Format d'instruction fixe et simple - Cache d'instruction..20
2.4.3 Architecture chargement-rangement…….....20
2.5 Les limitations du pipeline……...…….....20
2.5.1 Aléas de données…….....….21
Instructions UAL……...……..…....23
Autres instructions……...…….…....24
2.6 Aléas de contrôle……...……..25
2.6.1 Les différents types d'aléas de contrôle…….....25
Décodage des sauts et branchements…….....26
Calcul de l'adresse cible……...……..…...26
Calcul des conditions……...……..……26

2.6.2 Traitement des aléas de contrôle…….....27
Annulation d'instruction……..…..……..27
Les sauts et branchements retardés…….....27
2.7 Instructions flottantes……..…….....…….28

3. caches…….……..29
localité temporelle…….....30
localité spatiale…….…….30
3.1 Implémentation du cache……...30
3.2 TLB…….....31
3.3 Hiérarchie de caches……..31

II- L'Architecture Superscalair……..…….....34
Introduction ……..……...……...…….....34
1. L'architecture superscalair : ……...…….....38
1.1 Les limites fondamentales de l'architecture superscalair : ……....39
1.1.1 les Aléas de données : ……...……..….39
1.1.2 les Aléas de contrôle…….....40
Caches d'adresses de branchement…….....40
Tampon de prédiction de branchement…….....41
Instructions prédiquées……...……...…42
1.1.3 les Aléas de ressources……...……43
1.2 Le niveau de parallélisme d'instruction
et le parallélisme de machine……43
1.3 Lancement d'instruction et parallélisme de machine ……....44
1.3.1 Lancement dans l'ordre avec
complétion dans l'ordre …….....45
1.3.2 Lancement dans l'ordre avec
complétion dans le désordre…….……46
1.3.3 Lancement dans le désordre avec
complétion dans le désordre…….46
1.4 Dépendances de stockage et
renommage de registres ……...…47
1.5 L'exécution Superscalair …….……...……..…....48

III- L'environnement expérimentale…..……..…….....52
1. Introduction……...……...……..….52

2. Simplescalar……...……...……..…….52

3. Le jeu d'instructions émulée par SimpleScalar……...…….....53

4. Les plateformes qui supportent le SimpleScalar…….....53

5. Le fonctionnement des différents simulateurs…….....54
5.1 Les simulateurs de base : *sim-fast* et *sim*-safe ……....54

5.2 Simulation de cache : *sim-cache* et *sim-cheetah*.....54
5.3 Génération de statistiques : *sim*-profile.....55
5.4 L'exécution des instructions dans le désordre : *sim*-outorder.....56
5.4.1 Le pipeline du Simplescalar.....56

6 L'architecture du Simplescalar.....57
6.1 Réordonnancement dynamique du code par *RUU*.....57
6.2 Gestion de la mémoire.....60
6.3 Le cache dans SimpleScalar.....60

7. Les Benchmarks.....61
7.1 Les Benchmarks de la SPEC Cpu 61
7.2 La suite des benchmarks MediaBench.....61
7.3 **La suite des benchmarks MiBench.....61**

IV- Notre Contribution.....64
1. Introduction 64

2. Les caractéristiques des accès mémoire.....65
2.1 L'anatomie d'un chargement.....65
2.2 L'impacte de la latence du chargement.....66
2.3 L'anatomie d'un rangement.....67

3. Les types de dépendance mémoire.....68

4. Les travaux réalisés.....70
4.1 La prédiction des dépendances.....70
4.2 La prédiction des adresses.....71
4.3 Prédiction des valeurs 72
4.4 Renommage de mémoire.....72

5. notre contribution.....73
5.1 La classification des instructions d'accès mémoire: 73
5.2 La solution proposée.....75
5.3 L'anatomie du banc de registre: 75
5.4 L'architecture proposée.....77
5.5 La partie expérimentale.....78
L'environnement expérimental.....78
5.5.1 Caractéristiques des benchmarks.....79
5.5.2 Les différents types d'accès mémoire.....80
5.5.3 Les caractéristiques de l'architecture proposée.....82
Chargement.....82
Rangement.....83
Les résultats globaux 85

6. Conclusion 86

Conclusion et perspectives 87