Quelques questions

Bruno Malenfant

4 décembre 2003
dernière modification: 4 février 2004

1 Introduction.

1. Qu'elles sont les caractéristiques d'une famille de processeur?
2. Comment peut-on comparer deux architectures différentes?

2 Performance.

1. Il est possible d'effectuer un profil d'un programme afin d'obtenir le nnombre d'instructions exécutés par celui-ci. Un profil plus détaillé va nous indiquer la composition du programme. Par exemple, pour un programme P_1 nous pouvons obtenir le profil suivant:

Instruction	nombre de cycle	nombre d'instruction
Load	5	100 000
Store	4	80 000
Add	4	260 000
Beq	3	160 000
Jump	3	50 000

Donc lors de l'exécution de ce programme, 650 000 lignes de codes sont exécutés. La collone *nombre de cycle* indique combien de cycle d'horloge sont necéssaire pour eécuter cette instruction.

   (a) Si nous avons un ordinateur qui utilise une vitesse d'horloge de 1 200 MHz, en combien de temps le programme P_1 sera-t'il exécuté?

   (b) Quel est la performance MIPS dans ce cas?

   Nous compilons le même programme pour un deuxème ordinateur et nous obtenons le profil suivant:

Instruction	nombre de cycle	nombre d'instruction
Load	5	90 000
Store	4	70 000
Add	4	280 000
Beq	3	140 000
Jump	3	40 000

(c) Quel est le temps d'exécution pour ce programme si nous avons une vitesse d'horloge de 1 200 MHz?

(d) Quel est la performance MIPS?

(e) Comparez les performances réelles (temps d'exécution) en divisant le temps du plus lent par le temps du plus vite.

(f) Comparez les performances MIPS en divisant la valeur la plus grande par la valeur la plus petite.

3 Jeux d'instructions.

1. Nous voulons construire un jeu d'instructions très limité. Notre machine comprendra 8 registres de 16 bits (R0 à R7) et un registre PC de 16 bits aussi. Les mots mémoires ont 8 bits (1 octet).

   (a) Nous voulons utiliser une architecture de type rangement/chargement. Construisez une liste minimale des instructions de transfert qui seront nécessaire.

   (b) Quels sont les modes d'adressages qui devront être disponible pour ces opérations ?

   (c) Nous voulons un jeux d'instructions arithmétiques minimal qui permet d'exécuter des additions, soustractions, multiplications et divisions. Construisez un tel jeux avec les modes d'adressages permit pour chaque opération. N'oubliez pas qu'une multiplication de 16 bits par 16 bits donne un résultat de 32 bits. Une division donne un reste et un quotient.

   (d) Il faut aussi construire des opérations de contrôle. Ces opérations doivent permettent des branchements conditionnels.

      i. Nous voulons utiliser des registres de conditions avec des opérations de comparaisons. Énumérez les différents code de conditions qui vous serons nécessaire.

      ii. Pour chaque instructions que vous avez construite, dites quel son les codes de conditions qui seront modifiés et comment.

      iii. Construisez les différentes opérations de comparaisons avec leurs mode d'adressage et leurs effet sur les codes de conditions.

iv. Construisez des opérations de branchement conditionnel en donnant la condition de branchement en fonction des bits de contrôle.

(e) Construisez des opérations de contrôles inconditionneles. Ces opérations doivent comprendre un mécanisme d'appel de procédure. Pour ce faire vous allez devoir définir une pile système.

(f) Maintenant que nous avons un jeux d'instructions, il faut construire sont encodage. Pour chaque instruction vous devez définir son encodage.

- Soyez constant: placer les codes opérations aux même endroit d'une instruction à l'autre.
- Vérifiez qu'il n'y ait pas de doute sur l'identité d'une instruction ou de ses opérandes.

(g) Essayez votre jeux d'instructions en écrivant une routine de Fibonacci et un programme principal qui contient un appel à cette fonction.

(h) Quel type d'instruction manque-t'il à notre jeux d'instruction ?

2. Lors de la construction d'un jeux d'instruction il peut être préférable de rendre la tâche des concepteurs et des programmeurs le plus simple possible. Quels facteurs pourraient nuire à cette tâche ?

4 Pipeline.

1. Une architecture pipeliné permet d'exécuter plusieurs instructions simultanément. Quelles en sont les désavantages ?

2. Pour obtenir des meilleurs performances il est possible d'utiliser plus de 1 UAL pour résoudre les opérations. Comment est-il possible de placer plus de 1 UAL sur le pipeline pour gagner de l'efficacité.

3. Pourquoi un jeux d'instructions de type rangement/chargement est-il plus efficace qu'un jeux d'instruction mémoire à mémoire sur un pipeline ?

4. Dans le cas du pipeline à 6 étages vu en classe (Mi, Con, U1, Md1, U2, Md2). Supposons que nous ayons un programme qui contient n instructions. 15% de ces instructions sont des branchements. Nous utilisons une technique de branchement dynamique à deux bits qui a un taux de succès de 90%. Quel sera la pénalité causée par les branchements ? (en nombre de NOP ajouté par le processeur.)

5 Architecture Risc.

1. Pourquoi l'utilisation d'un grand nombre de registres aide t'elle au performance d'un processeur ?

2. Une architecture RISC permet d'optimiser un mécanisqme interne des processeurs, lequel ? En quoi les architectures RISC permettent-t'elle d'optimizer ces architectures ?

3. Les architectures RISC contiennent souvent un banc de registre virtuel, comment les paramêtres d'une procédure sont-ils transmit pour une tel architecture ?

4. En quoi un grand nombre de registre est-il plus performant qu'une mémoire cache ?

5. Pourquoi un pipeline RISC est-il plus performent qu'un pipeline CISC ?

6 Superscalaire.

1. Quel sont les principes de base d'une architecture super-scalaire ?

2. À quels étapes de l'exécution peut-il y avoir un changement dans l'ordre des instructions ?

3. Quel unité doit-on ajouter au processeur pour pourvoir utiliser une politique de changement d'ordre des instructions avant leurs exécutions ?

4. Quel est l'utilité de renommer les registres lors de l'exécution d'un programme sur une architecture super-scalaire ?

5. Les mises en oeuvre super-scalaire utilisent plusieurs pipelines chacuns spécialisés pour l'exécution d'un certains types d'instructions. Pourquoi n'est-il pas préférable d'utiliser des pipelines généraux ?

6. À quel moment le processeur peut-il renommer les registres ?

7. Comment peut-on décider du nombre et de la spécialitée de chaque pipeline interne d'un super-scalaire ?

8. Quels sont les limites d'une architecture super-scalaire ?

7 IA-64.

1. Quel est le principe de base des architecture IA-64 ?

2. Á quel moment ce fait le réordonnancement des instructions ?

3. Quel est l'avantage d'une architecture IA-64 ?

4. Quel est l'impact des règles d'encodages d'instruction utilisèes par l'architecture IA-64 ?

5. Le systême d'évaluation par prédicat demande l'utilisation du double des resources de la machine par rapport à une évaluation plus conventionnelle. En quoi cette évaluation est-elle plus performante ?

6. Comment fonctionne le chargement spéculatif ?

7. L'architecture IA-64 permet d'effectuer une rotation virtuelle des registres, quel mécanisme cela émule-t-il ?

8 Unité de contrôle.

1. Quel sont les composantes d'un RLP ?

2. Quel sont les entrées et les sorties d'un contrôleur ?

3. Quel sont les avantages d'un contrôleur contenu dans une mémoire ROM plutôt qu'un contrôleur sous la forme d'un circuit imprimé ? Quel sont les désavantages ?