

# Questions des examens antérieurs

Bruno Malenfant

22 février 2006

1. L'architecture MIPS comprend trois(3) modes d'adressage : immédiat, registre et relatif. Un mode très utilisé pour l'encodage des tableaux est le modes indexé. Comment est-il possible de réaliser ce mode (indexé) sur l'architecture MIPS? Donnez un exemple pour illustrer votre explication.
2. Le jeu d'instructions MIPS comprend un encodage sur un nombre constant de bits (32 bits par instruction). D'autres architectures utilisent un encodage sur un nombre variable de bits. Quelles sont les avantages et désavantages d'un encodage sur un nombre variable de bits?
3. Pour chaque instruction (style VAX) présentée plus bas, donnez une séquence d'instructions machines MIPS ayant l'effet indiqué. Si vous avez besoin de registres temporaires, utilisez \$0, \$1, \$2 et \$3. Sauf pour ces registres, ne modifiez un registre que si cela est absolument requis pour réaliser l'effet de l'instruction.
  - (a) `movl ($7)[$8], -($6)`    #  $\$6 = \$6 - 4$ ;  $\text{Mem}[\$7 + \$8 * 4] = \text{Mem}[\$6]$
  - (b) `clrl @($4)+`    #  $\text{Mem}[\text{Mem}[\$4]] = 0$ ;  $\$4 = \$4 + 4$
  - (c) `addl3 $6, 112($5), @4($9)`    #  $\$6 = \text{Mem}[112 + \$5] + \text{Mem}[\text{Mem}[4 + \$9]]$
4. Si vous possédez les données sur les performances MIPS d'une architecture alors vous pouvez calculer le temps d'exécution d'un programme pour cette architecture.

VRAI

FAUX

5. Une première organisation d'une architecture nous donne les CPI suivants:

Classe	CPI
Arithmétique	4
Chargement	5
Rangement	4
Branchement	3
Saut	3

Cette machine utilise une horloge de 2.0 GHz. Un premier compilateur à été écrit et nous donne un programme de 100 000 instructions pour un logiciel L. Ces instructions sont distribuées de la façon suivante:

Classe	Fréquence
Arithmétique	40 %
Chargement	20 %
Rangement	15 %
Branchement	15 %
Saut	10 %

- (a) Quel est le CPI de ce logiciel sur cette organisation?  
 (b) L'équipe de développement prédit pouvoir obtenir une augmentation de la vitesse d'horloge de 40% avec les nouveaux CPI suivant:

Classe	CPI
Arithmétique	5
Chargement	6
Rangement	5
Branchement	4
Saut	3

Quel sera le nouveau CPI pour la machine améliorée?

- (c) L'équipe propose aussi une amélioration du compilateur qui permettrait de diminuer le nombre d'instruction du logiciel L de 20 000 instructions. Cette amélioration donnerait les nouvelles distributions suivantes:

Classe	Fréquence
Arithmétique	45 %
Chargement	15 %
Rangement	15 %
Branchement	10 %
Saut	15 %

Les améliorations du matériel prendraient 3 mois et celles du logiciel en prendraient 4. Il est possible d'améliorer les deux en 6 mois. Sachant que les performances sur le marché s'améliore de 5 % par mois, lesquels des deux améliorations devraient être utilisées pour rester compétitif?

6. Soit le code MIPS suivant:

```

    add $1, $2, $3
suite: lw  $4, 392($5)
    slt $7, $1, $4
  
```

```

        beq $7, $0, fin
        j   suite
fin:    sub $3, $3, $4

```

Sachant que l'étiquette "suite" représente l'adresse 1000, complétez l'encodage en remplissant les cases vides:

0	2	3	1	0	32
35	5	4			
0	1	4	7	0	42
4	7	0			
2					
0	3	4	3	0	34

7. Voici une table de vérité.

$a$	$b$	$e_i$	$s$	$e_f$
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	1
1	0	0	1	1
1	0	1	1	0
1	1	0	0	1
1	1	1	1	1

- En utilisant une table de Karnaugh, trouvez l'équation logique nécessaire pour calculer  $e_f$  par rapport à  $a, b$  et  $e_i$ ?
- Dessinez le circuit logique qui donne en sortie le bit  $s$  avec en entrée les bits  $a, b$  et  $e_i$ .

8. Soit une porte "nand" qui utilise la table de vérité suivante:

a	b	a nand b
0	0	1
0	1	1
1	0	1
1	1	0

- Dessinez le circuit de la porte **nand** à l'aide de porte de base (ou, et, non). (Il n'est pas obligatoire d'utiliser les trois.)
- Construisez la fonction **Non (a)** à l'aide de la nouvelle porte **nand** seulement. Ici vous ne pouvez utiliser les portes et, ou, non. Vous pouvez utiliser autant de porte **nand** que vous le voulez.

(c) Dessinez la fonction Ou (a, b) à l'aide de la nouvelle porte **nand** seulement. Vous ne pouvez utiliser les portes et, ou, non. Vous pouvez utiliser autant de porte **nand** que vous le voulez.

9. Calculer la valeur absolue d'un nombre consiste à appliquer la formule suivante:

```
abs(n) =  
  si n >= 0 alors retourner n  
  sinon retourner -1 * n
```

Dessinez le circuit logique qui reçoit une valeur  $n$  signée de 32 bits et retourne la valeur absolue de  $n$ . Vous pouvez utiliser tout les circuits vus en classe.

10. Soit en entrée une valeur  $n$  signée de 32 bits. Nous voulons faire le calcul suivant:

```
f(n) =  
  si n >= 16 alors retourner n - 16  
  sinon retourner n
```

Dessinez le circuit logique qui reçoit une valeur  $n$  signée de 32 bits et retourne la valeur  $f(n)$ . Vous pouvez utiliser tout les circuits vus en classe.

11. Multiplication.

(a) En utilisant un multiplicateur 16 bits avec l'algorithme de Booth, quel est la caractéristique des nombres qui causeront les pire temps d'exécution?

(b) Effectuez la multiplication suivante en utilisant l'algorithme de Booth:

1001 1010 \* 0111 0110

remarque: montrez et expliquez les étapes intermédiaires.