

**NOLLE**

**Damien**

**L3 – Informatique**

**ADO – Devoir 3 :**

<b><u>Note :</u></b>	<b><u>Observation :</u></b>
<b><u>/20</u></b>	

Exercice 1)

Q1)

On sait que la mémoire cache contient 8 blocs, 1 bloc équivaut à 32 bits.

Déplacement = puissance de la taille d'un bloc = 32 bits =  $32 / 8 = 4$  octets =  $2^2 = 2$  bits

- Fonction correspondance directe :

Index = Puissance du nombre de blocs en mémoire cache = 8 blocs =  $2^3 = 3$  bits

Nb blocs mémoire cache = TC / NOBP

$$8 = TC/4$$

$$8 * 4 = 32 \text{ octets}$$

$$32/4 = 8 = 2^3$$

Etiquette = taille adresse – taille déplacement – taille index

$$= 15 - 2 - 3 = 10 \text{ bits}$$

- Fonction correspondance associative par ensemble de 4 :

Index = puissance (de 2, pour tous) du nombre d'ensemble dans la mémoire cache = NEC = NBC / NBPE  
 $= 8 / 4 = 2$  ensembles dans la mémoire cache =  $2^1 = 1$  bits.

Etiquette = taille adresse – taille déplacement – taille index = 15 – 2 – 1 = 12 bits

Q2)

0x2000 = 0010 0000 0000 0000

0100000000 000 00

0001 0000 0000 = 0x100 (Etiquette)

000 = 0x0 (index)

0x2010 = 0010 0000 0001 0000

0100000000 100 00

100 = 4 = 0x4 (Index)

01 0000 0000 = 0x100 (Etiquette)

0x2020 = 0010 0000 0010 0000

0100000001 000 00

Index = 000 = 0 = 0x0

Etiquette = 01 0000 0001 = 0x101

Calcul du taux de succès :

1/6

$1 * 8 / 6 * 8 = 8 / 48$

Q3 :

0x2000 = 0010 0000 0000 0000

010000000000 0 00

Etiquette = 0100 0000 0000 = 0x400

Index = 0

0x2010 = 0010 0000 0001 0000

010000000010 0 00

Etiquette = 0100 0000 0010 = 0x402

Index = 0

0x2020 = 0010 0000 0010 0000

010000000100 0 00

Etiquette = 0100 0000 0100 = 0x404

Index = 0

Calcul du taux de succès :

3/6

$3 * \frac{8}{6} * 8 = 24/48$

Soit le programme assembleur suivant:

```
.data
tab1: .byte 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16
tab2: .byte 1,2,3,4,5,6,7,8,9,20,21,22,23,24,25,26
tab3: .space 32
.text
addiu $a0,$0,0x2000
addiu $a1,$0,16
jal test
addiu $v0,$0,10
syscall
test:
lbu $t0,0($a0)
lbu $t1,16($a0)
addu $t0,$t0,$t1
sb $t0,32($a0)
addiu $a0,$a0,1
addiu $a1,$a1,-1
bne $a1,$0,test
jr $ra
```

Pour évaluer les performance de ce programme, nous allons utiliser un cache contenant des blocs de 1 mot (32 bits - 4 octets), contenant 8 blocs.

Nous utiliserons une fonction de correspondance directe, une fonction de correspondance associative par ensemble de 4.

La taille des adresses est de 15 bits.

1. Donnez la taille des trois champs (nombre de bits)

Directe    Etiquette=  , Index=  , Déplacement=

Associative    ensemble de 4 Etiquette=  , Index=

2. Calculez le taux de succès pour la fonction de correspondance directe. Pour cela, complétez les valeurs suivantes, pour les deux premières itérations.

Adresse=0x2000, Etiquette=0x  , Index=  , Succès=

Adresse=0x2010, Etiquette=0x  , Index=  , Succès=

Adresse=0x2020, Etiquette=0x  , Index=  , Succès=

Adresse=0x2001, Etiquette et Index= idem 0x2000, Succès=

Adresse=0x2011, Etiquette et Index= idem 0x2010, Succès=

Adresse=0x2021, Etiquette et Index= idem 0x2020, Succès=

D'où le Taux de succès =  /48

3. Calculez le taux de succès pour la fonction de correspondance associative. Pour cela, complétez les valeurs suivantes, pour les deux premières itérations.

Adresse=0x2000, Etiquette=0x  , Ensemble=  , Succès=

Adresse=0x2010, Etiquette=0x  , Ensemble=  , Succès=

Adresse=0x2020, Etiquette=0x  , Ensemble=  , Succès=

Adresse=0x2001, Etiquette et Ensemble= idem 0x2000, Succès=

Adresse=0x2011, Etiquette et Ensemble= idem 0x2000, Succès=

Adresse=0x2021, Etiquette et Ensemble= idem 0x2000, Succès=

D'où le Taux de succès =  /48

## Exercice 2)

Q1 :

Instruction/cycle	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
lbu \$t0,0(\$a0)	LI	DI	EX	M	ER													
lbu \$t1,16(\$a0)		LI	DI	EX	M	ER												
addu \$t0,\$t0,\$t1			LI	DI	DI	DI	EX	M	ER									
sb \$t0,32(\$a0)				LI	LI	LI	DI	DI	DI	EX	M	ER						
addiu \$a0,\$a0,1							LI	LI	LI	DI	EX	M	ER					
addiu \$a1,\$a1,-1										LI	DI	EX	M	ER				
bne \$a1,\$0,test											LI	DI	DI	DI	EX	M	ER	
lbu \$t0,0(\$a0)												LI	LI	LI	DI	EX	M	ER

$$5 + (n - 1) \text{ (nombre d'étages - 1)} = 5 + (8 - 1) = 12$$

3<sup>ème</sup> instruction : dépendance \$t1, \$t0

4<sup>ème</sup> instruction : dépendance \$t0

5<sup>ème</sup> instruction : dépendance \$a1

6<sup>ème</sup> instruction : dépendance : \$a0, \$t0

On a donc une dépendance sur \$t1, \$t0, \$a0 et \$a1.

XXLDDDEM W

XXXLLLDDDEM W

XXXXXXLLLDEM W

XXXXXXXXXXLDEM W

XXXXXXXXXXLDDDEM W

XXXXXXXXXXLLLDEM W

1. En utilisant un pipeline sans envoi, donnez l'état du pipeline sur le modèle des deux premières lignes (en majuscule et sans espace)

lbu \$t0,0(\$a0)	LDEM W
lbu \$t1,16(\$a0)	XLDEM W
addu \$t0,\$t0,\$t1	XXLDDDEM W
sb \$t0,32(\$a0)	XXXLLLDDDEM W
addiu \$a0,\$a0,1	XXXXXXLLLDEM W
addiu \$a1,\$a1,-1	XXXXXXXXXXLDEM W
bne \$a1,\$0,test	XXXXXXXXXXLDDDEM W
lbu \$t0,0(\$a0)	XXXXXXXXXXLLLDEM W

En déduire le nombre de cycle par itération si le branchement est pris

14

Q2 :

Instruction/cycle	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
lbu \$t0,0(\$a0)	LI	DI	EX	M	ER													
lbu \$t1,16(\$a0)		LI	DI	EX	M	ER												
addu \$t0,\$t0,\$t1			LI	DI	DI	DI	EX	M	ER									
sb \$t0,32(\$a0)				LI	LI	LI	DI	DI	DI	EX	M	ER						
addiu \$a0,\$a0,1							LI	LI	LI	DI	EX	M	ER					
addiu \$a1,\$a1,-1										LI	DI	EX	M	ER				
bne \$a1,\$0,test											LI	DI	DI	DI	EX	M	ER	
lbu \$t0,0(\$a0)												LI	LI	LI	DI	EX	M	ER

Instruction/cycle	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
lbu \$t0,0(\$a0)	LI	DI	EX	M	ER													
lbu \$t1,16(\$a0)		LI	DI	EX	M	ER												
addu \$t0,\$t0,\$t1			LI	DI	EX	EX	M	ER										
sb \$t0,32(\$a0)				LI	DI	DI	EX	M	ER									
addiu \$a0,\$a0,1					LI	LI	DI	EX	M	ER								
addiu \$a1,\$a1,-1							LI	DI	EX	M	ER							
bne \$a1,\$0,test								LI	DI	EX	M	ER						
lbu \$t0,0(\$a0)									LI	DI	EX	M	ER					

XXLDEEMW

XXXLDDEMw

XXXXLLDEMw

XXXXXXLDEMw

XXXXXXLDEMw

XXXXXXLDEMw

Instruction/cycle	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
lbu \$t0,0(\$a0)	LI	DI	EX	M	ER													
lbu \$t1,16(\$a0)		LI	DI	EX	M	ER												
addu \$t0,\$t0,\$t1			LI	DI	EX	EX	M	ER										
sb \$t0,32(\$a0)				LI	DI	DI	EX	M	ER									
addiu \$a0,\$a0,1					LI	LI	DI	EX	M	ER								
addiu \$a1,\$a1,-1							LI	DI	EX	M	ER							
bne \$a1,\$0,test								LI	DI	EX	M	ER						
lbu \$t0,0(\$a0)									LI	DI	EX	M	ER					

2. En utilisant un pipeline avec l'envoi et le branchement au plus tôt, donnez l'état du pipeline sur le modèle des deux premières lignes.

lbu \$t0,0(\$a0)	LDEMW
lbu \$t1,16(\$a0)	XLDEMW
addu \$t0,\$t0,\$t1	XXLDEEMW
sb \$t0,32(\$a0)	XXXLDDEMW
addiu \$a0,\$a0,1	XXXXLLDEMW
addiu \$a1,\$a1,-1	XXXXXXLDEMW
bne \$a1,\$0,test	XXXXXXXXLDEMW
lbu \$t0,0(\$a0)	XXXXXXXXLDEMW

En déduire le nombre de cycle par itération si le branchement est pris

11

Q3 :

Instruction/cycle	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
lbu \$t0,0(\$a0)	LI	DI	EX	M	ER													
lbu \$t1,16(\$a0)		LI	DI	EX	M	ER												
addu \$t0,\$t0,\$t1			LI	DI	EX	EX	M	ER										
addiu \$a1,\$a1,-1				LI	DI	EX	M	ER										
sb \$t0,32(\$a0)					LI	DI	EX	M	ER									
addiu \$a0,\$a0,1						LI	DI	EX	M	ER								
bne \$a1,\$0,test							LI	DI	EX	M	ER							
lbu \$t0,0(\$a0)								LI	DI	EX	M	ER						

3. Réorganiser le code en utilisant le branchement retardé d'un cycle

lbu \$t0,0(\$a0)

lbu \$t1,16(\$a0) ⇅

addu \$t0,\$t0,\$t1 ⇅

addiu \$a1,\$a1,-1 ⇅

sb \$t0,32(\$a0) ⇅

addiu \$a0,\$a0,1 ⇅

bne \$a1,\$0,test ⇅