



**POLYTECHNIQUE  
MONTRÉAL**

# INF1600

## Architecture des micro-ordinateurs

Groupe 1

Laboratoire 5

**Soumis par:**

Joe Abdo - 1 939 689

Jean-Sébastien Patenaude – 1 961 302

Le 9 avril 2019

**EXERCICE 1 :**

Voir les fichiers.

**EXERCICE 2 :**

1)

Mémoire principale de 1024Ko =  $2^{20}$  octets → adressage sur 20 bits

Mémoire cache de 16Ko =  $2^{14}$  octets

Lignes de 16 octets =  $2^4$  octets

Write-back, write-allocate et remplacement LRU

a) Direct :

- Il y a 1024 ( $2^{10}$ ) blocs de mémoire cache ( $2^{14}$  octets /  $2^4$  octets = 1024), donc 10 bits réservés pour l'ensemble.

- Il y a 16 octets ( $2^4$ ) par bloc (ligne), donc 4 bits réservés pour l'octet.

- Il y a  $2^6$  étiquettes (20 bits d'adressage – 10 bits ensemble – 4 bits octet = 6 bits), donc 6 bits réservés pour le tag.

Donc l'adresse est divisée ainsi :

tag	ensemble	octet
6 bits	10 bits	4 bits

b) Associative par ensemble de 2 blocs :

- Il y a 512 ( $2^9$ ) blocs de mémoire cache par ensemble, car ( $2^{14}$  octets /  $2^4$  octets =  $2^{10}$  blocs) et ( $2^{10}$  blocs / 2 ensembles =  $2^9$  blocs), donc 9 bits réservés pour l'ensemble.

- Il y a 16 octets ( $2^4$ ) par bloc (ligne), donc 4 bits réservés pour l'octet.

- Il y a  $2^7$  étiquettes, car (20 bits d'adressage – 9 bits ensemble – 4 bits octet = 7 bits), donc 7 bits réservés pour le tag.

Donc l'adresse est divisée ainsi :

tag	ensemble	octet
7 bits	9 bits	4 bits

c) Associative par ensemble de 4 blocs :

- Il y a 256 ( $2^8$ ) blocs de mémoire cache par ensemble, car ( $2^{14}$  octets /  $2^4$  octets =  $2^{10}$  blocs) et ( $2^{10}$  blocs / 4 ensembles =  $2^8$  blocs), donc 8 bits réservés pour l'ensemble.

- Il y a 16 octets ( $2^4$ ) par bloc (ligne), donc 4 bits réservés pour l'octet.

- Il y a  $2^8$  étiquettes, car (20 bits d'adressage – 8 bits ensemble – 4 bits octet = 8 bits), donc 8 bits réservés pour le tag.

Donc l'adresse est divisée ainsi :

tag	ensemble	octet
8 bits	8 bits	4 bits

2)

	Direct				2 blocs				4 blocs			
Accès	Tag	Set	Hit	w-b	Tag	Set	Hit	w-b	Tag	Set	Hit	w-b
WR 0x5EF1D	17	2F1			2F	0F1			5E	F1		
WR 0x19C7C	06	1C7			0C	1C7			19	C7		
RD 0x5EF1B	17	2F1	x		2F	0F1	x		5E	F1	x	
RD 0x8CDB0	23	0DB			46	0DB			8C	DB		
WR 0x3CDB3	0F	0DB		x	1E	0DB			3C	DB		
WR 0x5EF15	17	2F1	x		2F	0F1	x		5E	F1	x	
RD 0x68DBF	1A	0DB		x	34	0DB		x	68	DB		
WR 0xCAFF1C	32	2F1		x	65	0F1			CA	F1		
RD 0x39C7E	0E	1C7		x	1C	1C7			39	C7		
WR 0xCAFF1A	32	2F1	x		65	0F1	x		CA	F1	x	

0x5EF1D : 0101 1110 1111 0001 1101 donc :

direct : tag      ensemble    octet

01 0111 | 10 1111 0001 | 1101

2 blocs: tag      ensemble    octet

010 1111 | 0 1111 0001 | 1101

4 blocs: tag      ensemble    octet

0101 1110 | 1111 0001 | 1101

0x19C7C : 0001 1001 1100 0111 1100

direct : tag      ensemble    octet

00 0110 | 01 1100 0111 | 1100

2 blocs: tag      ensemble    octet

000 1100 | 1 1100 0111 | 1100

4 blocs: tag      ensemble    octet

0001 1001 | 1100 0111 | 1100

Les autres adresses suivent la même logique.

3)

direct :

Set	Tag0
2F1	32*
1C7	0E
0DB	1A

2 blocs :

Set	Tag0	Tag1
0F1	2F*	65*
1C7	0C*	1C*
0DB	1E*	34

4 blocs :

Set	Tag0	Tag1	Tag2	Tag3
F1	5E*	CA*		
C7	19*	39*		
DB	8C*	3C*	68*	

4)

Formule : temps d'accès effectif =  $h \cdot t_p + (1-h) \cdot t_s$ 

avec h : taux de succès d'accès (succès/total)

temps accès effectif =  $(\text{succès} \cdot 8\text{ns} + \text{défauts} \cdot 100\text{ns} + \text{accès mémoire principale} \cdot 100\text{ns}) / \text{total}$ 

direct :

$$3 \cdot 8\text{ns} + 7 \cdot 100\text{ns} + 4 \cdot 100\text{ns} = 1124\text{ns}$$

$$\rightarrow \text{temps accès effectif} = 1124\text{ns} / 10\text{accès} = 112,4\text{ns en moyenne}$$

2 blocs :

$$3 \cdot 8\text{ns} + 7 \cdot 100\text{ns} + 1 \cdot 100\text{ns} = 824\text{ns}$$

$$\rightarrow \text{temps accès effectif} = 824\text{ns} / 10\text{accès} = 82,4\text{ns en moyenne}$$

4 blocs :

$$3 \cdot 8\text{ns} + 7 \cdot 100\text{ns} = 724\text{ns}$$

$$\rightarrow \text{temps accès effectif} = 724\text{ns} / 10\text{accès} = 72,4\text{ns en moyenne}$$

5) Si la politique de placement de la cache avait été complètement associative, alors la structure de l'adresse n'aurait comporté que deux sections : une pour le tag et l'autre pour l'octet. Ainsi, la section pour l'ensemble aurait disparue. Il y aurait encore eu 4 bits réservés pour l'octet, car il y aurait encore des lignes de 16 ( $2^4$ ) octets en cache. Par contre, il y aurait eu 16 bits réservés pour le tag, car la mémoire principale comprend  $2^{20}$  octets et qu'il n'y a pas d'ensembles (20 bits d'adressage – 4 bits octet = 16 bits tag).