

TP5

Memoire Cache

Julien Legault, 1847125
Billy Bouchard, 1850477
B1

Un Travail présenté à :
Abdellatif Amrani



Département Génie informatique et Logiciel
Polytechnique Montreal
le 4 décembre

1 Memoire Cache

- pour le direct : 6 bit pour le tag | 10 bit pour l'ensemble | 4 bits pour l'octet
pour le 2 bloc : 7 bit pour le tag | 9 bit pour l'ensemble | 4 bits pour l'octet
pour le 4 bloc : 8 bit pour le tag | 8 bit pour l'ensemble | 4 bits pour l'octet

	Direct				2 blocs				4 blocs			
Accès	Tag(6)	Set(10)	Hit	W-B	Tag(7)	Set(9)	Hit	W-B	Tag (8)	Set(8)	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			4E	0DB			8C	DB		
WR 0x3CDB3	0F	0DB			1E	0DB			3C	DB		
WR 0x5EF15	17	2F1	x		2F	0F1	x		5E	F1	x	
RD 0x68DBF	1A	0DB		x	34	0DB			68	DB		
WR 0xCA1C	32	2F1		x	65	0F1			CA	F1		
RD 0x39C7E	0E	1C7		x	1C	1C7			39	C7		
RD 0xCA1A	32	2F1	x		65	0F1	x		CA	F1	x	

3.

Mem	17 2F1 D	06 1C7 C	17 2F1 B	23 0DB 0	0F 0DB 3	17 2F1 5	1A 0DB F	32 2F1 C	0E 1C7 E	32 2F1 A	Tags fin
2f1	17	W	R	R	W	W	R	W	R	R	32
1c7	6				0E						
0db	23				0F				1A		

Mem	2F 0F1 D	0C 1C7 C	2F 0F1 B	4E 0DB 0	1E 0DB 3	2F 0F1 5	34 0DB F	65 0F1 C	1C 1C7 E	65 0F1 A	Tags fin
0F1	2F	W	R	R	W	W	R	W	R	R	17 32
1C7	C				65				65		
0DB	23				0F				1A		
									0E		

Mem	5E F1 D W	19 C7 C W	5E F1 B R	8C DB 0 R	3C DB 3 W	5E F1 5 W	68 DB F R	CA F1 C W	39 C7 E R	CA F1 A R	Tags fin
F1	5E		5E			5E		CA		CA	5E CA
C7		19							39		19 39
DB				8C	3C		68				8C 3C 68

4. Calcul du temps d'accès effectif de la cache.

Temps accès effectif = $h \cdot t_p + (1-h) \cdot t_s$

Placement direct : $3/10r \times 8ns + 7/10e \times 100ns + 3/10w-b \times 100ns = 102,4ns$

Associative par ensembles de 2 : $3/10r \times 8ns + 7/10e \times 100ns = 72,4ns$

Associative par ensembles de 4 : $3/10r \times 8ns + 7/10e \times 100ns = 72,4ns$

ou r représentent les réussites à la cache et e les échecs.

5. Les données auraient été séparées de la sorte : 18 — 4, soit 18 bits pour le tag et 4 bits pour l'octet. En effet, ce mode de placement permet de mettre le bloc n'importe où il y a de la place dans la cache et ne prend pas une partie de l'adresse pour spécifier l'ensemble.

ex