M2-SEM
Concepts Avancés d'Architecture

Travaux dirigés sur les caches
TD 2.1

Année 2020-2021

Pr R. BOUDOUR

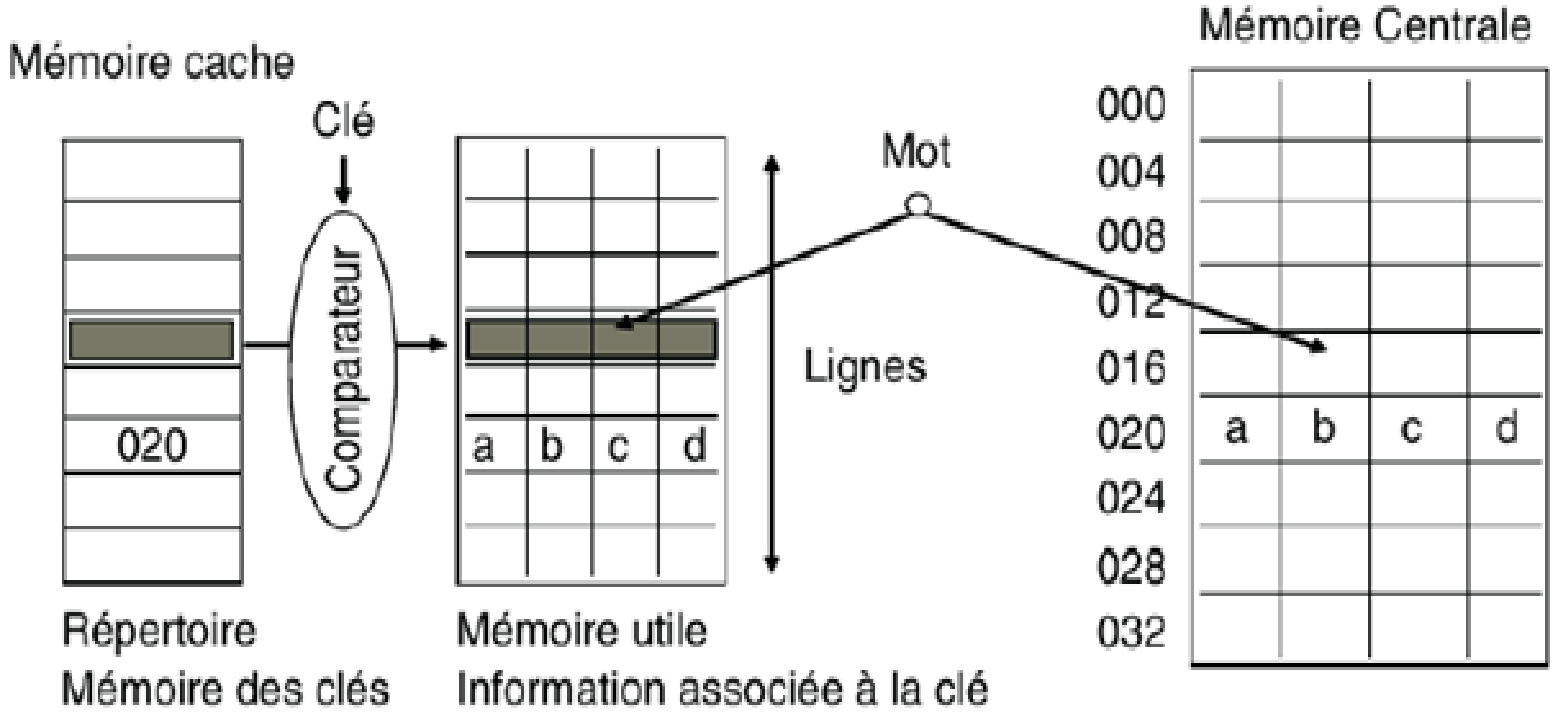
SRAM VS DRAM

Pour rappel :

- Design optimisé d'un SRAM
  - 4-6 transistors par bit
  - Temps d'accès très court
- Design optimisé d'un DRAM
  - 1 transistor par bit (plus de densité d'information donc moins $ par bit)
  - Temps d`accès plus long

Relation étiquette et numéro bloc

Relation clé (tag) cache et numéro bloc MP ? Observer la clé 020



Localité temporelle/spatiale

- Les données

```
for (i=0; i<N; i++) {
  for (j=0; j<N; j++) {
    y[i] = y[i] + a[i][j] * x[j]
  }
}
```

- Le programme

```
                ...
05  LOOP        LDR R1, R0, #3
06              ADD R1, R1, #5
07              STR R1, R0, #30
08              ADD R0, R0, #1
09              ADD R3, R0, R2
0A              BRn LOOP
```

i) **Indiquer le type de localité pour les données : y[i], a[i][j] et x[j] ?**

ii) **Soit le fragment de programme ci-dessus, expliquer localité spatiale et localité temporelle**

Localité temporelle/spatiale

- Les données

```
for (i=0; i<N; i++) {
  for (j=0; j<N; j++) {
    y[i] = y[i] + a[i][j] * x[j]
  }
}
```

- y[i]: propriétés de localités temporelle et spatiale.
- a[i][j]: propriété de localité spatiale.
- x[j]: propriété de localité temporelle et spatiale.

- Le programme

```
                ...
05  LOOP        LDR R1, R0, #3
06              ADD R1, R1, #5
07              STR R1, R0, #30
08              ADD R0, R0, #1
09              ADD R3, R0, R2
0A              BRn LOOP
                ...
```

Boucle : réutilisation des instructions : localité temporelle

Instructions consécutives en mémoire : localité spatiale

Exercice 1 : Q1 à Q8

Question 1

- **Pourquoi utilise-t-on des mémoires caches ?**

Réponse 1

- **Pour accélérer l'accès aux données en rapprochant les données du processeur dans des mémoires plus rapides mais plus coûteuses.**

Pour Q2 à Q8

- **Soit une mémoire cache de niveau L1 ayant les caractéristiques suivantes :**

  - 32 mots par ligne (mots de 2 octets)
  - Taille de 32ko
  - L1 et L2 sont inclusifs
  - 4-way set associatifs.
  - Remplacement LRU.
  - Association par poids faible.
  - Taille de bus d'adresse : 32 bits.

Question 2

- **Combien y a-t-il de lignes dans cette mémoire cache ?**

Réponse 2

$$\frac{TailleCache}{Taille\_Mot \times Nombre\_mots\_ligne} = \frac{32ko}{2 \times 32} = 512$$

Question 3

- **Combien y a t-il d'ensembles associatifs dans cette mémoire cache ?**

Réponse 3

$$\frac{Nombre\_lignes}{Nombre\_lignes\_par\_ens} = \frac{512}{4} = 128$$

Question 4

- **Si la mémoire cache de niveau L2 a une taille de 2 Mo, combien y a-t-il de blocs de la mémoire cache L2 par bloc de la mémoire cache L1 ?**

Réponse 4

$$\frac{Taille\_L2}{Taille\_L1} = \frac{2048}{32} = 64$$

Question 5

Si la mémoire fait 1 Go, combien d'adresses correspondront à un ensemble du cache L1 ?

Réponse 5

17	7	6
Etiquette	**Index**	**Offset**

$$\frac{Taille\_Mem}{Nombre\_ens \times Taille\_mot \times Taille\_ligne} = \frac{1Go}{128 \times 2o \times 32} = 131072$$

Question 6

Dans quels blocs (lignes) du cache peut-on trouver les blocs suivants :

Adresse décimale	Adresse hexadécimale	Numéro du bloc décimal	Numéro du bloc hexadécimal
0	00000000h		
64	00000040h		
640	00000280h		
4096	00001000h		
16384	00004000h		
64000	0000FA00h		

Réponse 6

Il suffit de considérer la structure de l'adresse dans l'organisation associative par ensemble de 4 lignes :
L'adresse de 32 bits se décompose ainsi :

Pour trouver le numéro du bloc (ensemble) on décompose l'adresse selon la structure ci-dessus et on identifie le champ index de 7 bits
Il me semble qu'il y a une erreur dans les adresses décimales 4096 et 16364 (conversion hexadécimale erronée aussi), numéro du bloc respectivement est 64 puis 127.

0	0000000 000000	0	
64	1 000000	1	
640	1010 000000	10	
4096	1000000 000000	64	erreur, avec la valeur hex le résultat est 0
16364	1 1111111 101100	127	erreur
16448	10 0000001 000000	1	
64000	111 1101000 000000	104	

Question 7

Si un bloc n'est pas présent en cache L1, combien de lignes de L1 aura-t-on parcouru ?

Réponse 7

- **4**

Question 8

Quelle est la probabilité de trouver un bloc quelconque présent dans L2 dans la mémoire cache L1 ?

- **Si L1 et L2 sont des caches inclusives ?**
- **Si L1 et L2 sont des caches exclusives ?**

Réponse 8

- $L1 \subset L2,\ P = \frac{1}{64} = 1.56\%$
- $L1 \cap L2 = \emptyset,\ P = 0$

Exercice 2

Soit les hypothèses suivantes :

taille cache 4k blocs,
taille bloc 4 mots (16 bytes),
adresse 32 bits.

Combien de bits sont nécessaires pour ranger les étiquettes si le cache :

- possède un mappage direct
- associatif par ensemble de deux blocs (2-way set associative)
- associatif par ensemble de 4 blocs (4-way set associative)
- complètement associatif (fully associative)

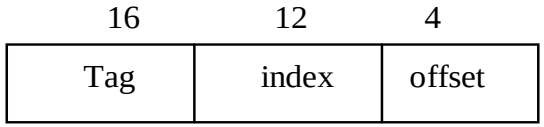
Réponse ex 2

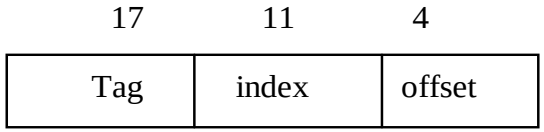
Mappage direct
- 16x4k=64 kbits

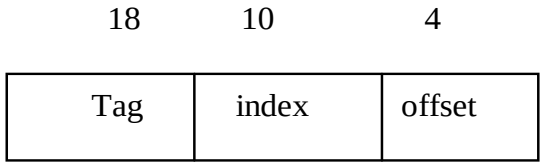
2-way
- 17x4k=68kbits

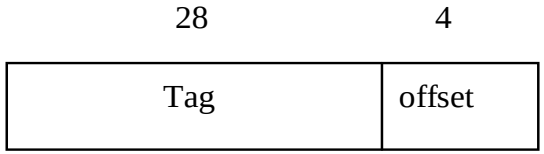
4-way
-18x4k=72kbits

fully associative
-28x4k=112kbits









Elever l'associativité ⇒ nombre de bits du champ index diminue, nombre de bits du champ étiquette augmente

Exercice 3

Un cache possède une capacité de 32 ko, des lignes de 128 octets et un degré d'associativité de 4. Le système contenant le cache utilise des adresses de 32 bits.

i) *Combien de lignes et d'ensembles possède le cache ?*

ii) *Combien d'entrées sont requises dans le tableau d'étiquettes ?*

iii) *Combien de bits d'étiquette sont requis pour chaque entrée dans le tableau d'étiquettes ?*

iv) Si le cache est de type write through, *combien de bits sont requis pour chaque entrée du tableau d'étiquettes et quelle quantité de mémoire totale est requise pour le tableau dans le cas d'une politique de remplacement LRU ? Qu'en serait-il s'il s'agissait d'un cache write-back ?*

v) Pour chacune des adresses ci-dessous, *indiquez le numéro de l'ensemble qui sera examiné afin de déterminer si l'adresse est contenue dans le cache et celui de l'octet référencé dans la ligne de cache :*

a) 0xABC8998 b) 0x32651987 c) 0x228945DB d) 0x48569CAC

Réponse ex 3

i) Le nombre de lignes est 32 ko/128 = $2^{15}/2^{7} = 2^{8}$ = 256 lignes, le nombre d'ensembles est : 256/4 = 64 ensembles

ii) Il y a 256 entrées dans le tableau d'étiquettes

iii) Longueur de l'étiquette : 32 – (6 + 7) = 19 bits

iv) Si le cache est :

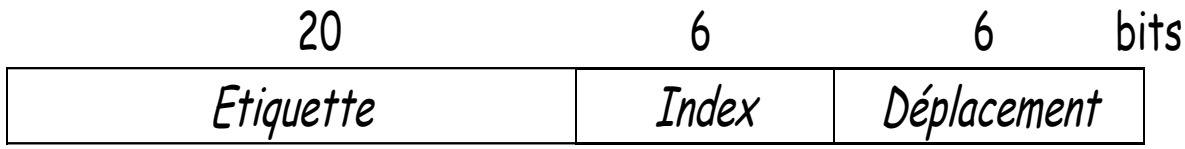
write through, l'étiquette sera 1 + 19+ 2 = 22 bits, taille du tableau : 23*256 = 5632 bits

write back, l'étiquette sera : 1 + 1 +2 + 19 = 23 bits, taille du tableau : 23*256 = 5888 bits

v) Numéro d'ensemble et numéro de cache : 19 et 24, 51 et 7, 11 et 91, 57 et 44

Exercice 4

Un cache associatif par ensembles décompose une adresse provenant de l'unité centrale (CPU) comme suit :



a) Déterminer la quantité maximale de mémoire adressable par la CPU
b) Déterminer la taille d'un bloc
c) Déterminer le nombre d'ensembles
d) Combien d'adresses correspondront à un ensemble du cache ?
e) Dans quelle entrée du cache sera chargé le mot d'adresse ABCDE8F8 ?
f) Peut-on déterminer la taille du cache ? Justifier votre réponse.

Réponse ex 4

a) Déterminer la quantité maximale de mémoire adressable par la CPU
2^{32} octets

b) Déterminer la taille d'un bloc
$2^{\text{déplacement}} = 2^6$ octets

c) Déterminer le nombre d'ensembles
$2^{\text{index}} = 2^6$ ensembles

d) Combien d'adresses correspondront à un ensemble du cache ?
Taille mémoire/nombre d'ensembles x taille ligne
$2^{32}/2^6 x 2^6 = 2^{20}$ adresses

e) Dans quelle entrée du cache sera chargé le mot d'adresse ABCDE8F8 ?
10101011110011011110 100011 111000
L'entrée 35 du cache à l'emplacement 56 de la ligne

f) Peut-on déterminer la taille du cache ? Justifier votre réponse.
non, on ne peut déterminer la taille du cache car on n'a pas d'information sur le nombre de blocs par ensemble.