M2-SEM
Concepts Avancés d'Architecture

Série de TDs sur les caches
TD2-2 (Suite)

Année 2020-2021
Pr R. BOUDOUR

Exercice 5

1) Les tableaux ci-dessous résument les actions entreprises lors d'un succès ou d'un échec, suivant la stratégie adoptée pour un accès en écriture.
Caractériser chacun des deux caches ?

	Lecture	Ecriture
Succès	Lire le mot dans le cache	Ecrire le mot dans le cache Ecrire le mot dans la MP
Echec	Charger la ligne Lire le mot dans le cache	Ecrire le mot en MP

Tableau 1

	Lecture	Ecriture
Succès	Lire le mot dans le cache	Ecrire le mot dans le cache
Echec	Décharger ligne si modifiée Charger la ligne Ecrire le mot dans le cache	Décharger ligne si modifiée Charger la ligne Lire le mot dans le cache

Tableau 2

2) Expliquer le problème de cohérence.

Réponse ex 5

* Ecriture allouée : elle consiste d'abord à charger le bloc manquant dans le cache, avant d'effectuer l'écriture.
* Ecriture non allouée : elle écrit uniquement dans la MP.
*Le tableau 1 caractérise un cache en écriture simultanée avec écriture non allouée.
* Le tableau 2 décrit un cache en réécriture avec mémoire allouée..
* Problème de cohérence : le cache et la MP peuvent contenir deux versions différentes de la même adresse mémoire : *le cache peut contenir une version plus récente dans le cas de la réécriture ; la cohérence est alors assurée par le microprocesseur via le mécanisme de déchargement.*
La MP peut aussi contenir une version plus récente, par exemple si un contrôleur d'E/S dans une architecture monoprocesseur a modifié le contenu d'une ligne de la MP qui est aussi présente dans le cache.

Exercice 6

Un cache possède une capacité de 64 Ko, des lignes de 128 octets et un degré d'associativité de 4. Le système contenant le cache utilise des adresses de 32 bits.

Donnez l'adresse du premier mot dans la ligne contenant l'adresse suivante :

- 0xA23847EF
- 0x7245E824
- 0xEEFABCD2

Réponse ex 6

Si les lignes du cache font 128 octets, alors les 7 bits de poids faible de l'adresse indiquent à quel octet dans la ligne une adresse fait référence.

Les lignes étant alignées, l'adresse du premier mot dans la ligne peut être trouvée en positionnant à 0 les bits spécifiant les octets dans la ligne.

En conséquence, les adresses des premiers octets dans les lignes contenant les adresses précitées sont les suivantes :

- *0xA2384780*
- *0x7245E800*
- *0xEEFABC80*

Exercice 7

Soit un cache possédant les mêmes caractéristiques que celui de l'exercice précédent. Pour chacune des adresses mentionnées ci-après, 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.

Nous supposerons que les bits utilisés pour sélectionner un octet à l'intérieur de la ligne sont les bits de poids faible de l'adresse et que les bits utilisés pour sélectionner l'ensemble sont les bits suivants de poids plus élevé.

- 0xABC89987
- 0x32651987
- 0x228945DB
- 0x48569CAC

Réponse ex 7

Dans l'exercice précédent, nous avons conclu que 7 bits étaient requis pour sélectionner un octet dans la ligne de cache, aussi les 7 bits de poids faible de l'adresse détermineront l'octet de référence. Sept bits sont également requis pour sélectionner l'ensemble, qui correspondra aux bits 7 à 13 de l'adresse. Considérant cela, les réponses sont les suivantes :

- *0xABC89987*
  Octet dans la ligne = 0x7, ensemble = 0x33 (51).
- *0x32651987*
  Les 14 bits de poids faible de cette adresse sont les mêmes que ceux de la question précédente, aussi, l'octet et l'ensemble seront-ils les mêmes.
- *0x228945DB*
  Octet dans la ligne = 0x5B (91), ensemble = 0xB (11).
- *0x48569CAC*
  Octet dans la ligne = 0x2C (44), ensemble = 0x39 (57).

Exercice 8

Un programme accède à 1 000 000 de références mémoire. Lorsqu'il tourne sur un système donné, le cache obtient un taux de miss de 7 %, dont un quart sont des échecs obligatoires, un autre quart des échecs de capacité et la moitié des échecs de conflits.

i) Si la seule modification que vous pouvez apporter au cache consiste à augmenter son degré d'associativité, quel est le nombre maximal de miss que vous pouvez espérer éliminer ?

ii) Si vous avez la possibilité d'augmenter la taille du cache en plus de son degré d'associativité, quel est le nombre maximal de miss que vous pouvez espérer éliminer ?

Réponse Ex 8

i) Le fait d'augmenter le degré d'associativité permettra de :

- réduire le nombre d'échecs de conflits (des miss qui surviennent parce que les lignes de cache se font concurrence pour des emplacements dans le cache),

- mais n'affectera pas le nombre d'échecs de capacité (les miss qui surviennent parce qu'un programme référence plus de données que ne peut en contenir le cache).

- En conséquence, le mieux que nous puissions espérer en augmentant le degré d'associativité du cache est d'éliminer tous les échecs de conflit.

Le taux de miss étant de 7 % et le programme opérant 1 000 000 de références mémoire, le nombre total de miss est de 70 000.

La moitié d'entre eux sont des échecs de conflit. Le nombre maximal que nous pouvons éliminer en augmentant le degré d'associativité du cache est donc de 35 000.

Réponse Ex 8

ii) En augmentant la capacité ainsi que le degré d'associativité du cache, nous pouvons éliminer les miss de capacité et les miss de conflit à la fois, soit les trois quarts de l'ensemble des miss, pour un total de :
35000 + 17500 = 52 500 miss.

Exercice 9

I. Vous êtes consulté pour faire une recommandation pour une nouvelle architecture d'ordinateur aux ingénieurs impliqués dans le design du cache et à ceux responsables de la configuration de la mémoire. L'ordinateur doit être conçu pour exécuter des programmes ou tâches informatiques qui changent dans des intervalles de temps extrêmement courts pour supporter des applications multitâches à très haute fréquence. A chacun des changements de tâche, la mémoire cache doit être vidée (flushed) de son contenu. En discutant autour de la table de réunion, vous entendez plusieurs propositions venant du personnel de la compagnie.

Exercice 9

1) *Parmi ces propositions, laquelle est vraie ?*

a) Ceci semble être des échecs de type obligatoire et donc dépendants de la dimension du cache

b) Ceci semble être des échecs de type capacité et donc dépendants de la dimension du cache

c) Ceci semble être des échecs de type obligatoire et sont indépendants de la dimension du cache

d) Ceci semble être des échecs de type capacité et donc indépendants de la dimension du cache

2) *Que recommanderiez-vous ?*

a) Augmenter la dimension du bloc

b) Diminuer la dimension du bloc

c) Augmenter le degré d'associativité dans la mémoire cache

d) Diminuer le degré d'associativité dans la mémoire cache

Exercice 9

II. Le nombre de cycles horloge pour l'exécution de chaque instruction est une moyenne qui dépend des détails de l'organisation de chaque ordinateur et de découlement des instructions. Quelles sont les variables qui peuvent faire varier le nombre de cycles requis pour exécuter une instruction ?

1. un échec au cache
2. l'introduction de bulles dans le pipeline pour éviter un aléa de donnée
3. un succès au cache

a. 1 seulement ; b. 2 et 3 ; c. 1, 2 et 3 ; d. 3 seulement e. 1 et 2

Réponse ex 9

- I

  1) c

  2) a

- II

  e. 1 et 2

Exercice 10

On suppose que le processeur utilisé a un cache données de 64 Ko, avec des lignes de 64 octets. Le cache utilise la réécriture avec écriture allouée (il y a des défauts de cache en écriture),
Le processeur a des adresses sur 32 bits. Le programme travaille sur deux tableaux de doubles X[N] et Y[N] situés en mémoire aux adresses :

&X[0] = F000 0000H, &Y[0] = &X[N-1] + 8

1) Quel est pour ce cache le nombre de bits pour l'adresse dans la ligne, le nombre de bits d'index et le nombre de bits d'étiquette dans les 2 cas suivants :

a) correspondance directe,
b) associativité 4 voies (4 lignes par ensemble) avec remplacement LRU.

2) Dans quelles lignes vont les flottants X[0] et Y[0] en correspondance directe pour :

N = 128, N = 1024 et N = 8192 ?

Réponse ex 10

Longueur adresse : 32 bits ; Taille cache : 64ko ;
Longueur ligne : 64 octets
Adresse dans la ligne est de 6 bits (64 octets/ligne) et il y a 1024 lignes

Correspondance directe

- Adresse dans la ligne : 6 bits
- Index : 10 bits
- Etiquette : 16 bits

Associativité 4 voies :

- Adresse dans la ligne : 6 bits
- Index : 8 bits
- Etiquette : 18 bits

Réponse ex 10

Il suffisait de remarquer que la représentation des flottants était en double précision sur 8 octets et que l'adresse porte sur l'octet.
En correspondance directe, l'emplacement est déterminé par la relation :

N° index ***mod*** *nbre lignes cache*

Ainsi :
&X[0] = F000 0000, en binaire : 1111 0000 0000 0000 0000 0000 0000 0000
identification donne avec le format de l'adresse en correspondance directe: 111100000000000 **0000000000** 000000 , index = 0 (0 mod 1024=0), il sera placé dans la ligne 0.

Pour Y[0] = ? nous savons que : &Y[0] = &X[127] + 8
&X[127] + 8 = & [127 * 8 + 8] = &[1024], en binaire 1024_{10} = 10000 000000_2
après identification : 16 mod 1024 = 16
donc Y[0] sera placé dans la ligne 16.
Idem pour N = 1024 puis N=8192

Réponse ex 10

Ainsi :

&X[0] = 1111 0000 0000 0000 | 0000 0000 00|00 0000

N= 128

&Y[0] = 1111 0000 0000 0000 | 0000 0100 00|00 0000

X[0] va dans la ligne 0 et Y[0] va dans la ligne 16

N=1024

&Y[0] = 1111 0000 0000 0000 | 0010 0000 00|00 0000

X[0] va dans la ligne 0 et Y[0] va dans la ligne 128

N= 8192

&Y[0] = 1111 0000 0000 0001 | 0000 0000 00|00 0000

X[0] va dans la ligne 0 et Y[0] va dans la ligne 0

Exercice 11

On considère qu'un processeur a un cache de données de 8 Ko, avec des blocs (lignes) de 16 octets. Il utilise la réécriture (write back) avec écriture allouée (write allocate). Le processeur exécute le programme C suivant :

```
float x[n], y[n], z[n] ;
        int i ;
        for (i = 0 ; i<100 ; i++)
        z[i] = x[i] + y[i] ;
```

Les adresses de début des tableaux sont respectivement :

@X[0] = F000 0000H
@Y[0] = F000 2000H
@Z[0] = F000 4000H

i) Donner le taux d'échecs cache de la boucle (n = 100) dans les hypothèses suivantes :
   a) le cache est à correspondance directe
   b) le cache est associatif quatre voies (4 blocs par ensemble)
   c) le cache est associatif deux voies (2 blocs par ensemble)

ii) Reprendre la question i) en supposant que les adresses de début des tableaux sont :

@X[0] = F000 0200H
@Y[0] = F000 2430H
@Z[0] = F000 4550H

Réponse ex 11

i)

a) **Correspondance directe**

31 13	12 4	3 0
1111 0000 0000 0000 000	0 0000 0000	0 0000
1111 0000 0000 0000 001	0 0000 0000	0 0000
1111 0000 0000 0000 010	0 0000 0000	0 0000

b) **4-way set associative**

31 11	10 4	3 0
1111 0000 0000 0000 0000 0	000 0000	0 0000
1111 0000 0000 0000 0010 0	000 0000	0 0000
1111 0000 0000 0000 0100 0	000 0000	0 0000

c) **2-way set associative**

31 12	11 4	3 0
1111 0000 0000 0000 0000	0000 0000	0 0000
1111 0000 0000 0000 0010	0000 0000	0 0000
1111 0000 0000 0000 0100	0000 0000	0 0000

Réponse ex 11

En correspondance directe, les adresses de X[i], Y[i] et Z[i] correspondent au même bloc du cache. Avec les caches associatifs 2 ou 4 voies, les adresses de X[i], Y[i] et Z[i] correspondent au même ensemble.

Réponse ex 11

a) correspondance directe : il y a trois défauts de cache par itération, un pour chaque élément de chaque tableau

Réponse ex 11

) associativité quatre voies : il y a quatre blocs différents pour chaque ensemble. Chaque
loc ayant 4 flottants, il y a trois défauts de cache pour 4 itérations, soit 0,75
défauts/itération

Réponse ex 11

c) associativité 2 voies : il y a deux blocs par ensemble. Compte tenu du LRU, il y a 3 défauts de cache par itération (voir ci-dessous)

Itération	1er ensemble	2ème ensemble
0	**X[0]** X[1] X[2]X[3]	**Y[0]** Y[1] Y[2]Y[3]
0-1	**Z[0]** Z[1] Z[2]Z[3]	X[0] **X[1]** X[2]X[3]
1	Y[0] **Y[1]** Y[2]Y[3]	Z[0] **Z[1]** Z[2]Z[3]
2	X[0] X[1] **X[2]**X[3]	Y[0] Y[1] **Y[2]**Y[3]
2-3	Z[0] Z[1] **Z[2]**Z[3]	X[0] X[1] X[2]**X[3]**
3	Y[0] Y[1] Y[2]**Y[3]**	Z[0] Z[1] Z[2]**Z[3]**

Réponse ex 11

En correspondance directe, les adresses de X[i], Y[i] et Z[i] correspondent au même bloc du cache. Avec les caches associatifs 2 ou 4 voies, les adresses de X[i], Y[i] et Z[i] correspondent au même ensemble.

a) correspondance directe : il y a trois défauts de cache par itération, un pour chaque élément de chaque tableau

b) associativité quatre voies : il y a quatre blocs différents pour chaque ensemble. Chaque bloc ayant 4 flottants, il y a trois défauts de cache pour 4 itérations, soit 0,75 défauts/itération

c) associativité 2 voies : il y a deux blocs par ensemble. Compte tenu du LRU, il y a 3 défauts de cache par itération (voir ci-dessous)

Itération	1er ensemble	2ème ensemble
0	**X[0]** X[1] X[2]X[3]	**Y[0]** Y[1] Y[2]Y[3]
0-1	**Z[0]** Z[1] Z[2]Z[3]	X[0] **X[1]** X[2]X[3]
1	Y[0] **Y[1]** Y[2]Y[3]	Z[0] **Z[1]** Z[2]Z[3]
2	X[0] X[1] **X[2]**X[3]	Y[0] Y[1] **Y[2]**Y[3]
2-3	Z[0] Z[1] **Z[2]**Z[3]	X[0] X[1] X[2]**X[3]**
3	Y[0] Y[1] Y[2]**Y[3]**	Z[0] Z[1] Z[2]**Z[3]**

Réponse ex 11

ii)

a)

31 13	12 4	3 0
1111 0000 0000 0000 000	0 0010 0000 32	0000
1111 0000 0000 0000 001	0 0100 0011 67	0000
1111 0000 0000 0000 010	0 0101 0101 85	0000
Etiquette	**Index**	**Dépl.**

b)

31 11	10 4	3 0
1111 0000 0000 0000 0000 0	010 0000 32	0000
1111 0000 0000 0000 0010 0	100 0011 67	0000
1111 0000 0000 0000 0100 0	101 0101 85	0000
Etiquette	**Index**	**Dépl.**

c)

31 12	11 4	3 0
1111 0000 0000 0000 0000	0010 0000 32	0000
1111 0000 0000 0000 0010	0100 0011 67	0000
1111 0000 0000 0000 0100	0101 0101 85	0000
Etiquette	**Index**	**Dépl.**

Réponse ex 11

En correspondance directe, les adresses de X[i], Y[i] et Z[i] correspondent à des blocs différents de cache.

Réponse ex 11

Avec les caches associatifs 2 ou 4 voies, les adresses de X[i], Y[i] et Z[i] correspondent à des ensembles différents.

Réponse ex 11

Dans les trois cas, il y a donc 3 défauts pour 4 itérations, soit 0,75 défauts par itération.

Donc 75 défauts pour 100 itérations.

Réponse ex 11

En correspondance directe, les adresses de X[i], Y[i] et Z[i] correspondent à des blocs différents
du cache. Avec les caches associatifs 2 ou 4 voies, les adresses de X[i], Y[i] et Z[i] correspondent
à des ensembles différents.
Dans les trois cas, il y a donc 3 défauts pour 4 itérations, soit 0,75 défauts par itération.

Donc 75 défauts pour 100 itérations.

Exercice 12

Un programme est formé d'une boucle de 48 instructions à exécuter 2 fois ; les instructions se trouvant dans l'ordre, aux adresses mémoire : 1 à 8, 145 à 160, 9 à 16, 145 à 160.

Ce programme doit tourner sur une machine possédant une mémoire cache d'une taille de 24 instructions. Le temps de cycle de la mémoire principale est M et le temps de cycle du cache est C. Le cache est totalement associatif et la stratégie de remplacement utilisée est LRU (on remplace le bloc le moins récemment utilisé).

Le cache possède 3 blocs de 8 instructions : les blocs que l'on peut transférer sont : 1-8, 9-16, …, 137-144, 145-152, 153-160 …

i) Quel est le temps total d'exécution du programme en ne tenant pas compte des temps de calcul ?

ii) Application numérique : M = 50 ns, C = 5 ns

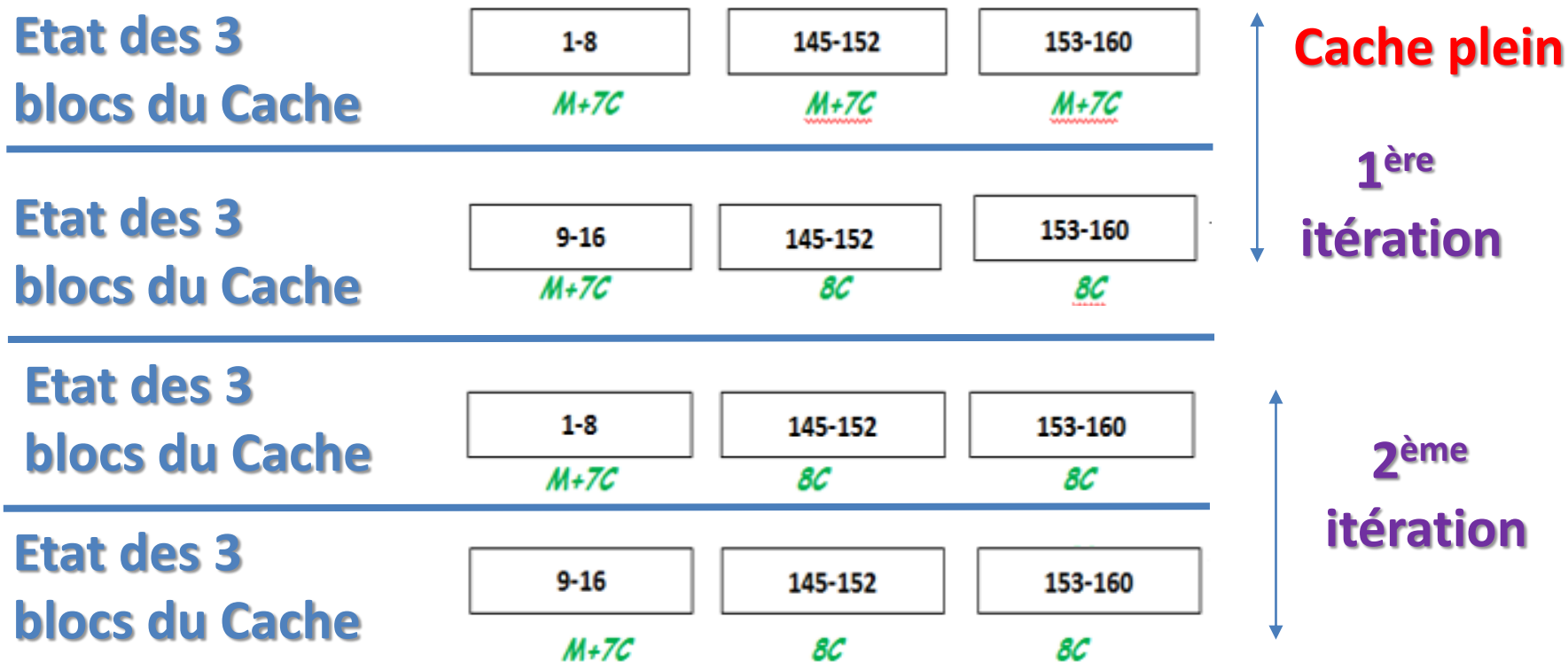
Réponse ex 12

- **Mémoire**

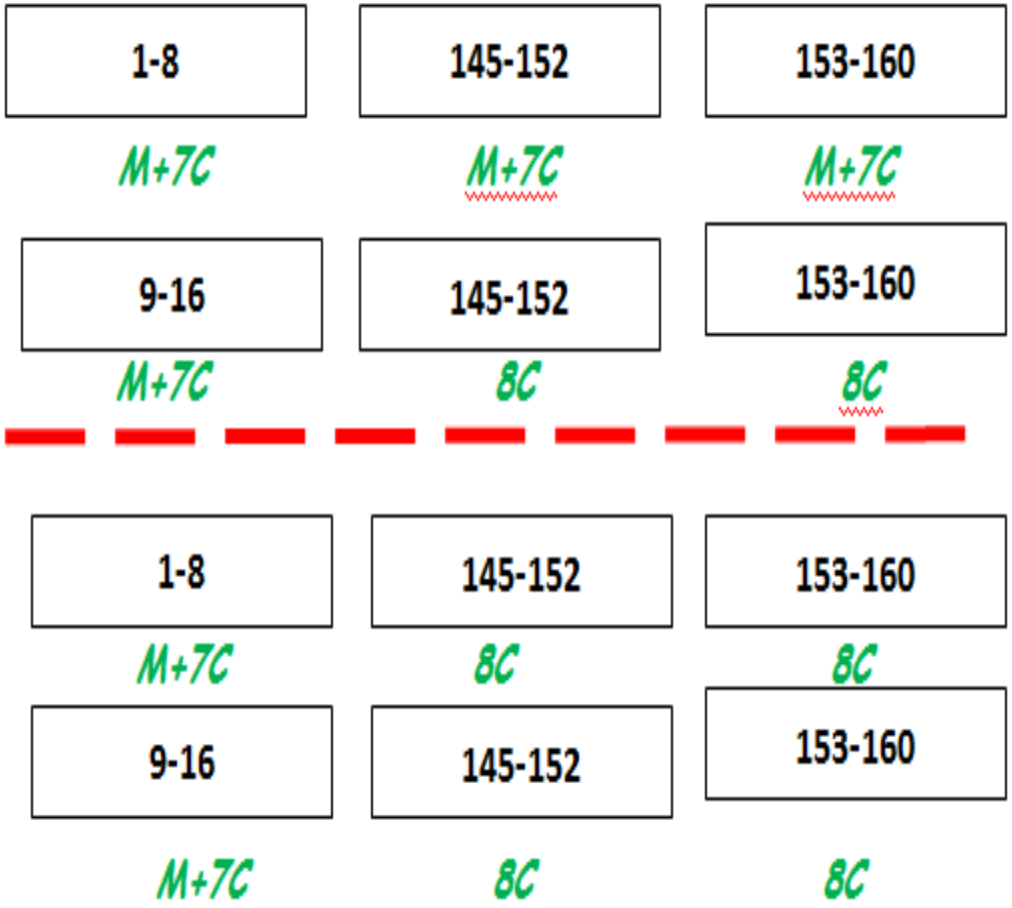
Ordre	Adresse	Instruction
1	1-8	Instructions
3	9-16	Instructions
2 / 4	145-160	Instructions

- **Boucle de 48 instructions**
- **Cache associatif de 3 blocs**
- **Un bloc comporte 8 instructions**

Réponse ex 12



Réponse ex 12



Temps total d'exécution = 6M+90C

Pour M=50ns et C=6ns

Temps total d'exécution = 6x50 + 90x5 = 750ns

Temps de calcul négligé

Doing nothing is very hard to do, you never know when you're finished.