



M2-SEM CAA

Série de TDs N° 3 sur la mémoire virtuelle

Année 2020-2021

Pr. R. BOUDOUR

Exercice 1

- Un ordinateur dispose d'un espace d'adressage virtuel accessible avec des adresses exprimées sur 32 bits. La taille des pages est de 4 ko. *Combien de pages composent l'espace d'adressage virtuel ?*
- La taille des pages doit-elle nécessairement être une puissance de 2 ?*
- Une page d'une taille de 4 000 octets peut-elle par exemple être implémentée sur le plan théorique ? Si oui, qu'en est-il dans la pratique ?*

Réponse 1

- ❑ *L'espace d'adressage couvre 2^{32} octets. Chaque page faisant 4 Ko, le nombre de pages est de :*

$$2^{32}/2^{12} = 2^{20} = 1\,048\,576 \text{ pages.}$$

- ❑ *En théorie, un système de mémoire virtuelle n'est qu'une correspondance entre des adresses virtuelles et des adresses physiques et l'on peut donc utiliser n'importe quelle taille de page.*
- ❑ *Cependant, en pratique, les MMU utilisent certains bits de l'adresse virtuelle comme index de la table des pages et cela ne peut se faire que si la taille des pages est une puissance de deux.*

Exercice 2

- ❑ Soit un système segmenté et paginé.
 - ❑ La taille moyenne d'un segment est de s mots
 - ❑ tandis que la taille des pages est de p mots.
 - ❑ Lorsqu'un segment est placé en mémoire, on perd un peu de place dans la dernière page.
 - ❑ De plus, on utilise s/p mots pour conserver la table des pages (un mot par entrée). Plus la taille des pages est petite, moins on perd de place dans la dernière page, mais plus la table des pages est grande.

Quelle est la taille des pages optimale (qui minimise l'espace perdu) ?

Réponse 2

□ Le coût de la fragmentation interne

- En moyenne, la dernière page d'un segment est remplie à moitié, On perd donc $p/2$ octets
- Le coût de la table des pages
 - Si chaque processus utilise en moyenne un espace s , il utilise s/p entrées dans la table des pages. On a une perte s/p
- Choix de l'optimum : On cherche à minimiser la perte $\approx p/2 + s/p$,
L'optimum est $p = \sqrt{2s}$ (entrée de table $e = 1$ octet)

- Si l'entrée de la table des pages est égale à e ($e \neq 1$) alors $p = ?$

Exercice 3

Soit une mémoire segmentée et paginée. Chaque adresse virtuelle comporte un numéro de segment de 2 bits, un numéro de page de 2 bits et un déplacement dans la page de 11 bits. La mémoire principale contient 32 ko, divisée en pages de 2 ko. Chaque segment est soit en mode lecture seule, soit en mode lecture-écriture, soit en mode lecture-écriture-exécution. On dispose de la table des pages et de protection suivante :

Segment 0		Segment 1		Segment 2	Segment 3	
Lecture seule		Lecture-exécution		Lect.-écrit.-exécution	Lecture-écriture	
Page virtuelle	Page réelle	Page virtuelle	Page réelle		Page virtuelle	Page réelle
0	9	0	sur disque	La table des pages n'est pas en mémoire centrale	0	14
1	3	1	0		1	1
2	sur disque	2	15		2	6
3	12	3	8		3	sur disque

Exercice 3

Pour chacun des accès suivants à la mémoire virtuelle, donner l'adresse physique calculée. Si une erreur se produit, donner son type :

Accès	Segment	Page	Déplacement dans la page
1. Charger les données	0	1	1
2. Charger les données	1	1	10
3. Charger les données	3	3	2047
4. Enregistrer les données	0	1	4
5. Stockage des données	3	1	2
6. Enregistrer les données	3	0	14
7. Sauter à l'adresse	1	3	100
8. Charger les données	0	2	50
9. Lecture les données	2	0	5
10. Sauter à l'adresse	3	0	60

Réponse 3

Les dix accès donnent les résultats suivants :

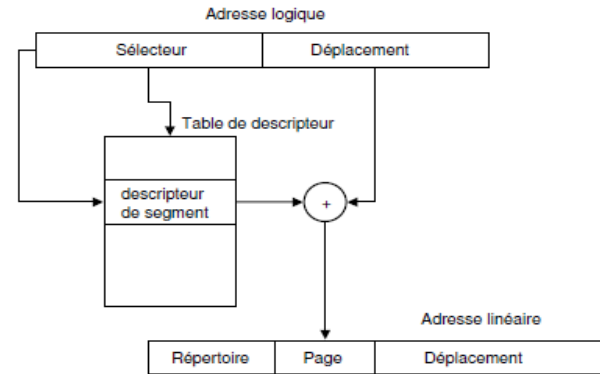
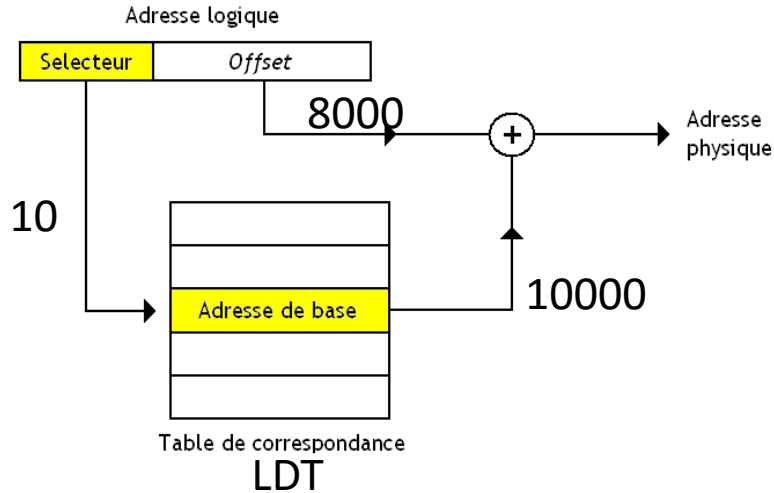
- (1) 6145,*
- (2) 10,*
- (3) défaut de page,*
- (4) défaut de protection,*
- (5) 2050,*
- (6) 28686,*
- (7) 16484,*
- (8) défaut de page,*
- (9) défaut de table des pages,*
- (10) défaut de protection. Notons que si un défaut de protection survient, il importe peu que la table des pages et la page soient en mémoire.*

Exercice 4

- ❑ Soit un programme qui référence sur le Pentium 4,
 - ❑ Le segment local 10, avec un déplacement de 8000.
 - ❑ Le champ BASE du segment LDT 10 contient 10000.

*Quelle entrée du répertoire de pages le Pentium 4 utilise t-il ?
Quels sont le numéro de page et le déplacement dans la page ?*

Réponse 4



❑ L'adresse linéaire est : $10\ 000 + 8\ 000 = 18\ 000$,

Selon le format Intel et identification des champs, on obtient :
ce qui signifie qu'on utilise :

DIR	PAGE	OFFSET
0000000000	0000000100	011001010000
0	4	1616

- ❑ l'entrée 0 du répertoire des pages,
- ❑ avec la page 4
- ❑ et le déplacement $18000 - 4 \times 4096 = 1616$.

Exercice 5

- A. Comparez la fragmentation interne avec la fragmentation externe. Comment les minimiser ?
- B. La mise en mémoire cache et la pagination sont en quelque sorte des procédures semblables qui se caractérisent par deux niveaux de mémoire (la mémoire cache et la mémoire principale pour la première, la mémoire principale et le disque dur pour la seconde).
Les arguments en faveur de l'utilisation de petites et de grandes pages sur le disque, s'appliquent-ils à la taille des lignes de la mémoire cache ?

Réponse 5

- A.
- ❑ On parle de fragmentation interne lorsque la dernière page d'un segment n'est pas pleine (fragmentation intra-segment)
 - ❑ La fragmentation externe correspond aux trous qui subsistent entre les segments (fragmentation inter-segment),
 - ❑ On peut diminuer la fragmentation interne en choisissant de petites tailles de page,
 - ❑ La fragmentation externe, se maîtrise en améliorant les techniques de placement des segments ou en faisant du compactage

Réponse 5

B.

- ❑ Puisqu'il n'y a pas de temps d'accès avant le chargement d'un bloc de cache, doubler la taille du bloc de cache double le temps d'accès.
 - ❑ Charger 128 octets prend deux fois plus de temps que d'en charger 64, Autrement dit, cela ne présente aucun avantage de charger 128 octets plutôt que deux fois 64 octets.
- ❑ Avec la pagination, il y a un long temps d'accès, souvent jusqu'à 20 ms, qui dépend du disque.
 - ❑ Du coup, charger deux fois 2Ko est beaucoup plus long que de charger 4Ko. Cela conduit à avoir de grandes tailles de pages, mais ne s'applique pas aux caches.

La tendance est aux grandes tailles de page ?!

Exercice 6

- I. Quelle est la différence entre une adresse virtuelle et une adresse physique ?
- II. Considérons le jeu de correspondances d'adresses ci-dessous pour une architecture dans laquelle les adresses virtuelles et physiques font 32 bits de long et dont les pages sont de 4 Ko, quelle est l'adresse physique qui correspond à chacune des adresses virtuelles suivantes ?
- a) 0x22433007
 - b) 0x13385abc
 - c) 0xabc89011

NP virtuelle	NP physique
0xabc89	0X97887
0x13385	0x99910
0x22433	0x00001
0x54483	0x1a8c2

Exercice 6

III. Les implémentations de tables de pages et de TLB font appel à un système de mémoire virtuelle à réécriture. Indiquez une raison pour laquelle les systèmes de mémoire virtuelle à écriture simultanée ne sont pas utilisés ?

Réponse 6

I.

- Les adresses physiques référencent directement des emplacements de la mémoire du système, Il s'agit des adresses que le processeur transmet au système mémoire
- Les adresses virtuelles sont des adresses que les programmes utilisent dans leurs opérations de chargement et de stockage
- Le système de mémoire virtuelle a pour charge de traduire les adresses virtuelles utilisées par un programme en adresses physiques utilisées par le système mémoire lors de l'exécution de chaque référence mémoire

Réponse 6

II.

- a) Dans ce système, les 12 bits de poids faible de l'adresse (0x007 dans ce cas) correspondent au décalage dans la page tandis que les 20 bits de poids fort (0x22433) définissent le numéro de page virtuelle. En observant le tableau, nous remarquons que le numéro de page physique correspondant ce numéro de page virtuelle est 0x00001. En concaténant le décalage dans la page avec le numéro de page physique, nous obtenons l'adresse physique qui correspond l'adresse virtuelle (0x00001007).
- b) 0x99910abc
- c) 0x97887011

Réponse 6

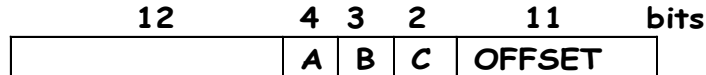
III.

- ❑ Les systèmes de mémoire virtuelle ne sont pas implémentés write through à cause de la différence de vitesse considérable entre la mémoire principale et les disques durs :
 - ❑ plusieurs ms pour disque dur
 - ❑ quelques dizaines voire des centaines de ns pour la mémoire principale

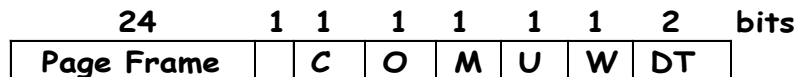
Exercice 7

Le 68030 de Motorola gère une mémoire virtuelle à l'aide de MMU (unité de gestion mémoire). Le 68030 commence avec une adresse de 32 bits engendrée par l'unité centrale et termine lorsqu'on a une adresse physique sur 32 bits. Le nombre de niveaux de tables utilisés varie de 0 à 4, le nombre de bits de chaque niveau peut-être déterminé par logiciel en affectant les champs d'un registre global (Translation Contrôle Registre). De plus, il est possible de demander au MMU d'ignorer les n bits de poids forts.

Supposons une découpe de l'adresse virtuelle selon le format suivant :



- i) Et si on ignore les 12 bits de poids forts, quelle serait la taille de l'espace d'adressage virtuel ? Puis la taille de la page ?
- ii) Ecrire l'adresse virtuelle dans le format ci-dessus : 000AEOC7H
- iii) Avec les 3 niveaux d table A,B et C : faire un schéma représentant le mécanisme de conversion utilisé par le MMU pour convertir une adresse virtuelle en adresse physique
- iv) La table finale de l'arbre des tables contient des descripteurs de pages réelles de la forme :



Indiquer le rôle de chacun des indicateurs : C, M, U, W et DT

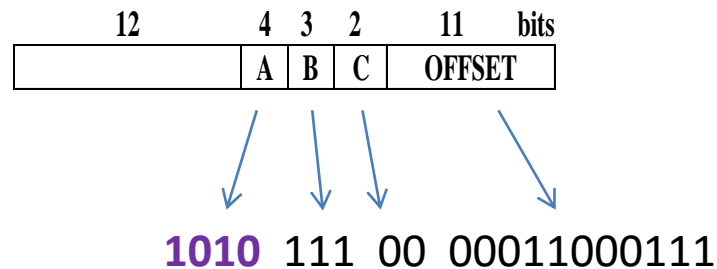
- Comment la MMU sélectionne-t-elle une table A parmi les 8 tables utilisées des 16 tables possibles pour déclencher la recherche de l'adresse physique.

Réponse 7

i) La taille de l'espace d'adressage serait de : 2^{20}

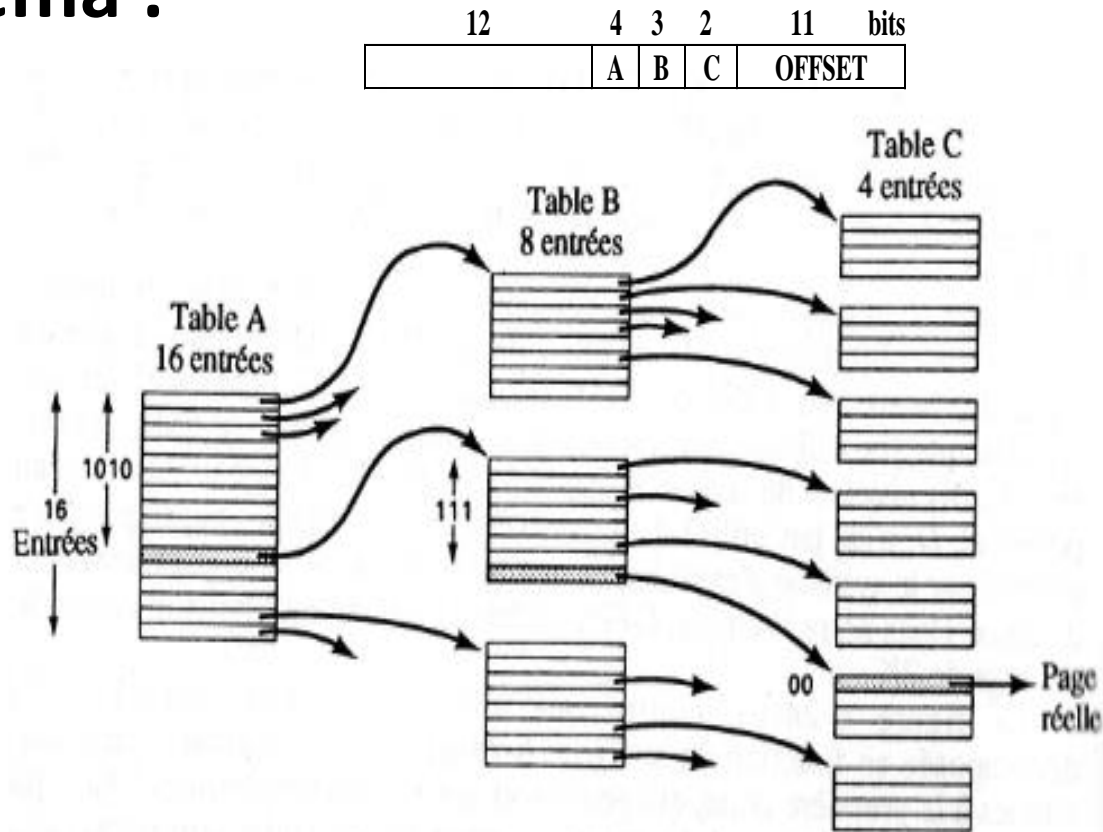
La taille la page : 2^{11}

ii) Ecriture de l'adresse virtuelle : 000AE0C7H dans le format ci-dessus :



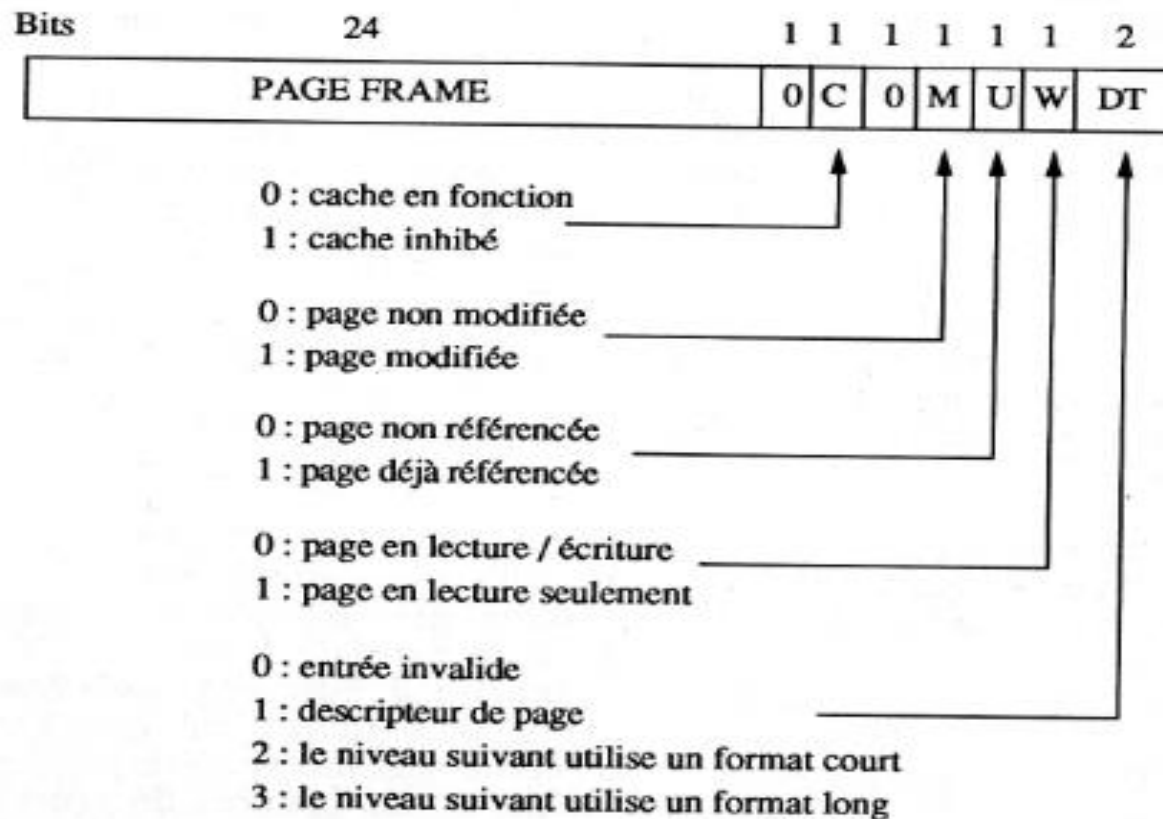
Réponse 7

iii) Schéma :



Réponse 7

iv) Rôle des indicateurs

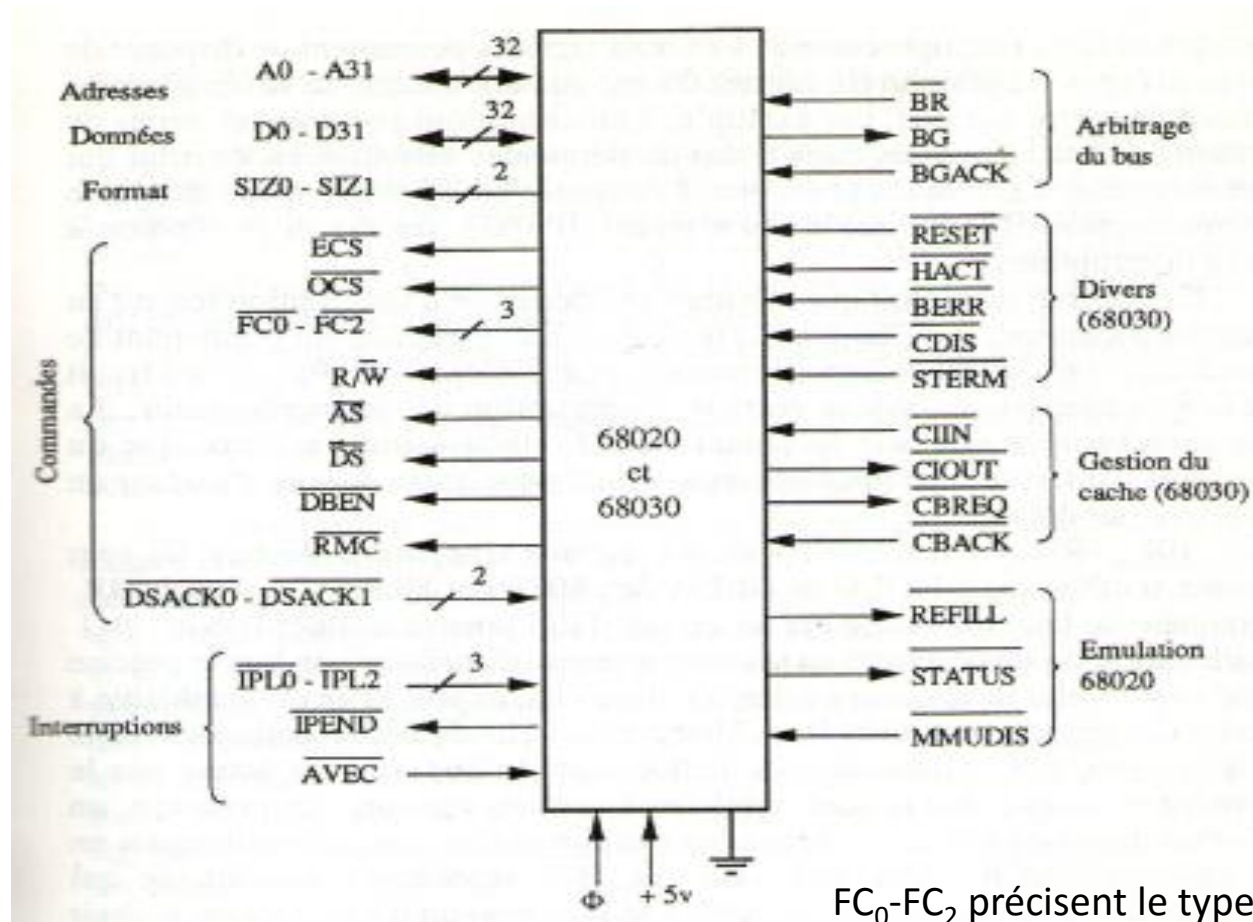


Réponse 7

v) Comment le MMU localisait la table A ?

Le matériel gère un tableau de 8 registres correspondant aux différents codes de la fonction FCx que l'on peut trouver au niveau des broches du circuit 68030. Suivant le code de fonction, on sélectionne une des 8 tables A potentielles qui peut être sélectionnée pour commencer la recherche

Architecture externe du 68030



$\overline{FC0} - \overline{FC2}$ précisent le type de cycle de bus courant (écriture, lecture, ...)

Exercice 8

1. Un processeur donné possède des adresses virtuelles et physiques de 32 bits. La taille de page est de 1 ko et le tampon de traduction anticipée (TLB) du processeur possède 128 entrées, pour un degré d'associativité de 4. Combien de mémoire requiert le tampon de traduction ? Nous supposons que le tampon de traduction n'arrondit pas ses entrées à l'octet supérieur.
2. Si le tampon de traduction anticipée d'un processeur possède un taux de hit de 90 % et qu'il faut 200 cycles pour effectuer une recherche dans la table des pages, quel est le temps de traduction moyen ? Nous supposons que le cache est étiqueté virtuellement et adressé physiquement afin de pouvoir masquer le temps de traduction si un hit TLB a lieu.

Réponse 8

- ❑ *Un TLB est un cache dédié pour les entrées de la table des pages.*
- ❑ *Souvent, il est organisé par ensembles d'entrées .*
 - ❑ *Une entrée de table des pages (une ligne de la table des pages) doit être accédée pour prélever l'adresse physique correspondant à l'adresse virtuelle.*
 - ❑ *Pour faire une traduction rapide, le TLB conserve les entrées récentes de la table des pages. Ceci évite d'aller en mémoire centrale à chaque fois pour les références futures*

Réponse 8

1. *Puisqu'il y a 128 entrées dans le tampon de traduction, la mémoire requise est de 128 fois la taille de chaque entrée. Chaque entrée doit contenir un bit de validité, un bit de modification, le NPP et le NPV moins le nombre de bits utilisés pour sélectionner l'ensemble dans le tampon.*

Avec des adresses de 32 bits et des pages de 1 ko, les NPV et NPP font 22 bits de long chacun. Pour 128 entrées et un degré d'associativité de 4, on conclut que le tampon contient 32 ensembles, aussi 5 bits du NPV sont utilisés pour sélectionner un ensemble. En conséquence, nous devons simplement stocker 17 bits du NPV pour déterminer si un hit a eu lieu, mais nous avons besoin de tous les 22 bits du NPP pour déterminer l'adresse physique d'une adresse virtuelle. Ceci nous donne un total de 41 bits par entrée du tampon de traduction anticipée.

2. *En utilisant notre équation standard pour le temps d'accès moyen, le temps de traduction moyen est :*

$$(T_{hit} \times P_{hit}) + (T_{miss} \times P_{miss}).$$

Etant donné que $T_{hit} = 0$ dans ce cas, l'équation se simplifie en

$$T_{miss} \times P_{miss} = 200 \text{ cycles} \times 0.1 = 20 \text{ cycles}$$

Exercice 9

Le TLB d'un processeur prend 2,2 ns pour traduire une adresse suite à un cache-hit, L'accès au tableau d'étiquettes du cache prend 2,5 ns. La logique hit/miss prend 1,0 ns, le tableau de données possède un champ d'accès de 3,4 ns et il faut 0,5 ns pour retourner les données au processeur en cas de hit. Quelle est la latence de cache-hit du cache en cas de hit TLB

- i) Si le cache est virtuellement adressé et virtuellement étiqueté ?
- ii) S'il est virtuellement adressé et physiquement étiqueté ?
- iii) S'il est physiquement adressé et physiquement étiqueté
- iv) Avantages et inconvénients de chacune des combinaisons

Réponse 9

- i) Si le cache propose un adressage et un étiquetage virtuels, nous n'avons pas besoin d'opérer une traduction d'adresse lorsqu'un cache-hit survient. En conséquence, la latence du cache-hit correspond au plus long temps entre celui qu'il faut pour déterminer si un hit est survenu et celui pour l'accès au tableau de données, à quoi s'ajoute le temps de retourner les données une fois la détection hit/miss opérée. Le temps d'accès au tableau d'étiquettes est de 2,5 ns et la logique hit/miss prend 1 ns supplémentaire, aussi faut-il 3,5 ns pour déterminer si un hit a eu lieu, ce qui est plus long que le temps d'accès au tableau de données. En conséquence, la latence du cache-hit est de :

$$3,5 \text{ ns} + 0,5 \text{ ns (temps de retourner les données)} = 4 \text{ ns}$$

Réponse 9

- ii) Dans le cas d'un adressage virtuel associé à un étiquetage physique, le temps pour déterminer si un hit a eu lieu est égal à la latence de la logique hit/miss, ajoutée au temps le plus long que représente le temps de recherche dans le tableau d'étiquettes ou le temps de traduction du TLB. Etant donné que le chemin critique du cache est la détection hit/miss, nous pouvons négliger le temps d'accès au tableau de données. Le temps d'accès au tableau d'étiquettes est de 2,5 ns, tandis que le temps de traduction anticipée est de 2,2 ns, aussi la latence dans ce cas sera-t-elle de :
- $$2,5 \text{ ns (temps d'accès au tableau d'étiquettes)} + 1 \text{ ns (détection hit/miss)} + 0,5 \text{ ns (retour de données)} = 4 \text{ ns}$$

Réponse 9

iii) Dans le cas de l'adressage et de l'étiquetage physiques, la traduction d'adresse doit être réalisée avant que l'accès au tableau d'étiquettes puisse être initié, aussi le temps d'accès est-il de :

$$2,2 \text{ ns (TLB)} + 2,5 \text{ ns (tableau d'étiquettes)} + 1 \text{ ns (logique hit/miss)} + 0,5 \text{ ns (retour des données)} = 6,2 \text{ ns}$$

Exercice 10

- ❑ Un disque dur :
 - ❑ est doté d'un unique plateau qui tourne à 15000 tpm
 - ❑ et possède 1024 pistes, chacune constituée de 2048 secteurs.
 - ❑ La tête de lecture/écriture du disque commence à la piste 0 (les pistes sont numérotées de 0 à 1023). Le disque reçoit une requête pour accéder à un secteur aléatoire ou une piste aléatoire.
 - ❑ Le temps de recherche de la tête de lecture du disque est de 1 ms pour 100 pistes à traverser.

- i) Quel est le temps de recherche moyen ?
- ii) Quelle est la latence rotationnelle moyenne ?
- iii) Quel est le temps de transfert pour un secteur ?
- iv) Quel est le temps moyen total pour résoudre une requête ?

Réponse 10

- ❑ Etant donné que la tête de lecture commence à la piste 0, elle devra traverser 0 piste pour gérer une requête ciblant la piste 0, 1 piste pour une requête ciblant la piste 1, et ainsi de suite, jusqu'à 1023 pour la requête ciblant la piste 1023. En moyenne, la tête de lecture devra parcourir la moitié du chemin vers la piste la plus à l'extérieur, soit 511.5 pistes. A 100 pistes/mn, cela donne un temps moyen de recherche de 5.115 ms.
- ❑ A 15 000 tpm, chaque tour prend 4 ms. La latence rotationnelle moyenne est moitié du temps de rotation pour un tour complet, soit 2 ms.
- ❑ Chaque tour prend 4 ms. Il y a 2048 secteurs par piste, aussi chaque secteur prend $4 \text{ ms} / 2048 = 1.95$ microsecondes pour passer sous la tête de lecture/écriture. En conséquence, le temps de transfert est de 1.95 microsecondes.
- ❑ Le temps d'accès moyen est simplement la somme de trois composants, soit 7.717 ms(en arrondissant à trois chiffres après la virgule). Ce résultat ne tient pas compte du temps requis pour transmettre les données au processeur.