

## Fiche de TD N° 4 Architecture des Ordinateurs (AO)

### Exercice 1

Soit un ordinateur à architecture bus système, composé d'un processeur travaillant à une fréquence de 1000 MHz et d'une mémoire centrale avec le nombre de lignes de données égale à 16 et le nombre de lignes d'adresse égale à 32.

- 1- Donner le taux de transfert (théorique) d'un bus de donnée d'un processeur.
- 2- Donner la taille (en Bits) de : RAM (Registre d'Adresse Mémoire), RDM (Registre de Données Mémoire), CO (Compteur Ordinal) et ACC (Accumulateur).
- 3- Donner la capacité de cette mémoire en Méga Octets.
- 4- Donner la plage d'adressage de cette mémoire (en hexadécimale/ base 16).
- 5- Si la taille du mot est égale à 32 bits, donner le nombre de mots adressables et la plus haute adresse possible.

### Exercice 2

Soit une machine dotée d'une mémoire centrale de 1024 K mot de 32 bits.

- 1- Combien de bits, d'octets, de Kilo octets et de Méga octets contient cette mémoire
- 2- Combien de valeur différente peut prendre un mot de cette mémoire ?
- 3- Déterminer la plage d'adressage de cette mémoire (en hexadécimale/ base 16)
- 4- On veut stocker sur cette mémoire des nombres réels où chaque nombre est représenté sur 64 bits. Calculer l'adresse du 9ème nombre sachant que le premier est stocké à l'adresse FF(16)

### Exercice 3

I/ Soit une machine munie d'une mémoire ayant les caractéristiques suivantes :

- L'adresse maximale en hexadécimal pouvant être prise par un mot mémoire est « FFFFFF »
- La capacité mémoire est de 256 Méga bits

1- Calculer la taille du bus d'adresse qui permet d'accéder à cette mémoire.

2- Calculer la taille d'un mot mémoire.

3- Proposer une architecture réduite pour cette mémoire

II/ Supposant d'on a augmenté la taille de cette mémoire et la mémoire est extensible jusqu'à 1 Giga bits.

1- Calculer le nombre total des mots mémoire après l'extension de cette mémoire.

2- Calculer le nombre de bits réservés à l'adressage (après extension)

### Exercice 4

I/ Classez les mémoires suivantes par taille, par rapidité : CD-ROM, Registre d'Instruction, Cache L3. Disques durs, ROM, Cache L1, USB, Cache L2.

II/ Soit une mémoire cache de niveau L1 ayant les caractéristiques suivantes :

- 64 mots par ligne (mots de 4 octets)
- Taille de 512 Ko
- L1 et L2 sont inclusifs

- 4-associatifs

- Remplacement LFU

- Association par poids faible

- Taille de bus d'adresse : 64 bits

1. Combien y-a-t-il de lignes dans cette mémoire cache ?
2. Combien y-a-t-il de blocs associatifs dans cette mémoire cache ?
3. Si la mémoire cache de niveau L2 a une taille de 8 Mo, combien y-a-t-il de blocs de la mémoire cache L2 par bloc de la mémoire cache L1 ?
4. Si un bloc n'est pas présent en cache L1, combien de lignes de L1 aura-t-on parcouru ?

**III/ A** partir des performances du tableau ci-dessous calculer le temps d'exécution moyen d'une instruction pour chaque niveau sachant que durée d'un cycle horloge est T.

Niveau	Temps d'accès succès (ns)	Taux de succès (%)	Pénalité d'échec (Cycles)	Taille
Cache L1	<b>3</b>	<b>80%</b>	<b>5</b>	<b>256 Ko</b>
Cache L2	<b>5</b>	<b>90%</b>	<b>10</b>	<b>512 Ko</b>

### Exercice 5

**I/** Soit un ordinateur à architecture bus système composé d'un processeur travaillant à une fréquence de 800 MHz et d'une mémoire centrale de 256 Méga mot mémoire de deux octets. Son registre d'instruction se compose de 16 bits.

1. Calculer la capacité maximale de la mémoire (en Octets).
2. Trouver la taille (en Bits) de : RAM (Registre d'Adresse Mémoire), RDM (Registre de Données Mémoire), CO (Compteur Ordinal) et ACC (Accumulateur).
3. Calculer le taux de transfert (théorique) d'un bus de donnée d'un processeur.

**II/** Soit une mémoire cache à deux niveaux ayant les caractéristiques suivantes :

Niveaux	Temps d'accès succès (ns)	Taux d'échec	Pénalité d'échec (Cycles)	Taille	Nombre de mots par ligne	Taille d'un mot
Cache L1	<b>2</b>	<b>20%</b>	<b>5</b>	<b>128 Ko</b>	<b>32 mots</b>	<b>4 Octets</b>
Cache L2	<b>4</b>	<b>10%</b>	<b>10</b>	<b>2 Mo</b>	<b>32 mots</b>	<b>4 Octets</b>

1. Calculer la taille d'une ligne ? Combien y-a-t-il de lignes dans cette mémoire cache (L1 et L2) ?
2. Combien y-a-t-il de blocs de la mémoire cache L2 par bloc de la mémoire cache L1 ?
3. Calculer le temps d'exécution moyen d'une instruction pour chaque niveau sachant que durée d'un cycle horloge est T.

### Exercice 6

Un processeur à 2 Go de mémoire principale avec des mots de 1 octet. Pour les différents caches ci-dessous :

1. Cache de 2 Mo à correspondance directe et écriture simultanée avec des lignes de 16 octets
  2. Cache de 4 Mo à correspondance directe, réécriture et lignes de 32 octets
  3. Cache de 4 Mo associatif 4 voies (4 lignes par ensemble), réécriture et lignes de 32 octets
- a. Quelle est la décomposition d'une adresse mémoire ?
  - b. Donner les différentes parties d'une ligne (bloc) de cache (nombre de bits des différentes parties). Quel est le surcoût lié aux bits de contrôle et d'étiquette (par rapport à la partie « données » du cache)

## Fiche de TD N° 4 Architecture des Ordinateurs (AO) (Solution)

### Exercice 1

Soit un ordinateur à architecture bus système, composé d'un processeur travaillant à une fréquence de 1000 MHz et d'une mémoire centrale avec le nombre de lignes de données égale à 16 et le nombre de lignes d'adresse égale à 32.

1. Donner le taux de transfert (théorique) d'un bus de donnée d'un processeur.

**Taux de transfert (Mo/s) = Fréquence (en MHz) × Largeur du bus (en octets)**

Largeur du bus = Taille d'un bus de données = nombre de lignes de données=16 bits = 2 octets

Taux de transfert (Mo/s) =  $1000 * 2 = 2000 \text{ Mo/s}$

2. Donner la taille (en Bits) de : RAM (Registre d'Adresse Mémoire), RDM (Registre de Données Mémoire), CO (Compteur Ordinal) et ACC (Accumulateur).

**RAM = Nbre de ligne d'adresse = 32 bits , RDM = Nbre de ligne de données = 16 bits**

**CO=Nbre de ligne d'adresse =32 bits , ACC= Nbre de ligne de données= 16 bits**

3. Donner la capacité de cette mémoire en Méga Octets.

**Capacité =  $2^{\text{nbre lignes d'@}} * \text{nbre de lignes de données}$ , Capacité =  $2^{32} * 2 = 2^{33}$  Octets=  $2^{13}$  MOcts**

4. Donner la plage d'adressage de cette mémoire (en hexadécimale/ base 16).

**Plage d'adressage est le calcul de l'adresse minimale et l'adresse maximale**

Nombre de lignes d'adresse =32

Adresse minimale = 0000000000... (2) sur 32 bits puisque 0000(2) = 0 (16) donc Adresse minimale = 00000000(16)

Et adresse maximale = 2nbre de lignes d'@ -1 = 1111111 ...1111(2) sur 32 bits Puisque 1111(2) = F (16) Donc Adresse maximale = FFFFFFFF(16)

5. Si la taille du mot est égale à 32 bits, donner le nombre de mots adressables et la plus haute adresse possible.

puisque il y a une augmentation dans la taille du mot mémoire donc le nombre de mots ne change pas ==> Le nombre de mots =  $2^{32}$  mots donc la taille du mot n'a pas d'influence sur le nombre mot.

La plus haute adresse =  $2^{32}-1 = 11111111...11111$  (2) (sur 32 bits) = FFFFFFFF (16) □

Donc là aussi la taille du mot n'a pas d'influence sur la plus haute adresse

### Exercice 2

Soit une machine dotée d'une mémoire centrale de 1024 K mot de 32 bits.

1- Combine de bits, d'octets, de Kilo octets et de Méga octets contient cette mémoire

**Capacité (bits)= Nombre de mots \* taille du mot =  $1024 * 32 = 2^{10} * 2^{10} * 2^5 = 2^{25}$  bits**

**Cap(octs)= $2^{25}/ 2^3 = 2^{22}$ octs||Cap(Kocts)=  $2^{22}/ 2^{10} = 2^{12}$ octs|| Cap(Mocts)=  $2^{12}/ 2^{10} = 2^2$ Mocts**

Combien de valeur différente peut prendre un mot de cette mémoire ?

**Nombre de valeur différente d'un mot mémoire=  $2^{\text{taille du mots}} = 2^{32}$  valeurs**

2- Déterminer la plage d'adressage de cette mémoire (en hexadécimale/ base 16)

Nombre de mots =  $2^{\text{nombre de lignes d'adresses}} = 1024K = 2^{10} * 2^{10} = 2^{20} \rightarrow$  nombre de lignes d'adresse = 20 lignes

adresse minimale (0000000000..) sur 20 bits = 00000(16)  
et adresse maximale ( $2^{20}-1$ ) = 1111111...1111 sur 20 bits = FFFF(16)

3- On veut stocker sur cette mémoire des nombres réels et chaque nombre est représenté sur 64 bits. Calculer l'adresse du 9<sup>ème</sup> nombre sachant que le premier est stocké à l'adresse FF(16)

Rep : chaque nombre dans la mémoire prend 2 mots mémoire (taille du nombre réel / taille du mot = 64/32 = 2 mot)

@n<sup>ème</sup>nombre = @1<sup>er</sup>nombre + (nombre de mots de chaque nombre) \* (n-1)

@1<sup>er</sup>nombre = FF(16) = (15\*16 + 15)<sub>(10)</sub> = 255<sub>(10)</sub>

@9<sup>ème</sup>nombre = @1<sup>er</sup>nombre + (8 x 2) = 255<sub>(10)</sub> + 16<sub>(10)</sub> = 271<sub>(10)</sub> = 10F<sub>(16)</sub>

### Exercice 3

I/ Soit une architecture de Von Neumann munie d'une mémoire ayant les caractéristiques suivantes :

- L'adresse maximale hexadécimal pouvant être pris par un mot mémoire est « FFFFFF »

- La capacité mémoire est de 256 Méga bits

1/ Calculer la taille du bus d'adresse qui permet d'accéder à cette mémoire.

L'adresse maximale hexadécimal = FFFFFF = (1111 1111 1111 1111 1111 1111)<sub>2</sub> écrit sur 24 bits

→ donc la taille du bus d'adresse est 24 bits

3/ Calculer la taille d'un mot mémoire.

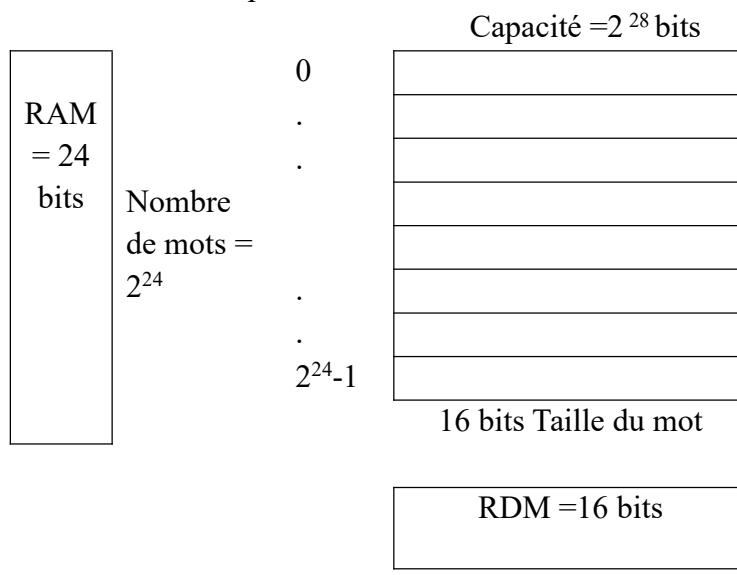
**Capacité= Nombre de mots \* taille du mot → taille du mot = Capacité/ Nombre de mots**

**Nombre de mots =  $2^{\text{nombre de lignes d'adresses}}$**  sachant que nombre de lignes d'adresse= taille du bus d'adresse

Donc Nombre de mots =  $2^{24}$  mots

Taille du mot = 256 M bits /  $2^{24}$  =  $2^8 * 2^{20} / 2^{24} = 2^4$  bits

4/ Proposer une architecture réduite pour cette mémoire



Nombre de lignes d'adresses = 24 lignes = taille RAM

Nombre de lignes de données = 16 lignes = taille RDM

II/ Supposant d'on a augmenté la taille de cette mémoire et la mémoire est extensible jusqu'à 1 Giga bits.

1/ Calculer le nombre total des mots mémoire après l'extension de cette mémoire.

**Nombre de mots mémoire= Taille mémoire après l'extension/ taille du mot mémoire**

$$\text{NMM} = 2^{30} / 2^4 = 2^{26} \text{ mots}$$

2/ Calculer le nombre de bits réservés à l'adressage (après extension)

**Nombre de mots =  $2^{\text{nombre de lignes d'adresse}}$**

$$= 2^{26} \rightarrow 26 \text{ bits d'adressage}$$

#### Exercice 4

I/ Classez les mémoires suivantes par taille, par rapidité :

Par taille : RI < L1 < L2 < L3 < ROM < CD < USB < DD.

Par vitesse : RI > L1 > L2 > L3 > ROM > DD > USB > CD.

Car le type de mémoire de :

- RI est un registre, c'est le plus rapide et de petite taille.
- L1 est une mémoire cache de niveau 1, elle est très rapide mais de petite capacité de stockage.
- L2 est une mémoire cache de niveau 2, elle a une plus grande capacité que L1 et est plus lente.
- ROM est une mémoire centrale à semi-conducteurs, son temps d'accès et sa taille sont beaucoup plus importants que ceux des registres et du cache.
- USB est un mémoire de masse de type mémoire Flash, le temps d'écriture est similaire à celui d'un disque dur mais avec un espace réduit, généralement une capacité de stockage qui varie de 1 à 32 Go.
- Disque dur est un mémoire de masse. Disque dur interne traditionnel HDD de type mémoire magnétique entre 500 Go et 4 To pour une utilisation normale et disque dur interne SSD de type mémoire flash entre 256 Go et 1 To. Pour les disques durs externes sur le site inmac wstore, 500 Go à 8 To
- CD est une mémoire optique de type mémoire de masse, il existe des CD de 700 Mo, 800 Mo ou 900 Mo

II/ Soit une mémoire cache de niveau L1 ayant les caractéristiques suivantes :

- 64 mots par ligne (mots de 4 octets)- Taille de 512 Ko- L1 et L2 sont inclusifs - 4-associatifs -

Remplacement LFU- Association par poids faible- Taille de bus d'adresse : 64 bits

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

**Nombre de lignes = Taille cache / (Taille mot \* nombre de mots par ligne)**

$$= 512 \text{ KOct} / (4 \text{ Oct} * 64) = 2^9 * 2^{10} / 2^2 * 2^6 = 2^{11} = 2048 \text{ lignes}$$

2- Combien y-a-t-il de blocs associatifs dans cette mémoire cache ?

**Nombre de blocs = Nombre de lignes / Nombre de lignes par bloc**

$$= 2^{11} / 4 = 2^{11} / 2^2 = 2^9 = 512 \text{ blocs}$$

3- Si la mémoire cache de niveau L2 a une taille de 8 Mo, combien y-a-t-il de blocs de la mémoire cache L2 par bloc de la mémoire cache L1 ?

**Taille cache L2 / Taille cache L1 = 8 MOct / 512 KOct =  $2^{23} / 2^{19} = 2^4 = 32$**

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

**4 lignes**

III/ A partir des performances du tableau ci-dessous calculer le temps d'exécution moyen d'une instruction pour chaque niveau sachant que durée d'un cycle horloge est T.

Niveau	Temps d'accès succès (ns)	Taux de succès (%)	Pénalité d'échec (Cycles)	Taille
Cache L1	3	80%	5	256 Ko
Cache L2	5	90%	10	512 Ko

**temps d'accès mémoire moyen = temps d'accès succès + taux d'échec x pénalité d'échec**

**taux d'échec = 1 - taux de succès**

**temps d'accès mémoire moyen Cache L1 =  $3 + (1-80\%) \times 5 = 4T$**

**temps d'accès mémoire moyen Cache L2 =  $5 + (1-90\%) \times 10 = 6T$**

### Exercice 5

I/ Soit un ordinateur à architecture bus système composé d'un processeur travaillant à une fréquence de 800 MHz et d'une mémoire centrale de 256 Méga mot mémoire de deux octets. Son registre d'instruction se compose de 16 bits.

1- Calculer la capacité maximale de la mémoire (en Octets).

**Capacité=Nombre de mots\*taille du mot=256M\*2oct=512 MOct=2<sup>9</sup>\*2<sup>20</sup> Octs=2<sup>29</sup> Octs**

2- Trouver la taille (en Bits) de : RAM (Registre d'Adresse Mémoire), RDM (Registre de Données Mémoire), CO (Compteur Ordinal) et ACC (Accumulateur).

**RAM= Nbre de ligne d'adresse**

**→ Nombre de mots=2<sup>nombre de lignes d'adresse</sup> = 256 Méga = 2<sup>8</sup>\*2<sup>20</sup> = 2<sup>28</sup> donc R@M = 28 bits**

**RDM =taille mot = 2 octets = 16 bits**

**CO=Nbre de ligne d'adresse= 28 bits**

**ACC= Nbre de ligne de données = 16 bits**

3- Calculer le taux de transfert (théorique) d'un bus de donnée d'un processeur.

**Taux de transfert (Mo/s) = Fréquence (en MHz) × Largeur du bus (en octets)**

**Largeur du bus = Taille d'un bus de données = taille d'un mot= 2 octets**

**Taux de transfert (Mo/s) = 800 \* 2 = 1600 Mo/s**

**Ou Largeur du bus = 2 octets = 16 bits**

**Taux de transfert (Mbits/s) = 800 \*16 = 12800 Mbits/s**

II/ Soit une mémoire cache à deux niveaux ayant les caractéristiques suivantes :

Niveaux	Temps d'accès succès (ns)	Taux d'échec	Pénalité d'échec (Cycles)	Taille	Nombre de mots par ligne	Taille d'un mot
Cache L1	2	20%	5	128 Ko	32 mots	4 Octets
Cache L2	4	10%	10	2 Mo	32 mots	4 Octets

1- Calculer la taille d'une ligne ? Combien y-a-t-il de lignes dans cette mémoire cache (L1 et L2) ?

**Taille d'une ligne (L1 et L2) = Taille mot\* Nombre de mots par ligne  
= 32 \* 4 = 2<sup>5</sup> \* 2<sup>2</sup> = 2<sup>7</sup> Octets ou 2<sup>10</sup> bits**

**Nombre de lignes = Taille cache / Taille d'une ligne**

**Nombre de lignes (L1) = 128 KOct / 2<sup>7</sup> Oct = 2<sup>7</sup>\*2<sup>10</sup>/2<sup>7</sup> = 2<sup>10</sup> Lignes**

**Nombre de lignes (L2) = 2 MOct / 2<sup>7</sup> Oct = 2\* 2<sup>20</sup>/2<sup>7</sup> = 2<sup>14</sup> Lignes**

2- Combien y-a-t-il de blocs de la mémoire cache L2 par bloc de la mémoire cache L1 ?

$$\text{Taille cache L2 / Taille cache L1} = 2 \text{ MOct}/128 \text{ KOct} = 2^{21}/2^{17} = 2^4 = 16$$

3- Calculer le temps d'exécution moyen d'une instruction pour chaque niveau sachant que durée d'un cycle horloge est T.

$$\text{Temps d'accès mémoire moyen} = \text{Temps d'accès succès} + \text{Taux d'échec} \times \text{Pénalité d'échec}$$

$$\text{Temps d'accès mémoire moyen Cache L1} = 2 + (20\%) \times 5 = 2 + (0.2) \times 5 = 3T$$

$$\text{Temps d'accès mémoire moyen Cache L2} = 4 + (10\%) \times 10 = 4 + (0.1) \times 10 = 5T$$

## Exercice 6

Un processeur à 2 Go de mémoire principale. Pour les différents caches ci-dessous :

1. Cache de 2 Mo à correspondance directe et écriture simultanée avec des blocs de 16 octets
2. Cache de 4 Mo à correspondance directe, réécriture et blocs de 32 octets
3. Cache de 4 Mo associatif 4 voies (4 blocs par ensemble), réécriture et blocs de 32 octets

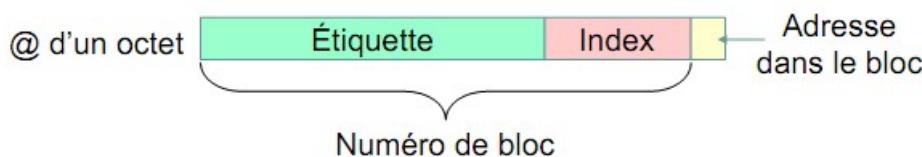
a. Quelle est la décomposition d'une adresse mémoire (nombre de bits des différentes parties) ?

Rappels :

- L'index indique dans quelle ligne va la donnée
- L'adresse dans le bloc (ou offset) indique où est la donnée dans la ligne
- L'étiquette sert à comparer si la donnée que l'on cherche est bien celle qui est stockée actuellement dans le cache ou non

+ Rappels du cours :

Format d'adresse mémoire cache (pour des mots de 1 octet) directe et associatif par ensemble (mixte entre direct et totalement associatif) est la même



Car dans placement direct → Le champ « numéro de bloc » est scindé en deux parties : l'étiquette et l'index.

Et dans cache associatif par ensemble → Le choix d'un ensemble est associatif (c.a.d. Le numéro de bloc est utilisé comme étiquette) et chaque ensemble est géré comme dans le cache à correspondance direct.

### Solution :

La taille d'adressage d'un mot :

La taille d'un mot = 1 oct = 8 bits

Nbre de mots = capacité/taille d'un mot =  $2^{30}/1\text{oct} = 2^{30}$

Nombre de mots =  $2^{\text{taille d'adressing d'un mot}} = 2^{31} \rightarrow$  donc la taille d'adresse d'un mot = **31 bits**

1. Cache de 2 Mo =  $2 \times 2^{20} = 2^{21}$  octs et une ligne = 16 octs =  $2^4$  octs,

➤ L'index étant la partie qui permet de savoir à quelle ligne est la donnée dans le cache  
Nbre de lignes = taille cache / taille ligne =  $2^{21}/2^4 = 2^{17}$  lignes → il faut **17 bits d'index**

➤ L'adresse dans bloc pour savoir à quel endroit est la donnée dans la ligne  
Nbre de mot dans une ligne = taille ligne / taille d'un mot =  $2^4$  octs / 1 oct =  $2^4$  mots par ligne  
→ il faut **4 bits d'adresse dans bloc**

➤ Enfin, l'étiquette = 31-17- 4= **10 bits**

2. Même explication que la précédente

Cache de 4 Mo=  $4 \times 2^{20} = 2^{22}$  octs, puisqu'une ligne = 32 octs =  $2^5$  octs,

- Nbre de lignes =  $2^{22}/ 2^5 = 2^{17}$  lignes ➔ **17 bits d'index**
- Nbre de mot dans une ligne =  $2^5$  octs/ 1 oct =  $2^5$  mots par ligne ➔ il faut **5 bits d'adresse dans bloc**
- **Etiquette = 31-17- 5= 9 bits**

3. Même explication que précédemment, sauf pour l'index.

- Comme le cache est associatif par 4 voies ➔ donc un  $4=2^2$  lignes ont la même adresse  
Et nbre de lignes =  $2^{22}/ 2^5 = 2^{17}$  lignes, les  $2^{17}$  lignes sont par groupes de  $2^2$ , soit  $2^{17}/ 2^2= 2^{15}$  groupes ➔ donc **15 bits d'index**
- Nbre de mot dans une ligne =  $2^5$  mots par ligne ➔ il faut **5 bits d'adresse dans bloc**
- **Etiquette = 31-15- 5= 11 bits**

b. Donner les différentes parties d'une ligne (bloc) de cache (nombre de bits des différentes parties). Quel est le surcoût lié aux bits de contrôle et d'étiquette (par rapport à la partie « données » du cache)

Rappels :

- L'étiquette est la même que celle de la question précédente.
- Il y a 1 ou 2 bits de contrôles : 1 bits de validité (la donnée est-elle utilisée actuellement ?) et 1 bit de modification en cas de réécriture (la donnée a-t-elle été modifiée sans avoir été modifiée aussi en mémoire principale ?)
- Les lignes de données sont celle de la question précédente

1. Etiquette = 10 bits

Contrôle = 1 bit (écriture = 1 bit de validité)

Données = taille bloc = taille ligne =  $16 * 8 = 2^7 = 128$  bits

**Surcoût = (bits contrôle + bits d'étiquette) / (bits contrôle + bits d'étiquette + bits données)**  
**= (1+10)/ (1+10+128) = 11/139 = 7,91%**

2. Etiquette = 9 bits

Contrôle = 2 bit (réécriture = 1 bit validité + 1 bit modifié)

Données =  $32 * 8 = 2^8 = 256$  bits

**Surcoût = (2+9)/ (2+9+256) = 11/267= 4,12%**

3. Etiquette = 11 bits

Contrôle = 2 bit (réécriture = 1 bit validité + 1 bit modifié)

Données =  $32 * 8 = 2^8 = 256$  bits

**Surcoût = (2+11)/ (2+11+256) = 13/269= 4,83%**