

Exercice 0 : QCM

Choisissez une et une seule réponse pour les questions suivantes.

Dans le processeur du cours, le registre MAR :

- Stocke des adresses
- Stocke des instructions
- Est relié au bus de données de la mémoire
- N'intervient que dans la phase FETCH
- Ne fait rien qu'à embêter les autres

Dans un chronogramme de bus asynchrone :

- Il n'y a pas de dispositif maître
- Le signal WAIT est géré par le maître
- Le signal MSYN est toujours active avant SSYN
- On voit apparaître le signal de l'horloge du bus
- On voit des images subliminales de publicité pour la RATP

Quand le clavier émet une demande d'interruption :

- C'est qu'elle souhaite signaler au processeur que des données sont prêtes
- Cette interruption est prioritaire sur toutes les autres
- Le processeur doit la traiter immédiatement
- Il devra en émettre une deuxième à la demande du processeur
- Il doit crier très fort pour que le CPU l'entende

Dans une machine petit-boutiste :

- On échange l'ordre des bits dans les octets
- Les données de type long sont stockées à l'envers
- La numérotation des octets dans les mots mémoire est l'inverse de cette numérotation dans une machine grand-boutiste
- On échange l'ordre des mots dans la mémoire
- C'est pareil que dans une machine moyen-boutiste, mais en mieux, et réciproquement.

Exercice 1 : Processeur

a) Quelle est la signification des flags Z, N, C et V associés à l'UAL ?

Z : 1 si le résultat est 0, 0 sinon.
 N (negative) : 1 si le résultat est négatif, 0 sinon.
 C (carry) : 1 si il y a une retenue, 0 sinon.
 V (overflow) : 1 si il y a dépassement de capacité, 0 sinon.

b) soit le jeu d'instructions suivant pour le microprocesseur du cours :

- SET# N : met la constante N dans l'accumulateur D0
- LDA N : prend la valeur située en mémoire à l'adresse N et la met dans D0
- ADD N : prend la valeur située en mémoire à l'adresse N et l'additionne à la valeur stockée dans D0. Le résultat de l'addition est écrit dans D0
- SUB N : prend la valeur située en mémoire à l'adresse N et la soustrait à la valeur stockée dans D0. Le résultat de l'addition est écrit dans D0
- ADD# N : ajoute la constante N à la valeur stockée dans D0. Le résultat de l'addition est stocké dans D0
- CMP N : effectue la soustraction entre l'accumulateur D0 et la valeur située en mémoire à l'adresse N, mais ne stocke pas le résultat dans D0 : utilisée pour mettre à jour les flags Z, N, C et V
- CMP# N : effectue la soustraction entre l'accumulateur D0 et la constante N, mais ne stocke pas le résultat dans D0 : utilisée pour mettre à jour les flags Z, N, C et V
- STA N : prend la valeur stockée dans D0 et l'écrit en mémoire à l'adresse N
- BRA N : effectue un saut de N instructions (N peut être positif ou négatif) pour le registre PC
- BRZ N : effectue un saut de N instructions si le flag Z vaut 1. Par exemple BRZ +4 effectuera un saut de 4 instructions si le flag Z vaut 1, et ne fait pas le saut si le flag Z vaut 0.

A partir de ces instructions de base, écrivez des programmes permettant de réaliser les instructions du langage C suivantes, en sachant que chaque variable est associée à une adresse qui porte le même nom que la variable, par exemple la variable a sera à l'adresse A.

b1) $a = 4 + \{2 * b\};$

```

LDA B
ADD 2
ADD #4
STA A
STOP
  
```

```

b2) IF (A==B)
    |
    |   a = 0;
    |
    |
  
```

<pre> LDA A CMP B BRZ +2 STOP SVA A STOP </pre>	<pre> LDA A CMP B BRZ +1 STOP SET #0 SVA A STOP </pre>
---	--

1

```

b3) x = 5;
while (x > 0)
    |
    |   i = i+2;
    |   x = x-1;
    |
  
```

<pre> SET #5 SVA X LDA X CMP #0 BRN 3 LDA Z ADD #2 SVA Z LDA X ADD #-1 </pre>	<pre> STA X BRA -9 STOP </pre>
---	--

1

b4) Comment, à votre avis, peut-on traiter un appel de fonction avec cette architecture de processeur ? (vous avez le droit d'ajouter un registre qui stocke l'adresse actuelle de la zone de pile par exemple).

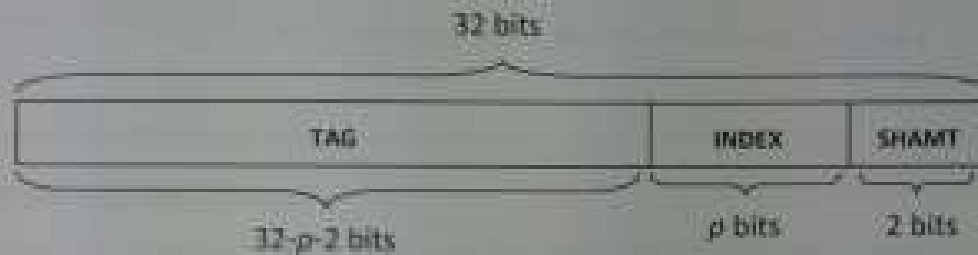
Avec cette architecture de processeur l'appel de fonction doit se faire à l'aide de BRA (sans conditionnel) qui exécutera le code de la fonction parce qu'elle suit à la suite du programme.
 On pourrait aussi intégrer chaque fonction dans le code où elle est appelée puisque une fonction est un morceau de code.
 Comment ?



Exercice 2 : Mémoire & Cache Mémoire

On considère un cache à correspondance directe à 256 emplacements.

On rappelle que dans ce cas, toute adresse 32 bits émise par le processeur sera découpée de la manière suivante :

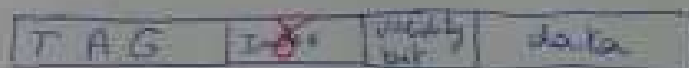


- a) Quelle est la taille p de l'index utilisé pour utiliser correctement le cache à 256 emplacements ? Quel est le rôle de ces bits d'index ?

$256 = 2^8$ donc 8 bits d'index.
 Ces bits d'index permettent d'aller chercher la ligne de cache correspondante.

0,5

- b) Que trouve-t-on exactement au sein d'un emplacement du cache (détailler les 4 parties de chaque emplacement, faites un schéma éventuellement) ?



Le tag et l'index constituent l'adresse.
 le validity bit est à 1 lorsqu'il y a une donnée et 0 quand il n'y a pas.
 la data est la donnée.

0,5

c) Expliquez le comportement du cache et de la mémoire lors :

- D'une requête de lecture

lors d'une lecture on interroge le cache (indépendamment, puis éventuellement) pour vérifier si la donnée est située dans le cache.
Si oui \rightarrow HIT on lit depuis le cache.
Sinon \rightarrow MISS on va chercher l'information en mémoire.

- D'une requête d'écriture

lors d'une écriture, deux situations sont possibles.
 \rightarrow write through : on modifie dans le cache et cela est aussi modifié dans la mémoire (si elle n'y trouve rien).
 \rightarrow write back : on modifie dans le cache et cela sera mis en mémoire lorsque le cache sera vide (HIT ou MISS).

0,5

d) On indique que le temps moyen t_{moy} d'accès à une donnée est donné par deux formules:

$$1. t_{\text{moy}} = h \cdot t_c + (1-h)(t_c + t_m)$$

$$2. t_{\text{moy}} = t_c + (1-h)t_m$$

Quelle est la bonne formule ? Justifiez votre réponse.

$$1. t_{\text{moy}} = h \cdot t_c + (1-h)(t_c + t_m) = h t_c + t_c (1-h) + t_m (1-h) = t_c + (1-h)t_m$$

$$2. t_{\text{moy}} = t_c + (1-h)t_m$$

Donc :

les deux formules sont bonnes avec h la probabilité de trouver l'information dans le cache.

1

e) indiquez les effets sur le cache, initialement vide, des requêtes suivantes (pour simplifier, les seize bits de poids fort de l'adresse seront considérés comme égaux à 0 et ne seront pas notés dans les adresses fournies en hexadécimal). Pour chaque requête, vous indiquerez : le numéro de l'emplacement de cache concerné, la valeur du tag (en binaire), le succès ou l'échec, ainsi que l'état de l'emplacement après la requête.

Lecture à l'@ B9B4 puis Lecture à l'@ 4F39 puis Lecture à l'@ B9B4 puis Lecture à l'@ B9B6 puis Lecture à l'@ 7A3A

<p>• $B9B4_{16} = 0011100110110100_2$ 1011100110110100 Tag : 00111001 → <u>MISS</u> (car le bit de sélection de cache est égal à 0) (emplacement 00 dans le cache) ↓ $4F39_{16} = 0100111100111001_2$ Tag : 01001111 → <u>MISS</u> (car le bit de sélection de cache est égal à 1) ↓ $B9B4_{16} = 1011100110110100_2$ → <u>HIT</u> (car le bit de sélection de cache est égal à 0) emplacement 00</p>	<p>• $B9B6_{16} = 1011100110110110_2$ Tag : 00111001 Emplacement 00 (même emplacement que B9B4 car même tag) → <u>HIT</u> ↓ $7A3A_{16} = 0111101000111010_2$ Tag : 01111010 Emplacement 01 dans le cache → <u>MISS</u> (car le bit de sélection de cache est égal à 1)</p>
---	--

3

Quelle remarque pouvez-vous faire sur cette suite de requêtes ?

On remarque qu'une adresse différente peut renvoyer à un même emplacement dans le cache (exemple B9B4 et B9B6).

1

Exercice 3 : Chronogramme de bus

On souhaite schématiser une opération (ou transaction) de lecture de données en mémoire par l'intermédiaire d'un bus synchrone.

- a) Indiquez quel est le dispositif maître et quel est le dispositif esclave pour cette transaction.

le processeur est le dispositif maître et la mémoire est l'esclave.

1

- b) Parmi les dispositifs maître et esclave, indiquez lequel contrôle chacun des signaux suivants (le signal d'horloge est contrôlé par l'horloge de bus, donc ni par le maître, ni par l'esclave)

- Bus d'adresse
- Bus de données
- \overline{RD} (signal de lecture)
- \overline{WAIT}
- \overline{MREQ}

maître	
maître	
à mémoire	
maître	
esclave	

- c) Combien de cycles d'attente seront nécessaires pour que la mémoire ait réalisé la lecture qui lui est demandée ?

ça dépend de la fréquence d'horloge.
Il faut 3 cycles d'attente.

d) Etablissez le chronogramme complet de cette transaction de bus (vous pouvez utiliser la page en mode "paysage")

