

**ESCRIT**

Le Jeudi 12 Avril 2012

Durée : 2 heures

35-

Sans documents, sans calculatrice

Vous répondrez aux questions dans le temps imparti. Toute copie ne sera prise en compte.

NOM

SAYADA

PRENOM

Eyad

GROUPE

B

### 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  $\overline{\text{WAIT}}$  est géré par le maître
- Le signal  $\overline{\text{MSYN}}$  est toujours activé avant  $\overline{\text{SSYN}}$
- 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 : zero  
N : negative  
C : carry  
V : ~~validité~~

0,5

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
- BRcc N : effectue un saut de N instructions si le flag cc 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)  $a = 4 + (2 * b);$

SET #4  
ADD B  
ADD B  
STA A

-1

```
b2) if (a==b)
{
    a = 0;
}
```

```
LDA A
CMP B
BRZ #2 // On saute la mise à 0 de a si a-b=0
SET #0
STA A
```

```
b3) x = 5;
while (x > 0)
{
    z = z+2;
    x = x-1;
}
```

<pre>SET #5 // On affecte 5 à x STA X  CMP #0 // On compare à 0 BRN #7 // Si ... négatif SET #2 ADD Z SET #(-1) ADD X</pre>	<pre>LDA (-X) CMP #0 BRN -6 // Si X ... négatif  <del>SET #2 ADD Z SET #(-1) ADD X LDA X CMP #0 BRZ -5</del></pre>
---	--

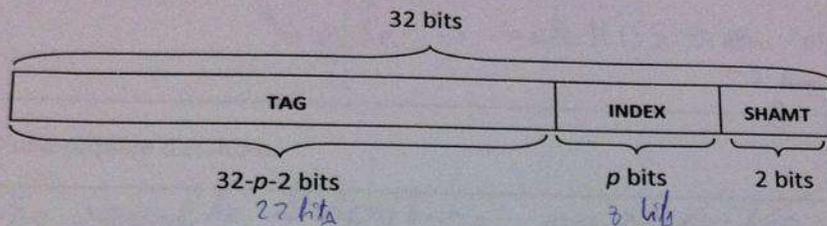
b4) Comment, à votre avis, peut-on traiter un appel de fonction avec cette architecture de processeur ? (vous avez le droit d'ajouter un registre qui stocke l'adresse actuelle de la zone de pile par exemple).

Par un appel de fonction, on stocke l'adresse actuelle *de quoi?* dans un registre.  
 Puis, on effectue les calculs *actions* de notre fonction, on retourne la valeur si besoin est dans le registre D0.  
 Enfin lorsque l'on revient au programme principal il suffit de chercher la valeur retournée dans ce registre si besoin.

**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 ?

$p = \log_2(256) = \log_2(2^8) = 8 \text{ bits}$   
Le rôle de l'index est de stocker l'adresse.

0,5

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

On trouve 4 parties dans ce cache :

- ~~index~~ : permet de stocker l'adresse.
- TAG : Il précise car l'index ne donne qu'une partie de l'adresse TAG + index = @
- data : la donnée stockée
- bit de validité (V) : un bit de validité, indiquant ou non la data du cache (pour la lecture)

0,5

- c) Expliquez le comportement du cache et de la mémoire lors :
- D'une requête de lecture

Demande de lecture → succès → cache HIT → lecture  
 (prévu et bit de validité)

↓ échec → cache MISS → on recherche dans la RAM  
 ⇒ une copie est faite dans le cache.

0,5

- D'une requête d'écriture

Une requête d'écriture va respecter les principes de localité temporelle (nombre accès mémoire) et spatiale (proche des autres emplacement).

Elle va toujours être accompagnée d'une écriture dans la mémoire (RAM)

0,5

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

1.  $t_{glob} = h.t_c + (1-h)(t_c + t_m)$

2.  $t_{glob} = t_c + (1-h).t_m$

Quelle est la bonne formule ? Justifiez votre réponse.

La bonne formule est :  $t_{glob} = t_c + (1-h)t_m$ .

⇒ temps d'accès au cache plus court et fait de toute manière.

Probabilité de  $(1-h)$  qu'il y ait accès dans la mémoire.

- 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

Lecture @ B9B4 : Si adresse existe et est  
 valide à 1, cache HIT, copie dans le cache.  
 À chaque lecture on recopiera dans le  
 cache la donnée, avec son TAG, INDEX et SHART.

Lecture @ 4F39 :

Lecture @ B9B4 : → accès direct dans  
 le cache car stockée auparavant.

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

Lorsque l'on appelle plusieurs fois une donnée <sup>DS</sup>  
 dans la RAM, elle est stockée dans le cache  
 et ainsi on pourra y accéder plus vite (localité  
 temporelle) ⇒ exemple avec @ B9B4

### 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.

maître : microprocesseur  
esclave : mémoire

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	---
esclave -	---
maître -	---
esclave -	---
maître -	---

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

ici, on a eu besoin de 4 cycles.

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

