

DEVOIR ECRIT

Le Vendredi 21 Mai 2010

Durée : 2 heures

Sans documents, sans calculatrices

Exercice 0 : QCM

Choisissez une et une seule réponse pour les questions suivantes, et n'oubliez pas de rendre cette page avec votre copie.

Dans le processeur du cours, le registre PC :

- Stocke des adresses
- Stocke des instructions
- Est un opérande
- N'intervient que dans la phase Fetch
- Ne fait rien qu'à embêter les autres

Dans un chronogramme de bus synchrone :

- Le signal WAIT est géré par le maître
- Il n'y a pas de dispositif maître
- Il n'y a pas de signal SSYN
- Le signal MYSYN est toujours activé avant SSYN
- On doit dessiner un smiley quand la transaction est terminée

Quand le disque dur émet une demande d'interruption :

- Cette interruption est prioritaire sur toutes les autres
- Cette interruption est moins prioritaire que toutes les autres
- C'est qu'il souhaite signaler au processeur que des données sont prêtes
- Il devra en émettre une deuxième à la demande du processeur
- Il doit l'envoyer en recommandé avec accusé de réception

Dans un ordinateur de type multiordinateurs :

- La mémoire est partagée
- La mémoire est distribuée
- Les performances augmentent linéairement en fonction du nombre de processeurs
- Il n'y a pas besoin de bus
- Tous les affichages se font en noir et blanc

Exercice 1 : Processeur

a) Indiquez le rôle des éléments suivants dans le microprocesseur CISC vu en cours:

- PC
- IR
- MAR
- Séquenceur (unité de contrôle)
- UAL
- MBR
- DO

b) Ecrivez un petit programme en assembleur de ce processeur qui réalise l'instruction C suivante (x, y et z sont des variables définies stockées aux adresses X, Y et Z)

$x = y+z+4;$

Rappel de quelques instructions de base pour écrire ce programme

- SET# N : met la constante N dans l'accumulateur DO
- LDA N : prend la valeur située en mémoire à l'adresse N et la met dans DO
- ADD N : prend la valeur située en mémoire à l'adresse N et l'additionne à la valeur stockée dans DO. Le résultat de l'addition est écrit dans DO
- ADD# N : ajoute la constante N à la valeur stockée dans DO. Le résultat de l'addition est stocké dans DO
- STA N : prend la valeur stockée dans DO et l'écrit en mémoire à l'adresse N
- BRA N : effectue un saut de N adresses (N peut être positif ou négatif) pour le registre PC

Exercice 2 : Mémoire & Cache Mémoire

On considère un cache à correspondance directe à 64 emplacements.

- a) On considère que le processeur émet des adresses sur 32 bits, et qu'à chaque adresse correspond un octet. On rappelle également qu'un mot mémoire a une taille de 32 bits. Rappelez comment il est possible d'accéder à un octet individuellement au sein d'un mot mémoire de 32 bits.
- b) Quelle est la taille de l'index utilisé pour utiliser correctement le cache à 64 emplacements ? Quel est le rôle de ces bits d'index ?

c) Que trouve-t-on exactement au sein d'un emplacement du cache (détaillez les 4 parties de chaque emplacement) ?

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

- D'une requête de lecture
- D'une requête d'écriture

e) 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.

Exercice 3 : Chronogramme de bus

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

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

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 la maître, ni par l'esclave)

- Bus d'adresse
- Bus de données
- \overline{WR} (signal d'écriture)
- \overline{MREQ}
- \overline{SSYN}
- \overline{MSYN}

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

d) Établissez le chronogramme complet de cette transaction de bus.

19

Exercice 1

PC: program counter. registre qui stocke l'adresse de la prochaine instruction du programme. 0,5

IR: Instruction register: stocke l'instruction à réaliser: op-code + opérande. 0,5

MAR: memory address register: stocke l'adresse de la prochaine du prochain accès à la mémoire (écriture ou lecture). 0,5

Sequencem: distribue des signaux pour hiérarchiser l'exécution des instructions (par exemple). 0,5

UAL: unité arithmétique et logique: responsable des calculs +, -, ET, OU... 0,5

MBR: memory buffer register: stocke la donnée qu'on veut écrire en mémoire ou bien la donnée lue et mémorisée. 0,5

DR: accumulateur ou registre de donnée, stocke par défaut les résultats du calcul de l'UAL. 0,5

b) LDA Y
ADD Z
ADD #4
STA X

1,5

Exercice 2

a) A chacune des 2^{32} adresses correspond 1 octet

Un mot mémoire occupe 32 bits soit 4 octets.

Pour accéder à un octet en particulier du mot mémoire il suffit d'envoyer une requête (écriture ou lecture) à l'adresse correspondante. α

b) La taille de l'index est de 6 bits car il faut un index différent de 0 à 63 (2^6)

Le n° de l'index est de permettre au processeur d'accéder directement à la ligne du cache correspondante à sa requête
par exemple si le processeur veut lire la donnée à l'adresse

110100111...111 il regardera dans le cache
← 32 bits →

à la ligne d'index 110100.

c) Dans un emplacement on a :

l'index : permet de accéder directement à un emplacement.

le TAG : permet de vérifier si l'adresse complète est correcte (dans notre cas il est constitué de $32 - 6 = 26$ bits)

les data : permet de stocker la donnée contenue à l'adresse (index + TAG) de la mémoire.

un bit de validité : permet de savoir si la donnée de l'emplacement est valide ou non

d) Requête de lecture.

Le ~~cache~~ processeur regarde dans le cache si l'index correspond à la donnée qu'il veut lire, puis le TAG est vérifié.

Si c'est bon* alors on a un cache HIT (la donnée est trouvée) le processeur y accède directement à partir du cache sans aller dans la mémoire.

Si le TAG ne correspond pas,** on a un cache MISS, le processeur a accès à la mémoire pour chercher la donnée (qui est également copiée dans le cache).

* et si le bit de validité est à 1 (donnée valide)

** ou si le bit de validité est à 0 (donnée ~~pas~~ invalide).

Requête d'écriture.

Il existe deux modèles :

le write-through : la donnée est écrite à la fois dans le cache et dans la mémoire.

le write-back : la donnée est écrite dans la mémoire que lorsque le cache est libéré (on vérifie le dirty bit pour savoir si la donnée a été modifiée ou non).

e) la bonne formule est : $t_{glob} = t_c + (1-h) \times t_m$

car le processeur a toujours accès au cache en premier (t_c : temps d'accès au cache) et si il ne trouve pas la donnée (probabilité $1-h$ avec h la proba de trouver la donnée) alors il a ensuite recours à la mémoire (t_m temps d'accès à la mémoire).

OK

Exercice

a) le dispositif maître est le processeur
le dispositif esclave est la mémoire. 1

b) Bus d'adresse : maître /
Bus de données : esclave /

\overline{WR} : on souhaite réaliser une lecture donc le signal \overline{WR} n'est pas émis. (s'il s'agit d'une erreur alors: \overline{RD} maître)

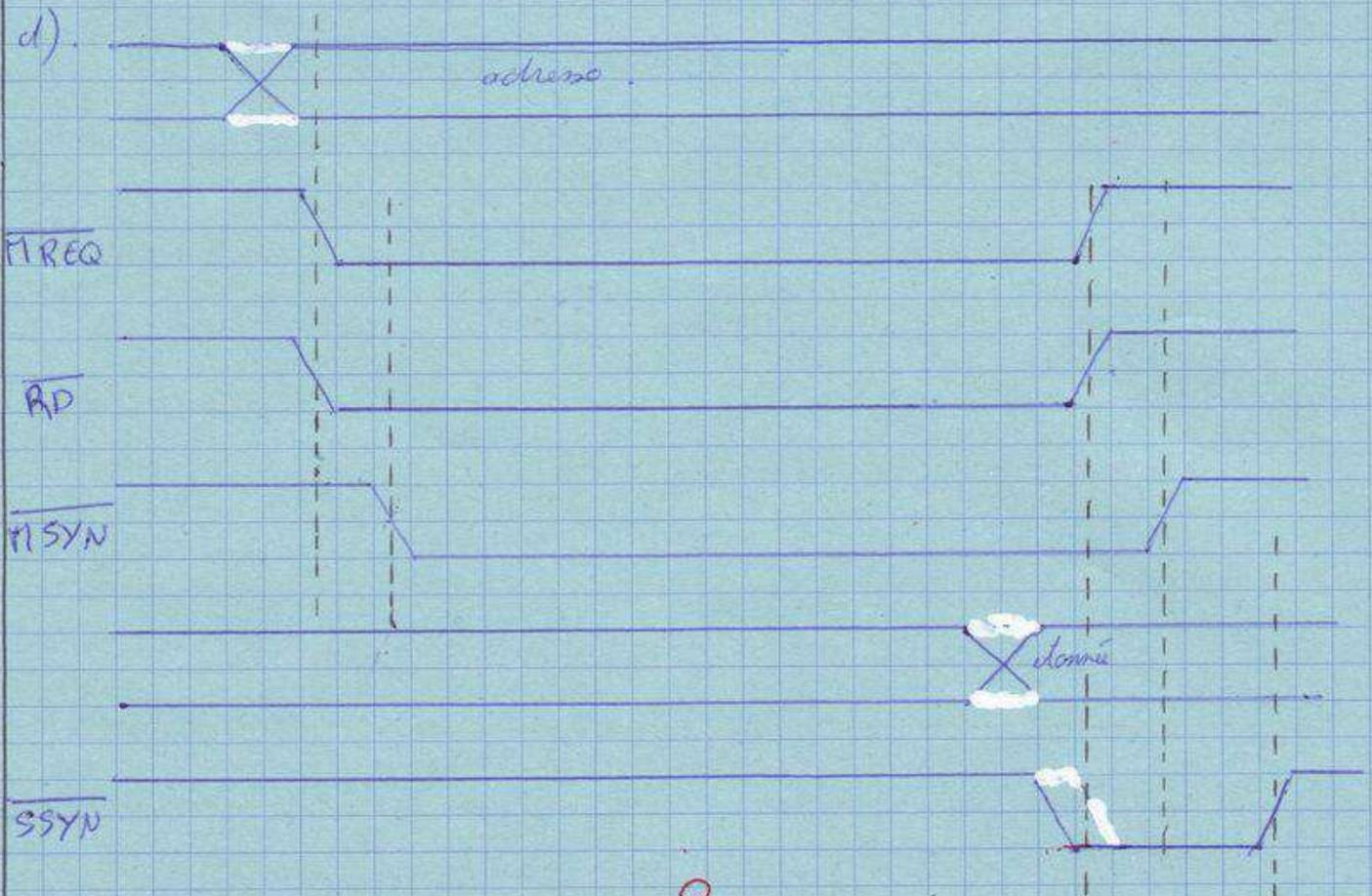
\overline{MREQ} : maître /

\overline{SSYN} : esclave /

\overline{MSYN} : maître /

2

c) Il s'agit d'un bus asynchrone par conséquent il n'y a pas de cycles d'attente à proprement parler. 1



2