

Examen Final

IFT-1215

20 avril 2015

Directives

- Documentation autorisée : une page manuscrite recto.
- Pas de calculatrice, téléphone, ou autre appareil électronique autorisé.
- Répondre sur le questionnaire dans l'espace libre qui suit chaque question.
Utiliser le verso de la page si nécessaire.
- Vous ne devez faire que 5 des 6 questions (à choix).
- Chaque question vaut 5 points pour un total maximum de 25 points.
- Les questions ne sont pas placées par ordre de difficulté.

0 Nom et prénom (1 point de bonus)

Écrire son nom et prénom et son code permanent en haut de chaque page.

1 Circuits logiques

Mettre au point un *compteur à saturation* sur 2 bits.

Il s'agit d'un circuit séquentiel avec un bit I en entrée et deux bits C_0 et C_1 en sortie (plus une horloge, bien sûr). Les deux bits C_0 et C_1 représentent un nombre entier C entre 0 et 3.

À chaque cycle, si I est 0, alors C est décrémenté de 1, et si I est 1, alors C est incrémenté de 1. Si C vaut déjà 3 lorsqu'on veut l'incrémenter, il reste simplement à 3 (ce qui justifie le nom de *saturation*), et de même Si C vaut déjà 0 lorsqu'on veut le décrémenter, il reste simplement à 0. L'état initial est sans importance.

Par exemple. Si I vaut successivement : 0 0 0 0 1 1 1 1 0
alors C_1C_0 devrait rendre les valeurs : ?? 0? 00 00 01 10 11 11 10
où ? signifie que cela dépend de l'état initial.

1. De combien de registres (bascules D) a-t-on besoin ?
2. Donner la table de vérité de la partie combinatoire du circuit.
3. Donner une fonction booléenne qui décrit la partie combinatoire du circuit.
4. Dessiner une implantation possible du circuit avec ses portes logiques et ses registres.

2 LMC

Rappels :

Encodage des instructions LMC				Décomposition de l'instruction LDA	
HLT	000	BR	6xx	PC	→ MAR
ADD	1xx	BRZ	7xx	MDR	→ IR
SUB	2xx	BRP	8xx	IR[adr]	→ MAR
STO	3xx	IN	901	MDR	→ Acc
LDA	5xx	OUT	902	PC + 1	→ PC

Soit l'état suivant du LMC :

... 18=518, 19=118, 20=219, ..., PC=18, Acc=0

1. Montrer le contenu des registres IR, PC, MAR, MDR, et Acc après avoir exécuté les instructions aux adresses 18 et 19.
2. Montrer le contenu de chacun de ces registres à chacun des pas de la décomposition de l'instruction suivante (i.e. à l'adresse 20).

3 Mémoires caches

Soit une mémoire cache à correspondance directe. Ce cache prend des adresses de 24bit (pour une mémoire où chaque adresse contient un *byte* (B), comme d'habitude) et les divise en trois parties : un *index* qui indique la ligne de cache correspondante, un *offset* qui indique le byte dans cette ligne, et une *étiquette* qui contient le reste. La capacité du cache est de 8KB, et la longueur de chaque ligne est de 16B.

1. Combien de *lignes* peuvent être gardées dans ce cache ?
2. Quelle est la taille de chaque *index*, *offset*, et *étiquette* ?
3. Laquelle des trois parties contient les bits les plus significatifs de l'adresse (MSB), et laquelle contient les bits les moins significatifs (LSB) ?
4. Quelle est la taille (en bits) de la mémoire auxiliaire (appelée *répertoire*) nécessaire pour stocker les *étiquettes* de chaque ligne ?
5. Quel type de *localité* bénéficierait si on doublait la taille des lignes ?
6. Cite deux désavantages d'avoir des lignes deux fois plus longues.

4 Performance des disques

Soit un disque dur de 1TB avec des blocs de 4096B, qui tourne à 7200rpm, avec un temps d'accès (*seek time*) moyen de 5ms et une bande passante maximale de 120MB/s.

1. En présumant que le nombre de secteurs par cylindre est constant, combien de blocs ce disque a-t-il sur chaque piste ?
2. Combien de temps faudra-t-il pour lire séquentiellement un fichier de 1GB disposé de manière optimale sur le disque ?
3. Combien de temps faudra-t-il si ce fichier est divisé en morceaux de 512KB, où les blocs d'un morceau sont bien placés séquentiellement, mais les morceaux sont répartis aléatoirement sur le disque ?
4. Combien de temps faudra-t-il si le fichier est disposé séquentiellement comme avant, mais qu'on le lit par blocs de 32KB en commençant par la fin du fichier ?
5. Combien de blocs peut-on lire par seconde (IO/s) si ces blocs sont disposés de manière aléatoire sur le disque ?

Note : comme d'habitude vous pouvez présumer que $1024 \simeq 1000$.

5 Faute de page

Soit une station de travail avec des processus P2 et P3 *prêts* et un processus P1 *élu*. La station n'a qu'un seul processeur et a un disque dont une partie est utilisée comme *mémoire virtuelle*.

Mettre dans l'ordre les différents événements et étapes (certaines peuvent apparaître plusieurs fois) lors de l'exécution d'une instruction LOAD qui accède à une adresse présentement stockée sur le disque :

- (A) Chargement du processus Pn depuis son *PCB* vers le processeur
- (B) Envoi d'une requête au disque
- (C) Mettre à jour la table des pages du processus Pn
- (D) Recherche de l'instruction dans le cache
- (E) Recherche de la donnée dans le cache
- (F) Réception d'une interruption
- (G) Saut à la routine d'ordonnancement
- (H) Saut à la routine de défaut de page
- (I) Saut à la routine de gestion des interruptions
- (J) Sauvegarde du processus Pn dans son *PCB*
- (K) Traduction d'adresse logique en adresse physique
- (L) Trouver une case libre

Inclure ce qui se passe au niveau du processeur, du disque, du système d'exploitation, et de leurs interactions, et ce depuis le début de l'exécution de l'instruction jusqu'à ce qu'elle s'exécute avec succès.

6 QCM

Attention! bonne réponse : 0.5 points
 pas de réponse : 0 points
 mauvaise réponse : -0.25 points

Les tables de pages à un seul niveau ne sont pas utilisées parce qu'elles sont pas assez rapides

vrai faux

$$A + \overline{B} + A = \overline{(B \bullet \overline{A})}$$

vrai faux

Tant que les données sont plus petites que le cache, un cache à correspondance directe fonctionne aussi bien qu'un cache purement associatif.

vrai faux

En complément à 2 sur 8 bits, 100 est égal à -156.

vrai faux

0.4 ne peut pas être représenté exactement dans la représentation virgule flottante IEEE standard.

vrai faux

Si le futur est un miroir du passé, l'algorithme de remplacement dit *FIFO* est optimal.

vrai faux

Au TP2, la bande passante mesurée en mode séquentiel varie très peu avec la taille du tableau.

vrai faux

Les systèmes *temps-réels* ont besoin de caches très rapides et très sophistiqués.

vrai faux

L'algorithme "du tourniquet" (*round robin*) n'est pas utilisé en pratique car il peut causer des *famines*.

vrai faux

Faire une copie d'une grande table de nombres est une opération qui offre une bonne localité spatiale.

vrai faux