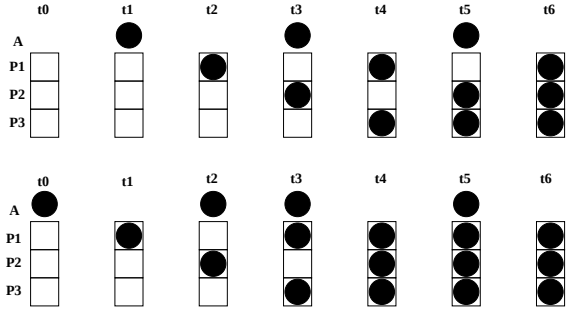
Université Paul Cézanne (Aix-Marseille III)

Licence Math-Info 1ère année - I2 - Examen de mai 2007
Durée : 1h30 - Aucun document autorisé - Calculatrice interdite

Exercice 1 : Circuits logiques (7 points)

Le but final de l'exercice est de réaliser un circuit avec mémoire qui indique l'état global d'une colonne de jetons qu'on complète au cours du temps. La figure ci-dessous indique deux exemples de l'évolution d'une colonne au cours du temps, pendant 6 unités de temps. La colonne peut contenir 3 jetons au maximum, dont les positions sont P1, P2 ou P3. On remplit cette colonne en ajoutant quand on veut un jeton par le haut, ce qu'indique la position A. Après qu'un jeton soit ajouté, il descend d'un cran dans la colonne à chaque unité de temps, jusqu'à toucher le sol (position P3) ou un jeton qui ne peut plus descendre (car ce dernier est sur le sol ou sur un jeton qui ne peut descendre). L'exemple du haut de la figure indique une succession d'états de la colonne où on ajoute un nouveau jeton au temps t1, au temps t3 et au temps t5. Le premier jeton se trouve en position A au temps t1, en position P1 au temps t2, en position P2 au temps t3 et en position P3 à partir du temps t4. Dans l'exemple du bas de la figure, le 3e jeton est ajouté au temps t3 (en position A) et reste bloqué en position P1 à partir du temps t4 car le 2e jeton (ajouté au temps t2) reste bloqué en postion P2 à partir du temps t4. Au temps t5, on essaie d'ajouter un 4e jeton mais il n'y a plus de place donc rien ne change dans la colonne.



Dans un premier temps, nous allons construire un circuit C avec 4 bits d'entrée A, P_1, P_2 et P_3 et 3 bits de sortie P'_1, P'_2 et P'_3. Ce circuit indique ce que sera l'état global d'une colonne au temps suivant, c'est-à-dire, le contenu de chacune des trois positions de la colonne, en fonction de ce que contenait précédemment la colonne en ses trois positions et du fait qu'on ajoute ou non un nouveau jeton. Quand un bit est à 1, cela signifie qu'il y a un jeton à la position qu'il indique. Ex : si on a en entrée $A = 1$, $P_1 = 0$, $P_2 = 1$ et $P_3 = 0$ (ce qui correspond à l'état au temps t3 de la colonne de l'exemple du haut de la figure) alors on a en sortie $P'_1 = 1$, $P'_2 = 0$ et $P'_3 = 1$ (contenu de la colonne au temps t4) ; si on a en entrée $A = 1$, $P_1 = 1$, $P_2 = 1$ et $P_3 = 1$ (ce qui correspond à l'état de la colonne de l'exemple du bas de la figure au temps t5) alors on a en sortie $P'_1 = 1$, $P'_2 = 1$ et $P'_3 = 1$ (contenu de la colonne au temps t6).

1.1 *Ecrire la table de vérité de la fonction logique qui indique les valeurs des sorties P'_1, P'_2 et P'_3 en fonction des valeurs d'entrées A, P_1, P_2 et P_3.*

1.2 *En utilisant la méthode des tables de Karnaugh, écrire les fonctions booléennes correspondant aux valeurs de sorties de P'_1, P'_2 et P'_3.*

1.3 *Dessiner le circuit C permettant d'obtenir P'_1, P'_2 et P'_3 à partir de A, P_1, P_2 et P_3.* Pour ceci, on n'a le droit d'utiliser que la porte logique unaire NON et les portes logiques OU et ET.

1.4 Maintenant, on veut réaliser le circuit à mémoire simulant une succession d'états d'une colonne de jetons. Ce circuit possède une entrée d'horloge nommée CK, deux entrées a et CLR et trois sorties p_1, p_2 et p_3. Les sorties p_1, p_2 et p_3 indique l'état de la colonne de jetons. L'entrée a indique si on ajoute un jeton ou non. Lorsque l'entrée CLR est à 0, la colonne est vidée, c'est-à-dire, les sorties p_1, p_2 et p_3 sont à 0. Lorsque CLR est à 1, à chaque impulsion d'horloge les sorties sont modifiées en fonction de a et des précédentes valeurs de p_1, p_2 et p_3.

Dessiner le circuit décrit ci-dessus. Pour ceci, on n'utilisera que le circuit C (défini dans la question **1.3**), des bascules D et des portes logiques NON, OU ou ET. Dans cette question, le circuit C est considéré comme une boîte noire, on le dessinera comme un rectangle (à quatre entrées et trois sorties) et on ne redessinera donc pas les portes logiques qui le composent.

Exercice 2 : Représentation des nombres et codage en mémoire (3 points)

2.1 *Donnez la représentation binaire et la représentation hexadécimale du nombre 123 (représenté en notation décimale).*

2.2 *Donnez la valeur en représentation décimale de l'entier relatif codé sur un octet dont la représentation hexadécimale est E3.*

2.3 Un texte ASCII contient une suite de 500 caractères. *Combien d'octets sont utilisés pour stocker ce fichier ? Si ces caractères n'étaient que des chiffres (entre 0 et 9), combien de bits suffirait-il pour coder chaque chiffre ? Combien d'octets pour stocker ces 500 chiffres ?*

Exercice 3 : Langage machine (4 points)

Dans cet exercice, on se place dans le cadre du microprocesseur MP0 et du langage machine LM0 décrit en cours. On rappelle que MP0 est un microprocesseur possédant deux registres d'adresses 8 bits A0 et A1 et deux registres de données 8 bits D0 et D1. LM0 possède notamment les instructions de transfert d'octet MOVE, d'addition ADD, de soustraction SUB. Ces instructions ont deux opérandes. Le premier est l'opérande source et le deuxième est l'opérande destination, qui sera modifié grâce à l'opérande source. LM0 a 3 types d'adressage : immédiat (on préfixe l'opérande avec #), direct (l'opérande est laissé tel quel) et indirect (l'opérande, qui est toujours un registre d'adresse, est mis entre parenthèses).

3.1 On considère que la mémoire contient la suite de dix valeurs 56, 34, 5, 88, 61, 63, 16, 45, 72 et 36 à partir de l'adresse 50. On exécute la série d'instructions suivante stockée à partir de l'adresse 100 :

```
100 : MOVE #50, D0
102 : MOVE D0, A1
104 : MOVE A1, A0
106 : ADD #1, A1
108 : CMP (A1), (A0)
110 : JGT #114
112 : MOVE A1, A0
114 : CMP #59, A1
116 : JGT #106
118 : MOVE D0, A1
120 : MOVE (A1), D1
122 : MOVE (A0), (A1)
124 : MOVE D1, (A0)
126 : ADD #1, D0
128 : CMP #59, D0
130 : JGT #102
```

Quelles seront les 10 valeurs stockées à partir de l'adresse 50 après l'exécution de ces instructions ? Expliquez pourquoi. On rappelle que CMP est l'instruction de comparaison. JGT est l'instruction conditionnelle de saut si la première valeur de l'instruction CMP précédente est strictement supérieure à sa deuxième valeur.

3.2 On exécute la série d'instructions suivante stockée à partir de l'adresse 60 :

```
60 : MOVE #65, A0
62 : ADD #2, (A0)
64 : ADD #2, A0
66 : ADD #2, (A0)
68 : MOVE #60, D0
70 : MOVE D0, 100
```

Quelle valeur y aura-t'il à l'adresse 100 après l'exécution de ces instructions ? Expliquez pourquoi. On rappelle que chaque instruction est codée sur deux octets et que le deuxième octet contient un nombre apparaissant dans un des deux opérandes.

Exercice 4 : Utilisation d'Unix (6 points)

On se trouve dans un répertoire contenant uniquement le fichier `grosminet.txt` et le répertoire `cage`, ce dernier ne contenant que le fichier `titi.txt`. Le fichier `grosminet.txt` contient l'unique ligne de texte `Miaou` et le fichier `titi.txt` contient l'unique ligne de texte `Cuicui`.

4.1 *Indiquez les effets (ce qui est affiché, le ou les fichiers modifiés, etc) de l'exécution des commandes Unix suivantes à partir d'un shell. Signalez les erreurs éventuelles.* Chaque question est indépendante : on fera comme si les commandes des questions précédentes n'avaient pas été exécutées.

a) `$ ls`
b) `$ ls grosminet.txt`
c) `$ ls cage`
d) `$ ls titi.txt`
e) `$ ls *`
f) `$ cat grosminet.txt`
g) `$ cat *`
h) `$ cat < grosminet.txt`
i) `$ cat grosminet.txt > titi.txt`
j) `$ cat > titi.txt`
k) `$ cat`
l) `$ echo Ouah >> grosminet.txt`
m) `$ cat */* >> grosminet.txt`
n) `$ cat grosminet.txt > grosminet.txt`
o) `$ cat grosminet.txt >> grosminet.txt`
p) `$ cat grosminet.txt cage/titi.txt grosminet.txt | head -n2`
q) `$ cat grosminet.txt | cat > titi.txt`
r) `$ cat grosminet.txt | cat >> grosminet.txt`
s) `$ head -n2 | tail -n1`
t) `$ ls chien.txt > chien.txt`

4.2 *Indiquez la suite de commandes permettant de permuter les fichiers* `grosminet.txt` *et* `titi.txt`.

4.3 *Indiquez la suite de commandes permettant de créer le répertoire* `maison` *dans le répertoire courant et d'y déplacer* `grosminet.txt` *ainsi que le répertoire* `cage`.