ORDRE DES INGÉNIEURS DU QUÉBEC

SESSION DE MAI 2022

Toute documentation permise
Calculatrices : modèles autorisés seulement
Durée de l'examen : 3 heures

16-EL-A4 – Systèmes numériques et ordinateurs

**Question 1 (40 points)** Soit cette table de vérité pour un circuit logique ayant pour entrées A, B, C, D et pour sortie S :

| A | B | C | D | S |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

a) Donnez l'équation canonique de ce circuit (10 points)

b) Donnez la table de Karnaugh représentant ce circuit (10 points)

| CD\AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | | |
| 01 | | | | |
| 11 | | | | |
| 10 | | | | |

c) Tracez le diagramme comportant **le plus petit nombre de portes logiques** pour réaliser ce circuit en incluant l'équation booléenne correspondante. Les portes ET et OU peuvent avoir plus de deux entrées. (20 points)

**Question 2 (30 points)** Vous devez concevoir une machine à états finis capable de détecter le motif binaire « 111 » dans sa séquence d'entrée. Au troisième bit du motif, votre machine doit avoir la sortie « 1 ». La sortie sera toujours « 0 » dans les autres cas. Donc, si votre machine reçoit la séquence :

0,1,1,1,1,1,0,1,1,1

La sortie devrait être :

0,0,0,1,1,1,0,0,0,1

Notez que la sortie « 1 » est répétée après le troisième si l'entrée est toujours 1.

a) Tracez le diagramme de votre machine à états finis. **Attention :** la machine doit être conçue pour recevoir un seul bit à la fois (et non un groupement de bits) (10 points)

b) Donnez la table d'états-transitions de la machine (5 points)

c) Donnez la séquence de sortie du circuit pour l'entrée suivante (5 points) :

« 100011100101 »

d) Traduisez le circuit en un programme en C où l'entrée peut être obtenue par la fonction `int E(void)` et la sortie changée par la fonction `void S(int)`. (10 points).

Par exemple, si la séquence d'entrée est « 100 » et que la sortie devrait être « 000 », la fonction `E()` devrait agir ainsi :

```
int v = 0;
v = E();        // v = 1
v = E();        // v = 0
v = E();        // v = 0
```

Le programme doit fonctionner indéfiniment. Il pourrait donc avoir une structure de la sorte :

```
while (true) {
     // À faire : traitement de l'entrée avec E()
     int out = 0;   // À changer selon l'état en cours
     S(out);
}
```

(Dans l'exemple précédent, sans modifications, la sortie serait toujours à 0).

**Question 3 (20 points) :** Soit le programme en assembleur MIPS suivant :

```
li   $t0, 0
li   $t1, 10
li   $t2, 0x2FC0

start:
addi $t0, $t0, 3
sub  $t3, $t0, $t1
bgtz $t3, end
j    start
end:
sw   $t0, 0($t2)
```

a) Donnez le contenu du registre $t0 à la fin de l'exécution du programme (5 points)

b) Traduisez ce programme en langage C ou C++ (15 points)

**Question 4 (10 points)** Dans l'architecture MIPS et dans beaucoup d'autres architectures de type RISC, une instruction du genre :

```
beqm $t0, 0($t1), target
```

, qui ferait un branchement si le contenu du registre $t0 et le contenu en mémoire à l'adresse représentée par $t1 sont égaux, n'existe pas. Pourquoi ? Donnez également la séquence d'opération équivalente en assembleur MIPS légal.