Examen S4-CLM Mai 2010

Tous documents autorisés - Calculatrices Autorisées
Les questions sont indépendantes. Durée 2h

Représentation des réels [2 pt]

On considère la représentation IEEE 754 des flottants simple précision.
Q1. Quel est le plus grand positif X strictement inférieur à 2 représentable. Donner une expression formelle et la représentation IEEE 754 **en hexadécimal**. La représentation décimale n'est **PAS** demandée.

Compteurs et automates [4 pt]

Q2. Réaliser un compteur par 10. On donnera
a) le nombre de bascules D nécessaires ;
b) la table de transition (table de vérité de la logique de transition).
Q3. On veut réaliser un automate qui contrôle une porte de garage. La spécification est la suivante.
- Signal d'entrée E sur 1 bit.
- Commande d'ouverture Z sur 1 bit. $Z = 1$ correspond à l'ouverture de la porte ; tant que Z vaut 1, la porte reste ouverte. $Z = 0$ correspond à la fermeture de la porte ; tant que Z vaut 0, la porte reste fermée.
- Commande d'avertisseur A.
- Comportement : lorsque E passe à 1,
  - La commande d'avertisseur A passe à 1, et doit rester à 1 un temps T_a.
  - Au bout de ce temps T_a, la porte de garage s'ouvre (on néglige le temps d'ouverture), et A s'éteint.
  - Lorsque E passe à 0, la porte se ferme sans déclencher d'avertissement. Lorsque E passe à 0 pendant l'avertissement, celui-ci s'éteint et l'ouverture ne se produit pas.

On dispose d'un compteur réinitialisable, qui compte T_a : entrée R (reset), sortie C. C est à 1 pendant 1 cycle, lorsque le compteur a compté T_a depuis le dernier reset.

a) Compléter diagramme de l'automate de la fig. 1 ;

b) Compléter la table 1 pour définir la table de transition et le PLA des commandes de cet automate. On ajoutera autant de lignes que nécessaire. Attention : lorsqu'une entrée est indifférente, par exemple C dans l'état 0, il est impératif d'utiliser la notation X.

Jeu d'instructions [10 pt]

Dans toute cette partie, on considère le jeu d'instructions DIDE.
Q4. Donner le codage hexadécimal des instructions AND R4, R5, R6 et STORE R1, -4(R15).
Q5. Donner le fragment de code qui initialise

a) le registre R1 à 0x10000000
b) le registre R2 à 0x00001000
c) le registre R3 à 0x98128723

Q6. Que calcule le code suivant, lorsque R1 contient un entier naturel inférieur à 2^{16} .

```
SLL R2, 5, R1
SUB R2, R2, R1
```

Q7 On veut effectuer la multiplication d'un entier naturel contenu dans le registre R1 par 31, avec résultat sur 64 bits dans R3 (partie haute) et R2 (partie basse).

a) Pourquoi a-t-on besoin de 64 bits ?
b) Donner le principe de votre méthode - en français courant
c) Donner le code correspondant.

Q8. On veut compiler la conditionnelle

`si ((x > y) ou (x > 7)) alors x =1 sinon x = 0`

En supposant que la valeur de `x` est dans `R1` et celle de `y` dans `R2`, les instructions suivantes sont celles du programme correspondant à cette condionnelle, mais elles sont listées dans le désordre. On demande de les remettre dans l'ordre.

```
          BLE   suite
          XOR   R3 R3 R3
          CMPI  R1 7
          CMP   R1 R2
          BGT   vrai
vrai :    ADDI  R3 R3 1
suite :
```

Q9. On veut compiler la boucle

```
int X[100], Y[100] ;
for (i = 0 ; i <100 ; i++)
    Y[i] = Y[i-1] + X[i] ;
```

`X` est implanté à partir de l'adresse 0x00001000 et `Y` à partir de l'adresse 0x00002000. On supposera que le registre R1 contient 0x00001000 et que le registre R2 contient 0x00002000 ; le code correspondant n'est pas demandé. **Un code non commenté ne sera pas corrigé**.

Mémoire [4 pt]

Q10. On considère la déclaration en C suivante :

```
char c[3] ;
double x[12] ;
short k ;
int tab[10] ;
```

En supposant que l'allocation mémoire se fait dans l'ordre des déclarations et à partir de l'adresse 0, donner :

a) la taille en octets de la zone mémoire correspondante
b) l'adresse de c[2], x[10] et de tab[3] **en notation décimale**.

Rappel : les données sont alignées.

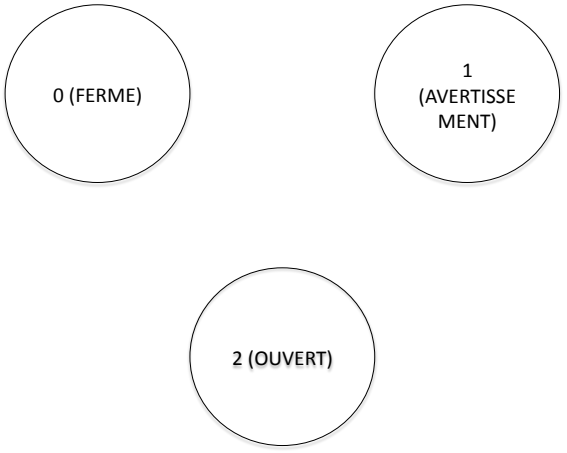

FIGURE 1 – Automate d'ouverture de porte

TABLE 1 – Réalisation de l'automate d'ouverture de porte

Q_1	Q_0	E	C	D_1	D_0	Z	A	R
0	0	0	X	0	0	0	0	0