E.N.S.E.I.R.B. Année 2006-2007
Filière Informatique 1ère année
Filière Télécom 1ère année

STRUCTURE DES ORDINATEURS

EXAMEN

2 heures
avec documents de cours

N.B. : - Les réponses aux questions doivent être argumentées et aussi concises que possible.
- Le barème est donné à titre indicatif.

Question 1 (6 points)

On considère l'architecture ISA d'un processeur 32 bits disposant, entre autres, des instructions suivantes :
- « LDI R*i*, *val* » pour le chargement immédiat d'une valeur entière dans le registre R*i* ;
- « LD R*i*, (R*j*) » pour le chargement indirect par registre dans R*i* d'une valeur entière contenue en mémoire à l'adresse donnée par R*j* ;
- « ADD R*d*, R*s1*, R*s2* » pour l'addition entière dans R*d* des valeurs contenues dans R*s1* et R*s2*, et
- « SHL R*d*, R*s*, R*l* » pour le décalage vers la gauche (« *shift left* ») du contenu du registre R*s*, décalé de R*l* positions, le résultat étant placé dans le registre R*d*.

On suppose que toutes les adresses sont exprimées en octets.

On considère que le registre R1 contient l'adresse de début d'un tableau `tab` de structures de données « *à la C* »(commençant donc à l'indice 0), la structure de base étant constituée de deux entiers `x` et `y`.

(1.1) (2 points)

Au moyen des instructions précédentes, écrivez le fragment de code machine permettant de récupérer dans le registre R3 le contenu de `tab[i].y`, où la valeur de `i` est contenue dans le registre R2.

On suppose maintenant que l'architecture ISA possède en plus une instruction LDBX permettant le chargement par adressage basé indexé.

(1.2) (2 points)

Expliquez ce qu'est l'adressage basé indexé, et proposez une écriture des opérandes de l'instruction LDBX.

(1.3) (2 points)

Utilisez votre écriture de l'instruction LDBX pour réécrire de façon plus compacte le fragment de code précédent.

Question 2 (14 points)

Les ordinateurs ne savent manipuler que des nombres binaires, alors que les humains expriment les nombres sous forme décimale. Lors d'un calcul interactif avec un ordinateur, il faut donc convertir les nombres décimaux en nombres binaires, puis à la fin du calcul binaire convertir le résultat en décimal, ce qui est très coûteux. Pour éviter cela, certains anciens processeurs disposaient d'instructions pour travailler directement avec des nombres décimaux entiers, en utilisant une représentation appelée « BCD », pour « *Binary-Coded Decimal* », soit en Français « *Décimal Codé en Binaire* ». L'idée du codage BCD est de coder chaque chiffre décimal sous forme binaire et de manipuler directement ce codage en machine, pour que chaque nombre ainsi calculé puisse se lire directement sous forme décimale. Par exemple, si on codait chaque nombre décimal sur un octet, le nombre $815_{(10)}$ s'écrirait alors $00001000\ 00000001\ 00000101_{(2)}$.

(2.1) (2 points)

On veut coder chaque chiffre BCD de la façon la plus compacte possible. Quel est le nombre minimal de bits b nécessaire au codage d'un chiffre BCD ? Justifiez votre réponse.

Dans la suite de cet exercice, quand on manipulera le codage BCD, on codera toujours chaque chiffre sur ce nombre minimal de bits.

(2.2) (2 points)

Pour chacun des nombres décimaux suivants, donnez leur écriture binaire BCD, ainsi que l'écriture hexadécimale de leur codage BCD :

- $42_{(10)}$;
- $9203_{(10)}$.

(2.3) (1 point)

Que peut-on remarquer au sujet de l'écriture hexadécimale d'un nombre codé en BCD ?

On suppose que l'on dispose déjà d'un additionneur binaire classique sur b bits, avec une retenue entrante C_{IN}, une retenue sortante C_{OUT}, deux groupes d'entrées, étiquetées de A_0 à A_{b-1} et de B_0 à B_{b-1} pour les deux nombres binaires A et B d'entrée à additionner (le bit 0 est le bit de poids le plus faible), et d'un groupe de sorties étiquetées de S_0 à S_{b-1} pour le résultat. Dans la suite, cette fonction pourra être représentée schématiquement par une boîte A. On dispose en outre des portes logiques classiques.

(2.4) (2 points)

Construisez une table donnant, pour chaque chiffre décimal compris entre 0 et 9 :

- la représentation binaire BCD de ce chiffre ;
- le résultat de l'ajout de 1 à cette représentation BCD, codée sous la forme de deux chiffres BCD ;
- le résultat de l'ajout de 1 à cette représentation BCD, au moyen de l'additionneur binaire classique ;
- la différence décimale entre ces deux résultats, interprétés tous les deux en binaire.

(2.5) (2 points)

Par extension des résultats contenus dans la table ci-dessus, quelles sont les valeurs de la différence entre la somme BCD et la somme binaire de deux chiffres BCD ? Quand la différence entre les deux résultats est-elle non nulle, et que vaut-elle dans ce cas ?

(2.6) (3 points)

Donnez et justifiez la formule logique, puis faites le schéma, d'un circuit qui détecte quand la somme BCD calculée par un additionneur b bits classique est fausse (c'est-à-dire quand la différence étudiée à la question précédente est non nulle). Ce circuit a comme entrées la valeur S de la somme calculée par l'additionneur ainsi que la retenue sortante. Sa sortie E vaut 1 s'il y a erreur, et 0 sinon. Dans la suite, cette fonction pourra être représentée schématiquement par une boîte E.

(2.7) (2 points)

En utilisant le circuit de la question précédente et deux additionneurs binaires à b bits, dessinez un additionneur BCD qui prend en entrée deux chiffres BCD et une retenue d'entrée, et produit un chiffre BCD et une retenue de sortie.