Université de Bordeaux
Licence Informatique 2ème année

U.F. Informatique
Année 2023 – 2024

ARCHITECTURE DES ORDINATEURS

Devoir Surveillé 1

1 heure et 20 minutes
sans documents

N.B. : - Travaillez calmement. Ne bâclez pas vos réponses : il vaut mieux traiter correctement moins de questions que tout faire de travers.
- Les réponses aux questions doivent être argumentées et aussi concises que possible.
- Le barème est donné à titre indicatif.

Exercice 1 (50 points)

On considère des entiers codés sur 16 bits, interprétés comme des entiers relatifs codés en complément à deux. Chacun de ces entiers peut (et doit) s'écrire avec quatre chiffres hexadécimaux (on omettra le préfixe « 0x »).
Soient les cinq entiers suivants :

M	N	P	Q	R
1991	9A33	B713	46F2	7827

(1.1) (10 points)
Parmi ces 5 entiers, quels sont ceux qui sont négatifs ? Expliquez pourquoi.

(1.2) (10 points)
Classez ces entiers du plus petit au plus grand (par exemple : « M < N < P < Q < R », mais cette réponse choisie arbitrairement n'est probablement pas la bonne...). Justifiez la réponse sur votre copie.

(1.3) (10 points)
Calculez $N + R$, et donnez le résultat en hexadécimal. Pour cela, effectuez l'addition en binaire directement sur la copie — une réponse brute ne rapportera aucun point. Y a-t-il débordement (*overflow*) ?

(1.4) (20 points)
Calculez $-Q$ et $N - Q$, et donnez les résultats en hexadécimal. Y a-t-il débordement (*overflow*) ? Comme pour la question précédente, le détail des calculs en binaire doit figurer sur la copie.

Exercice 2 (150 points)

La page suivante contient le texte d'un programme écrit en langage y86. Tous les commentaires ayant malencontreusement été effacés, il vous faut partir de zéro pour comprendre ce que fait ce programme, en répondant aux questions ci-dessous.
Attention : les réponses qui, au lieu d'expliquer, paraphrasent simplement le code (telles que « on soustrait edx à ebx ») ne rapporteront aucun point.

(2.1) (20 points)
Quels sont les paramètres de la fonction `mystere` ? Expliquez quelle partie du code permet de répondre à cette question.
Si ce code avait été produit par un compilateur C, quel aurait été le prototype de la fonction `mystere` ?

(2.2) (20 points)
Quel est le rôle de l'instruction d'adresse `0x032` ? Pourquoi cette instruction prend-elle en paramètre l'entier 8 ?

(2.3) (10 points)
Quelle est la valeur qui sera stockée à l'adresse d'étiquette `v` à la fin de l'exécution du programme ? Justifiez votre réponse.

(2.4) (10 points)
Quelle aurait été la valeur retournée par la fonction `mystere` si, dans l'instruction d'adresse `0x00e`, on avait mis la valeur immédiate `3` au lieu de `7` ?

(2.5) (10 points)
Quel est le rôle de l'instruction d'adresse `0x023` ? Quel lien existe-t-il entre cette instruction et l'appel de la fonction `mystere` ?

(2.6) (30 points)
Dans la fonction `mystere`, quelles sont les rôles respectifs des différents registres utilisés ? Justifiez vos réponses.

(2.7) (10 points)
Dans la fonction `mystere`, quels sont les rôles respectifs des étiquettes `myst1`, `myst2` et `myst3` ?

(2.8) (10 points)
Dites, en une phrase, ce que renvoie la fonction `mystere`, en prenant en compte les connaissances obtenues de la question précédente.

(2.9) (10 points)
En vous basant sur le codage de certaines instructions (dites lesquelles), quels sont les numéros sur 4 bits correspondant aux registres `ebx`, `edx` et `edi` ?

(2.10) (20 points)
Expliquez le codage binaire des six octets de l'instruction d'adresse `0x05c`.

```
0x000:               |           .pos 0
0x000: 30f400020000 | init:     irmovl pile,%esp
0x006: 30f705000000 |           irmovl 5,%edi
0x00c: a07f         |           pushl  %edi
0x00e: 30f307000000 |           irmovl 7,%ebx
0x014: a03f         |           pushl  %ebx
0x016: 30f66c000000 |           irmovl t,%esi
0x01c: a06f         |           pushl  %esi
0x01e: 8030000000   |           call   mystere
0x023: c0f40c000000 |           iaddl  12,%esp
0x029: 400f88000000 |           rmmovl %eax,v
0x02f: 10           |           halt
                    |
0x030: a03f         | mystere:  pushl  %ebx
0x032: 502408000000 |           mrmovl 8(%esp),%edx
0x038: 50140c000000 |           mrmovl 12(%esp),%ecx
0x03e: 500410000000 |           mrmovl 16(%esp),%eax
0x044: c1f101000000 | myst1:    isubl  1,%ecx
0x04a: 7269000000   |           jl     myst3
0x04f: 503200000000 |           mrmovl (%edx),%ebx
0x055: 6103         |           subl   %eax,%ebx
0x057: 7667000000   |           jg     myst2
0x05c: c0f204000000 |           iaddl  4,%edx
0x062: 7044000000   |           jmp    myst1
0x067: 6030         | myst2:    addl   %ebx,%eax
0x069: b03f         | myst3:    popl   %ebx
0x06b: 90           |           ret
                    |
0x06c:              |           .align 4
0x06c: 04000000     | t:        .long  4
0x070: 02000000     |           .long  2
0x074: 03000000     |           .long  3
0x078: 06000000     |           .long  6
0x07c: 01000000     |           .long  1
0x080: 07000000     |           .long  7
0x084: 05000000     |           .long  5
0x088: 00000000     | v:        .long  0
                    |
0x200:              |           .pos 0x200
0x200:              | pile:
```