

# Rattrapage

## Circuits et architecture des ordinateurs

— Master d'informatique —

Juin 2016, durée 2h.

*L'examen se compose de quatre exercices indépendants. Les documents sont interdits à l'exception d'une feuille de mémento (double A4). Tous les appareils électroniques y compris les montres sont interdits.*

► **Exercice 1** Dans tout cet exercice, on considère des entiers codés en complément à 2 sur 8 ou 16 bits. Remplir le tableau suivant en donnant les codages des entiers

Entier	Codage sur 8 bits	Codage sur 16 bits
1		
-1		
143		
223		
-157		

► **Exercice 2** Le but de cet exercice est d'écrire une routine qui convertit une chaîne de caractères contenant uniquement des caractères '0' et '1' en la valeur numérique obtenue en interprétant la chaîne comme un nombre écrit en binaire. On rappelle qu'en LC-3 les chaînes sont, comme en C, terminée par le caractère '\0' et que chaque caractère occupe un mot mémoire de 16 bits. La seconde question ne sera corrigée que si la première question est correctement traitée.

- Écrire une fonction `a2b` en C ou en pseudo-code qui prend en paramètre une chaîne de caractères sous forme d'un tableau de caractères terminé par '\0' (de type `char*` en C) et qui retourne la valeur numérique (de type `int` en C). Le prototype de `a2b` en C est donc `int a2b(char* str)`.
- Écrire une routine en langage d'assembleur LC-3 qui réalise la même opération. La routine devra prendre l'adresse de la chaîne dans le registre R0 et retourner la valeur dans le même registre R0.

► **Exercice 3** On suppose qu'on a ajouté au processeur LC-3 une instruction de rotation circulaire ROT qui admet les mêmes syntaxes que l'instruction AND. On veut utiliser cette instruction pour réaliser des décalages vers la gauche ou la droite. La différence entre un décalage et une rotation vers la gauche est que les bits les plus à gauche sont perdus et que les bits les plus à droite prennent la valeur 0.

On suppose d'abord que  $n$  est une constante fixe qui vaut 1, 2, 3, ou 4.

- Donner une séquence de deux instructions qui réalisent un décalage vers la gauche de  $n$  bits. On pourra utiliser l'instruction ROT et toutes les autres instructions du LC-3.
- Donner une séquence de deux instructions qui réalisent un décalage vers la droite de  $n$  bits. On pourra utiliser l'instruction ROT et toutes les autres instructions du LC-3.

On suppose maintenant que  $n$  est une constante fixe qui vaut entre 5 et 8.

- Donner une séquence de 4 instructions ou moins qui réalisent un décalage vers la gauche de  $n$  bits. On pourra utiliser l'instruction ROT et toutes les autres instructions du LC-3.
- Donner une séquence de 4 instructions ou moins qui réalisent un décalage vers la droite de  $n$  bits. On pourra utiliser l'instruction ROT et toutes les autres instructions du LC-3.

On suppose maintenant que  $n$  est supérieur à 9.

- Expliquer comment réaliser un décalage de  $n$  bits vers la droite ou vers la gauche.

► Exercice 4 On considère le programme suivant écrit en assembleur LC-3.

```
.ORIG x3000
mystere: LD R0,n
          LD R6,sp
          JSR myst
          TRAP x25      ; HALT
n:        .FILL 5
sp:       .FILL 0x4000

myst:     AND R0,R0,R0
          BRnz fini
          ADD R6,R6,-1
          STR R7,R6,0
          ADD R0,R0,-1
          JSR myst
          ADD R0,R0,1
          LDR R7,R6,0
          ADD R6,R6,1
          RET
fini:     AND R0,R0,0
          RET
          .END
```

- Quel est le contenu en binaire du registre R0 lorsque le programme atteint l'instruction TRAP ?
- Quel serait le contenu du registre R0 lorsque le programme atteint l'instruction TRAP si la valeur chargée dans le registre R0 à la première instruction était respectivement 3, 4, 6 et 7.
- Que calcule la routine myst quand la valeur contenue au départ dans R0 est positive ?