

Examen (Sujet A)

Circuits et architecture des ordinateurs

— Master d'informatique —

Janvier 2012, durée 2h30.

L'examen se compose de trois exercices indépendants. Les seuls documents autorisés sont une feuille de memento (double A4). Tous les appareils électroniques sont interdits à l'exception des montres.

► **Exercice 1** Dans tout cet exercice, on considère des entiers codés en complément à 2 sur 8 ou 16 bits.

a) Remplir le tableau suivant en donnant les codages des entiers

Entier	Codage sur 8 bits	Codage sur 16 bits
1	00000001	00000000
-1	11111110	
143	10001111	
223	11011111	
-157	01000111	

On utilise des additionneurs 8 et 16 bits pour calculer les sommes. Remplir le tableau suivant en donnant le résultat de l'addition en binaire puis la valeur en décimal.

Entier	Addition sur 8 bits	Valeur en déc.	Addition sur 16 bits	Valeur en déc.
143 + 223				
143 - 157				

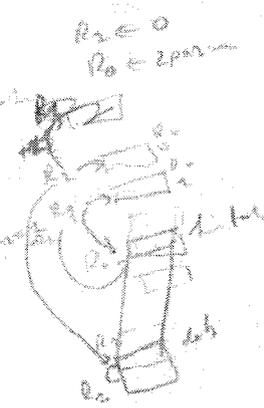
► **Exercice 2** On considère le programme suivant écrit en assembleur LC-3.

```

.ORIG x3000
myst: LD R0,param
      JSR myst0
      TRAP x25 ; HALT
param: .FILL 10

myst0: AND R1,R1,0
      AND R0,R0,R0
      BRz fini0z
      ADD R1,R1,1
      ADD R6,R6,-1
      STR R2,R6,0
loop0: AND R2,R1,R0
      BRnp fini0
      ADD R1,R1,R1
      BR loop0
fini0: LDR R2,R6,0
      ADD R6,R6,1
fini0z: RET

myst1: AND R1,R1,0
      AND R0,R0,R0
      BRz fini1z
      ADD R6,R6,-2
      STR R2,R6,0
      STR R3,R6,1
      ADD R2,R1,1
loop1: AND R3,R2,R0
      BRz next1
      ADD R1,R2,0
next1: ADD R2,R2,R2
      BRnp loop1
fini1: LDR R3,R6,1
      LDR R2,R6,0
      ADD R6,R6,2
fini1z: RET
      .END
  
```



- Donner les valeurs en décimal et en binaire des registres R0 et R1 lorsque le programme atteint l'instruction TRAP.
- La valeur 10 de .FILL est remplacée par une valeur quelconque n. Donner les valeurs successives du registre R1 lors du passage à l'étiquette loop0.

- c) Expliquer ce que calcule la routine `myst0`.
- d) Donner le rôle des instructions `ADD R6,R6,-1` et `STR R2,R6,0`.
- e) Reprendre les questions a), b) et c) en remplaçant l'instruction `JSR myst0` par `JSR myst1`.

► **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.

- a) 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.
- b) 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.

- c) 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.
- d) 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.

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

► **Exercice 4** On suppose que le processeur LC-3 est réalisé par un pipeline à 5 étages. On suppose que le registre `R0` contient la valeur 10. Donner les aléas de pipeline rencontrés par l'exécution des 15 premières instructions après le début de la routine `myst0` de l'exercice 2. Pour chacun de ces aléas, donner son type, comment le résoudre et s'il provoque l'apparition de bulles dans le pipeline.