

CC3 - LIF6 Printemps 2013

Durée prévue : 1h15

Aucun document autorisé. Calculatrices, téléphones et ordinateurs portables interdits.

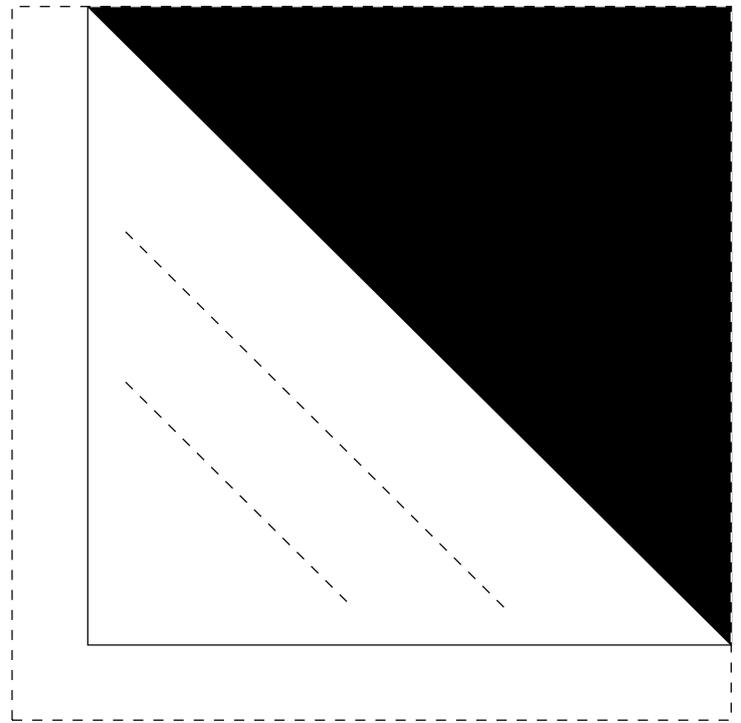
N'oubliez pas de noter votre nom, votre prénom et votre numéro d'étudiant sous le cache rabattable.

Justifiez brièvement et précisément vos réponses, tout en respectant les emplacements prévus.

Le barème est donné à titre indicatif, et pourra être modifié.

Ne rien écrire dans ce cadre.

Partie 1 :
Partie 2 :
Partie 3 :
Partie 4 :



1 Leading zero count (6 pts)

Le but est d'écrire une routine `lzc`, permettant de compter le nombre de bits égaux à zéro en tête de la représentation binaire sur 16 bits d'un entier naturel strictement positif. Par exemple, $lzc((0000\ 0000\ 0010\ 1000)_2) = 10$. Il vous est demandé pour cela de traduire l'algorithme suivant :

```
paramètre d'entrée   : R0, entier naturel strictement positif
paramètre de sortie  : R1, nombre de 0 en tête de l'entier dans R0
registres temporaires : R2, R3
On traduit l'algo. suivant :
R2 <- R0
R1 <- 0
R3 <- R2 & x8000 // x8000 est une constante hexadécimale
while(R3 == 0) {
    R1++
    R2 <- R2 << 1 // décalage de un bit vers la gauche
    R3 <- R2 & x8000 // & désigne le ET bit-à-bit
}
```

Complétez le programme ci-dessous dans le langage d'assemblage du LC3. Vous devez respecter les commentaires présents dans le code, et ajouter vos propres commentaires. N'oubliez pas de compléter le programme principal.

```
        .ORIG x3000        ; adresse de début de programme
; partie dédiée au code
        .....           ; charge l'entier désigné par n dans R0
        .....           ; appel à la routine lzc
        .....           ; stocke le résultat à l'adresse r
        HALT              ; termine le programme

; partie dédiée aux données
n:      .FILL 5            ; entier strictement positif
r:      .BLKW 1           ; emplacement pour le résultat
m:      .FILL x8000       ; masque

; Compte le nombre de bits à 0 en tête de la représentation binaire d'un entier naturel.
lzc:    .....           ; .....
        .....           ; .....
        .....           ; .....
```