Circuits et Architecture (CA7)
Examen – Session n° 1 : Sujet A
12 Janvier 2018, durée 3 heures

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

Exercice 1 – Représentation des nombres et opérations élémentaires

Dans cet exercice, les entiers sont codés en complément à deux sur 8 bits. Pour deux entiers m et n , on note $m + n$ la vraie valeur de la somme des deux nombres et $m \oplus n$ la somme calculée par un additionneur 8 bits. De même, $m - n$ dénote la vraie différence et $m \ominus n$ est la différence calculée par un circuit soustracteur sur 8 bits.

1. Donnez un circuit de l'additionneur à 8 bits $ADD_8(A, B, C_0)$ en utilisant l'additionneur à 1 bit avec retenue, noté SC en cours. Les entrées A et B ont chacune 8 bits et C_0 est une entrée d'un bit. Les deux sorties de l'additionneur sont S à 8 bits représentant la somme calculée et C_1 à 1 bit représentant la retenue générée.
2. Etant donné un circuit additionneur à 8 bits comme ci-dessus, rappelez comment est obtenu un circuit $SUB_8(A, B)$ qui calcule la différence entre ses entrées.
3. Reproduisez sur votre copie le tableau ci-dessous en le complétant. Donnez une explication chaque fois que la vraie valeur diffère de celle obtenue par le circuit.

m		n		$m + n$	$m \oplus n$	$m - n$	$m \ominus n$
décimal	binaire	décimal	binaire	décimal	binaire	décimal	binaire
40	00101000	27	00011011	67		13	
67		80					
-40		-80					
-67		-27					

Exercice 2 – Programmation LC-3

L'objectif de cet exercice est d'écrire une routine LC-3 qui calcule la *parité* d'un mot de 16 bits, c'est à dire le nombre de bits à 1 modulo 2. La parité est donc 0 si le nombre de bits à 1 du mot est pair et 1 si le nombre de bits à 1 est impair.

1. Donnez la séquence d'instructions LC-3 qui permet d'incrémenter le registre R1 si le bit de poids fort du registre R0 est 1.
2. Donnez la séquence d'instructions LC-3 qui permet de déplacer d'un cran à gauche (vers les poids forts) les bits contenus dans le registre R0.
3. Écrivez une routine *parite* qui calcule la parité d'un mot de 16 bits contenu dans le registre R0. Le résultat (valeur 0 ou 1) est retourné dans ce même registre.
4. Modifiez la routine précédente pour que le seul registre modifié par celle-ci soit le registre R0 (on ne prend pas en compte le registre R7).
5. Écrivez la séquence d'instructions qui modifie le bit 15 d'un mot de 16 bits contenu dans le registre R1 de telle sorte que le nombre de bits à 1 devient pair.

Exercice 3 – Programme mystère

On considère le programme suivant écrit en assembleur LC-3. On rappelle que LC-3 est un micro-processeur sur 16 bits.

```

        .ORIG x3000
mystere: LEA R0, str
        JSR myst
        TRAP x25          ; HALT
str:    .STRINGZ "Un deux trois "
sp:    .FILL -32          ; code caractère espace est 32

myst:   LD R3, sp
        ADD R1, R0, 0
loop:   LDR R2, R0, 0
        BRz endloop
        ADD R4, R2, R3
        BRz next
        STR R2, R1, 0
        ADD R1, R1, 1
next:   ADD R0, R0, 1
        BR loop
endloop: NOT R0, R0
        ADD R0, R0, 1
        ADD R0, R1, R0
        BRz end
        AND R0, R0, 0
        STR R0, R1, 0
end:   RET
        .END

```

1. Quelle est la valeur du registre R1 lorsque l'exécution du programme atteint l'instruction TRAP ? Et la valeur du registre R0 ?
2. Quel est le contenu de la mémoire entre les adresses `str` et `sp` lorsque le programme atteint l'instruction TRAP ? Et si à l'adresse `str` se trouve la chaîne de caractères Architecture ?
3. Que calcule la routine `myst` ?
4. Comment modifier la routine `myst` pour que le contenu des registres autres que R7 soit préservé à la sortie de la routine ?

Exercice 4 – Pipeline

On suppose que le processeur LC-3 est réalisé par un pipeline à 5 étages comme cela a été vu en cours. On suppose que les instructions de saut sont exécutées partiellement sur les deux premiers étages. On suppose que le programme chargé est celui de l'exercice précédent mais que l'exécution commence à l'adresse `myst` avec le registre R0 contenant la valeur `x3003`, c'est à dire celle de l'étiquette `str`. Donner les aléas de pipeline rencontrés par l'exécution des 10 premières instructions après le début de la routine `myst`. Pour chacun de ces aléas,

1. donner son type ;
2. dire s'il est possible de l'éliminer résoudre ;
3. préciser s'il provoque l'apparition de bulles dans le pipeline.