

NOM, Prénom, numéro d'étudiant :

CC-TD-1B - LIF6 - lundi 7/3/16 - noté sur 10 - durée : 30 minutes

Aucun document autorisé; calculatrices, téléphones et ordinateurs interdits. Le barème pourra être modifié.

EXERCICE 1. Un processeur est doté d'une mémoire de 256 o : à chaque adresse correspond une case de 1 octet. Il dispose de 4 registres généraux, indicés de 0 à 3 et notés R0...R3. Les instructions sont codées sur 1 ou 2 octets :

assembleur	action	premier octet								second octet	
		7	6	5	4	3	2	1	0	7...0	
ST RS,adr	mem[adr] <- RS	0	1	0	1	RS	0	0	adr		
LD RD,adr	RD <- mem[adr]	0	1	0	0	RD	0	0	adr		
NOT RD,RS	RD <- ~RS en complément à 2 sur 8 bits	1	0	1	1	RD	RS		—		
ADD RD,RS	RD <- RD + RS en complément à 2 sur 8 bits	1	0	1	0	RD	RS		—		
INC RD,RS	RD <- RS + 1 en complément à 2 sur 8 bits	1	0	0	1	RD	RS		—		
HALT	Met fin à l'exécution du programme	0	0	0	0	0	0	0	0	—	

Dans ce tableau :

- RD désigne un registre de destination et RS un registre source (R0...R3);
- adr désigne une adresse codée sur 1 octet, et mem[adr] est la case mémoire d'adresse adr;
- ~RS est la négation bit-à-bit de RS. Par exemple, $\sim(01001101)_2 = (\overline{01001101})_2 = (10110010)_2$.

Le programme suivant calcule mem[c] <- mem[a] - mem[b] (il n'y a pas de dépassement de capacité ici) :

```
LD R0,a      // R0 <- mem[a]
LD R1,b      // R1 <- mem[b]
NOT R2,R1    // R2 <- ~R1
INC R2,R2    // R2 <- R2 + 1
ADD R2,R0    // R2 <- R0 + R2 = mem[a] - mem[b] (en l'absence de dépassement)
ST R2,c      // mem[c] <- R0
HALT        // fin du programme
```

```
a:      -32      // constante décimale stockée en complément à 2 sur 8 bits
b:       81      // constante décimale stockée en complément à 2 sur 8 bits
c:        0      // emplacement où sera stocké le résultat
```

Question 1 (1 pt). Que vaut R2 après l'exécution des instructions NOT R2,R1 et INC R2,R2 (on attend une formule, pas un nombre) ?

Question 2 (1 pt). Comment sont codés $(-32)_{10}$ et $(81)_{10}$ en complément à 2 sur 8 bits? Posez vos calculs, et notez proprement vos résultats.

.....

.....

.....

.....

.....

.....

$(-32)_{10} =$ $(81)_{10} =$

Question 3 (1,5 pt). Posez, en complément à 2 sur 8 bits, l'opération $(-32)_{10} + (-81)_{10}$; encadrez votre résultat. Convertissez votre résultat en décimal en posant vos calculs.

.....

.....

.....

.....

.....

.....

Question 4 (0.5 pt). Après l'exécution de l'instruction `ADD R2,R0`, donnez le contenu du registre R2 en binaire.

.....

Question 5 (3 pts). Le tableau du début de l'exercice est reproduit ici :

assembleur	action	premier octet								second octet	
		7	6	5	4	3	2	1	0	7...0	
<code>ST RS,adr</code>	<code>mem[adr] <- RS</code>	0	1	0	1			RS	0	0	adr
<code>LD RD,adr</code>	<code>RD <- mem[adr]</code>	0	1	0	0			RD	0	0	adr
<code>NOT RD,RS</code>	<code>RD <- ~RS en complément à 2 sur 8 bits</code>	1	0	1	1			RD		RS	—
<code>ADD RD,RS</code>	<code>RD <- RD + RS en complément à 2 sur 8 bits</code>	1	0	1	0			RD		RS	—
<code>INC RD,RS</code>	<code>RD <- RS + 1 en complément à 2 sur 8 bits</code>	1	0	0	1			RD		RS	—
<code>HALT</code>	Met fin à l'exécution du programme	0	0	0	0	0	0	0	0	0	—

Donnez le codage du programme : pour chaque ligne, donnez l'adresse du 1er octet, le codage en binaire, et le codage en hexadécimal de l'instruction ou de la donnée. Rayez les octets qui ne sont pas utilisés.

adresse hexadécimale	instruction ou donnée	premier octet								second octet								codage hexadécimal
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	
<code>1B</code>	<code>LD R0,a</code>																	
	<code>LD R1,b</code>																	
	<code>NOT R2,R1</code>																	
	<code>INC R2,R2</code>																	
	<code>ADD R2,R0</code>																	
	<code>ST R2,c</code>																	
	<code>HALT</code>																	
	<code>a: -32</code>																	
	<code>b: 81</code>																	
	<code>c: 0</code>																	

EXERCICE 2. Cet exercice est disjoint du précédent !

Question 6 (1 pt). Quelle est l'écriture de $(1201)_5$ en décimal ? Posez votre calcul.

.....

Question 7 (2 pts). Quelle est l'écriture de $(5,45)_{10}$ en binaire ? Posez votre calcul.

.....

