

CC1 - LIF6 Automne 2013

Durée prévue : 2h00

Aucun document autorisé. Calculatrices, téléphones et ordinateurs portables interdits. N'oubliez pas de noter sur le sujet qui est à rendre votre numéro de copie. Justifiez brièvement et précisément vos réponses, tout en respectant les emplacements prévus. Le barème est donné à titre indicatif, et pourra être modifié.

1 Langage machine (4,5 pts)

Un processeur 8 bits est doté d'un espace mémoire de 64 Kio ; à chaque adresse en mémoire centrale correspond une case de 1 octet. Le processeur dispose de 4 registres de travail, indicés de 0 à 4 et notés R0...R4. Les instructions sont codées sur 1 ou 2 octets, comme indiqué dans le tableau ci-dessous :

assembleur	action	premier octet								second octet
		7	6	5	4	3	2	1	0	7...0
HALT	Met fin à l'exécution du programme	0	0	0	0	0	0	0	0	—
ADD RD,RS	RD <- RD + RS	0	1	0	1	RD		RS		—
LD RD,adr	RD <- Mémoire[adr]	1	0	1	0	RD		0	0	adr
ST RS,adr	Mémoire[adr] <- RS	1	0	1	1	RS		0	0	adr

Dans ce tableau :

- RD désigne un registre de destination et RS un registre source (R0...R4) ;
- adr désigne une adresse en mémoire, codée sur 1 octet ;
- Mémoire[adr] désigne la case mémoire d'adresse adr.

Un programmeur qui écrit en langage d'assemblage n'a pas besoin de calculer lui-même les adresses de son programme : il utilise simplement une étiquette pour désigner l'adresse d'une instruction ou d'une donnée, et l'assembleur se charge du calcul des adresses. Dans le programme ci-dessous, les étiquettes zero, a et b désignent l'adresse de données ou de résultats en mémoire.

```

LD R0,zero // .....
LD R1,a    // .....
ADD R1,R1  // .....
ADD R0,R1  // .....
ADD R1,R1  // .....
ADD R0,R1  // .....
ST R0,b    // .....
HALT      // .....
zero:    0 // constante décimale 0 sur 8 bits
a:       43 // constante décimale 43 sur 8 bits
b:       0 // case de 8 bits où est stocké le résultat

```

1. (1 point) Ajouter des commentaires au programme ci-dessus, en indiquant l'action effectuée par chaque instruction. Vous utiliserez les étiquettes a et b comme des noms de variables pour clarifier vos commentaires. Exprimez ensuite la valeur de b en fonction de a en fin de programme.

.....

2. (0.5 points) On regarde le contenu des registres et des cases mémoire comme des entiers naturels codés en binaire sur 8 bits. Donnez la valeur décimale que prend b après l'exécution du programme. Détaillez vos calculs.

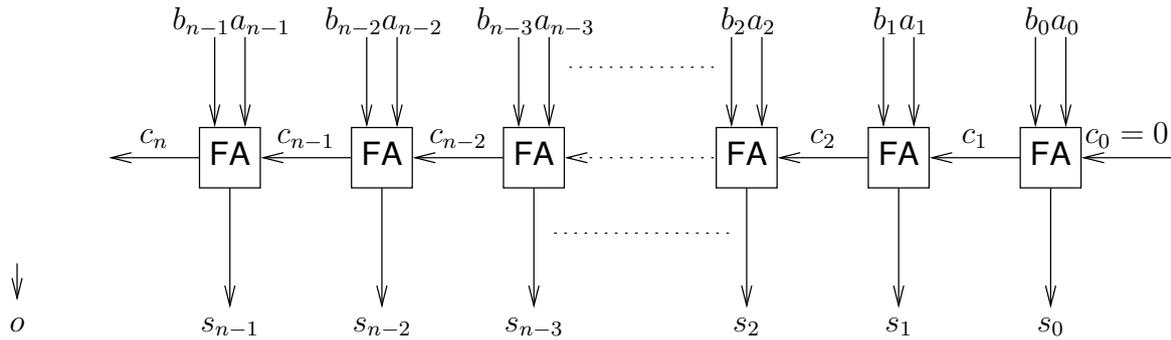
.....

3. (1 point) On suppose que le programme sera rangé en mémoire à partir de l'adresse (1A)_H. Complétez le tableau ci-dessous, en indiquant pour chaque instruction l'adresse de son premier octet et pour chaque donnée son adresse.

adresse	instruction
(1A) _H	LD R0,zero
	LD R1,a
	ADD R1,R1
	ADD R0,R1
	ADD R1,R1
	ADD R0,R1
	ST R0,B
	HALT
	zero: 0
	a: 43
	b: 0

3 Addition-soustraction en complément à 2 sur n bits (7,5 pts)

Soient a et b deux entiers relatifs connus par leur codage en complément à 2 sur n bits ($n \geq 2$) : $a = (a_{n-1}a_{n-2} \dots a_1a_0)_2$ et $b = (b_{n-1}b_{n-2} \dots b_1b_0)_2$. On étudie dans un premier temps l'addition de a et de b à l'aide d'un additionneur à propagation de retenue classique, fabriqué à l'aide de composants *full-adders* (FA) :



On rappelle les éléments suivants sur la représentation en complément à 2.

- Par définition, $(a_{n-1}a_{n-2} \dots a_1a_0)_2 = -a_{n-1}2^{n-1} + \sum_{i=0}^{n-2} a_i2^i$.
- Le bit a_{n-1} est appelé le bit de signe dans la représentation de a en complément à 2 sur n bits.
- Soit $s = (s_{n-1}s_{n-2} \dots s_1s_0)_2$ le résultat calculé par l'additionneur représenté ci-dessus. On dit qu'il y a dépassement de capacité dans le calcul de $a + b$ en complément à 2 sur n bits si $s \neq a + b$ (cela se produit si $a + b$ est soit « trop grand », soit « trop petit » pour être représenté en complément à 2 sur n bits).

3.1 Dépassement de capacité (5 pts)

On souhaite ajouter à notre additionneur une sortie o (comme *overflow*) permettant de détecter si on se trouve en présence d'un cas de dépassement de capacité.

- (1 point) Posez, en complément à 2 sur 8 bits, les additions avec les opérandes a et b indiquées ci-dessous ; A chaque fois, donnez la valeur des opérandes et celle du résultat calculé (qui peut être différent du résultat exact) en décimal, en précisant bien le signe de ces valeurs. Indiquez clairement les opérations pour lesquelles il y a eu dépassement de capacité.

$$\begin{array}{r}
 (\underline{1} \underline{0} \underline{0} \underline{0} \underline{1} \underline{1} \underline{1} \underline{1})_2 \\
 + (\underline{1} \underline{0} \underline{0} \underline{1} \underline{0} \underline{0} \underline{0} \underline{0})_2 \\
 \hline
 (\underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad})_2 \\
 \hookrightarrow (\underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad})_2
 \end{array}
 \qquad
 \begin{array}{r}
 (\underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad})_{10} \\
 + (\underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad})_{10} \\
 \hline
 (\underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad})_{10}
 \end{array}$$

$$\begin{array}{r}
 (\underline{0} \underline{1} \underline{1} \underline{1} \underline{1} \underline{1} \underline{1} \underline{1})_2 \\
 + (\underline{1} \underline{0} \underline{0} \underline{0} \underline{0} \underline{0} \underline{0} \underline{0})_2 \\
 \hline
 (\underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad})_2 \\
 \hookrightarrow (\underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad})_2
 \end{array}
 \qquad
 \begin{array}{r}
 (\underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad})_{10} \\
 + (\underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad})_{10} \\
 \hline
 (\underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad})_{10}
 \end{array}$$

$$\begin{array}{r}
 (\underline{0} \underline{1} \underline{1} \underline{1} \underline{1} \underline{1} \underline{1} \underline{1})_2 \\
 + (\underline{1} \underline{0} \underline{0} \underline{0} \underline{0} \underline{0} \underline{0} \underline{0})_2 \\
 \hline
 (\underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad})_2 \\
 \hookrightarrow (\underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad})_2
 \end{array}
 \qquad
 \begin{array}{r}
 (\underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad})_{10} \\
 + (\underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad})_{10} \\
 \hline
 (\underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad})_{10}
 \end{array}$$

3.2 Additionneur-soustracteur (2,5 pts)

On souhaite maintenant compléter le circuit de manière à ce qu'il puisse aussi effectuer des soustractions en complément à 2 sur n bits. Pour cela, on ajoute une entrée e qui indiquera si le circuit doit calculer l'addition ou la soustraction des deux entiers placés sur ses entrées. En l'absence de dépassement de capacité,

- si $e = 1$ alors $s = a + b$,
- si $e = 0$ alors $s = a - b$.

On note $d = (d_{n-1}d_{n-2} \cdots d_1d_0)$ le mot de n bits tel que

- si $e = 1$ alors $d = b$,
- si $e = 0$ alors $d = \bar{b}$, où \bar{b} désigne le complément à 1 de b .

1. (1 point) Rappelez comment peut être effectuée la soustraction $a - b$ en complément à 2 sur n bits. Vous supposerez dans vos explications qu'aucun problème de dépassement de capacité se produit.

.....

.....

.....

.....

2. (0,5 points) Soit i un entier tel que $0 \leq i < n$. Complétez la table de vérité ci-dessous, et exprimez d_i par une fonction booléenne.

b_i	e	d_i
0	0	
0	1	
1	0	
1	1	

.....

3. (1 point) Donnez le circuit d'un additionneur-soustracteur en complément à 2 sur 4 bits. Veillez à annoter votre schéma (il faut faire apparaître en particulier : les a_i, b_i, c_i, d_i, s_i , ainsi que e et o).

