

**EXERCICES DIRIGES 3-4****LANGAGE MACHINE  
CORRIGE****Exercice 1. Manipulation des modes d'adressages**

LOAD D R0 400	$R0 \leftarrow 2000$
LOAD Im R1 1002	$R1 \leftarrow 1002$
ADD Rg2 R0 R1	$R0 \leftarrow R0 + R1 = 3002$
NEG Rg1 R1	$R1 \leftarrow -1002$
ADD I R1 404	$R1 \leftarrow R1 + ((404)) = -1002 + 3000 = 1998$
STORE B R1 900	$(1000) \leftarrow 1998$

**Exercice 2. Manipulation des modes d'adressage**

	LOAD Im R1 6	$R1 \leftarrow 6$
	LOAD Im R0 0	$R0 \leftarrow 0$
	LOAD I RB 1024	$RB \leftarrow 996$
Boucle:	LOAD B R2 4	$R2 \leftarrow 15, 45, 32, 1256, 253, 256$
	ADD Rg2 R0 R2	$R0 \leftarrow R0 + R2 = 15, 60, 92, 1348, 1601, 1857$
	ADD Im RB 4	$RB \leftarrow RB + 4 = 1000, 1004, 1008, 1012, 1016, 1020$
	ADD D R1 1032	$R1 \leftarrow R1 - 1 = 5, 4, 3, 2, 1, 0$
	JMPZ Fin	NR, NR, NR, NR, NR, Rfin
	JMP Boucle	Rboucle, Rboucle, Rboucle, Rboucle, Rboucle
Fin	STORE B R0 16	$R0 \rightarrow (1036)$

*Rboucle, saut à boucle ; Rfin, saut à fin ; NR saut non réalisé.*

Ce programme additionne les contenus des cases mémoires 1000 à 1020 et stocke le résultat final à l'adresse 1036.

**Exercice 3**

Q1/ **deux accès mémoire**

Q2/ **FAUX**, il contient l'adresse de la prochaine instruction à exécuter

Q3/ 4 Mmots de 32 bits =  $2^2 * 2^{20} * 2^2$  octets =  $2^{24}$  octets, **donc au minimum 24 bits.**

## Exercice 4. Code en langage d'assemblage

### Question 1

```
LOAD D R1 A
LOAD D R2 B
ADD Im R2 6
MUL Im R1 5
ADD Rg2 R1 R2
STORE D R1 B
```

### Question 2

Ce petit programme consiste en une boucle qui ajoute la valeur 1 au contenu du registre R1. A chaque ajout de 1, un test sur l'occurrence d'un overflow est réalisé (instruction JMPO 20). Si un overflow a été détecté, le programme sort de la boucle d'ajout de 1 et va enregistrer le contenu de R1 à l'adresse indirecte  $0=(24)$ ). Comme la machine travaille en complément à 2 sur 32 bits, le plus grand nombre positif représentable est  $2^{31} - 1$  soit 2147483647. On en déduit qu'il faudra ici 8 tours de boucle pour sortir en overflow.

## Exercice 5.

### Question 1.

Adresse mémoire	Mot mémoire	Commentaire éventuel
A :		Valeur de la case A non initialisée
B		Valeur de la case B non initialisée
	IN D A	
	LOAD Im R1 -1	R1 est chargé avec la valeur -1
	LOAD D R2 A	R2 est chargé avec le contenu du mot d'adresse A
	JMP Addition	Saut à l'instruction nommée Addition
Fin	STORE D R2 A	R2 est écrit à l'adresse A en mémoire centrale
	STOP	Fin de l'exécution
Addition	ADD Rg2 R2 R1	$R2 = R2 + R1$
	JMP Fin	Saut à l'instruction nommée Fin

Ce programme décrémente d'une unité la valeur placée dans le mot d'adresse A.

**Question 2.**

**A/** Les indicateurs S, C, O, Z permettent de positionner les propriétés du dernier calcul réalisé par l'UAL. Ainsi S permet d'indiquer si le résultat produit par l'UAL est positif ( $\geq 0$ ) ou négatif, O permet d'indiquer l'occurrence d'un dépassement de capacité, C d'un carry. Enfin Z permet d'indiquer si le résultat produit par l'UAL est nul ou non.

**B/** Lors de l'exécution du programme, l'instruction `ADD Rg2 R2 R1` effectue l'opération  $1 + (-1)$  en complément à 2 sur 8bits.

$$\begin{array}{r} 0000\ 0001 : (1)_{10} \\ + 1111\ 1111 : (-1)_{10} \\ \hline 1\ 0000\ 0000 : (0)_{10} \end{array} \quad \text{: il y a **carry** et **résultat nul**}$$

Les indicateurs du registre PSW sont alors positionnés comme suit :

**S** indique un résultat positif (**S est à 0**); **C à 1** indique qu'il y a un carry généré par l'addition des deux bits de poids le plus fort; **O à 0** indique qu'il n'y a pas de dépassement de capacité ;

**Z** indique un résultat nul (**Z est à 1**);

S	C	O	Z
0	1	0	1

**C/** Lors de l'exécution du programme, l'instruction `ADD Rg2 R2 R1` effectue l'opération  $-128 + (-1)$ . Le résultat de cette opération est  $(-129)_{10}$ . Les nombres étant codés selon la convention en complément à 2 sur 8 bits, un overflow est généré car l'intervalle de représentation est  $[-128 ; +127]$ .

Posons cette opération en binaire.

$$\begin{array}{r} 1000\ 0000 : (-128)_{10} \\ + 1111\ 1111 : (-1)_{10} \\ \hline 1\ 0111\ 1111 : (+127)_{10} \end{array} \quad \text{: il y a **carry** et **overflow**}$$

Les indicateurs du registre PSW sont alors positionnés comme suit :

**S à 0** indique un résultat positif ; **C à 1** indique qu'il y a carry ; **O à 1** indique qu'il y a un dépassement de capacité ; **Z à 0** indique un résultat non nul ;

S	C	O	Z
0	1	1	0

**Question 3.**

Adresse mémoire	Mot mémoire
A :	
B :	
	IN D A
	LOAD Im R1 -1
	LOAD D R2 A
	JMP Addition
Fin	STORE D R2 A
	STOP
Addition	ADD Rg2 R2 R1
	<b>JMPO Overflow</b>
	<b>JMP Fin</b>
<b>Overflow</b>	<b>STORE D R2 B</b>
	<b>STOP</b>