

# T.P. 1 – Corrigé

## Premiers pas en assembleur 68000

### Étape 5

1. Déterminez manuellement (sans l'aide de l'assembleur ni du débogueur) le résultat des additions suivantes, ainsi que le contenu des bits **N**, **Z**, **V** et **C** du registre d'état.

- Addition sur 8 bits : \$B4 + \$4C

\$B4 + \$4C = \$100 (le résultat sur 8 bits est \$00.)

**N = 0, Z = 1, V = 0 et C = 1**

- Addition sur 16 bits : \$B4 + \$4C

\$00B4 + \$004C = \$0100

**N = 0, Z = 0, V = 0 et C = 0**

- Addition sur 16 bits : \$4AC9 + \$D841

\$ 4AC9 + \$D841 = \$1230A (le résultat sur 16 bits est \$230A.)

**N = 0, Z = 0, V = 0 et C = 1**

- Addition sur 32 bits : \$FFFFFFFF + \$00000015

\$FFFFFFFF + \$00000015 = \$100000014 (le résultat sur 32 bits est \$00000014.)

**N = 0, Z = 0, V = 0 et C = 1**

- Le *flag* **N** est le bit de signe du résultat. Il prend donc la valeur du bit de poids fort du résultat.
- Le *flag* **Z** est à 1 si le résultat est nul.
- Le *flag* **C** est à 1 s'il y a une retenue.
- Le *flag* **V** se détermine en faisant la **supposition** que les nombres à additionner sont signés. Il est à 1 uniquement si l'une des deux conditions suivantes est vraie :
  - On additionne deux nombres positifs et le résultat est négatif ;
  - On additionne deux nombres négatifs et le résultat est positif.

2. Servez-vous du débogueur afin de vérifier que vos réponses à la question précédente sont correctes. Pour cela, réalisez un programme en assembleur 68000 qui effectue les quatre additions ci-dessus. Assemblez votre programme et exécutez-le à l'aide du débogueur après avoir localisé où ce dernier affiche les bits du registre d'état.

- Il existe plusieurs solutions possibles. Vous en trouverez une ci-après.
- Exécutez le code pas à pas afin de vérifier les sommes et les *flags*.

```

org      $4
Vector_001 dc.l   Main

```

```

org      $500

```

```

Main
; Addition sur 8 bits.
move.b  #$b4,d0
move.b  #$4c,d1
add.b   d0,d1

; Addition sur 16 bits.
move.w  #$b4,d0
move.w  #$4c,d1
add.w   d0,d1

; Addition sur 16 bits.
move.w  #$4ac9,d0
move.w  #$d841,d1
add.w   d0,d1

; Addition sur 32 bits.
move.l  #$ffffffff,d0
move.l  #$15,d1
add.l   d0,d1

```

## Étape 6

On souhaite réaliser l'addition de deux nombres entiers codés sur 128 bits. Proposez un programme en assembleur 68000 qui effectue cette addition. Vous respecterez les indications suivantes :

Entrées : **D3:D2:D1:D0** = Entier sur 128 bits (**D0** étant les 32 bits de poids faible).

**D7:D6:D5:D4** = Entier sur 128 bits (**D4** étant les 32 bits de poids faible).

Sortie : **D3:D2:D1:D0** = **D3:D2:D1:D0** + **D7:D6:D5:D4**

	C3	C2	C1	C0		
	⋮		D3	D2	D1	D0
+	⋮		D7	D6	D5	D4
	↓					
	C3		D3	D2	D1	D0

```

add.l   d4,d0      ; D4 + D0      -> D0, C0 -> X
addx.l  d5,d1      ; D5 + D1 + X -> D1, C1 -> X
addx.l  d6,d2      ; D6 + D2 + X -> D2, C2 -> X
addx.l  d7,d3      ; D7 + D3 + X -> D3, C3 -> X

```

**Étape 7**

En utilisant uniquement des instructions de rotation, donnez quelques instructions qui modifient la valeur de **D1** afin de lui donner les valeurs ci-dessous. Pour chaque cas, la valeur initiale de **D1** est \$76543210.

- **D1** = \$76543120

```

ror.w #4,d1 ; D1 = $ 7654 3210
ror.b #4,d1 ; D1 = $ 7654 0321
rol.w #4,d1 ; D1 = $ 7654 0312
rol.w #4,d1 ; D1 = $ 7654 3120

```

- **D1** = \$75640213

```

rol.w #4,d1 ; D1 = $ 7654 3210
ror.b #4,d1 ; D1 = $ 7654 2103
ror.w #4,d1 ; D1 = $ 7654 2130
swap d1 ; D1 = $ 7654 0213
ror.w #4,d1 ; D1 = $ 0213 7654
ror.w #4,d1 ; D1 = $ 0213 4765
rol.w #4,d1 ; D1 = $ 0213 4756
swap d1 ; D1 = $ 0213 7564
swap d1 ; D1 = $ 7564 0213

```

- **D1** = \$54231067

```

ror.l #8,d1 ; D1 = $ 7654 3210
ror.b #4,d1 ; D1 = $ 1076 5432
ror.b #4,d1 ; D1 = $ 1076 5423
swap d1 ; D1 = $ 5423 1076
ror.b #4,d1 ; D1 = $ 5423 1067

```

- **D1** = \$05634127

```

ror.l #4,d1 ; D1 = $ 7654 3210
ror.b #4,d1 ; D1 = $ 0765 4321
ror.l #8,d1 ; D1 = $ 0765 4312
ror.b #4,d1 ; D1 = $ 1207 6543
ror.l #4,d1 ; D1 = $ 1207 6534
ror.l #8,d1 ; D1 = $ 3412 0765
ror.b #4,d1 ; D1 = $ 3412 0756
ror.l #8,d1 ; D1 = $ 5634 1207
ror.b #4,d1 ; D1 = $ 5634 1270
ror.l #4,d1 ; D1 = $ 5634 1270
ror.l #4,d1 ; D1 = $ 0563 4127

```