



Systèmes Embarqués 1 & 2

Classes T-2/I-2 // 2017-2018

a.05 – Traitement des nombres

Solutions

1 Exercices sur les nombres entiers non-signés

Exercice 1

Convertir en binaire les nombres suivants

- a. 125_{10}
- b. 377_8
- c. $ADE1_{16}$

Solution:

- a. $125_{10} \rightarrow 0111'1101$
- b. $377_8 \rightarrow 1111'1111$
- c. $ADE1_{16} \rightarrow 1010'1101'1110'0001$

Exercice 2

Convertir en décimal les nombres suivants

- a. 10011000_2
- b. 177_8
- c. $25E1_{16}$

Solution:

- a. $10011000_2 \rightarrow 152$
- b. $177_8 \rightarrow 127$
- c. $25E1_{16} \rightarrow 9697$

Exercice 3

Additionner les nombres binaires suivants et donner l'état des flags C & Z

- a. $10011000_2 + 10011000_2$
- b. $11111101_2 + 00000011_2$
- c. $00011000_2 + 10011100_2$

Solution:

- a. $10011000_2 + 10011000_2 \rightarrow 0011'0000_2 \quad C = 1, Z = 0$
- b. $11111101_2 + 00000011_2 \rightarrow 0000'0000_2 \quad C = 1, Z = 1$
- c. $00011000_2 + 10011100_2 \rightarrow 1011'0100_2 \quad C = 0, Z = 0$

**Exercice 4**

Soustraire les nombres binaires suivants et donner l'état des flags C & Z

- a. $10011000_2 - 10011000_2$
- b. $11111101_2 - 00000011_2$
- c. $00011000_2 - 10011100_2$

Solution:

- a. $10011000_2 - 10011000_2 \rightarrow 0000'0000_2 \quad C = 1, Z = 1$
- b. $11111101_2 - 00000011_2 \rightarrow 1111'1010_2 \quad C = 1, Z = 0$
- c. $00011000_2 - 10011100_2 \rightarrow 0111'1100_2 \quad C = 0, Z = 0$

Exercice 5

Donner l'état des flags C & Z pour les comparaisons suivantes

- a. `cmp 125, 128`
- b. `cmp 77, 26`
- c. `cmp 254, 254`
- d. `cmp 255, 0`

Solution:

- a. `cmp 125, 128` $\rightarrow 0111'1101_2 - 1000'0000_2 = 1111'1101_2 \quad C = 0, Z = 0$
- b. `cmp 77, 26` $\rightarrow 0100'1101_2 - 0001'1010_2 = 0011'0011_2 \quad C = 1, Z = 0$
- c. `cmp 254, 254` $\rightarrow 1111'1110_2 - 1111'1110_2 = 0000'0000_2 \quad C = 1, Z = 1$
- d. `cmp 255, 0` $\rightarrow 1111'1111_2 - 0000'0000_2 = 1111'1111_2 \quad C = 1, Z = 0$



2 Exercices sur les nombres entiers signés

Exercice 1

Convertir en binaire les nombres suivants et indiquer l'état du flag N

- a. -125_{10}
- b. 271_8
- c. $50F_{16}$

Solution:

- a. $-125_{10} \rightarrow 1000'0011 \quad N = 1$
- b. $271_8 \rightarrow 1011'1001 \quad N = 1$
- c. $50F_{16} \rightarrow 0101'0000'1111'0001 \quad N = 0$

Exercice 2

Convertir en décimal les nombres suivants

- a. 10011000_2
- b. 177_8
- c. $85E_{16}$

Solution:

- a. $10011000_2 \rightarrow -104 \quad (152)$
- b. $177_8 \rightarrow 127 \quad (127)$
- c. $85E_{16} \rightarrow -31263 \quad (34273)$

Exercice 3

Additionner les nombres binaires suivants et donner l'état des flags V & N & Z

- a. $10011000_2 + 10011000_2$
- b. $11111101_2 + 00000011_2$
- c. $00011000_2 + 10011100_2$

Solution:

- a. $10011000_2 + 10011000_2 \rightarrow 0011'0000_2 \quad V = 1, N = 0, Z = 0$
- b. $11111101_2 + 00000011_2 \rightarrow 0000'0000_2 \quad V = 0, N = 0, Z = 1$
- c. $00011000_2 + 10011100_2 \rightarrow 1011'0100_2 \quad V = 0, N = 1, Z = 0$

Exercice 4

Soustraire les nombres binaires suivants et donner l'état des flags V & N & Z

- a. $10011000_2 - 10011000_2$
- b. $11111101_2 - 00000011_2$
- c. $00011000_2 - 10011100_2$

Solution:

- a. $10011000_2 - 10011000_2 \rightarrow 0000'0000_2 \quad V = 0, N = 0, Z = 1$
- b. $11111101_2 - 00000011_2 \rightarrow 1111'1010_2 \quad V = 0, N = 1, Z = 0$
- c. $00011000_2 - 10011100_2 \rightarrow 0111'1100_2 \quad V = 0, N = 0, Z = 0$

Exercise 5

Donner l'état des flags V & N & Z pour les comparaisons suivantes

- cmp 127, -125
- cmp 77, -26
- cmp -30, -34
- cmp 55, 66

Solution:

- | | | | | |
|----|---------------|---|---|-----------------------|
| a. | cmp 127, -125 | → | $0111'1111_2 - 1000'0011_2$ | |
| | | → | $0111'1111_2 + 0111'1101_2 = 1111'1100_2$ | $V = 1, N = 1, Z = 0$ |
| b. | cmp 77, -26 | → | $0100'1101_2 - 1110'0110_2$ | |
| | | → | $0100'1101_2 + 0001'1010_2 = 0110'0111_2$ | $V = 0, N = 0, Z = 0$ |
| c. | cmp -30, -34 | → | $1110'0010_2 - 1101'1110_2$ | |
| | | → | $1110'0010_2 + 0010'0010_2 = 0000'0100_2$ | $V = 0, N = 0, Z = 0$ |
| d. | cmp 55, 66 | → | $0011'0111_2 - 0100'0010_2$ | |
| | | → | $0011'0111_2 + 1011'1110_2 = 1111'0101_2$ | $V = 0, N = 1, Z = 0$ |



3 Evaluation de petits codes en assembleur

Prévoir l'état des fanions Z, C, N et V ainsi que le résultat contenu dans le registre R2 suite à l'exécution des instructions assembleur ci-dessous

Remarque : on considère que le μP est capable de traiter des mots de 8bits !

Exercice 1

```
ldr    r2, =128
ldr    r1, =-128
cmp    r2, r1
```

Solution:

→ $1000'0000_2 - 1000'0000_2$
 → $1000'0000_2 + 1000'0000_2 = 0000'0000_2$

N=0 Z=1 C=1 V=1 R2 (signé)=-128 R2(non signé)=128

Exercice 2

```
ldr    r0, =64
ldr    r1, =-128
adds   r2, r0, r1
```

Solution:

→ $0100'0000_2 + 1000'0000_2$
 → $0100'0000_2 + 1000'0000_2 = 1100'0000_2$

N=1 Z=0 C=0 V=1 R2 (signé)=-64 R2(non signé)=192

Exercice 3

```
ldr    r0, =228
ldr    r1, =128
subs   r2, r0, r1
```

Solution:

→ $1110'0100_2 - 1000'0000_2$
 → $1110'0100_2 + 1000'0000_2 = 0110'0100_2$

N=0 Z=0 C=1 V=0 R2 (signé)=100 R2(non signé)=100

Exercice 4

```
ldr    r0, =240
ldr    r1, =-16
subs   r2, r0, r1
```

Solution:

→ $1111'0000_2 - 1111'0000_2$
 → $1111'0000_2 + 0001'0000_2 = 0000'0000_2$

N=0 Z=1 C=1 V=0 R2 (signé)=0 R2(non signé)=0

**Exercice 5**

```
ldr    r2, =0
ldr    r1, =0
cmp    r2, r1
```

Solution:

→ $0000'0000_2 - 0000'0000_2$

→ $0000'0000_2 + 0000'0000_2 = 0000'0000_2$

N=0

Z=1

C=1

V=0

R2 (signé)=0

R2(non signé)=0



4 Interprétation de petits codes en assembleur

Considérer les 5 codes assembleurs ci-dessous. Pour chacun d'eux définir l'état des fanions Z, C, N et V ainsi que l'interprétation du résultat en valeur signée et non-signée.

Remarque : on considère que le μP est capable de traiter des mots de 8bits !

Exercice 1

```
ldr    r0, #-7
ldr    r1, #249
adds   r2, r0, r1
```

Solution:

→ $1111'1001_2 + 1111'1001_2$
 → $1111'1001_2 + 1111'1001_2 = 1111'0010_2$

N=1 Z=0 C=1 V=0 R2 (signé)=-14 R2(non signé)=242

Exercice 2

```
ldr    r0, #248
ldr    r1, #-128
adds   r2, r0, r1
```

Solution:

→ $1111'1000_2 + 1000'0000_2$
 → $1111'1000_2 + 1000'0000_2 = 0111'1000_2$

N=0 Z=0 C=1 V=1 R2 (signé)=120 R2(non signé)=120

Exercice 3

```
ldr    r0, #128
ldr    r1, #0
adds   r2, r0, r1
```

Solution:

→ $1000'0000_2 + 0000'0000_2$
 → $1000'0000_2 + 0000'0000_2 = 1000'0000_2$

N=1 Z=0 C=0 V=0 R2 (signé)=-128 R2(non signé)=128

Exercice 4

```
ldr    r0, #62
ldr    r1, #200
subs   r2, r0, r1
```

Solution:

→ $0011'1110_2 - 1100'1000_2$
 → $0011'1110_2 + 0011'1000_2 = 0111'0110_2$

N=0 Z=0 C=0 V=0 R2 (signé)=118 R2(non signé)=118

**Exercice 5**

```
ldr    r0, #-8
ldr    r1, #-96
subs   r2, r0, r1
```

Solution:

$$\rightarrow 1111'1000_2 - 1010'0000_2$$

$$\rightarrow 1111'1000_2 + 0110'0000_2 = 0101'1000_2$$

N=0 Z=0 C=1 V=0 R2 (signé)=88 R2(non signé)=88



5 Conversion de nombres à virgules flottantes

Exercice 1

Représenter en hexadécimal sur 32 bits (simple précision) les valeurs réelles suivantes

(a) 1'048'576

Solution:

$$\begin{aligned}\rightarrow 1'048'576_{10} &= 10'0000_{16} = 1 * 2^{20} \\ \rightarrow S &= 0, E = 20 + 127 = 147 = 1001'0011_2, T = 0...0_2 \\ \rightarrow 1'048'576_{10} &= 0x4980'0000\end{aligned}$$

(b) 2048

Solution:

$$\begin{aligned}\rightarrow 2048_{10} &= 800_{16} = 1 * 2^{11} \\ \rightarrow S &= 0, E = 11 + 127 = 138 = 1000'1010_2, T = 0...0_2 \\ \rightarrow 2048_{10} &= 0x4500'0000\end{aligned}$$

(c) 55.75

Solution:

$$\begin{aligned}\rightarrow 55.75_{10} &= 11'0111.11_2 * 2^0 = 1.1011'1110_2 * 2^5 \\ \rightarrow S &= 0, E = 5 + 127 = 132 = 1001'0100_2, T = 1011'1110_2 \\ \rightarrow 55.75_{10} &= 0x425f'0000\end{aligned}$$

(d) 5/4096

Solution:

$$\begin{aligned}\rightarrow 5_{10}/4096_{10} &= 101_2 * 2^0 / 2^{12} = 1.0100_2 * 2^{-10} \\ \rightarrow S &= 0, E = -10 + 127 = 117 = 0111'0101_2, T = 010...0_2 \\ \rightarrow 5_{10}/4096_{10} &= 0x3aa0'0000\end{aligned}$$

(e) -25/2

Solution:

$$\begin{aligned}\rightarrow -25_{10}/2_{10} &= -1'1001_2 * 2^0 * 2^{-1} = -1.1001_2 * 2^3 \\ \rightarrow S &= 1, E = 3 + 127 = 130 = 1000'0010_2, T = 1001'0...0_2 \\ \rightarrow -25_{10}/2_{10} &= 0xc148'0000\end{aligned}$$