Systèmes Embarqués 1 & 2

Classes T-2/I-2 // 2018-2019

a.14 – Traitement des nombres

Exercices

1 Exercices sur les nombres entiers non-signés

Exercice 1

Convertir en binaire les nombres suivants

a. 125_{10}
b. 377_{8}
c. $ADE1_{16}$

Exercice 2

Convertir en décimal les nombres suivants

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

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. $00011000_2 + 10011100_2$

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. $00011000_2 - 10011100_2$

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

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. $50F1_{16}$

Exercice 2

Convertir en décimal les nombres suivants

a. 10011000_{2}
b. 177_{8}
c. $85E1_{16}$

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. $00011000_{2} + 10011100_{2}$

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. $00011000_{2} - 10011100_{2}$

Exercice 5

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

a. cmp 127, -125
b. cmp 77, -26
c. cmp -30, -34
d. cmp 55, 66

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
```

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

Exercice 2

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

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

Exercice 3

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

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

Exercice 4

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

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

Exercice 5

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

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

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
```

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

Exercice 2

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

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

Exercice 3

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

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

Exercice 4

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

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

Exercice 5

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

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

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

(b) 2048

(c) 55.75

(d) 5/4096

(e) -25/2