

TD n° 1 : REPRESENTATION DES NOMBRES

1. Représentation des entiers

Soient des entiers en représentation en complément à 2 où un nombre N correspond à

$$N = -a_{n-1}2^{n-1} + \sum_{i=0}^{i=n-1} a_i 2^i$$

Rappel : dans cette représentation, on peut obtenir l'opposé -N d'un nombre N en prenant le complément à 1 de N (complémentation bit à bit), puis en ajoutant 1.

Additions n bits + n bits et soustractions n bits – n bits

Faire les additions suivantes sur un octet et indiquer les cas de débordement :

15 _H + 48 _H	72 _H + F9 _H	7D _H – 3F _H
F5 _H + AF _H	47 _H + 3A _H	79 _H – 89 _H
15 _H + A3 _H	81 _H + 95 _H	A9 _H – FF _H

Dans quelles conditions le résultat est-il correct ?

Additions n bits + p bits (avec p < n)

Faire les additions suivantes :

1560 _H + 48 _H	7200 _H + F9 _H
F500 _H + AF _H	47F0 _H + 3A _H
15FF _H + A3 _H	8100 _H + 95 _H

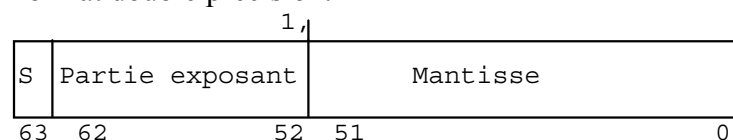
2. Nombres en représentation “virgule flottante” (Standard IEEE 754)

Le codage est donné dans la table ci-dessous où s est le bit de signe, e est la partie exposant et f représente la partie fractionnaire après le 1 implicite.

e	f	représente
0 ; s ± 1	0	±0
0	≠ 0	s x 0,fx2 ^{E_{min}}
0 < e < e _{max}	quelconque	s x 1,fx2 ^E
e _{max}	0	±∞
e _{max}	≠ 0	NaN

La partie exposant en double précision représente l'exposant réel + 1023. La partie exposant en simple précision représente l'exposant réel + 127

Format double précision:



En simple précision, l'exposant occupe 8 bits et la mantisse 23.

1) Quels nombres simple précision correspondent aux mots de 32 bits suivants :

- a) 41300000H
- b) 41E00000H
- c) BF800001H
- d) 00A00000H

2) Ecrire 1 et -1000 de façon normalisée.

3) Donner le plus grand positif et son prédécesseur, indiquer leur écart; le plus petit positif normalisé et dénormalisés; le plus grand et le plus petit négatif.

3. CONVERSIONS

Soient les déclarations C suivantes :

```
int x ;float f ; double d ;
```

où x est un entier sur 32 bits, f est un flottant 32 bits (simple précision) et d un flottant 64 bits double précision.

On utilise les opérateurs de conversion de C.

Indiquer si les assertions suivantes sont vraies ou fausses, en justifiant

- a) $x == (int)(float) x$
- b) $x == (int)(double) x$
- c) $f == (float)(double) f$
- d) $d == (float) d$
- e) $f == -(-f)$
- f) $2/3 == 2/3.0$
- g) $d < 0.0 == (2*d) < 0.0$
- h) $d > f == -f < d$
- i) $d*d >= 0.0$
- j) $(d+f) - d == f$

4. Représentation flottante sur processeur traitement du signal

Certains processeurs enfouis (ex : Blackfin chez Analog Devices) n'ont pas d'opérateurs flottants. Le processeur Blackfin peut effectuer des opérations sur des entiers 16 bits (opérations arithmétiques, logiques et décalage). On utilise 2×16 bits pour représenter un nombre flottant.

La fraction est représentée en complément à 2, dans un format fixe de type 1,15, où le premier bit est le bit de signe et les quinze bits suivants sont les bits après la virgule. La fraction est normalisée quand les deux premiers bits d'un nombre positif sont 01 et que les deux premiers bits d'un nombre négatif sont 10. L'exposant est représenté sur 16 bits en complément à 2.

Quels sont les nombres entiers représentables avec le format fixe 1,15 ?

Quels sont les plus petit (normalisés) et plus grand nombres positifs représentables ? Même question pour les nombres négatifs. Quel est le plus petit positif dénormalisé représentable ? Même question pour le plus grand négatif. Comment peut on représenter 0 avec une telle représentation ? Peut-on représenter l'infini ?