**MATHS NUMERATION**

**Ex 5 (du pdf) :**

1010 1101

2^0 + 2^2+2^3+2^5+2^7 = 173

Flip flop :

**B :** 10000101

Flip flop : 01111010

+1 : 01111011

= -123

**C :** 10011111

Flip flop : 01100000

+1 : 01100001

= -97

**2) d** : 1001 0010 1100 0111

Flip flop : 0110 1101 0011 1000

+1 : 0110 1101 0011 1001

= : -27961

**E**: 0011 10001 0010 1111.

=

**6)**

-53 : 1110101

**Soustraction en binaire :**

Soustraire c’est additionner l’opposé

2-15=-13

2 = 0000 0010

15 = 0000 1111

* Flip flop = 1111 0000
* +1 = 1111 0001

Ou :

#

-15 =(1) 111 0001

=

(1) 111 0011

= 128 – 13 = 115

**Codage binaire des décimaux :**

Quand un nb décimal est un nb réél qui ne contient qu’un nombre fini de chiffre après la virgule

Ex : 121,35

La partie E(12135) = 121

121 en binaire : 1111001

On code alors : 0.35 en binaire de la façon suivante :

0.35

0,35 x 2 = **0**,70

0,70 \* 2 = **1**,40

0,40 \* 2 = **0**,80

0,80 \*2 = **1**,60

0,60 \* 2 = **1**,20

0,20\*2 = **0**,40

0,40 \* 2 = **0**,80

121,35 = 1111001 , 01 0110 0110 0110 …

**Exercice 8 : conversion de parties fractionnaires**

1. Coder en binaire sur 8 bits les nombres a = 0; 57812510 et b = 0; 8510.

2. Convertir en décimal les nombres c = 0; 10 11 00 002 et d = 0; 11 01 10 012.

1. 0,578125

…

2. Convertir en décimal les nombres c = 0; 10 11 00 002 et d = 0; 11 01 10 012.

c) 0,10110000

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |  |
| 0 | ½ | ¼ | 1/8 | 1/16 |  |  |  |  |  |

= 0,5+0,25+0,125+0,0625

=0,9375

d) 0,11011001

0+1/2+1/4+0+1/16+1/32+0+0+1/256

=0+0.5+0.25+0+0.0625+0.03125+0+0+0.00390625 =

0,84765625

**COURS**

Norme IEEEE 754 pour coder les réelles à virgule flotantes

Soit le nombre réel 11,8

Ratio entiere : 11, en binaire : 1011

On a donc 11,85 (base 10) qui donne en binaire pour 1011,11 01110 0110 … (= x)

* 1,0

On donne les règles en première position (=y)

… 1111

… 0110 0110

On à donc déplacé la virgule de trois positions à partir de la gauche

On suppose que l’on dispose d’un mot de 32 bits (simple prénom) en norme IEEE 754 on par :

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 = positif, 0 négatif | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

Bit de signe ----------------------------------------------------------------------------------------- = 1 octet pour l’exposant : et dans la norme on calcule 127 + 3 = 130 et l’on code en binaire 130 dans cet octet

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **0** | 1111 | 0110 | 0110 | 0110 |
|  |  |  |  |  |

🡨----------------------------------------------------------------------------🡪 23 bits de plus = mantisse

Exercice 9 : virgule flottante

Coder (sur 32 bits, avec la norme IEEE 754) les nombres a = 40, b = -0; 078125, c = 13; 625 et d = -87; 375.

40,0

40 : 32+8 = 00101000

Soit

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |

0 : 0,0000000000000000000000

Donc : 40,0 = 101000, 0000000000000000000000

Ou : ~~1~~,0100000 \*2^5 = 5+127 = 132

= **0 (car positif) 1**000100 (exposant) **010 … 0 (mantisse : 23)**