Architecture des Ordinateurs, corrigé TD 1

Numération élémentaire

Exercice 1. Calculer 2^8, 2^9, 2^{10}, 2^{15}, 2^{16}, 2^{32}.

Correction.

$$\begin{aligned}
2^8 &= 2^7 \times 2 = 128 \times 2 = 256 \\
02^9 &= 2^8 \times 2 = 256 \times 2 = 512 \\
2^{10} &= 2^8 \times 2^2 = 256 \times 4 = 1024 \\
2^{15} &= 2^{10} \times 2^5 = 1024 \times 32 = 32768 \\
2^{16} &= 2^{15} \times 2 = 32768 \times 2 = 65536 \\
2^{32} &= 2^{16} \times 2^{16} = 65536 \times 65536 = 4294967296
\end{aligned}$$

Exercice 2. Convertir en binaire, puis en octal, et enfin en héxadécimal les nombres suivants : 100, 127, 128, 256, 1000, 1023, 1024, 10000.

Correction. *La méthode des divisions successives par deux est longue et fastidieuse... On lui préférera la méthode des approximations successives par les puissances de deux.*

- *Conversion de* 100 *:*

  $$\begin{array}{lll}
  64 \leq 100 < 128 & \textit{donc reste} & 100 - 64 = 36 \\
  32 \leq 36 < 64 & \textit{donc reste} & 36 - 32 = 4 \\
  4 \leq 4 < 8 & \textit{donc reste} & 4 - 4 = 0
  \end{array}$$

  Par conséquent 100 *s'écrit en binaire* 1100100_2, 144_8 *en octal,* 64_{16} *en héxadécimal.*

- *Conversion de* 127 *:*

  $$\begin{array}{lll}
  64 \leq 127 < 128 & \textit{donc reste} & 127 - 64 = 63 \\
  32 \leq 63 < 64 & \textit{donc reste} & 63 - 32 = 31 \\
  16 \leq 31 < 32 & \textit{donc reste} & 31 - 16 = 15 \\
  8 \leq 15 < 16 & \textit{donc reste} & 15 - 8 = 7 \\
  4 \leq 7 < 8 & \textit{donc reste} & 7 - 4 = 3 \\
  2 \leq 3 < 4 & \textit{donc reste} & 3 - 2 = 1 \\
  1 \leq 1 < 2 & \textit{donc reste} & 1 - 1 = 0
  \end{array}$$

  Par conséquent 127 *s'écrit en binaire* 1111111_2, 177_8 *en octal,* $7F_{16}$ *en héxadécimal.*

- *Conversion de* 128 *:* $128 = 2^7$ *donc un bit à un suivi de 7 zéros :* 10000000_2 *en binaire,* 200_8 *en octal,* 80_{16} *en héxadécimal.*

  Remarque : $127 = 128 - 1$, *or* $128 = 2^7$ *donc un bit à un suivi de 7 zéros.*

- *Conversion de* 256 *:* $256 = 2^8$ *donc un bit à un suivi de 8 zéros :* 100000000_2 *en binaire,* 400_8 *en octal,* 100_{16} *en héxadécimal.*

- *Conversion de* 1000 *:*

$$\begin{array}{lll}
512 \leq 1000 < 1024 & \textit{donc reste} & 1000 - 512 = 488 \\
256 \leq 488 < 512 & \textit{donc reste} & 488 - 256 = 232 \\
128 \leq 232 < 256 & \textit{donc reste} & 232 - 128 = 104 \\
64 \leq 104 < 128 & \textit{donc reste} & 104 - 64 = 40 \\
32 \leq 40 < 64 & \textit{donc reste} & 40 - 32 = 8 \\
8 \leq 8 < 16 & \textit{donc reste} & 8 - 8 = 0
\end{array}$$

  Par conséquent 1000 *s'écrit en binaire* 1111101000_2*,* 1750_8 *en octal,* $3E8_{16}$ *en héxadécimal.*

- *Conversion de* 1023 *:* $1023 = 1024 - 1$ *or* $1024 = 2^{10}$ *donc un bit suivi de 10 zéros. Par conséquent* 1023 *s'écrit en binaire* 1111111111_2*,* 1777_8 *en octal,* $3FF_{16}$ *en héxadécimal.*

- *Conversion de* 1024 *:* $1024 = 2^{10}$ *donc un bit suivi de 10 zéros. Par conséquent* 1024 *s'écrit en binaire* 10000000000_2*,* 2000_8 *en octal,* 400_{16} *en héxadécimal.*

- *Conversion de* 10000 *:*

$$\begin{array}{lll}
8192 \leq 10000 < 16384 & \textit{donc reste} & 10000 - 8192 = 1808 \\
1024 \leq 1808 < 2048 & \textit{donc reste} & 1808 - 1024 = 784 \\
512 \leq 784 < 1024 & \textit{donc reste} & 784 - 512 = 272 \\
256 \leq 272 < 512 & \textit{donc reste} & 272 - 256 = 16 \\
8 \leq 16 < 32 & \textit{donc reste} & 16 - 16 = 0
\end{array}$$

  Par conséquent 10000 *s'écrit en binaire* 10011100010000_2*,* 23420_8 *en octal,* 2710_{16} *en héxadécimal.*

Exercice 3. Convertir en binaire, puis en octal, et enfin en décimal les nombres suivants : $5A_{16}$, $CFBA_{16}$, $E10D_{16}$, FF_{16}, $B00_{16}$, $F000_{16}$, $FFFF_{16}$.

Correction.

- *Conversion de* $5A_{16}$ *:* $5A_{16}$ *s'écrit en* 01011010_2 *en binaire,* 132_8 *en octal, enfin* $5A_{16} = 5 \times 16^1 + 10 \times 16^0 = 80 + 10 = 90$.

- *Conversion de* $CFBA_{16}$ *:* $CFBA_{16}$ *s'écrit en* 1100111110111010_2 *en binaire,* 147672_8 *en octal, enfin* $CFBA_{16} = 12 \times 16^3 + 15 \times 16^2 + 11 \times 16^1 + 10 \times 16^0 = 12 \times 4096 + 15 \times 256 + 11 \times 16 + 10 = 49152 + 3840 + 176 + 10 = 53178$.

- *Conversion de* $E10D_{16}$ *:* $E10D_{16}$ *s'écrit en* 1110000100001101_2 *en binaire,* 160415_8 *en octal, enfin* $E10D_{16} = 14 \times 16^3 + 1 \times 16^2 + 0 \times 16^1 + 13 \times 16^0 = 57344 + 256 + 13 = 57613$.

- *Conversion de* FF_{16} *:* FF_{16} *s'écrit en* 11111111_2 *en binaire,* 377_8 *en octal, enfin* $FF_{16} = 15 \times 16^1 + 15 \times 16^0 = 240 + 15 = 255$.

- *Conversion de* $B00_{16}$ *:* $B00_{16}$ *s'écrit en* 101100000000_2 *en binaire,* 5400_8 *en octal, enfin* $B00_{16} = 11 \times 16^2 = 11 \times 256 = 2816$.

- *Conversion de* $F000_{16}$ *:* $F000_{16}$ *s'écrit en* 1111000000000000_2 *en binaire,* 170000_8 *en octal, enfin* $F000_{16} = 15 \times 16^3 = 15 \times 4096 = 61440$.

- *Conversion de* $FFFF_{16}$ *:* $FFFF_{16}$ *s'écrit en* 1111111111111111_2 *en binaire,* 177777_8 *en octal, enfin* $FFFF_{16} = 15 \times 16^3 + 15 \times 16^2 + 15 \times 16^1 + 15 \times 16^0 = 15 \times 4096 + 15 \times 256 + 15 \times 16 + 15 = 65535$. *On peut remarquer que* $FFFF_{16}$ *c'est* $10000_{16} - 1 = 65536 - 1 = 65535$.

Exercice 4. Soit x une base quelconque,

- montrer que 10101_x est un multiple de 111_x ;
- exprimer le quotient dans les bases 2, 8, 10, 16.

Correction. *Dans la base* x*, le nombre* 10101_x *s'écrit* $1 \times x^4 + 0 \times x^3 + 1 \times x^2 + 0 \times x^1 + 1 \times x^0$ *soit* $x^4 + x^2 + 1$*. le nombre* 111_x *s'écrit* $1 \times x^2 + 1 \times x^1 + 1 \times x^1$ *soit* $x^2 + x + 1$*. On montre que* $x^4 + x^2 + 1$ *est un multiple de* $x^2 + x + 1$*. Pour cela on considère la division euclidienne de* $x^4 + x^2 + 1$ *par* $x^2 + x + 1$ *et dans ce cas, en effectuant la division, on obtient* $x^4 + x^2 + 1 = (x^2 + x + 1)(x^2 - x + 1)$ *:*

```
  4       2             2
 x      + x       + 1  | x  + x + 1
                       |-----------
  4   3   2            |  2
-(x + x  + x )         | x  - x + 1
--------------         |
      3                |
   - x   + 1           |
      3   2            |
 -(- x  - x  - x)      |
 ----------------      |
          2            |
         x  + x + 1    |
          2            |
       -(x  + x + 1)   |
       -------------   |
                       |
              0        |
```

- *en base 2 :* $2^2 - 2 + 1 = 3$*, soit* 11_2
- *en base 8 :* $8^2 - 8 + 1 = 57$*, soit* 71_8
- *en base 10 :* $10^2 - 10 + 1 = 91$
- *en base 16 :* $16^2 - 16 + 1 = 241$ *soit* $F1_{16}$

Nombres signés

Exercice 5. Quel est l'équivalent décimal des nombres signés suivants 101_2 (sur trois bits), 1011_2 (sur quatre bits), 00111001_2 (sur huit bits), 10111001_2 (sur huit bits).

Correction.

- *Sur trois bits de représentation* 101_2 *est un nombre négatif. Son complément à deux est* 010_2 *(complément à un)* $+1$*, soit* 011_2 *(*$+3_{10}$*), donc* 101_2 *vaut* -3_{10}*.*
- *Sur quatre bits de représentation* 1011_2 *est un nombre négatif. Son complément à deux est* 0100_2 *(complément à un)* $+1$*, soit* 0101_2 *(*$+5_{10}$*), donc* 1011_2 *vaut* -5_{10}*.*
- *Sur huit bits de représentation* 00111001_2 *est un nombre positif dont la valeur décimale est* $2^5+2^4+2^3+2^0$*, soit* 57_{10}*.*
- *Sur huit bits de représentation* 10111001_2 *est un nombre négatif. Son complément à deux est* 01000110_2 *(complément à un)*$+1$*, soit* 01000111_2 *(*$2^6+2^2+2^1+2^0=+71_{10}$*), donc* 10111001_2 *vaut* -71_{10}*.*

Exercice 6. Ecrire les compléments à 1 puis à 2 des nombres binaires suivants : 1010101_2, 0111000_2, 0000001_2, 10000_2, 0000_2. Commenter…

Correction.

	Complément à 1	*Complément à 2*
1010101_2	0101010_2	*Compl(1)+1 :* 0101011_2
0111000_2	1000111_2	*Compl(1)+1 :* 1001000_2
0000001_2	1111110_2	*Compl(1)+1 :* 1111111_2
10000_2	01111_2	*Compl(1)+1 :* $\mathbf{1}0000_2$
0000_2	1111_2	*Compl(1)+1 :* $\mathbf{1}0000_2$

Remarquer la quatrième ligne (0 *modulo* 2^4*) et la cinquième ligne (débordement sur le cinquième bit).*

Arithmétique des nombres signés

Exercice 7. Effectuer les opérations arithmétiques suivantes sur 6 bits, les nombres représentés étant signés, puis donner les résultats en décimal :

- $001110_2+110010_2$, $101011_2+111000_2$, $111001_2+001010_2$;
- $010101_2-000111_2$, $111001_2-001010_2$, $101011_2-100110_2$.

Correction.

```
  001110        101011        111001
+ 110010      + 111000      + 001010
--------      --------      --------
 1000000       1100011       1000011


  010101        111001        101011
- 000111      - 001010      - 100110
--------      --------      --------
  001110        101111        000101
```

Remarque :

- *Un résultat dont le* 6ème *bit à gauche (bit de poids fort) est à* 1*, représente en complément à 2 un entier négatif ;*
- *Un résultat dont le* 6ème *bit à gauche (bit de poids fort) est à* 0*, représente en complément à 2 un entier positif ;*
- *Les résultats issus des additions sont donnés sur 7 bits, il y a donc dépassement ; mais en complément à 2, les entiers sont codés modulo* 2^n *(ici* $n = 6$*), et par conséquent l'équivalent décimal d'un résultat sur 7 bits est égal à l'équivalent décimal du même résultat sur 6 bits (autrement dit, il suffit d'enlever le 7ème bit à gauche, bit de poids fort)...*
- *les additions :*
  $C_2(1000000) = C_2(000000 \textit{ modulo } 2^6) = 0$ *sur 6 bits.*
  $C_2(1100011) = C_2(100011 \textit{ modulo } 2^6) = C_1(100011) + 1 = 0011100 + 1 = 0011101 = 2^4 + 2^3 + 2^2 + 2^0 = +29$*, donc la valeur en décimal de* 100011 *(6 bits) est* -29*.*
  $C_2(1000011) = C_2(000011 \textit{ modulo } 2^6) = 2^1 + 2^0 = +3$ *sur 6 bits.*
- *les soustractions :*
  $C_2(001110) = 2^3 + 2^2 + 2^1 = +14$ *sur 6 bits.*
  $C_2(101111) = C_1(101111) + 1 = 010000 + 1 = 010001 = 2^4 + 2^0 = +17$ *donc la valeur en décimal de* 101111 *(6 bits) est* -17*.*
  $C_2(000101) = +5$ *sur 6 bits.*

Exercice 8. Effectuer les opérations arithmétiques suivantes directement en héxadécimal, puis vérifier le résultat en binaire :

- $B7AD_{16} + 51E0_{16}$;
- $8BA2_{16} + 6A7_{16}$;
- $8BA2_{16} - 6A7_{16}$;

Correction.

- *Opérations en héxadécimal :*

$$\begin{array}{r} B7AD_{16} \\ +51E0_{16} \\ \hline 1098D_{16} \end{array} \qquad \begin{array}{r} 8BA2_{16} \\ +6A7_{16} \\ \hline 9249_{16} \end{array}$$

- *Vérification en binaire :*

```
     1011 0111 1010 1101         1000 1011 1010 0010
     0101 0001 1110 0000              0110 1010 0111

------------------------         -------------------
0001 0000 1001 1000 1101         1001 0010 0100 1001

  1    0    9    8    D            9    2    4    9
```

- $8BA2_{16} - 6A7_{16}$ *: Soustraire, c'est additionner à un nombre son opposé i.e. son complément à deux. Le complément à deux (sur 16 bits) de* $06A7_{16}$ *est* $F959_{16}$*. On a alors :*

$$\begin{array}{r} 8BA2_{16} \\ +F959_{16} \\ \hline 84FB_{16} \end{array}$$

Il est bien sûr aussi possible d'effectuer directement la soustraction, en tenant compte du fait que les retenues sont données en base 16. Ainsi dans la soustraction $8BA2_{16} - 6A7_{16}$ *on commence par enlever (en héxadécimal)* 7_{16} *de* 12_{16} *(qui est l'écriture héxadécimale de* 18*), il reste donc* B_{16} *(écriture héxadécimale de* 11*) avec une retenue qui se propage ; on obtient alors (colonne suivante)* $1_{16} + A_{16} = B_{16}$ *enlevé de* $1A_{16}$ *(c'est-à-dire* 11 *enlevé de* 26*) et il reste donc* F_{16} *(écriture héxadécimale de* 15*) avec une retenue qui se propage ; on obtient* $1_{16} + 6_{16} = 7_{16}$ *enlevé de* B_{16} *(écriture héxadécimale de* 11*), il reste donc* 4_{16} *; enfin, on enlève* 0_{16} *de* 8_{16} *et il reste donc* 8_{16}*. En bref :*

$$\begin{array}{r} 8BA2_{16} \\ -6A7_{16} \\ \hline 84FB_{16} \end{array}$$

Exercice 9. Convertir en binaire en passant par l'héxadécimal :

- -5 sur 16 bits, puis sur 32 bits ;
- -23 sur 32 bits.

Correction. *Sur 16 bits :*

$$\begin{array}{r} 10000_{16} \\ -00005_{16} \\ \hline FFFB_{16} \end{array}$$

Sur 32 bits :

$$\begin{array}{r} 100000000_{16} \\ -000000005_{16} \\ \hline FFFFFFFB_{16} \end{array} \qquad \begin{array}{r} 100000000_{16} \\ -000000017_{16} \\ \hline FFFFFE9_{16} \end{array}$$

Nombres fractionnaires

Exercice 10. Convertir en binaire, en virgule fixe :

- $0,48$ avec la partie fractionnaire exprimée sur 6 bits ;
- $0,83$ avec la partie fractionnaire exprimée sur 4 bits ;
- $37,62$ avec la partie fractionnaire exprimée sur 8 bits ;

Correction.

- *Conversion de* $0,48$*, partie décimale sur 6 bits :*

$$\begin{array}{rcl} 0,48 \times 2 & = & 0,96 \rightarrow 0 \\ 0,96 \times 2 & = & 1,92 \rightarrow 1 \\ 0,92 \times 2 & = & 1,84 \rightarrow 1 \\ 0,84 \times 2 & = & 1,68 \rightarrow 1 \\ 0,68 \times 2 & = & 1,36 \rightarrow 1 \\ 0,36 \times 2 & = & 0,72 \rightarrow 0 \end{array}$$

  $0,48$ *s'écrit* $0,011110_2$.

- *Conversion de* $0,83$*, partie décimale sur 4 bits :*

$$\begin{array}{rcl} 0,83 \times 2 & = & 1,66 \rightarrow 1 \\ 0,66 \times 2 & = & 1,32 \rightarrow 1 \\ 0,32 \times 2 & = & 0,64 \rightarrow 0 \\ 0,64 \times 2 & = & 1,28 \rightarrow 1 \end{array}$$

  $0,83$ *s'écrit* $0,1101_2$.

- *Conversion de* $37,62$*, partie décimale sur 8 bits :*
  - *Partie entière :*

$$\begin{array}{lll} 32 \leq 37 & \textit{donc reste} & 37 - 32 = 5 \\ 4 \leq 5 & \textit{donc reste} & 5 - 4 = 1 \\ 1 \leq 1 & \textit{donc reste} & 1 - 1 = 0 \end{array}$$

    donc $37 = 2^5 + 2^2 + 2^0$ *d'où en binaire* 100101_2.

– *Partie décimale :*

$$\begin{aligned}
0,62 \times 2 &= 1,24 \rightarrow 1 \\
0,24 \times 2 &= 0,48 \rightarrow 0 \\
0,48 \times 2 &= 0,96 \rightarrow 0 \\
0,96 \times 2 &= 1,92 \rightarrow 1 \\
0,92 \times 2 &= 1,84 \rightarrow 1 \\
0,84 \times 2 &= 1,68 \rightarrow 1 \\
0,68 \times 2 &= 1,36 \rightarrow 1 \\
0,36 \times 2 &= 0,72 \rightarrow 0
\end{aligned}$$

$0,62$ *s'écrit* $0,10011110_2$.

$37,62$ *s'écrit* $100101,10011110_2$.

Exercice 11. Donner la représentation flottante de $3,14159$ en simple précision dans la norme IEEE 754.

Correction. $3,14159_{10} \rightarrow 11,00100100_2$*, soit après normalisation :* $1,100100100_2 \times 2^1$*. On a donc :*

- 100100100 *représente la mantisse (avec* 1 *significatif implicite)*
- 1 *représente l'exposant*
- *l'exposant corrigé en simple précision est* 010000000 *(*$128 = 1 + 127$*).*

La représentation flottante de $3,14159$ *en simple précision est donc*

0 010000000 10010010000000000000000

.