Électronique Numérique

Numération & codes

POLYTECH ANGERS

Rémy Guyonneau
Sylvain Pezeril

Les systèmes de numération

- Numération décimale
  - Les chiffres en base 10 : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
  - Exemple : $7239 = 7.10^3 + 2.10^2 + 3.10^1 + 9.10^0$
  - $N = (a_{n-1} \ldots a_1\, a_0)_B$
    - La base B est notée en indice
    - N vaut en décimal :
      - $N = a_{n-1}B^{n-1} + \ldots + a_1B^1 + a_0B^0$

Les systèmes de numération

- Numération binaire
  - Les chiffres en base 2 : 0, 1
  - On écrit

    $(a_{n-1} \dots a_1 a_0)_2 = a_{n-1}2^{n-1} + \dots + a_1 2^1 + a_0 2^0$
  - Exemple :

    $(4)_{10} = 1.2^2 + 0.2^1 + 0.2^0$ On commence toujours au rang 0 à droite

    $(4)_{10} = (100)_2$
- Le bit de poids le plus fort est appelé MSB (Most Significant Bit)
- Le bit de poids le plus faible est appelé LSB (Least Significant Bit)

Les systèmes de numération

- Numération hexadécimale
  - Les chiffres en base 16 : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
  - On écrit

    $(N)_{16} = a_{n-1} \ldots a_1\ a_0$

    $(N)_{10} = a_{n-1}16^{n-1} + \ldots + a_1 16^1 + a_0 16^0$
  - Exemple :

    $(CA)_{16} = CA_H = \$CA = 0xCA$

    $(CA)_{16} = C.16^1 + A.16^0 = 12.16 + 10.1$

    $(CA)_{16} = (202)_{10}$

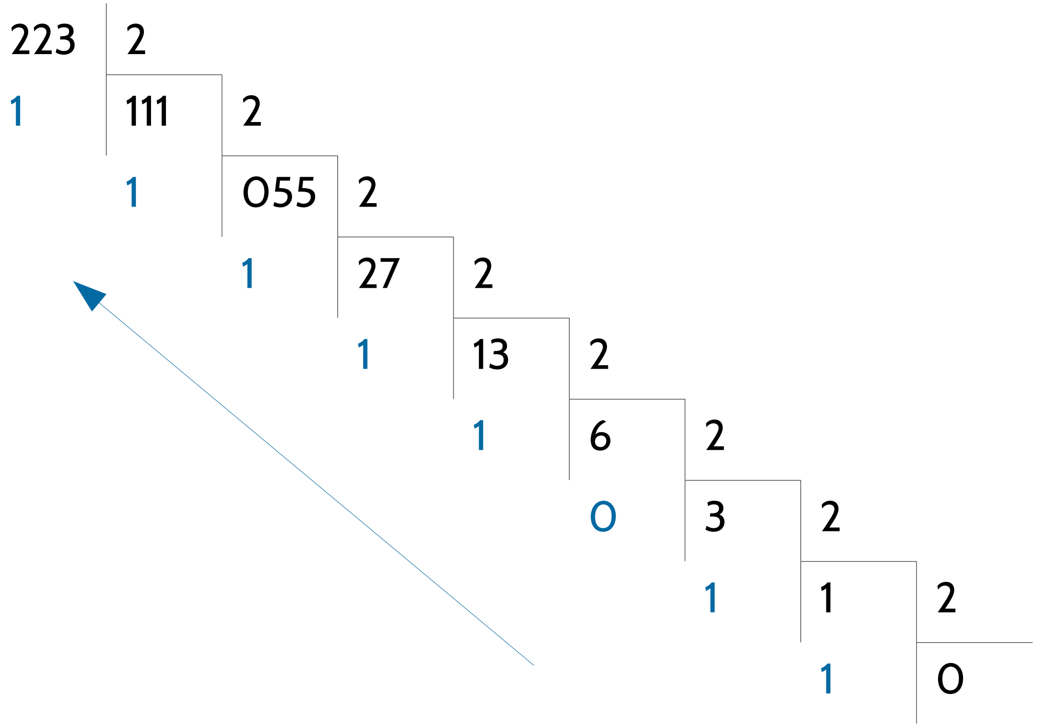
Les systèmes de numération

- Changement de base – Décimal vers binaire
  - Par soustraction successives :
    - On soustrait des puissances de 2
    - Exemple $(223)_{10} = (1101\ 1111)_2$

223 – 128 = 95	1 XXX XXXX
95 – 64 = 31	11 XX XXXX
32 > 31…	110 X XXXX
31 – 16 = 15	1101 XXXX
15 – 8 = 7	1101 1 XXX
7 – 4 = 3	1101 11 XX
3 – 2 = 1	1101 111 1

Les systèmes de numération

- Changement de base – Décimal vers binaire
  - Par divisions successives par 2:
    - Exemple $(223)_{10} = (1101\ 1111)_2$



Les systèmes de numération

- Changement de base – Décimal vers hexadécimal
  - Par divisions successives par 16:
    - Exemple $(223)_{10} = (DF)_{16}$

223	16
15	13

Soit $13.16^{1} + 15 = DF_{H}$

Les systèmes de numération

- Changement de base – base quelconque vers base décimale
  - Utiliser la forme $N = a_{n-1}B^{n-1} + \ldots + a_1B^1 + a_0B^0$
    - Exemple $(A1)_{16} = 10.16^1 + 1.16^0 = 10.16 + 1 = (161)_{10}$

POLYTECH ANGERS

Les systèmes de numération

- Changement de base – binaire vers hexadécimal
  - Exemple :

$$(110101110001)_2 = 1101 \,|\, 0111 \,|\, 0001$$

$$(110101110001)_2 = (D71)_{16}$$

Les systèmes de numération

- Changement de base – hexadécimal vers binaire
  - Exemple :

$(BC34)_{16} =$	B	C	3	4
$(BC34)_{16} =$	(1011	1100	0011	$0100)_2$

Codage des nombres

- Codes pondérés
  - Codes naturels

Décimal	Binaire	Hexadécimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F
16	1 0000	10

Codage des nombres

- Code BCD
  - Binary Coded Decimal
  - DCB
  - Représenter chaque chiffre décimal par un quartet de bits
    - $(7239)_{10} = (0111\ 0010\ 0011\ 1001)_{BCD}$

      7 2 3 9

Décimal	Binaire	Hexadécimal	BCD
0	0000	0	0000
1	0001	1	0001
2	0010	2	0010
3	0011	3	0011
4	0100	4	0100
5	0101	5	0101
6	0110	6	0110
7	0111	7	0111
8	1000	8	1000
9	1001	9	1001
10	1010	A	0001 0000
11	1011	B	0001 0001
12	1100	C	0001 0010
13	1101	D	0001 0011
14	1110	E	0001 0100
15	1111	F	0001 0101
16	1 0000	10	0001 0110

Codage des nombres

- Codes pondérés
  - Représentation des nombres négatifs : complément à 2

    $(-N) = C_2(N) = \overline{N} + 1$
  - Avec n bits
    - On représente des nombres positifs compris entre 0 et $2^{n-1}-1$
    - On représente des nombres négatifs allant de -1 à $-(2^{n-1})$

Décimal	Binaire signé
3	0 11
2	0 10
1	0 01
0	0 00
-1	1 11
-2	1 10
-3	1 01
-4	1 00

Codage des nombres

- Codes pondérés
  - Représentation des nombres négatifs : complément à 2
    - En informatique, un nombre signé et un nombre non signé ne seront pas codé de la même manière, par exemple

Signed char var1 = -125 ;
→ var1 est codé sur un octet en C2,
donc a pour limite [-128, 127]

Base 10	Code C2
+127	0111 1111
+125	0111 1101
0	0000 0000
-125	1000 0011
-128	1000 0000

unsigned char var2 = 125 ;
→ var2 est codé sur un octet,
donc a pour limite [0, 255]

Base 10	Base 2
+255	1111 1111
+130	1000 0011
+127	0111 1111
0	0000 0000

Codage des nombres

- Codes pondérés
  - Nombres en virgule flottante

    $\pm . M . 2^E$ où M (compris entre 1 et 2) est la mantisse et E l'exposant

  - En informatique, le type simple précision (float) est codé sur 32 bits :
    - 1 bit pour le signe
    - 8 bits pour l'exposant
    - 23 bits pour la mantisse

$$\text{Valeur} = (-1)^{\text{signe}} \times 1.\text{mantisse} \times 2^{\text{exposant}-127}$$

Codage des nombres

$$\text{Valeur} = (-1)^{\text{signe}} \times 1.\text{mantisse} \times 2^{\text{exposant}-127}$$

- Codes pondérés
  - Nombres à virgule flottante
  - Exemple 1 : float f = +1 ;
    - S = 0
    - Exp = 0111 1111 = 127
    - M = 000 0000 0000 0000 0000 0000
      - Valeur = $(-1)^0 \times 1.0 \times 2^{127-127} = 1 \times 1 \times 2^0 = 1 \times 1 \times 1 = 1$
  - Exemple 2 : float f = -0.25 ;
    - S = 1
    - Exp = 0111 1101 = 125
    - M = 000 0000 0000 0000 0000 0000
      - Valeur = $(-1)^1 \times 1.0 \times 2^{125-127} = -1 \times 2^{-2} = -0.25$

Codage des nombres

- Codes pondérés

$$\text{Valeur} = (-1)^{\text{signe}} \times 1.\text{mantisse} \times 2^{\text{exposant}-127}$$

  - Nombres à virgule flottante
  - Exemple 3 : float f = +115.2 ;
    - S = 0
    - Exp = 1000 0101 = 133
    - M = 111 1010 0000 1111 0101 1100 = 7999324
      - Valeur = $(-1)^0 \times 1.7999324 \times 2^{133-127} = 1.7999324 \times 2^6 = 115.195$

Codage des nombres

- Codes NON pondérés
  - Code cyclique : code binaire réfléchi / Code Gray

c	b	a
0	0	0
0	0	1
0	1	1
0	1	0
1	1	0
1	1	1
1	0	1
1	0	0

Codage des nombres

- Codes NON pondérés
  - Code redondants : détecteur et correcteur d'erreurs
    - Exemple : le code de Hamming (7,4)
      - 7 bits au total : 4 bits de données et 3 bits de contrôle (bits de parité)

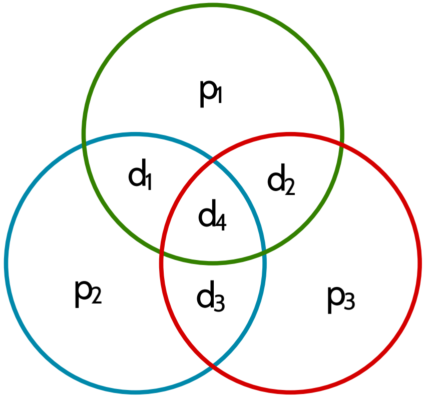
$d_1\ d_2\ d_3\ d_4\ p_1\ p_2\ p_3$

$p_1 = d_1 \oplus d_4 \oplus d_2$

$p_2 = d_1 \oplus d_4 \oplus d_3$

$p_3 = d_3 \oplus d_4 \oplus d_2$

    - Ce code permet d'identifier (détecter et localiser) une erreur d'un bit



Si $d_1 + d_2 + d_4$ est pair

alors $p_1 = 0$

sinon $p_1 = 1$

p_3	OK	OK	OK	NOK	NOK	NOK	NOK
p_2	OK	NOK	NOK	OK	OK	NOK	NOK
p_1	NOK	OK	NOK	OK	NOK	OK	NOK
Err :	p_1	p_2	d_1	p_3	d_2	d_3	d_4

Codage des nombres

- Codes non numériques
  - Code ASCII
    - ASCII : 7 bits
    - ASCII étendu : 8 bits
    - Utilisation : coder des catactères

Char	Dec	Hex	Bin
#	35	23	010 0011
$	36	24	010 0100
%	37	25	010 0101
0	48	30	011 0000
1	49	31	011 0001
@	64	40	100 0000
A	65	41	100 0001
B	66	42	100 0010
a	97	61	110 0001
b	98	62	110 0010
~	126	7E	111 1110
[DEL]	127	7F	111 1111

Exercices

$(11001100)_2 = (\qquad)_{10}$ $(10011110)_2 = (\qquad)_{10}$ $(110001101)_2 = (\qquad)_{10}$

$(11001100)_2 = (\qquad)_{16}$ $(10011110)_2 = (\qquad)_{16}$ $(110001101)_2 = (\qquad)_{16}$

$(D5)_{16} = (\qquad)_2$ $(8E)_{16} = (\qquad)_2$

$(143)_{10} = (\qquad)_2$ $(227)_{10} = (\qquad)_2$ $(1212)_{10} = (\qquad)_2$

$(223)_{10} = (\qquad)_{16}$

$(63)_{10} = (\qquad)_{BCD}$ $(11100)_2 = (\qquad)_{BCD}$

$(3D)_{16} = (\qquad)_{BCD}$ $(49)_{16} = (\qquad)_{BCD}$

Exercices

$(11001100)_2 = (204)_{10}$

$(10011110)_2 = (158)_{10}$

$(110001101)_2 = (397)_{10}$

$(11001100)_2 = (CC)_{16}$

$(10011110)_2 = (9E)_{16}$

$(110001101)_2 = (18D)_{16}$

$(D5)_{16} = (11010101)_2$

$(8E)_{16} = (10001110)_2$

$(143)_{10} = (10001111)_2$

$(227)_{10} = (11100011)_2$

$(1212)_{10} = (10010111100)_2$

$(223)_{10} = (DF)_{16}$

$(63)_{10} = (0110\ 0011)_{BCD}$

$(11100)_2 = (0010\ 1000)_{BCD}$

$(3D)_{16} = (0110\ 0001)_{BCD}$

$(49)_{16} = (0111\ 0011)_{BCD}$