Codification et Représentation de l’Information (CRI)

MI – USTHB – SEC4

Par Dr L.ABADA

abada.lyes@gmail.com

références : CRI - N.HADJI

- **Chapitre 1 : Codification et représentation des nombres,**
- **Chapitre 2 : Algèbre de Boole,**
- **Chapitre 3 : circuits combinatoires :**
  - L’Additionneur,
  - Le Décodeur,
  - Le Multiplexeur
  - …….

Introduction

- Notre langage écrit utilise un code basé sur 26 lettres (majuscules et minuscules), 10 chiffres, des symboles de ponctuation et des signes mathématiques.
- Grâce à ce code et à ces règles nous pouvons transmettre des informations, donner des instructions, dénombrer...
- Bien que les ordinateurs soient qualifiés "d'intelligence artificielle", ils n'ont aucune faculté d'appréhender le monde extérieur.

Introduction

- Bien que les ordinateurs soient qualifiés "d'intelligence artificielle", ils n'ont aucune faculté d'appréhender le monde extérieur.
- Leur seule intelligence réside dans leur rapidité d'exécution de combinaisons d'ordre à deux états (0 , 1)équivalent à (éteint , allumé).
- Le terme bit signifie « binary digit ». Il s'agit de la plus petite unité d'information manipulable par une machine numérique.
- Il est possible de représenter physiquement cette information binaire par un signal électrique ou magnétique

Introduction

- La **codification** consiste à établir une correspondance entre la représentation externe de l'information dont nous sommes utilisateurs et sa représentation interne dans la machine, qui est une suite de bits (suite de 0 et 1)

Systèmes de numération

- Un système de numération est défini par un ensemble de **symboles (chiffres ou lettres)** et des **règles d'écriture** pour le **positionnement** de ces symboles.
- L'exemple le plus répandu de système de numération est la **numération décimale**.
- Ce système est composé de dix chiffres : **0, 1, 2, 3, 4, 5, 6, 7, 8, 9**
- Un nombre est représenté par une succession de chiffres.
- Chaque chiffre possède **un poids**.

Systèmes de numération

$2_3\,3_2\,4_1\,1_0 = 2*10^3+3*10^2+4*10^1+1*10^0$

$abcd = a*10^3+b*10^2+c*10^1+d*10^0$

Base 10

0
1
2
3
4
5
6
7
8
9
10

Base	Base16	Base12	Base 10	Base 9	Base8	Base 5	Base 3	Base 2
	0	0	00	00	0	0	0	00
	1	1	01	01	1	1	1	01
	2	2	02	02	2	2	2	10
	3	3	03	03	3	3	10	11
	4	4	04	04	4	4	11	100
	5	5	05	05	5	10	12	101
	6	6	06	06	6	11	20	110
	7	7	07	07	7	12	21	111
	8	8	08	08	10	13	22	1000
	9	9	09	10	11	14	100	1001
	A	A	10	11	12	20	101	1010
	B	B	11	12	13	21	102	1011
	C	10	12	13	14	22	110	1100
	D	11	13	14	15	23	111	1101
	E	12	14	15	16	24	112	1110
	F	13	15	16	17	31	120	1111
	10	14	16	17	20		121	10000
	11	15	17	18	21		122	10001
	12	16	18	20	22		200	10010
	13	17	19	21	23		201	
		18	20	22	24	44		
		19	***21***	***23***	25	100		
	19	1A	22	24	26			
	1A	1B	23					
	1B	20	24					
	1C			88	77			
			99	100				
	21		100					
	1F							

Base 10	Base 8
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	10
9	11
10	12
11	13
12	14
13	15
14	16
15	17
16	20
17	21
18	22
19	23
20	***24***
21	25
22	26
23	27
24	30
25	31
26	32
27	33

$$(24)_8 = 2*8^1+4*8^0 =16+4 = (20)_{10}$$

Base 10	Base 8
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	10
9	11
10	12
11	13
12	14
13	15
14	16
15	17
16	20
17	21
18	22
19	23
20	***24***
21	25
22	26
23	27
24	30
25	31
26	32
27	33

$$(24)_8 = 2*8^1+4*8^0 = 16+4 = (20)_{10}$$

$$(a_3a_2a_1a_0)_{10} = a_3*10^3+a_2*10^2+a_1*10^1+a_0*10^0$$

$$(a_3\,a_2\,a_1\,a_0)_8 = a_3*8^3+a_2*8^2+a_1*8^1+a_0*8^0$$

$$(a_3\, a_2\, a_1\, a_0)_{10} = a_3 * 10^3 + a_2 * 10^2 + a_1 * 10^1 + a_0 * 10^0$$

$$(a_3\, a_2\, a_1\, a_0)_8 = a_3 * 8^3 + a_2 * 8^2 + a_1 * 8^1 + a_0 * 8^0$$

Bases

Un système de numération à base B est défini par B symboles (chiffres ou lettres)

Soit N un nombre de n chiffres représenté en base B

$$\mathbf{N = a_{n-1}a_{n-2}....a_i...a_0 \quad i \quad a_i < B}$$

(Tous les symboles sont strictement inférieurs à B)

Quelque soit la base, la forme polynomiale de N est :

$$\mathbf{N = a_{n-1} * B^{n-1} + a_{n-2} * B^{n-2} \ldots + a_i * B^i \ldots + a_0 * B^0}$$

Système Binaire (Base 2)

→{0,1}

Système Octal (Base 8)

→{0,1,2,3,4,5,6,7}

Système Hexadécimal (Base 16)

→{0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F}

Changement de base : Base B→Base 10:

il suffit de représenter le nombre sous sa forme polynomiale et calculer la somme de tous les termes

Bases

Un système de numération à base B est défini par B symboles (chiffres ou lettres)

Soit N un nombre de n chiffres représenté en base B

$$\mathbf{N = a_{n-1}a_{n-2}....a_i...a_0 \text{ i } \quad a_i < B}$$

(Tous les symboles sont strictement inférieurs à B)

Quelque soit la base, la forme polynomiale de N est :

$$\mathbf{N = a_{n-1} * B^{n-1} + a_{n-2} * B^{n-2} \; . \; . \; . \; . + a_i * B^i \; . \; . \; . \; . + a_0 * B^0}$$

Changement de base : Base B→Base 10:

Exemple :

B = 2

N= $(1111011)_2 = 1*2^6 + 1*2^5 + 1*2^4 +1*2^3 + 0*2^2 + 1*2^1 +1*2^0 = (123)_{10}$

B = 16

N = $(7B)_{16} = (7*16^1 + 11*16^0)_{10} = 123$

Changement de base : Base B→Base 10:

Pour les nombres décimaux, on utilisera des exposants négatifs.

Exemple :

B = 16

N = $(7_1 B_0, 8_1 4_2)_{16} = 7*16^1 + 11*16^0 + 8*16^{-1} + 4*16^{-2} = (123{,}515625)_{10}$

Changement de base : Base 10 →Base B:

La règle à suivre est celle des divisions successives,

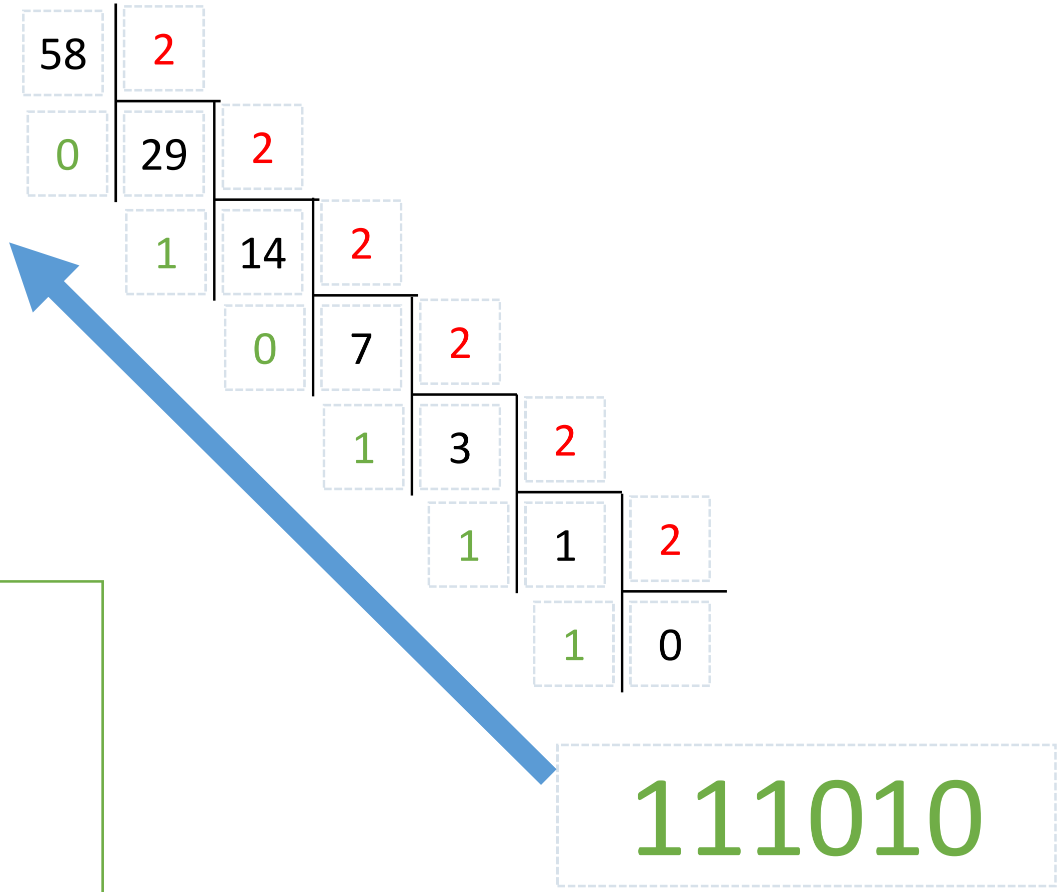
1. On divise le nombre par **B**
2. On sauvegarde **le reste** puis **on divise le quotient** par **B**
3. Ainsi de suite jusqu’à obtention d’un quotient nul

La suite des restes correspond au nombre de la base visée

Le premier reste correspond au **poids faible** et **le dernier** au **poids fort**

Changement de base : Base 10 → Base B

Exemple : binaire (base2) → $N=(58)_{10}$ → $(???)_2$



La règle à suivre est celle des divisions successives,

1. On divise le nombre par **B**
2. On sauvegarde **le reste** puis **on divise le quotient** par **B**
3. Ainsi de suite jusqu'à obtention d'un quotient nul

La suite des restes correspond au nombre de la base visée

Le premier reste correspond au **poids faible** et **le dernier** au **poids fort**

Changement de base : Base 10 → Base B:

Exemple : octale (base8) → $N=(58)_{10}$ → $(???)_8$

58	8	
2	7	8
	7	0

72

La règle à suivre est celle des divisions successives,

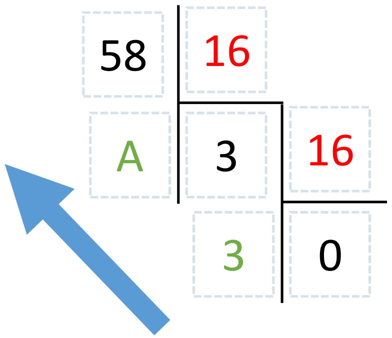
1. On divise le nombre par <u>**B**</u>
2. On sauvegarde <u>**le reste**</u> puis **on divise le quotient** par **B**
3. Ainsi de suite jusqu'à obtention d'un quotient nul

La suite des restes correspond au nombre de la base visée

Le premier reste correspond au **poids faible** et **le dernier** au **poids fort**

Changement de base : Base 10 →Base B:

Exemple : hexadécimale (base16) ➔ $N=(58)_{10}$ → $(???)_{16}$



3A

La règle à suivre est celle des divisions successives,

1. On divise le nombre par **B**
2. On sauvegarde **le reste** puis **on divise le quotient** par **B**
3. Ainsi de suite jusqu'à obtention d'un quotient nul

La suite des restes correspond au nombre de la base visée
Le premier reste correspond au **poids faible** et **le dernier** au **poids fort**

Changement de base : Base B1 → Base B2:

Pour passer d'une base B1 vers une base B2 il faut 2 opérations.

1. Il faut d'abord passer de la base B1 vers la base 10
2. puis de la base 10 vers la base B2

Changement de base : Base B1 → Base B2:

Exemple : Convertir (3141)5 vers la base 16

$(3141)_5$ = $(421)_{10}$
$(421)_{10}$ = $(1A5)_{16}$

→ $(3141)_5 = (1A5)_{16}$

Pour passer d'une base B1 vers une base B2 il faut 2 opérations.

1. Il faut d'abord passer de la base B1 vers la base 10
2. puis de la base 10 vers la base B2

Changement de base :

Applications aux bases 2, 8 et 16

2➔ 8

Pour convertir un nombre de la base 2 vers la base 8, il faut découper ce nombre en groupes de 3 bits et remplacer chaque groupe par sa valeur octale (en partant de la droite).

Exemple : Convertir $(1\ 011\ 010)_2$ vers la base 8

$(001\quad 011\quad 010)_2 = (1\ 3\ 2)_8$

Base 2	Base 8
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Changement de base :
Applications aux bases 2, 8 et 16

2 ← 8

Pour convertir un nombre de la base 8 vers la base 2, il suffit de transcrire chaque chiffre de ce nombre en binaire sur 3 bits (en partant du poids faible).

Exemple : Convertir $(645)_8$ vers la base 2

$(420)_8 = (\ 100\ \ \ 010\ \ 000)_2$

Base 2	Base 8
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Changement de base : Applications aux bases 2, 8 et 16

2 ➔ 16

Pour convertir un nombre de la base 2 vers la base 16, il faut découper le nombre en groupes de 4 bits et remplacer chaque groupe par sa valeur hexadécimale (en partant de la droite).

Exemple :

Convertir $(101 \quad 1010)_2$ vers la base 16

$(0101 \quad 1010)_2 = (5\ A)_{16}$

Base 2	Base 16
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Changement de base : Applications aux bases 2, 8 et 16

2 ← 16

Pour convertir un nombre de la base 16 vers la base 2, il suffit de transcrire chaque chiffre de ce nombre en binaire sur 4 bits en partant du poids faible.

Exemple : Convertir $(A15)_{16}$ vers la base 2

$(A15)_{16} = (1010\ \ 0001\ 0101)_2$

Base 2	Base 16
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Changement de base : Applications aux bases 2, 8 et 16

8 → 16

Pour convertir un nombre de la base 8 vers la base 16 ou inversement, on peut transiter par la base 10 mais on peut également passer par la base 2

Base8 → base2 → base16

Base16 → base2 → base8

Exemple : Convertir $(232)_8$ vers la base 16

$(232)_8 = (010\ \ 011\ \ 010)_2 = (\ 0000\ \ 1001\ \ 1010)_2 = (0\ 9\ A)_{16}$

Base 2	Base 16
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Changement de base : Applications aux bases 2, 8 et 16

des nombres décimaux

Pour la conversion des nombres décimaux, on sépare la partie entière de la partie décimale.
La partie entière est traitée comme indiqué précédemment.
La conversion de la partie décimale se fait de droite à gauche.

Base 2

Base 16

$(011101,010\ 110)_2 = (\ \underline{0}11\ \ 101,\ 010\ 11\underline{0}\ \ )_2 = (35,26)_8$
$(0001\ 1101\ \ ,0101\ \ 1000)_2 = (\ \underline{000}1\ \ 1101,\ 0101\ 1\underline{000}\ )_2 = (1D,58)_{16}$

Arithmétique binaire

(Exemples d'opérations réalisées en binaire)

1

Addition

```
 1 1 1 1
  1 1 1 0 1          29
+       1 1 1      +  7
 -----------       ------
 1 0 0 1 0 0         36
```

$0+0 = 0$

$0+1 = 1$

$1+0 = 1$

$1+1 = 10 = 2_{10}$

$1+1+1 = 11 = 3_{10}$

Arithmétique binaire

(Exemples d'opérations réalisées en binaire)

1

1

Soustraction

	1	1	1	1	1	0	1	27
-				1		1	1	- 9
		1		1				
	------------------							------
	1	0		1		1	0	22

0-0 = 0

0-1 =

1-0 = 1

1-1 = 0

Arithmétique binaire

(Exemples d'opérations réalisées en binaire)

Multiplication

```
      1 1 1 0 1
*         1 1 1
  -------------
      1 1 1 0 1
    1 1 1 0 1 *
  1 1 1 01 **
```

```
     1
 11101
11101
     1
1010111
11101
11001011
```

```
   29
*   7
------
  203
```

0*0 =

0*1 =

1*0 =

1*1 =

Arithmétique binaire

(Exemples d’opérations réalisées en binaire)

Division

$$\begin{array}{r|l} 11101 & 111 \\ \cline{2-2} 00001 & 100 \end{array}$$

$$\begin{array}{r|l} 29 & 7 \\ \cline{2-2} 1 & 4 \end{array}$$

0*0 =
0*1 =
1*0 =
1*1 =

1

Arithmétique binaire

(Exemples d'opérations réalisées en binaire)

Division

11000001	101
001000	100110
00110	
0011	

193	5
43	38
3	

0*0 =
0*1 =
1*0 =
1*1 =

1

Représentation des nombres entiers relatifs

Les entiers relatifs sont représentés en binaire dans un format fixe (on ne peut pas comparer par exemple un nombre de 5 bits avec un nombre de 8 bits).

Le bit de poids fort représente le signe, il est égal à 0 si le nombre est positif et il est égal à 1 si le nombre est négatif

Représentation des nombres entiers relatifs

- **Représentation en signe et valeur absolue (SVA)**
- **Représentation en complément à 1 (C1)**
- **Représentation complément à 2 (C2)**
- ~~**Représentation d'un nombre en virgule flottante dans le format IEEE 754**~~

Représentation en Signe et valeur absolue (SVA)

Le **<u>bit de poids fort</u>** représente le signe du nombre (**0 pour +** et 1 pour –)
Les autres bits représentent la valeur absolue du nombre.
X représenté sur un format de n bits $-(2^{n-1} - 1) < X < +(2^{n-1} - 1)$

000 = 7 = 2^n-1

Exemples : Représentation sur **<u>8 bits</u>**

A = +25

$00011001_{(SVA)}$

$-(2^7 - 1) < X < +(2^7 - 1)$ -

127,127

A = -25

$10011001_{(SVA)}$

Représentation en signe et valeur absolue (SVA)

Le bit de poids fort représente le signe du nombre (0 pour + et 1 pour –)
Les autres bits représentent la valeur absolue du nombre.
X représenté sur un format de n bits $-(2^{n-1} - 1) < X < +(2^{n-1} - 1)$
000

Exemples : Représentation sur n=**8 bits**
$B = +525 = 10\ 0000\ 1101_{(2)}$
$525 > 2^{11}-1 = 127$

525 :

N'est pas possible,

Représentation en complément à 1 (C1)

Le complément à 1 d'un nombre est obtenu en inversant tous les bits.
Le bit de poids fort représente le signe du nombre (0 pour + et 1 pour –).
X représenté sur un format de n bits $-(2^{n-1} - 1) < X < +(2^{n-1} - 1)$

Exemple : Représentation sur 8 bits

A = 1 0 0 1 1 0 0 1
C1(A) = 0 1 1 0 0 1 1 0

A + C1(A) = 11111111+1 =0000000

Représentation complément à 2 (C2)

Le complément à 2 d'un nombre est égal au complément à 1 (C1) du nombre auquel on ajoute 1 : C(2)= C(1)+1,
X représenté sur un format de n bits $-(2^{n-1}) < X < +(2^{n-1} - 1)$
(-X)= C2 (X)

Exemples : représentation sur **8 bits**
A = 25 = $00011001_{(2)}$
CA1(25)= $11100110_{(ca1)}$
CA2(25) = 11100110 +1 = $11100111_{(ca2)}$
CA2(25) = -25 (sur 8bit)

Représentation complément à 2 (C2)

Exemple avec n=4:

E = [-2^{4-1} , $2^{4-1}-1$]

E = [-8,7]

0= 0000 1111+1 = 0000

1= 0001 1110+1=1111

2= 0010 1101+1 = 1110

3= 0011 1100+1 = 1101

4= 0100

5= 0101

6= 0110

7= 0111 1000+1 = 1001

1111

1110 $\rightarrow$ 0001+1

= 0010

-0= C2(0) = 0000

-1= C2(1) = 1111

-2= C2(2) = 1110

-3= C2(3) = 1101

-4= C2(4) = 1100

-5= C2(5) = 1011

-6= C2(6) = 1010

-7= C2(7)= 1001

-8= C2(8)= 1000

Représentation complément à 2 (C2)

Soit A et B 2 nombres représentés en C2 sur 8 bits, trouver leurs valeurs décimales.

Représenter en CA2 :

A = 53

A = $00110101_{(2)}$= $00110101_{(ca2)}$

A = -53

A = -53 = CA2(53) = CA2(00110101) = CA1(00110101)+1

= 11001010+1 = $11001011_{(ca2)}$

Calculer le CA2 :

A = 13

CA2(A) = CA2(13) = CA1(00001101) +1 = 11110010+1 = 11110011 = -13

Opérations arithmétiques en complément à 2

Dans cette représentation la soustraction doit est traitée comme une addition. Pour les opérations arithmétiques on doit appliquer la règle suivante :

$$\forall\ (A,B) \in E \quad \text{si} \quad X = A + B \quad \text{alors} \quad X \in E$$

Exemples d'opérations arithmétiques avec n=4 (E = [- 8,+ 7]) :

Addition

```
    5            0101
+   1       +    0001
-------    -------------
                 0110
```

Dépassement ?

Pas de déplacement.

Opérations arithmétiques en complément à 2

Dans cette représentation la soustraction doit est traitée comme une addition. Pour les opérations arithmétiques on doit appliquer la règle suivante :

$$\forall (A,B) \in E \quad \text{si} \quad X = A + B \quad \text{alors} \quad X \in E$$

Exemples d'opérations arithmétiques avec n=4 (E = [- 8,+ 7]) :

Addition

$$\begin{array}{rr} & -7 \\ + & -4 \\ \hline & -4 \end{array} \qquad \begin{array}{rr} & 1001 \\ + & 1100 \\ \hline & \cancel{1}0101 \end{array}$$

Dépassement ?

Il y a un déplacement,
(+) + (+) = (-)

Opérations arithmétiques en complément à 2

Dans cette représentation la soustraction doit est traitée comme une addition. Pour les opérations arithmétiques on doit appliquer la règle suivante :

$\forall\ (A,B) \in E \quad \text{si} \quad X = A + B \quad \text{alors} \quad X \in E$

Exemples d'opérations arithmétiques avec n=4 (E = [- 8,+ 7]) :

Soustraction

$$\begin{array}{r} 5 \\ -\ \ 3 \\ \hline +2 \end{array} \qquad \begin{array}{r} 0101 \\ +\ \ 1101 \\ \hline (+2) = \cancel{1}0010 \end{array}$$

Dépassement ?

Traitement du dépassement de capacité pour une addition:

- Si les deux opérandes sont de même signe et que le résultat est du même signe que les opérandes, il n’y a pas de dépassement

➢ 0001 + 0101 = 0110 ⇔ 1+5= 6

➢ 1111 + 1011 = ~~1~~1010 ⇔ -1-5=-6

- Si les deux opérandes sont de même signe et que le résultat est du signe opposé alors il y a dépassement.

➢ 0101+0100=1001 ⇔ ~~5+4 = -7~~

- Si les deux opérandes sont de signes opposés, il n'y a jamais de dépassement

➢ 1110+0111=~~1~~0101 ⇔ -2 + 7 = 5

Traitement du dépassement de capacité pour une addition:

Remarque :

- Il ne faut pas confondre les 2 expressions :

1. « Représenter X en format C2 » : revient à écrire X en C2
2. « Donner le C2 de X » reviens à écrire l'opposé de X

Ex : X = 35

Représenter 35 en CA2

$35 = 00100011_{(ca2)}$

Représenter -35 en CA2

-35 = CA2(35) = CA1(00100011)+1 = 11011101

Calculer/donner le CA2 de 35

CA2(35) = -35 = CA1(00100011)+1 = 11011101

Le code BCD

Le code BCD, Binary Coded Decimal (Décimal Codé en Binaire) est un code qui s'applique uniquement aux chiffres de la base 10. Chaque chiffre décimal est représenté directement par sa valeur binaire sur un format de 4 bits

Table des codes BCD

DEC	→	BCD
0	→	0000
1	→	0001
2	→	0010
3	→	0011
4	→	0100
5	→	0101
6	→	0110
7	→	0111
8	→	1000
9	→	1001

Exemple :

$(987)_{10} = (1001 \ 1000 \ 0111)_{BCD}$

$(13)_{10} = (0001 \ 0011)_{BCD}$

Addition BCD

Binaire	
0 →	0000
1 →	0001
2 →	0010
3 →	0011
4 →	0100
5 →	0101
6 →	0110
7 →	0111
8 →	1000
9 →	1001
10 →	1010
11 →	1011
12 →	1100
13 →	1101
14 →	1110
15 →	1111
16 →	10000

	BCD
0000 ←	0
0001 ←	1
0010 ←	2
0011 ←	3
0100 ←	4
0101 ←	5
0110 ←	6
0111 ←	7
1000 ←	8
1001 ←	9
0001 0000 ←	10
0001 0001 ←	11
0001 0010 ←	12
0001 0011 ←	13
0001 0100 ←	14
0001 0101 ←	15
1 0110 ←	16

+

----

1 0011

Addition BCD

SI la somme de 2 chiffres codés en BCD est inférieure ou égale à 9 (1001) alors on ne fait rien sinon on ajoute 6 (0110) pour corriger et on retient 1.

Exemple : Réaliser l'addition A + B en BCD :

$A = 183_{10} = ($ 0001 1000 0011 $)_{BCD}$ $B = 375_{10} = ($ 0011 0111 0101 $)_{BCD}$

```
  111          111
  0001   1000   0011
  0011   0111   0101
          11
  0101   1111   1000
         0110
  0101   0101   1000
```

Addition BCD

1 0110

SI la somme de 2 chiffres codés en BCD est inférieure ou égale à 9 (1001) alors on ne fait rien sinon on ajoute 6 (0110) pour corriger et on retient 1.

Exemple 2

4 0000 0100

+ 3 0000 0011

= -----------------------------------

0000 0111

Exemple 3

4 0100

+ 9 1001

= -----------------------------

1101

0110

0001 0011

Le code GRAY

Le code de Gray est un code binaire qui permet de passer d’un nombre entier N au nombre suivant (N+1) en changeant un seul digit(bit). (On l’appelle également Code Réfléchi). Ce code sera utilisé principalement dans les Tableaux de Karnaugh pour simplifier les fonctions booléennes.

0000	**0**		**11**		**22**
0001	**1**		**12**		**23**
0011	**2**		**13**		**24**
0010	**3**		**14**		**25**
0110	**4**		**15**		**26**
0111	**5**		**16**		**27**
0101	**6**		**17**		**28**
0100	**7**		**18**		**29**
1100	**8**		**19**		**30**
1101	**9**		**20**		**31**
1111	**10**		**21**		**32**

Conversion du code binaire vers le code Gray

Soit $X = B_n\ B_{n-1}.....\ B_0$ représenté en binaire Pour convertir X en code de Gray il faut suivre les règles suivantes :

$G_n = B_n$

$G_i = 0$ si $B_i = B_{i+1}$

$G_i = 1$ si $B_i <> B_{i+1}$

Exemple : pour n = 4 X = 10001 en binaire X = $(11001)_{gray}$

1	0	0	0	1
B4	B3	B2	B1	B0

1	1	0	0	1
G4	G3	G2	G1	G0

Le code GRAY

Le code de Gray est un code binaire qui permet de passer d'un nombre entier N au nombre suivant (N+1) en changeant un seul digit. (On l'appelle également Code Réfléchi). Ce code sera utilisé principalement dans les Tableaux de Karnaugh pour simplifier les fonctions booléennes.

0000	**0**	1110	**11**	11101	**22**
0001	**1**	1010	**12**	11100	**23**
0011	**2**	1011	**13**	10100	**24**
0010	**3**	1001	**14**	10101	**25**
0110	**4**	1000	**15**	10111	**26**
0111	**5**	11000	**16**	10110	**27**
0101	**6**	11001	**17**	10010	**28**
0100	**7**	11011	**18**	10011	**29**
1100	**8**	11010	**19**	10001	**30**
1101	**9**	11110	**20**	10000	**31**
1111	**10**	11111	**21**	110000	**32**

Conversion du code Gray vers le code binaire

Soit $X = G_n\ G_{n-1} \ldots\ldots G_0$ représenté en code Gray

Pour convertir X en binaire il faut suivre les règles suivantes :

$Bn = Gn$

$B_i = 0$ si $B_{i+1} = G_i$

$B_i = 1$ si $B_{i+1} <> G_i$

Exemple : pour n = 4 X = 10101 en code Gray X = $(11001)_2$

1	0	1	0	1
G4	G3	G2	G1	G0
1	1	0	0	1
B4	B3	B2	B1	B0

Codification et représentation -Numérique

Codification et représentation –Numérique : Le code ASCII

Le code ASCII

Définition Le code ASCII : American Standard Code For Information Interchange,

Est une norme de codage en informatique mise au point dans les années 60.

Ce code définit 128 caractères représentés sur 8 bits.

Codification et représentation –Numérique : Le code ASCII

Cette table est présentée sous une forme condensée, fondée sur la base 16. Chaque caractère se trouve au croisement d'une ligne et d'une colonne. Le numéro de la ligne suivi du numéro de la colonne représentent le code du caractère en hexadécimal.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	SP	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

Codification et représentation –Numérique : Le code ASCII

Exemple : La lettre M se trouve au croisement de la ligne 4 et de la colonne D
son code est 4D en hexadécimal.
Sa représentation est donc : 0100 1101 en code ASCII 0

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	SP	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

Codification et représentation –Numérique : Le code ASCII

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US

Caractères de contrôle :

On peut considérer que l'ASCII dispose d’une trentaine de caractères de contrôle plus ou moins utilisés.

Les caractères usuels sont NUL, LF, CR, DEL et ESC

- **NUL :** indique la fin d’une chaine de caractères notamment en langage C.
- **LF** et **CR** : indiquent la fin d’une ligne.
- On utilisera **LF, CR** ou les 2 selon le système d’exploitation :
  - Sous Linux par exemple ce sera **LF**,
  - sous Mac OS on utilise **CR**
  - sous Windows ce sera **CR** suivi de **LF**.
- **ESC** indique la sortie d’un texte.
- **DEL** indique l’effacement d’un caractère.

Codification et représentation –Numérique : Le code ASCII

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	SP	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O

Exemple : Dans un éditeur de texte ou dans un champ de saisie taper sans relâcher la touche alt + le code de caractère en décimal :

$4D_{(16)} = 77_{(10)}$ ➔ M
$30_{(16)} = 48_{(10)}$ ➔ 0
$3F_{(16)} = 63_{(10)}$ ➔ ?
$45_{(16)} = 69_{(10)}$ ➔ E

Codification et représentation –Numérique : Le code ASCII

<u>Exercise N°7: </u>(à faire comme un exemple en cours)

1/ En code ASCII $(41)_{16}$ correspond à 'A' et $(30)_{16}$ correspond à '0', sans l'utilisation de la table du code ASCII déduire le codage du message suivant : Covid-19

2/ Décoder le message suivant :4269656E76656E757320656E204F51

Codification et représentation –Numérique : Le code ASCII

<u>Exercice N°7: </u>(à faire comme un exemple en cours)

1/ En code ASCII

$(41)_{16}$ ➔ 'A'

$(61)_{16}$ ➔ 'a'

$(30)_{16}$ ➔ '0',

$(2D)_{16}$ ➔ '-'

Sans l'utilisation de la table du code ASCII déduire le codage du message suivant :

Covid-19

‘C’ =   43   ‘o’ =   6F   ‘v’ =   76   ‘i’= 69   ‘d’ = 64

‘-’ = 2D   ‘1’= 31   ‘9’= 39

Codification et représentation –Numérique : Le code ASCII

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	SP	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

Exercice N°7:

2/ Décoder le message suivant :42 69 65 6E 76 65 6E 75 73 20 65 6E 20 4D 49

Codification et représentation –Numérique : Le code UNICODE

Le code UNICODE

’Unicode est une norme de codage mise au point dans les années 90;
il définit plus de **60000** caractères de plusieurs langues, codés sur **16 bits**.
Le code ASCII est inclus dans l’Unicode.

Le code ASCII est uniquement basé sur les lettres anglo-saxonnes, on n’y trouve pas les lettres accentuées de la langue française comme le **à** ou le **é** par exemple.
On peut retrouver ces lettres dans le tableau UNICODE suivant :

Codification et représentation –Numérique : Le code UNICODE

Le code UNICODE

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
008	PAD	HOP	BPH	NBH	IND	NEL	SSA	ESA	HTS	HTJ	VTS	PLD	PLU	RI	SS2	SS3
009	DCS	PU1	PU2	STS	CCH	MW	SPA	EPA	SOS	SGCI	SCI	CSI	ST	OSC	PM	APC
00A	NBSP	¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	SHY -	®	¯
00B	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
00C	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
00D	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
00E	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
00F	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Codification et représentation –Numérique : Le code UNICODE

Le code UNICODE

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
008	PAD	HOP	BPH	NBH	IND	NEL	SSA	ESA	HTS	HTJ	VTS	PLD	PLU	RI	SS2	SS3
009	DCS	PU1	PU2	STS	CCH	MW	SPA	EPA	SOS	SGCI	SCI	CSI	ST	OSC	PM	APC
00A	NBSP	¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	SHY -	®	¯
00B	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
00C	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
00D	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
00E	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
00F	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Codification et représentation –Numérique : Le code UNICODE

Le code UNICODE

	0	1	2	3	4	5	6	7	8	9	A	B	C	D
060	؀	؁	؂	؃	□	□	□	□	□	□	□	؋	،	؍
061	سؐ	سؑ	سؒ	سؓ	سؔ	سؕ	سؖ	سؗ	سؘ	سؙ	سؚ	؛	ALM	
062	□	ء	آ	أ	ؤ	إ	ئ	ا	ب	ة	ت	ث	ج	ح
063	ذ	ر	ز	س	ش	ص	ض	ط	ظ	ع	غ	□	□	□
064	ـ	ف	ق	ك	ل	م	ن	ه	و	ى	ي	سً	سٌ	سٍ
065	سِ	سّ	سْ	سٓ	سٔ	سٕ	سٖ	سٗ	س٘	سٙ	سٚ	سٛ	سٜ	سٝ
066	٠	١	٢	٣	٤	٥	٦	٧	٨	٩	٪	٫	٬	٭
067	سٰ	ٱ	ٲ	ٳ	ٴ	ٵ	ٶ	ٷ	ٸ	ٹ	ٺ	ٻ	ټ	ٽ
068	پ	ځ	ڂ	ڃ	ڄ	څ	چ	ڇ	ڈ	ډ	ڊ	ڋ	ڌ	ڍ
069	ڐ	ڑ	ڒ	ړ	ڔ	ڕ	ږ	ڗ	ژ	ڙ	ښ	ڛ	ڜ	ڝ
06A	ڠ	ڡ	ڢ	ڣ	ڤ	ڥ	ڦ	ڧ	ڨ	ک	ڪ	ګ	ڬ	ڭ
06B	ڰ	ڱ	ڲ	ڳ	ڴ	ڵ	ڶ	ڷ	ڸ	ڹ	ں	ڻ	ڼ	ڽ