République Algérienne Démocratique et Populaire
Ministére de l’Enseignement Supérieur et de la Recherche Scientifique
Universite de Mascara
Faculté des Sciences Exactes
Département d’Informatique

Codage et Représentation des données

BACHIR BOUIADJRA Rochdi

Table of contents

Système de numération

I

A. Conversion de la base 10 à la base B

1. Méthode des divisions succéssive

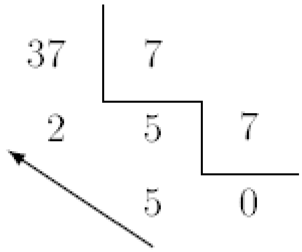
N est itérativement divisé par B jusqu'à obtenir un quotient égal à 0. La conversion du nombre N dans la base B est obtenue en notant les restes de chacune des divisions effectuées depuis la dernière division jusqu'à la première.

Note

Les restes sont obligatoirement inférieurs à $\boldsymbol{B}$.

Example

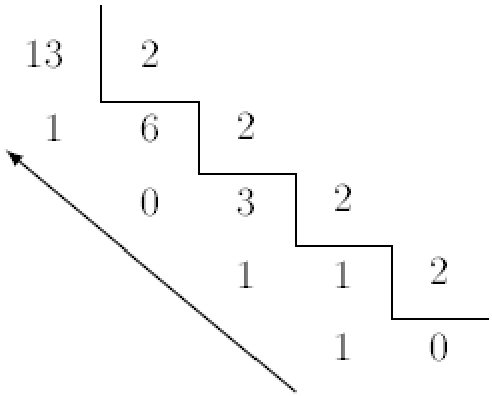
$(37)_{10} = (?)_7$, on a $37 \div 7 = 5$ reste $\mathbf{2}$, $5 \div 7 = 0$ reste $\mathbf{5}$, donc $(37)_{10} = (52)_7$.



$$(37)_{10} = (52)_7$$

Example

$(13)_{10} = (?)_2$, on a $13 \div 2 = 6$ reste $\mathbf{1}$, $6 \div 2 = 3$ reste $\mathbf{0}$, $3 \div 2 = 1$ reste $\mathbf{1}$ et $1 \div 2 = 0$ reste $\mathbf{1}$, donc $13_{10} = (1101)_2$



$$(13)_{10} = (1101)_2$$

2. Méthode des soustractions

La plus grande puissance de B qui est inférieure ou égale à N est soustraite à N. Le processus de soustraction est répété sur le reste de la différence, jusqu'à obtenir un résultat égal à 0. Le nombre N exprimé en base B est alors obtenu en notant le nombre de fois où une même puissance de B a été retirée et ce pour chaque puissance depuis la plus grande apparaissant, dans l'ordre décroissant des puissances.

Example

$123 = (?)_5$ (on a $5^2 = 25, 5^1 = 5, 5^0 = 1$)

$123 - 25 = 98$, $98 - 25 = 73$, $73 - 25 = 48$, $48 - 25 = 23$, (ou bien $123 - 4 \times 25 = 23$), $23 - 4 \times 5 = 3$, $3 - 3 \times 5^0 = 0$, donc $123 = (443)_5$

Example

$20 = (?)_3$ (on a $3^2 = 9, 3^1 = 3, 3^0 = 1$)

$20 - 9 = 11$, $11 - 9 = 2$, $2 - 3^0 = 1$, $1 - 3^0 = 0$, donc $20 = (202)_3$

B. Conversion de la base B à la base 10

1. Principe

Pour trouver la valeur décimale d'un nombre écrit dans la base B on utilise la formule de puissance suivante:

$(a_{n-1}a_{n-2}\cdots a_2a_1a_0)_B = a_{n-1}B^{n-1} + a_{n-2}B^{n-2} + \cdots + a_2B^2 + a_1B^1 + a_0B^0$

avec comme condition: $a_{n-1}, a_{n-2}, \cdots a_2, a_1, a_0 < B$ et $a_{n-1} \neq 0$.

Example

$(52)_7 = 5 \times 7^1 + 2 \times 7^0 = 35 + 2 = (37)_{10}$

$(110001)_2 = 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$

$= 32 + 16 + 1 = (49)_{10}$

2. Passage d'une B vers une Base C

Pour passer d'une B vers une autre base de numération C, on convertit tout d'abord le nombre écrit dans B vers la base décimale (10) puis le résultat sera convertit vers la base C: $B \rightarrow C \Leftrightarrow B \rightarrow 10 \rightarrow C$.

Example

$(155)_6 = (?)_9$

$(155)_6 = 1 \times 6^2 + 5 \times 6^1 + 5 \times 6^0 = 36 + 30 + 5 = (71)_{10}$.

$71 \div 9 = 7$ reste $\mathbf{8}$, $7 \div 9 = 0$ reste $\mathbf{7}$, donc $(71)_{10} = (78)_9$, finalement $(155)_6 = (78)_9$.

C. Bases utiles

1. Base Binaire B = 2

Cette base est utilisée par les ordinateurs, dans ce cas $B = 2$, et les chiffres utilisés sont $\{0, 1\}$. Chaque chiffre est appelé Bit.

Pour passer de la base binaire vers la base décimale en utilise la formule en puissance

$(110000)_2 = 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 = (48)_{10}$.

Reminder

Pour convertir un nombre décimal en binaire on peut utilisé la méthode des soustractions en se basant sur le tableau des puissances de 2

Poids	Valeur décimale
2^0	1
2^1	2
2^2	4
2^3	8
2^4	16
2^5	32
2^6	64
2^7	128
2^8	256
2^9	512
2^{10}	1024
2^{11}	2048
...	...

Example

$1753 = (?)_2$

$1753 - 1024 = 729$, $729 - 512 = 217$, $217 - 128 = 89$, $89 - 64 = 25$, $25 - 16 = 9$, $9 - 8 = 1$, $1 - 1 = 0$, donc $1753 = (11011011001)_2$.

Fundamental

Pour un nombre pair le dernier bit à droite est égale à 0.

Pour un nombre impair le dernier bit à droite est égale à 1.

2. Base Octale B = 8

Cette base est utilisée pour avoir une écriture plus simple d'un nombre binaire. Les chiffres utilisés sont $\{0, 1, 2, 3, 4, 5, 6, 7\}$.

Pour convertir un nombre décimal en base octale soit on utilise la division succéssive sur la base, soit en jutilise la méthode des soustractions.

Example

$(1753)_{10} = (?)_8$, on a $1753 \div 8 = 219$ reste $\mathbf{1}$, $219 \div 8 = 27$ reste $\mathbf{3}$, $27 \div 8 = 3$ reste $\mathbf{3}$,

$3 \div 8 = 0$ reste $\mathbf{3}$, donc $(1753)_{10} = (3331)_8$.

Example

$(6024)_{10} = (?)_8$, on a $6024 - 4096 = 1928$, $1928 - 3 \times 512 = 392$, $392 - 6 \times 64 = 8$, $8 - 8 = 0$, donc $6024 = (13610)_8$.

Fundamental

on convertit le nombre en base binaire puis on regroupe les bits de droite à gauche en groupe de 3 bits ensuite en convertit chaque groupe en base décimale selon le tableau suivant. Inversement pour convertir un nombre héxadécimal en binaire on convertit chaque chiffre en binaire sur 3 bits.

Nombre binaire	Valeur décimale
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Example

$(1753)_{10} = (11011011001)_2 = (\underline{011} \mid \underline{011} \mid \underline{011} \mid \underline{001})_2 = (3331)_8$

$(7632)_8 = (?)_2 = (?)_{10}$:

$(7632)_8 = (111 \mid 110 \mid 011 \mid 010)_2 = 3994$

3. Base Héxadécimale B = 16

Fundamental

Cette base utilise les chiffres: $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$.

Pour passer de la base décimale vers la base héxadécimale on utilise la division succéssive par 16, ou on utilise le système binaire en regroupant les bits de droite à gauche par ensemble de 04 bits et on convertit chaque groupe en décimal selon le tableau suivant.

Inversement pour convertir un nombre héxadécimal en binaire on convertit chaque chiffre en binaire sur 04 bits.

Nombre binaire	Valeur décimale
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	$A, (10)$
1011	$B, (11)$
1100	$C, (12)$
1101	$D, (13)$
1110	$E, (14)$
1111	$F, (15)$

Example

$(1753)_{10} = (?)_{16}$, on a $1753 \div 16 = 109$ reste **9**, $109 \div 16 = 6$ reste **13**, $6 \div 16 = 0$ reste **6**, donc $(1753)_{10} = (6D9)_{16}$.

Example

$(1753)_{10} = (11011011001)_2 = (\underline{0110} \mid \underline{1101} \mid \underline{1001})_2 = (6D9)_{16}$

Example

$(FAC29)_{16} = (?)_2$: $(FAC29)_{16} = (1111 \mid 1010 \mid 1100 \mid 0010 \mid 1001)_2$

D. Conversion d'un nombre réel fractionnaire

1. Conversion de la base décimale vers une base B

Lorsque le nombre N est fractionnaire, la conversion de sa partie entière vers une base B s'effectue avec l'une des deux méthodes que nous venons de voir (méthode des divisions et méthode des soustractions). La conversion de la partie fractionnaire, par contre, s'effectue en multipliant cette partie fractionnaire par B. La multiplication est itérée sur la partie fractionnaire du résultat obtenu. La conversion de la partie fractionnaire du nombre N est obtenue par la suite des parties entières de chacun des résultats des multiplications effectuées.

Example

- $(47.62)_{10} = (?)_5$:

  Partie entière: méthode des soustractions donne $47 = 25 + 4 \times 5 + 2 \times 5^0 = (142)_5$

  Partie fractionnaire/

  $0.62 \times 5 = 3.1$

  $0.1 \times 5 = 0.5$

  $0.5 \times 5 = 2.5$

  $0.5 \times 5 = 2.5$

  $0.5 \times 5 = 2.5$

  $\ldots$

  donc $(47.62)_{10} \backsimeq (142.30222)_5$
- $(42.3125)_{10} = (?)_2$:

  Partie entière: méthode des soustractions donne $42 = 2^5 + 2^3 + 2^1 = (101010)_2$

  Partie fractionnaire:

  $0.3125 \times 2 = 0.625$

  $0.625 \times 2 = 1.25$

  $0.25 \times 2 = 0.5$

  $0.5 \times 2 = 1.0$

  $0.0 \times 2 = 0.0$

  donc $(42.3125)_{10} = (101010.0101)_2$

2. Conversion de la base B vers la base décimale

Dans ce cas on utilise la formule des puissances suivante:

$(a_{n-1}a_{n-2}\cdots a_2a_1a_0, a_{-1}a_{-2}a_{-3}\cdots)_B =$

$a_{n-1}B^{n-1} + a_{n-2}B^{n-2} + \cdots + a_2B^2 + a_1B^1 + a_0B^0 + a_{-1}B^{-1} + a_{-2}B^{-2} + a_{-3}B^{-3} + \cdots$

avec comme condition: $a_{n-1}, a_{n-2}, \cdots a_2, a_1, a_0, a_{-1}, \cdots < B$ et $a_{n-1} \neq 0$.

Example

- $(101,011)_2 = 2^2 + 2^0 + 2^{-2} + 2^{-3} = 5.375$
- $(A2,B)_{16} = 10 \times 16^1 + 2 \times 16^0 + 11 \times 16^{-1} = 162.6875$

3. Conversion du binaire fractionnaire vers une base puissance de 2

a) Base Octale, n=3

Pour la partie entière on regroupe les bits de droite à gauche par groupe de 03 bits. si on manque de bits on rajoute des zéros. Ensuite on convertit chaque ensemble en décimal.

Pour la partie fractionnaire on regroupe les bits de gauche à droite par groupe de 03 bits. si on manque de bits on rajoute des zéros. Ensuite on convertit chaque ensemble en décimal.

Pour l'oprération inverse, chaque chiffre est convertit en binaire sur 3 bits.

Example

- $(1110101110, 011011101)_2 = (1656, 335)_8$
- $(5731, 026)_8 = (101111011001, 000010110)_2$

b) Base Héxadécimale, n=4

Pour la partie entière on regroupe les bits de droite à gauche par groupe de 04 bits. si on manque de bits on rajoute des zéros. Ensuite on convertit chaque ensemble en décimal.

Pour la partie fractionnaire on regroupe les bits de gauche à droite par groupe de 04 bits. si on manque de bits on rajoute des zéros. Ensuite on convertit chaque ensemble en décimal.

Pour l'oprération inverse, chaque chiffre est convertit en binaire sur 4 bits.

Example

- $(1110101110, 011011101)_2 = (3AE, 6E8)_{16}$
- $(ABC1, F59)_{16} = (1010101111000001, 111101011001)_2$

E. Exercices

1. Exercice 1

1. Convertir les nombres suivants en décimal:
   $(526)_7$; $(BAC)_{13}$; $(1101010011)_2$; $(2121)_3$; $(176)_8$; $(FADEC)_{16}$
2. Convertir les nombres ci-dessous (donnés en base 10) dans la base indiquée:
   173 en base 2 ; 254 en base 2 ; 1025 en base 16 ; 4218 en base 16 ; 5269 en base 8.
   (Utiliser deux méthodes pour chaque exemple)

2. Exercice 2

1. Déterminer la base $\boldsymbol{b}$ pour que les égalités ci-dessous soient vraies.
   $(132)_b = (30)_{10}$; $(A04)_b = (1444)_{10}$; $(2A)_{16} = (36)_b$

2. Déterminer les plus petites bases possibles ($\boldsymbol{a}$ et $\boldsymbol{b}$) pour que les égalités ci-dessous soient vraies:
$(101)_a = (401)_b$; $(101)_a = (10001)_b$; $(12)_a = (1002)_b$
3. Soit $\boldsymbol{x}$ le nombre entier naturel dont l'écriture binaire est $(10\cdots01)_2$ (n chiffres 0 encadrés par deux 1, n étant non nul). Comment s'écrivent x^2 et x^3 ?

3. Exercice 3

1. Conversion rapide vers une base 2^n:
$(B7AE)_{16} \rightarrow 2$; $(DCBF)_{16} \rightarrow 2$; $(56349)_{16} \rightarrow 2$; $(5567)_8 \rightarrow 2$; $(ABCD)_{16} \rightarrow 8$ $(2074)_8 \rightarrow 16$; $(1110100000111010)_2 \rightarrow 16$; $(1110100000111010)_2 \rightarrow 8$
2. Convertir le nombre 2021 en base 2, 8 et 16.
3. Écrire en base 16 le nombre dont l'écriture en base 4 est $(30123210)_4$ sans le convertir en base 10

4. Exercice 4

1. Convertir les nombres suivants en décimal : $(1101,011)_2$; $(765,42)_8$; $(FAC,9D)_{16}$
2. Convertir les nombres ci-dessous dans la base indiquée:
$(14,65)_{10} \rightarrow 8$ (3 chiffres après la virgule) ;
$(42,65625)_{10} \rightarrow 2$;
$(69,23)_{10} \rightarrow 16$ (3 chiffres après la virgule);
$(11011000110,001011011)_2 \rightarrow 16$;
$(1011110100,1110111)_2 \rightarrow 8$.

II

Calcul Arithmétique des nombres non signés

A. Arithmétique dans une base quelconque

1. Addition dans une base B

Méthode

Soient A et C deux nombres écrites dans la base B tel que:
$A = (a_{n-1}a_{n-2}\cdots a_1a_0)_B$ et $C = (c_{n-1}c_{n-2}\cdots c_1c_0)_B$. Que vaut $A + C$

L'addition des chiffres de rang i, a_i et c_i en base B se fait de la manière suivante:

- Si le résultat de l'addition est supérieure ou égale à B on le convertit en base B et on rapporte un retenu.
- Si le résultat est strictement inférieur à B on l'écrit tel qu'il est.

Example : Addition dans la base 7

$$\begin{array}{r} {\scriptstyle 1\ 1\ 1} \\ 6\ 5\ 3 \\ +\ 4\ 3\ 4 \\ \hline 1\ 4\ 2\ 0 \end{array}$$

On a $A = (653)_7$ et $C = (434)_7$ que vaut $(A + C)_7$?

On a $3 + 4 = 7 \geq 7$ et $7 = (10)_7$, donc on garde 0 et on rapporte 1 comme retenu.

On a $5 + 3 + 1 = 9$, on a $9 \geq 7$ et $9 = (12)_7$, donc on garde 2 et on rapporte 1 comme retenu.

On a $6 + 4 + 1 = 11 \geq 7$ et $11 = (14)_7$.

Finalement le résultat est $(1420)_7$.

Example : Addition dans la base Octale

On a $A = (754)_8$ et $C = (627)_8$ que vaut $(A + C)_8$?

On a $4 + 7 = 11 \geq 8$ et $11 = (13)_8$, donc on garde 3 et on rapporte 1 comme retenu.

On a $5 + 2 + 1 = 8 \geq 8$, et $8 = (10)_8$, donc on garde 0 et on rapporte 1 comme retenu.

On a $7 + 6 + 1 = 14 \geq 8$ et $14 = (16)_8$.

Finalement le résultat est $(1603)_8$.

$$\begin{array}{rcccc} & {\scriptstyle 1} & {\scriptstyle 1} & & {\scriptstyle 1} \\ & & 7 & 5 & 4 \\ + & & 6 & 2 & 7 \\ \hline & 1 & 6 & 0 & 3 \end{array}$$

L'opération d'addition dans la base octale est basée sur le tableau suivant. On lit ce tableau comme suit: l'addition, par exemple, de 5_8 et 7_8 est l'intersection entre la ligne (colonne) de 5_8 et la colonne (ligne) de 7_8, donc 14_8, qui est la conversion de $5 + 7 = 12$ en octale.

$+$	$(0)_8$	$(1)_8$	$(2)_8$	$(3)_8$	$(4)_8$	$(5)_8$	$(6)_8$	$(7)_8$
$(0)_8$	$(0)_8$	$(1)_8$	$(2)_8$	$(3)_8$	$(4)_8$	$(5)_8$	$(6)_8$	$(7)_8$
$(1)_8$	$(1)_8$	$(2)_8$	$(3)_8$	$(4)_8$	$(5)_8$	$(6)_8$	$(7)_8$	$(10)_8$
$(2)_8$	$(2)_8$	$(3)_8$	$(4)_8$	$(5)_8$	$(6)_8$	$(7)_8$	$(10)_8$	$(11)_8$
$(3)_8$	$(3)_8$	$(4)_8$	$(5)_8$	$(6)_8$	$(7)_8$	$(10)_8$	$(11)_8$	$(12)_8$
$(4)_8$	$(4)_8$	$(5)_8$	$(6)_8$	$(7)_8$	$(10)_8$	$(11)_8$	$(12)_8$	$(13)_8$
$(5)_8$	$(5)_8$	$(6)_8$	$(7)_8$	$(10)_8$	$(11)_8$	$(12)_8$	$(13)_8$	$(14)_8$
$(6)_8$	$(6)_8$	$(7)_8$	$(10)_8$	$(11)_8$	$(12)_8$	$(13)_8$	$(14)_8$	$(15)_8$
$(7)_8$	$(7)_8$	$(10)_8$	$(11)_8$	$(12)_8$	$(13)_8$	$(14)_8$	$(15)_8$	$(16)_8$

Example : Addition de deux nombres fractionnaires en octale

$$\begin{array}{rcccccc} & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & & & \\ & 1 & 2 & 4 & . & 7 & 0 \\ + & & 7 & 6 & . & 4 & 6 \\ \hline & 2 & 2 & 3 & . & 3 & 6 \end{array}$$

On désire calculer $124.70 + 76.46$

On suprime la virgule.

On a $0 + 6 = 6$, $7 + 4 = 11 = (13)_8$ on garde 3 et rapporte 1 comme retenu

On a $1 + 4 + 6 = 11 = (13)_8$, on garde 3 et rapporte 1 comme retenu

On a $1 + 2 + 7 = 10 = (12)_8$, on garde 2 et on rapporte 1 comme retenu

Finalement $1 + 1 = 2$, on remet la virgule à sa position, donc le résultat finale est 223.36

2. Soustraction dans une base B

Méthode

Soient A et C deux nombres écrites dans la base B tel que:
$A = (a_{n-1}a_{n-2} \cdots a_1a_0)_B$ et $C = (c_{n-1}c_{n-2} \cdots c_1c_0)_B$. Que vaut $A - C$

La soustraction des chiffres de rang i, a_i et c_i en base B se fait de la manière suivante:

- Si $a_0 > c_0$ la soustraction se fait comme dans le cas décimale et le résultat est écrit tel qu'il est.
- Si $a_0 < c_0$ on ajoute la valeur B à a_0 et 1 à c_1, puis on fait la soustraction $a_0 + B - c_0$
- Si $a_1 > c_1$ (ou $c_1 + 1$) la soustraction se fait comme dans le cas décimale et le résultat est écrit tel qu'il est.
- Si $a_1 < c_1$ (ou $c_1 + 1$) on ajoute la valeur B à a_0 et 1 à c_1, puis on fait la soustraction $a_1 + B - c_1$ ou $a_1 + B - (c_1 + 1)$
- On itère cette opération jusqu'à a_{n-1}

Example : Soustraction dans la base Octale

$$\begin{array}{rccc} & 6 & 5_{+8} & 3_{+8} \\ - & {}_{1+}4 & {}_{1+}6 & 4 \\ \hline & 1 & 6 & 7 \end{array}$$

On a $A = (653)_8$ et $C = (464)_5$ que vaut $(A - C)_8$?

On a $3 < 4$ on ajoute 8 à 3 et $11 - 4 = 7$, on ajoute 1 à 6.

On a $5 < 7$ donc on ajoute 8 à 5 et $13 - 7 = 6$, on ajoute 1 à 4.

Finalement $6 - 5 = 1$. Le résultat est 167_8.

Example : Soustraction dans la base Héxadécimale

Calcul de $(2021)_{16} - (BAC)_{16}$:

On a $1 < C$ on ajoute 16 à 1 et $17 - 12 = 5$, on ajoute 1 à A

On a $2 < 11$ on ajoute 16 à 2 et $18 - 11 = 7$, on ajoute 1 à B.

On a $0 < 12$, donc on ajoute 16 à 0 et $16 - 12 = 4$ et on ajoute 1.

Finalement, $2 - 1 = 1$. Le résultat est donc 1475_{16}.

$$\begin{array}{rcccc} & 2 & 0_{+16} & 2_{+16} & 1_{+16} \\ - & {}_{1} & {}_{1+}B & {}_{1+}A & C \\ \hline & 1 & 4 & 7 & 5 \end{array}$$

3. Multiplication dans une base B

Méthode

Soient A et C deux nombres écrites dans la base B tel que:
$A = (a_{n-1}a_{n-2} \cdots a_1a_0)_B$ et $C = (c_{n-1}c_{n-2} \cdots c_1c_0)_B$.

La multiplication des chiffres de rang i, a_i et b_i en base B se fait de la manière suivante:

- Si le résultat de la multiplication est supérieure ou égale à B on le convertit en base B et on rapporte un retenu.
- Si le résultat est strictement inférieur à B on l'écrit tel qu'il est.

Example : Multiplication en Octale

```
      1
      7 3
x     1 4
  _______
   1 1
    3 5 4
  + 7 3 .
  _______
  1 3 0 4
```

On a $A = (73)_8$ et $C = (14)_8$ que vaut $(A \times C)_8$?

On a $4 \times 3 = 12 \geq 8$ et $12 = (14)_7$, donc on garde 4 et on rapporte 1 comme retenu.

On a $4 \times 7 + 1 = 29$, on a $29 \geq 8$ et $29 = (35)_8$.

On a $1 \times 3 = 3 < 8$ et $1 \times 7 = 7 < 8$.

On addionne $354 + 730$. Donc $(A \times C)_8 = 1304_8$.

Example : Multiplication en Héxadécimale

On a $A = (2021)_{16}$ et $C = (BAC)_{16}$ que vaut $(A \times C)_{16}$?

On a $C \times 1 = C < 16$ donc on garde C.

On a $C \times 2 = 24$, on a $24 \geq 16$ et $24 = (18)_{16}$, donc on garde 8 et on rapporte 1 comme retenu.

On a $C \times 0 + 1 = 1 < 16$ et $C \times 2 = 24 = (18)_{16}$.

On a $A \times 1 = A < 16$ donc on garde A.

On a $A \times 2 = 20$, on a $20 \geq 16$ et $20 = (14)_{16}$, donc on garde 4 et on rapporte 1 comme retenu.

On a $A \times 0 + 1 = 1 < 16$ et $A \times 2 = 20 = (14)_{16}$.

```
        1,1
      2 0 2 1
x       B A C
  ___________
    1 8 1 8 C
  1 4 1 4 A .
1 6 1 6 B . .
  ___________
1 7 7 0 1 2 C
```

On a $B \times 1 = B < 16$ donc on garde B.

On a $B \times 2 = 22$, on a $22 \geq 16$ et $22 = (16)_{16}$, donc on garde 6 et on rapporte 1 comme retenu.

On a $B \times 0 + 1 = 1 < 16$ et $B \times 2 = 22 = (16)_{16}$.

On addionne $1818C + 1414A0 + 1616B00$. Donc $(A \times C)_8 = 177012C_{16}$.

4. Division dans une base B

Méthode

La division de deux nombres A et C, c.à.d. $A \div C$, dans une base quelconque B se fait en connaissant la table de multilication de C dans B.

Tout d'abord, à partir de A (de gauche à droite), on forme un nombre supérieur ou égale à C, puis on cherche dans la table de multiplication de C (dans la base de B) le multiplicateur de C inférieur ou égale au nombre formé, ensuite on fait une soustraction entre-eux. On ajoute un digit de A au résultat de cette soustraction. On itère ce processus jusqu'au dernier digit de A.

Example : Division dans la base 5

$$\begin{array}{r|l}
1444 & 23 \\ \cline{2-2}
124 & 34 \\ \hline
204 & \\
202 & \\ \hline
2 &
\end{array}$$

Que vaut $1444_5 \div 23_5$?:

La table de multiplication de 23_5 est :
$23_5 \times 0_5 = 0_5$, $23_5 \times 1_5 = 23_5$, $23_5 \times 2_5 = 101_5$, $23_5 \times 3_5 = 124_5$, $23_5 \times 4_5 = 202_5$, ….

Le nombre formé à partir de 1444 est 144.

D'après la table de multiplication, il y'a 3 fois 23 dans 144 et il reste 20, ($144 - 23_5 \times 3 = 20$).

On descend 4, on obtient 204 et d'après la table de multiplication, il y'a 4 fois 23 dans 204 et il reste 2, ($204 - 202_5 \times 4 = 2$).

Finalment on a $1444_5 \div 23_5 = 34$ et il reste 2_5.

Example : Division dans la base Octale

Que vaut $7625_8 \div 71_8$?:

La table de multiplication de 71_8 est :
$71_8 \times 0_8 = 0_8$, $71_8 \times 1_8 = 71_8$, …, $71_8 \times 5_8 = 435_8$, ….

Le nombre formé à partir de 7625 est 76.

D'après la table de multiplication, il y'a 1 fois 71 dans 76 et il reste 5, ($76 - 71_8 \times 1 = 5$).

On descend 2, on obtient 52 et d'après la table de multiplication, il y'a 0 fois 71 dans 52 et il reste 52.

On descend 5, on obtient 525 et d'après la table de multiplication, il y'a 5 fois 71 dans 525 et il reste 70, ($525 - 71_8 \times 5 = 70$).

Finalment on a $7625_8 \div 71_8 = 105_8$ et il reste 70_8.

Si on continu, on obtient $105,76_8$.

$$\begin{array}{r|l}
7625 & 71 \\ \cline{2-2}
71 & 105 \\ \hline
52 & \\
525 & \\
435 & \\ \hline
70 &
\end{array}$$

Example : Division dans la base Héxadécimale

```
2021 | CE
19C  |-----
----    27.ED
 661
 5A2
 ---
  BF0
  B44
  ---
   AC0
   A76
   ---
    4A
```

Que vaut $2021_{16} \div CE_{16}$?:

La table de multiplication de CE_{16} est :

$CE_{16} \times 2_{16} = 19C_{16}$, . . .,
$CE_{16} \times 7_{16} = 5A2_{16}$, ...

Le nombre formé à partir de 2021 est 202.

D'après la table de multiplication, il y'a 2 fois CE dans 202 et il reste 66, ($202 - CE_{16} \times 2 = 66$).

On descend 1, on obtient 661 et d'après la table de multiplication, il y'a 7 fois CE dans 661 et il reste BF,
$(661 - CE_{16} \times 7 = BF_{16})$

Finalment on a $2021_{16} \div CE_{16} = 27_{16}$ et il reste BF_{16}.

Si on continu, on obtient $27.ED_{16}$ et il reste $4A_{16}$

B. Arithmétiques des nombres binaires non signés

1. Addition des nombres binaires

Méthode

L'addition en binaire se fait de la même façon qu'au système décimale.

- On commence à additionner les bits de poids faibles qui se trouvent à droite;
- On a une retenue chaque fois que la somme est supérieure à deux. Cette retenue est reportée au bit de poids suivant et ainsi de suite jusqu'au bit le plus à gauche.

a_i	b_i	S	R
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Ce tableau donne la somme de deux bits a_i et b_i avec $S = a_i + b_i$ et R est la retenue.

Example

$$\begin{array}{lr} & 11111 \\ & 10001100 \\ + & 11111101 \\ \hline & 110001001 \end{array} \qquad \begin{array}{lr} & 1111111 \\ & 10011001 \\ + & 11111111 \\ \hline & 110011000 \end{array}$$

Advice

- Dans une colonne si le nombre de **1** est paire égale à $2m$, le résultat de la somme de cette colonne est toujours égale à 0 et le retenue est composée de $m\mathbf{1}$.
- Dans une colonne si le nombre de **1** est impaire égale à $2m+1$, le résultat de la somme de cette colonne est toujours égale à 1 et le retenue est composée de $m\mathbf{1}$.

Example

$$\begin{array}{lr} & 1\quad 1 \\ & 111111 \\ & 111111 \\ & 10101100 \\ & 11001100 \\ & 10001100 \\ + & 11111101 \\ \hline & 1100000001 \end{array} \qquad \begin{array}{lr} & 1\quad 1 \\ & 111111 \\ & 111111\ \ 1 \\ & 11101101 \\ & 10101100 \\ & 11001100 \\ & 10001100 \\ + & 11111101 \\ \hline & 1111101110 \end{array}$$

2. Sustraction des nombres binaires

Méthode

a_i	b_i	S	R
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Elle consiste à réaliser la différence entre deux nombres binaires selon le tableau ci-centre ou $S = a_i - b_i$ et R est la retenue. Si le résultat est négatif l'opération ne se termine jamais, d'où la nécessité d'introduire la notion des nombres binaires signés.

Example

$$\begin{array}{rr} & 11101100 \\ - & 10110001 \\ & 11\;\;\;\;11 \\ \hline & 00111011 \end{array} \qquad \begin{array}{rr} & 11101100 \\ - & 11110001 \\ \cdots & 11111\;\;\;\;11 \\ \hline \cdots & 1111111011 \end{array}$$

3. Multiplication/Division de deux nombres binaires

Méthode

Cette opération est similaire au cas décimale en utilisant le tableau ci-centre.

a_i	b_i	$a_i \times b_i$
0	0	0
0	1	0
1	0	0
1	1	1

Example

$$\begin{array}{cccccccc} & & & 1 & 1 & 0 & 1 \\ \times & & & & 1 & 0 & 1 \\ \hline & & & 1 & 1 & 0 & 1 \\ & & 0 & 0 & 0 & 0 & \bullet \\ + & 1 & 1 & 0 & 1 & \bullet & \bullet \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}$$

a) Multiplication de nombres en virgule fixe

- La multiplication se fait comme pour les nombres entiers.
- On fait d'abord la multiplication sans tenir en compte des virgules, puis à la fin on ajuste la virgule à la bonne place.

Note

La division s'effectue comme en décimale sauf que le résultat de division est soit égale à 0 ou 1.

C. Exercices

1. Exercice 1

- Effectuer les additions suivantes en binaire :
  $10101010 + 11001110$;
  $110111 + 101110 + 110011$;
  $1110011 + 1111111 + 1101011 + 101110$
- Effectuer les additions suivantes en octal (base 8) :
  $465 + 673$;
  $276 + 653 + 25$
- Effectuer les additions suivantes en hexadécimal (base 16) :
  $B796 + CAFE$;
  $8965 + 3979$;
  $324 + 697 + B2A$
- Si $137 + 276 = 435$, alors que vaut $731 + 672$

2. Exercice 2

Effectuer les soustractions suivantes en binaire
$11101101010 - 110111100$;
$10110001 - 10011111$;
$1101111 - 1110100$.

3. Exercice 3

- Effectuer les multiplications suivantes en binaire
  1101101×11001 ;
  11010110×101001.
- Effectuer les divisions suivantes en binaire
  a. $101100/101$ (5 chiffres après la virgule)
  b. $101011010/1101$ (4 chiffres après la virgule)

Représentation et Arithmétique des nombres signés

III

A. Représentation des nombres signés

Dans ce qui suit, on va s'intéressé aux nombres signés qui comportent un signe positif ou un signe négatif. On abordera tout d'abord les nombres binaires (dont la base est $B = 2$). Un nombre signé est représenté selon une séquence de bits d'une longueur fixée à n bits.

Plusieurs types de représentation des nombres binaires signés existent. Leur évolution est motivée par les incnvénients et limites d'applications. Mais on peut passer d'une représentation à une autre très facilement. Les représentations des nombres binaires signés sont:

1. Représentation en Signe-Valeur Absolue (SVA)

Definition

Dans cette représentation une séquence binaire $(a_{n-1}a_{n-2}\cdots a_1a_0)_2$ de n bits est codée comme suit:

- le bit le plus significatif (le plus à gauche): a_{n-1} est le bit de signe. Si va valeur est 0, alors le nombre est positif, si sa valeur est 1 , alors le nombre est négatif.
- les bits restantes $a_{n-2}\cdots a_1a_0$ codent la valeur absolue du nombre décimale.

Example

-27 est codé sur 6 bits par 111011, sur 8 bits par 10011011, car 11011 est le codage binaire de 27 (non signé) ou la valeur absolue.

$+27$ est codé sur 6 bits par 011011, sur 8 bits par 00011011, car 11011 est le codage binaire de 27 (non signé) ou la valeur absolue.

Advice

- L'inconvénient majeur de cette représentation est celle du 0. En effet, sur 8 bits $+0$ est codé par 00000000 et -0 par 10000000, ce qui nous donne deux codes pour le même nombre 0. D'ou la nécessité d'une autre représentation.
- L'intervalle des nombres codés en SVA est en fonction de la longueur de la chaîne binaire utilisée pour la représentation. Ainsi, sur 8 bits, l'intervalle des nombres représentables est $[11111111_2, 01111111_2]$, soit l'intervalle $[-127, +127]$. Donc toute opération arithmétique sur une machine de 8 bits dont le résultat est en dehors de cet intervalle est une opération faussée. Cela revient à demander à un enfant qui connait seulement les nombres de 1 à 10 de faire l'addition entre 5 et 8.

Longueur du nombre binaire	Intervalle dans la base décimale
2	$[-1, +1]$
3	$[-3, +3]$
4	$[-7, +7]$
8	$[-127, +127]$
16	$[-32\,767, +32\,767]$
32	$[-2\,147\,483\,647, +2\,147\,483\,647]$
$\dots$	$\dots$
n	$[-2^{n-1}+1, +2^{n-1}-1]$

2. Représentation en Complément à 1

Definition

Appelé aussi représentation en complément restreint, le complément à 1 (C_1) s'obtient en inversant les bits de la valeur absolue <u>*tout en gardant le bit de signe*</u>.

Example

-27 est codé en SVA sur 8 bits par 10011011, et en C_1 par 11100100

-127 est codé en SVA sur 8 bits par 11111111, et en C_1 par 10000000.

Advice

Le complément à 1 sur n bits d'un nombre négatif est le résultat de l'opération suivante:

$C_1(-N) = 2^n - N - 1$

Example

$C_1(-5)$ sur 4 bits. $C_1(-5) = 2^4 - 5 - 1 = 16 - 5 - 1 = 10 = 1010$.

Advice

Evaluation des nombres signés: Pour le complément à 1 d'un nombre négatif on affecte uns valeur négative pour le bit le plus significatif (MSB) et on fait la somme des points dont la valeur du bit est égale à 1, puis on ajoute 1.

Example

1. $11101000 = C_1(?)$. $-2^7 + 2^6 + 2^5 + 2^3 = -24$, on ajoute 1. Donc $11101000 = C_1(-23)$.
2. $11100100 = C_1(?)$. $-2^7 + 2^6 + 2^5 + 2^2 = -28$, on ajoute 1. Donc $11100100 = C_1(-27)$

Advice

L'intervalle des nombres codés en complément à 1 est le même que dans le cas du codage en SVA.

Advice

Le bit de poids fort a_{n-1} de la chaîne binaire $a_{n-1}a_{n-2}\cdots a_0$ peut être également interprété comme bit de signe. Ainsi :

- si $a_{n-1} = 0$, alors la chaîne binaire représente un nombre positif dont la valeur décimale s'obtient directement par conversion directe de la base binaire à la base décimale;
- si $a_{n-1} = 1$, alors la chaîne binaire représente un nombre négatif $-N$ dont la valeur décimale est celle du nombre positif associé $+N$ obtenu en complément à 1.

Example

$00110001 = C_1(?)$. On a $00110001 = (2^5 + 2^4 + 2^0) = C_1(+49)$

$11001110 = C_1(?)$. Pour le nombre positif on convertit les bits ce qui donne $00110001 = +49$, donc $11001110 = C_1(-49)$.

3. Représentation en Complément à 2

Definition

Le complément à 2 (C_2) d'un nombre négatif s'obtient en ajoutant 1 au complément à 1.

Example

N	SVA	C_1	C_2
-1	10000001	11111110	$11111110 + 1 = 11111111$
-55	10110111	11001000	$11001000 + 1 = 11001001$
-127	11111111	100000000	$10000000 + 1 = 10000001$

Advice

Le complément à 2 sur n bits d'un nombre négatif est le résultat de l'opération suivante:
$C_2(-N) = 2^n - N$

Example

$C_2(-5)$ sur 4 bits. $C_2(-5) = 2^4 - 5 = 16 - 5 = 11 = 1011$.

Advice

Evaluation des nombres signés: Pour le complément à 1 d'un nombre négatif on affecte uns valeur négative pour le bit le plus significatif (MSB) et on fait la somme des points dont la valeur du bit est égale à 1.

Example

1. $10100000 = C_2(?)$. $-2^7 + 2^5 = -96$. Donc $11101000 = C_1(-96)$.
2. $11111111 = C_2(?)$. $-2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 = -127$. Donc $11111111 = C_1(-127)$

Advice

L'intervalle des nombres codés en complément à 2 sur n bits est donné par $[-2^{n-1},\ 2^{n-1} - 1]$.

Advice

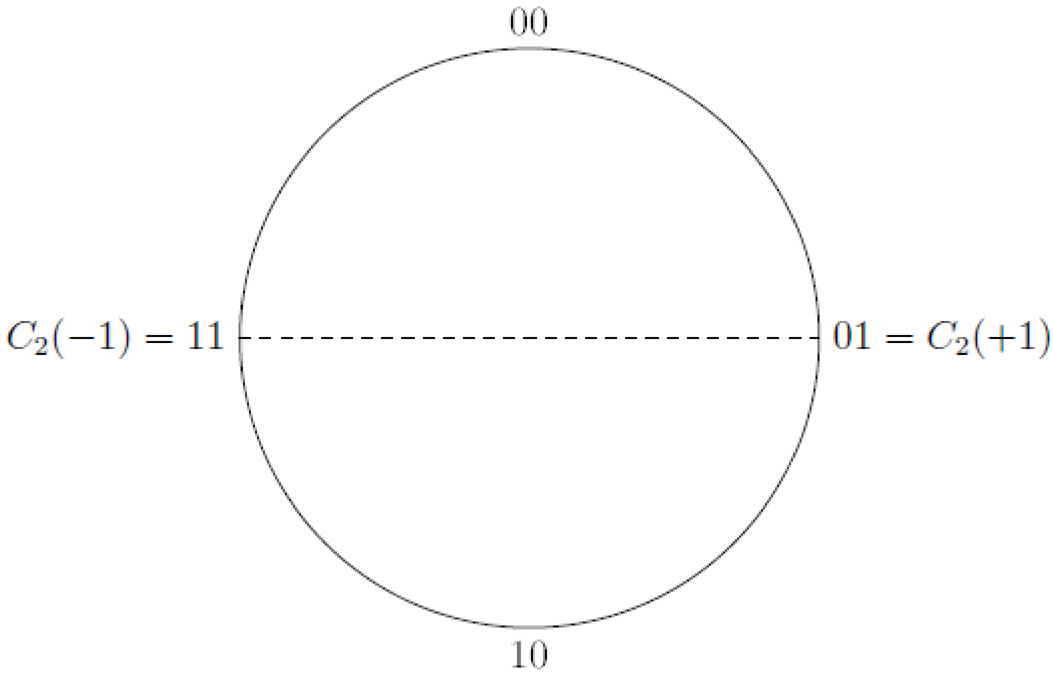
Pour calculer le complément à 2 d'un nombre négatif on procède comme suit: on parcourt les bits du nombre positif correspondant puis on inverse les bits juste après le premier 1 rencontré.

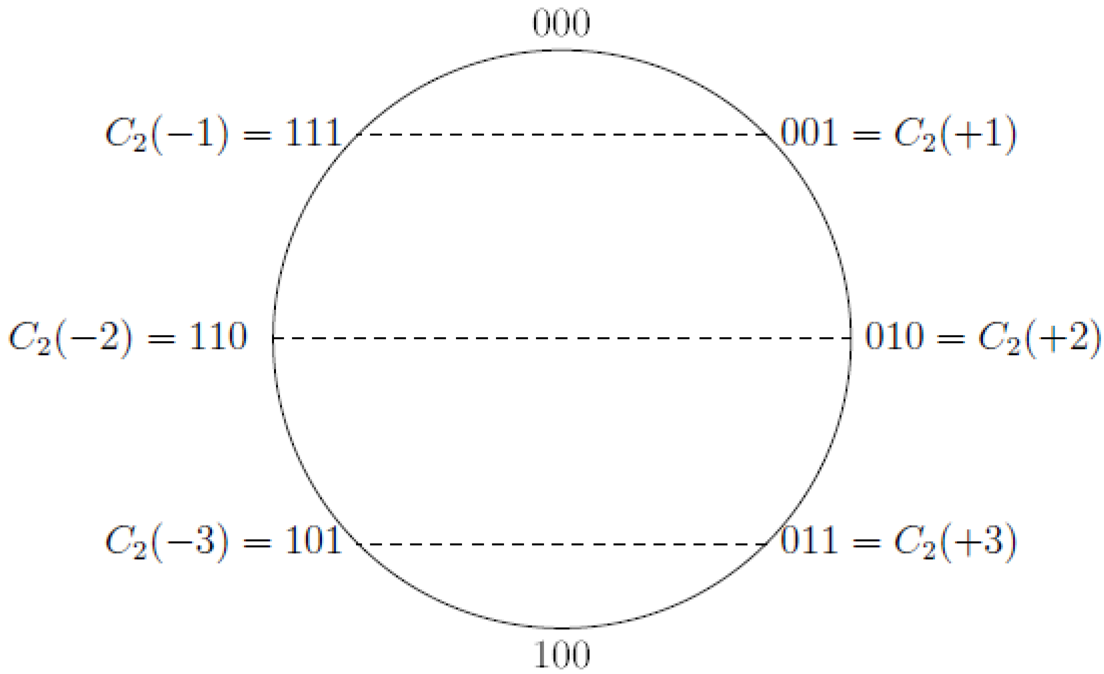
Example

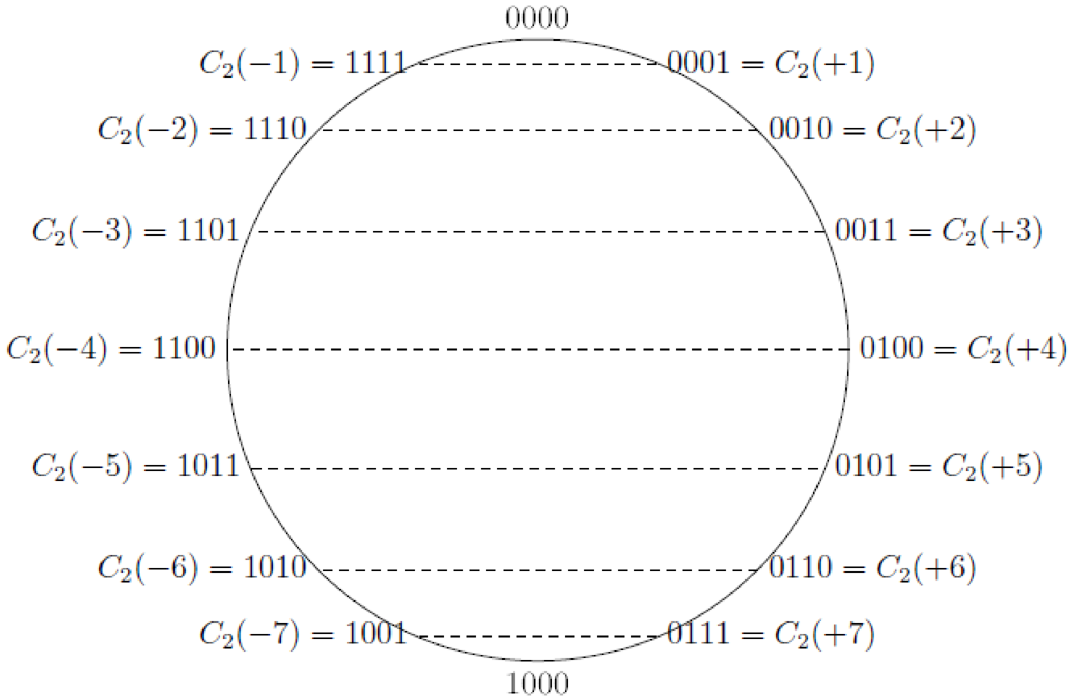
1. $C_2(-66) =?$. On a $+66 = 64 + 2 = 2^6 + 2^1 = 010000\mathbf{10}$, donc $C_2(-66) = 101111\mathbf{10}$
2. $C_2(-1) =?$. On a $+1 = 0000000\mathbf{1}$, donc $C_2(-1) = 1111111\mathbf{1}$.

Method : Méthode graphique

Il s'agit de placer des séquences binaires par ordre croissant sur le périmètre d'un cercle dans des positions équidistants. Les séquences binaires sue le demi-cercle droit sont les compléments à 2 des nombres positives correspondants, tandis que les séquences binaires sur le demi-cercle gauche sont les compléments à 2 des nombres négatifs correspondants.


00
$C_2(-1) = 11$
$01 = C_2(+1)$
10


000
$C_2(-1) = 111$
$001 = C_2(+1)$
$C_2(-2) = 110$
$010 = C_2(+2)$
$C_2(-3) = 101$
$011 = C_2(+3)$
100



B. Arithmétique des nombres signés

1. Notion de retenu et de dépassement de capacité

a) Notion de retenu

Definition

Lorsqu'on effectue une addition de deux séquences binaires de n bits et le résultat comporte $n + 1$ bits, le bit le plus significatif (MSB) représente une **retenu** et il n'est pas perdu mais sauvegardé par la machine.

Extra

La présence d'une **retenun'indique pas** que le résultat de l'addition est **érroné**.

b) Notion de dépassement de capacité

Definition

Dans une addition de deux nombres signées si le résultat déborde de l'intervalle $[-2^{n-1} - 1, 2^{n-1} - 1]$ pour le complément à 1, ou $[-2^{n-1}, 2^{n-1} - 1]$ pour le complément à 2, c'est-à-dire que ce résultat est en dehors de l'intervalle des nombres représentables sur n bits (par la convention choisie pour la représentation de ces nombres signés), alors le résultat obtenu est alors **érroné** au regard de son interprétation. On parle alors de **dépassement de capacité**.

Extra

La présence d'un **dépassement de capacité** implique que le **résultat** de l'addition est **érroné**.

2. Addition en complément à un

a) 2 nombres de signes opposés

Le résultat est représentable avec le nombre de bits fixés, **pas de dépassement de capacité** ;

- s'il n'y a pas de retenue, le résultat est un nombre négatif dont la valeur absolue est obtenue en inversant les bits ;
- s'il y a une retenue, le résultat est un nombre positif, on ajoute la retenue pour avoir sa valeur.

Example

On désire additionner $+45$ et -75 sur 08 bits:

On a $C_1(+45) = 00101101$ et $C_1(-75) = 10110100$, donc le résultat est 11100001, on a pas de retenu et $11100001 = C_1(-30)$

Example

On désire additionner $+83$ et -65 sur 08 bits:

On a $C_1(+83) = 01010011$ et $C_1(-65) = 10111110$, donc le résultat est 100010001, on a une retenu, et le résultat final est $00010001 + 1 = 00010010 = +18$

b) 2 nombres de même signes

Reminder

dépassement de capacité = résultat érroné

2 nombres de signe positif

- dépassement de capacité quand le bit de signe du résultat est 1 ;

2 nombres de signe négatif

- il y a une retenue puisque les deux bits de signe sont 1 ;
- on ajoute la retenue et
  a. un 0 pour le bit de signe signifie un dépassement de capacité ;
  b. un 1 pour le bit de signe, on a le bon résultat .

Example

On désire additionner $+15$ et $+125$ sur 08 bits:

On a $C_1(+15) = 00001111$ et $C_1(+125) = 1111101$, donc le résultat est 10001100, on a pas un dépassement de capacité car le bit de signe est 1.

Example

On désire additionner -37 et -56 sur 08 bits:

On a $C_1(-37) = 11011010$ et $C_1(-56) = 11000111$, donc le résultat est 110100001, on ajoute la retenu $10100001 + 1 = 10100010$. Le résultat final est bon : $10100010 = C_1(-93)$.

Example

On désire additionner -97 et -100 sur 08 bits:
On a $C_1(-97) = 10011110$ et $C_1(-100) = 10011011$, donc le résultat est 100111001, on a une retenu et le bit de signe est 0.Donc le résultat final est éronné (dépassement de capacité).

3. Addition en complément à deux

a) 2 nombres de signes opposés

Le résultat est représentable avec le nombre de bits fixés, **pas de dépassement de capacité** ;

- s'il y a une retenue, on l'oublie. On lit directement le résultat codé en complément à deux.

Example

On désire additionner $+45$ et -75 sur 08 bits:
On a $C_2(+45) = 00101101$ et $C_1(-75) = 10110101$, donc le résultat est 11100010, on a pas de retenu et $11100010 = C_2(-30)$

Example

On désire additionner $+83$ et -65 sur 08 bits:
On a $C_2(+83) = 01010011$ et $C_2(-65) = 10111111$, donc le résultat est 100010010, on a une retenu, et le résultat final est $00010010 = +18$

b) 2 nombres de même signes

- Il y a dépassement de capacité si la retenue est distincte du dernier bit de report (i.e. celui sur le bit de signe) ;
- S'il y a une retenue on l'oublie. On lit directement le résultat codé en complément à deux.

Example

1. Sur 8 bits, nous effectuons l'addition des nombres $+120_{10}$ et $+20_{10}$ représentés selon la convention du complément à 2 : $+120 = 01111000$ et $+20 = 00010100$, le résultat est $10100000 = C_2(-96)$, ce qui est un résultat erroné, donc on a un **dépassement de capacité** (DC).
2. Sur 8 bits, nous effectuons l'addition des nombres -1_{10} et -70_{10} représentés selon la convention du complément à 2: $C_2(-1) = 11111111$ et $C_2(-70) = 10111010$, le résultat est $10111001 = C_2(-71)$.

4. Soustraction en complément à 2

Toute opération de soustraction peut être transformée en une opération d'addition en utilisant la représentation en complément à deux:
$A - B = A + (-B) = A + C_2(-B)$
Dans le calcul du complément à deux de $-B$ le bit de signe doit aussi être inversé.

Example

1. $0111 - 0101 = 7 - 5 = +2$ et $0111 + C_2(-5) = 0111 + 1011 = \mathbf{1}\ 0010 = +2$
2. $1111 - 1001 = -1 - (-7) = +6$ et $1111 + C_2(+7) = 1111 + 0111 = \mathbf{1}\ 0110 = +6$
3. $11111111 - 00000010 = -1 - (+2) = -3$ et $11111111 + C_2(-2) = 11111111 + 11111110 = \mathbf{1}\ 11111101 = -3$

C. Exercices

1. Exercice 1

Compléter le tableau suivant:

Valeur décimale	SVA	C_1	C_2
-25			
	10010111		
		10001100	
			10001111

2. Exercice 2

Représentez le nombre -119 en binaire sur 8 bits dans le codage SVA et en C_2.

Suivant la représentation, quel est l'entier codé sur 8 bits par 11010110 ?

$11010110 = SVA(?)$, $11010110 = C_1(?)$, $11010110 = C_2(?)$

Effectuer les soustraction suivantes en complément à deux et dite si on a un dépassement de capacité ou non:

$11010111 - 11110000$; $-01000101 - 11111100$

Représentation et Arithmétique des nombres signés à virgule flottante

IV

A. La norme IEEE-754

1. Format normalisé d'un nombre

La norme **IEEE-754** est utilisée pour normaliser les nombres en virgule flottante. Elle définit un format standard utilisé dans la plupart des ordinateurs 32 ou 64 bits. D'ou, cette norme propose deux formats de représentation: un format simple précision sur 32 bits et un format double précision sur 64 bits.

Definition

Un nombre binaire est représenté en virgule flottante et écrit sous la forme suivante:

$\pm 1, M \times 2^{\pm e}$

ou M est dite **mantisse** et e est l'**exposant**.

Example

1. $-10.25_{10} = -1010,01_2 = -1,01001 \times 2^3$, donc $M = 01001$ et $e = +3$
2. $+0.85_{10} = +0,110110011011 \cdots = 1.10110011011 \cdots \times 2^{-1}$, donc $M = 10110011011 \cdots$ et $e = -1$

2. IEEE-754 Simple précision 32 bits

Cette norme utilise le format suivant:

- Bit de signe (1 bit): 0 si le nombre est positif et 1 si le nombre est négatif.
- Exposant normalisé $E = 127 + e$ codé sur 08 bits.

- Mantisse M codé sur 23 bits

Example

1. $-10.25_{10} = -1010,01_2 = -1,01001 \times 2^3$, donc $M = 01001$ et $e = +3$. $S = 1$, $E = 3 + 127 = 130 = 10000010_2$ et $M = 01001000 \cdots 000$, donc $-10.25_{10} = 1 \mid 10000010 \mid 01001000 \cdots 000_{IEEE-754}$
2. $+0.85_{10} = +0,110110011011 \cdots = 1.10110011011 \cdots \times 2^{-1}$, donc $M = 10110011011 \cdots$ et $e = -1$. $S = 0$, $E = -1 + 127 = 126 = 01111110_2$ et $M = 10110011011 \cdots$, donc $+0.85_{10} = 0 \mid 01111110 \mid 10110011011001101100110_{IEEE-754}$

Note

Il est judicieux de travailler avec le format hexadécimale en regroupant les groupes de 04 bits par un élément de $\{0, 1, 2, \cdots, 9, A, B, C, D, E, F\}$. En effet, les formes compactes des exemples précédents sont:

1. $-10.25_{10} = 1100 \mid 0001 \mid 0010 \mid 0100 \mid 0 \cdots 000_{IEEE-754}$ $= (C1240000)_{IEEE-754}$.
2. $+0.85_{10} = 0011 \mid 1111 \mid 0101 \mid 1001 \mid 1011 \mid 0011 \mid 0110 \mid 0110_{IEEE-754}$ $= (3F59B366)_{IEEE-754}$.

3. Cenversion inverse de la norme IEEE-754 SP vers le décimale

Pour passer de la représentation **IEEE-754** Simple Précision à la base décimale il suffit de parcourir le chemin inverse. En effet, il suffit de détecter le signe, l'exposé e de l'exposé étendu E par: $e = E - 127$ et finalement la mantisse. Les exemples suivants illustrent la procédure à suivre:

Example

1. $C0600000_{IEEE-754} = (N)_{10}$. On a $C0600000 = 1 \mid 10000000 \mid 11000 \cdots 000$, on déduit $S = 1$, $E = 10000000_2 = 128$ donc $e = 128 - 127 = 1$, et $M = 11$. Finalement $N = -1.11 \times 2^1 = -11.1_2 = -3.5_{10}$.
2. $C17B0000_{IEEE-754} = (N)_{10}$. On a $C17B0000 = 1 \mid 10000010 \mid 11110110 \cdots 000$, on déduit $S = 1$, $E = 10000010_2 = 130$ donc $e = 130 - 127 = 3$, et $M = 11110110$. Finalement $N = -1.1111011 \times 2^3 = -1111.1011_2 = -15.6875_{10}$.

B. Arithmetique dans la norme IEEE-754

1. Addition de deux nombres en IEEE-754 SP

L'addition de deux nombres écrits sous la forme IEEE-754 SP se fait de la manière suivante:

- On doit alligné les deux nombres pour qu'ils aient le même exposant, ensuite on additionne les mantisses après alignement.
- Généralement, l'alignement se fait pour le nombre ayant l'exposant le plus petit pour ne pas perdre en précision.

$A = \pm 1, M_1 \times 2^{e_1}$, $B = \pm 1, M_2 \times 2^{e_2}$, avec $e_1 < e_2$. On allighe A à l'exposant de B, soit

$A = \pm 1, M_1' \times 2^{e_2}$, donc $A + B = \pm 1, (M$

Example

$2.25 = (10,01)_2 = 1,001 \times 2^1$. On a $S = 0$ (signe positif), $e = 1, E = 127 + e = 128 = (10000000)_2$ et la mantisse $M = 0010 \cdots 0$.

$134.0625 = (10000110,0001)_2 = 1,00001100001 \times 2^7$. On a $S = 0$ (signe positif), $e = 7, E = 127 + 7 = 128 = (10000110)_2$ et la mantisse $M = 000011000010 \cdots 0$.

L'allignement: $2.25 = 0.000001001 \times 2^7$, la mantisse devienne $M = 0000010010 \cdots 0$. L'addition des mantisse se fait comme en binaire: $00000100100000000000000 + 00001100001000000000000 = 00010000101000000000000$.

L'addition des deux nombres donne: $01000011000010000101 0 \cdots 0 = 43085000)_{IEEE-754}$

V

Codage de l'information

L'action de codage consiste à faire correspendre des nombres de lettres ou des mots à un groupe spécial de symboles appelé "**code**". L'opération inverse est appelée "**décodage**".

A. Le codage BCD "Binary Coded Decimal"

1. Principe

Definition

Le code BCD (Binary Coded Decimal) qui signifie Décimal codé binaire permet de représenter un chiffre décimal de 0 à 9 par un ensemble de 4 bits. Le nombre de mots du code BCD est 10. Dans ce codage chaque bit est pondéré par des poids: le premier on lui associe un poids $P_0 = 1$, le second on lui associe un poids $P_1 = 2$, le troisième un poids $P_2 = 4$ et le quatrième un poids $P_3 = 8$. On l'appelle souvent le codage (8421).

Chiffre décimal	Mot BCD
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110

Chiffre décimal	Mot BCD
7	0111
8	1000
9	1001

Example

$(953)_{10} = (100101010011)_2$

Note

Une autre pondération peutêtre utilisée pour donnée naissance à un autre code comme celui d'**AIKEN**. Ce code utilise les poids **2421**

Les mots du code AIKEN sont donnés par le tableau suivant:

Chiffre décimal	Mot AIKEN
0	0000
1	0001
2	0010
3	0011
4	0100
5	1011
6	1100
7	1101
8	1110
9	1111

2. Arithmétique en BCD

a) Complément à 9 d'un nombre décimal

Le complément à 9 d'un nombre décimal se fait par complémentation de chaque chiffre par rapport à 9.

Example

$C_9(523) = 476$
$C_9(901) = 098$

b) Complément à 10 d'un nombre décimal

Le complément à 10 d'un nombre décimal se fait par ajout de 1 au complément à 9: $C_{10} = C_9 + 1$.

Example

$C_{10}(523) = 476 + 1 = 477$
$C_{10}(901) = 098 + 1 = 099$

c) Représentation d'un nombre signé en décimal

$\oplus \rightarrow 0$: le signe plusest codé par 0
$\ominus \rightarrow 9$: le signe moin est codé par 9

Example

$C_9(-54) = 945$, donc $C_{10}(-54) = 946$
$C_9(-17) = 982$, donc $C_{10} = 983$

Method

Pour effectuer une opération de soustraction en décimal on calcul le C_{10} des nombres négatifs: $A - B = A + C_{10}(-B)$

Example

$35 - 17 = 35 + C_{10}(-17) = 035 + 983 = 1018$. On ignore le chiffre supplémentaire (1), le résultat sera donc en $C_{10} = 018 = +18$

Advice

Si le résultat de l'opération d'addition ou de soustraction est positif on le lit directement, par contre si le résultat est négatif on recalcule son complément à 10.

Example

$24 - 54 = 24 + C_{10}(-54) = 024 + 946 = C_{10}(970) = -(029 + 1) = -30$

d) L'addition en BCD

L'addition en BCD consiste à convertir chaque chiffre décimal en binaire sur 4 bits puis effectuer l'addition des mots de code de chaque rang.

Example

$(14 + 24)_{10} = (00010100)_{BCD} + (00100100)_{BCD} = (00111000)_{BCD} = 38_{10}$

Note

On sait que les codes BCD allant de 0 à 9. Si le résultat de l'addition de deux mots est strictement supérieure à 9 on retranche 10 ce qui revient à ajouter 6 en BCD c.à.d 0110.

$$
\begin{array}{rr}
 & 16 \\
+ & 17 \\
\hline
 & 33
\end{array}
\qquad
\begin{array}{rr}
 & 0001\ 0110 \\
 & 0001\ 0111 \\
\hline
 & 0010\ 1101 \\
+ & 0110 \\
\hline
 & 0011\ 0011 \\
\hline
 & 3 \quad\quad 3
\end{array}
$$

e) La soustraction en BCD

Pour la soustraction en BCD on utilise le complément à 10. La soustraction en BCD est convertit toujours en une addition: $A - B = A + C_{10}(-B)$

Example

$38 - 28 = 38 + C_{10}(-28) = 038 + 972 = 010$ en BCD on a:

$$
\begin{array}{r}
0000\ 0011\ 1000 \\
1001\ 0111\ 0010 \\
\hline
1001\ 1010\ 1010 \\
0110\ 0110 \\
\hline
1010\ 0001\ 0000 \\
0110\ 0001\ 0000 \\
\hline
\cancel{1}\,0000\ 0001\ 0000
\end{array}
$$

B. Code binaire réfléchi ou code Gray

Le code Gray est un code construit de telle façon qu'a partir du chiffre 0 chaque nombre consécutif diffère du précédent immédiat d'un seul digit.

En l'exprimant autrement nous pouvons également dire que l'on change un seul bit à la fois quand un nombre est augmenté d'une unité.

De plus, nous opèrons de telle manière que le digit de transformation soit d'un poids faible. Si une erreur survient lors d'une transformation d'un nombre à un autre elle est ainsi minimisée.

1. Construction du code Gray

Method

Si $(b_{n-1}b_{n-2}\cdots b_0)_2 = (G_{n-1}G_{n-2}\cdots G_0)_{gray}$

- Le nombre de bits du code Gray est égale au nombre de bits du code binaire naturel
- Le bit du poids ford du code Gray est égale au bit du poids fort du code binaire naturel: $b_{n-1} = G_{n-1}$
- Pour $i \neq n-1$ on a $G_i = b_{i+1} \bigoplus b_i = 1$ si $b_{i-1} \neq b_i 0$ sinon

Example

$(11)_{10} = (1011)_2 = (G_3G_2G_1G_0)_{gray}$
$G_3 = b_3 = 1$
$G_2 = b_3 \bigoplus b_2 = 1$, $G_1 = b_2 \bigoplus b_1 = 1$, $G_0 = b_1 \bigoplus b_0 = 0$, d'ou
$(11)_{10} = (1011)_2 = (1110)_{gray}$

Example

$(95)_{10} = 2^6 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 = (1011111)_2 = (G_6G_5G_4G_3G_2G_1G_0)_{gray}$
$G_6 = b_6 = 1$
$G_5 = b_6 \bigoplus b_5 = 1$, $G_4 = b_5 \bigoplus b_4 = 1$, $G_3 = b_4 \bigoplus b_3 = 0$, $G_2 = b_3 \bigoplus b_2 = 0$,
$G_1 = b_2 \bigoplus b_1 = 0$ et $G_0 = b_1 \bigoplus b_0 = 0$, d'ou $(95)_{10} = (1011111)_2 = (1110000)_{gray}$

Fundamental : Tableaux des codes gray

En applicant la méthode précédante on construit les tableaux des codes gray pour différentes valeurs de n:

- Cas de $n = 2$
- Cas de $n = 3$
- Cas de $n = 4$

C. Code ASCII - American Standard Code of Information Interchange

A.S.C.I.I. est l'abréviation de American Standard Code for Information Exchange. Ce codage consiste à associer une valeur numérique binaire (interprétable en hexadécimal, décimal, ...) à chacun des caractères utilisables dans l'échange de données informatique : caractères alphabétiques et numériques (alphanumérique), ponctuation, codes de contrôles divers.

Developpé initialement par le **American National Standards Institute** (ANSI).

Code de 7 bits (128 entrees possibles, 95 graphiques et 33 de contrôle), stocké sur un octet (08 bits), c.à.d. de 0000000 au 1111111.

Le bit est quelquefois inutilise, utilisé comme bit de parité, ou pour coder 128 autres symboles.

1. Comment lire du tableau ASCII

Table de caractères et Codes hexadécimaux.

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Method

Le tableau contient:

- 95 codes graphiques de 20_{16} à $7E_{16}$ dont des codes alphabétiques, des codes numériques et des codes de ponctuation.
- 33 codes de contrôle de 00_{16} à $1F_{16}$ et $7F_{16}$

Example

La chaîne de caractère Hello, world ! , a pour code (en hexadecimal), ´ $48656C6C6F2C20776F726C6421$

a) Codes graphiques

a a le code hexadécimal 61_{16}. Pour convertir ce caractère en caractère majuscule (i.e., A), on doit soustraire au code 20_{16} (touche shift), on obtient 41_{16}. L'ordre des lettres est respecté (classement par ordre alphabétique.

Le caractère "**5**" codé par le 35_{16} est différent du nombre 5. Pour convertir le caractère en nombres on doit soustraire au code la valeur 30_{16}.

Circuits logiques

VI

Dans cette partie, on va décrire d'une manière un peu plus profonde un outil d'analyse des circuits logiques qui constituent le matériel du traitement de l'information qui est l'ordinateur. Cet outil est l'algèbre de Boole de son fondateur le mathématicien anglaisGeorge Boole (1815-1864) qui a proposéles principes de cet art.

A. Algèbre de Boole

Dans la réalité il existe des situations ou on a deux états (situations) possibles:

porte ouverte où fermée, lampe éteinte où allumée, condensateur chargé où déchargé, un bit $= 1$ où $= 0$. Pour étudier un phénomène à deux états on utilise la théorie de l'algèbre de Boole où algèbre booléen.

Pour chaque état on associe deux valeurs qui sont vraie (1) et faux (0) (logique positive).

Une situation alors est représentée par une variable logique(booléenne) notée $X \in E = \{0, 1\}$. Si on possède n variables logiques il existe donc un ensemble de 2^n permutations possibles de ces variables.

Dans cette algèbre, il existe que trois opérations: OU logique (représentée par le signe +), ET logique (représentée par le signe . ou une simple concaténation), et NOT logique (représentée par le signe bar $\bar{X}$. Les trois et uniques opérations sont des fonctions logiques schématisées par la table logique suivante (dite table de vérité):

X	Y	$\bar{X}$	$X + Y$	XY
0	0	1	0	0
0	1	1	1	0
1	0	0	1	0
1	1	0	1	1

1. Axiomes et théorèmes

a) Axiomes de l'algèbre de Boole

Soit $B = \{0, 1\}$ un ensemble muni par le triplet des opération $(\bar{}, +, \cdot)$. Les posulats ou axiomes sont cités ci-dessous.

i Posulat 1

La loi $(+)$ est une loi de composition interne: $\forall x, y \in B$ alors $x + y \in B$

ii Posulat 2

La loi $(\cdot)$ est une loi de composition interne: $\forall x, y \in B$ alors $x \cdot y \in B$

iii Postulat 3

Il existe un élement neutre noté 0 pour la loi $(+)$: $\forall x \in B$ alors $x + 0 = x$

iv Postulat 4

Il existe un élement neutre noté 1 pour la loi $(\cdot)$: $\forall x \in B$ alors $x \cdot 1 = x$

v Postulat 5

La loi $(+)$ est **commutative**: $\forall x, y \in B$ alors $x + y = y + x$

vi Postulat 6

La loi $(\cdot)$ est **commutative**: $\forall x, y \in B$ alors $x \cdot y = y \cdot x$

vii Postulat 7

La loi $(+)$ est distributive par rapport à la loi $(\cdot)$: $\forall x, y, z \in B$ alors $x + (y \cdot z) = (x + y)(x + z)$

viii Postulat 8

La loi $(\cdot)$ est distributive par rapport à la loi $(+)$: $\forall x, y, z \in B$ alors $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$

ix Postulat 9

La loi de négation $\overline{(\cdot)}$ appelée complément logique vérifie :
$\forall x \in B, \exists \overline{X} \in B / X + \overline{X} = 1$ et $X \cdot \overline{X} = 0$

b) Théorèmes fondamentaux de l'algèbre de Boole

i Théorème 1: Eléments absorbants

$\forall X \in B$ alors $X + 1 = 1$ et $X \cdot 0 = 0$
Preuve: **"P" signifie Postulat**

$X + 1 = (X + 1) \cdot 1$ en utilisant P4 $= (X + 1) \cdot (X + \overline{X})$ en utilisant P9 $= X + 1 \cdot \overline{X}$ en utilisant P7 $= X + \overline{X}$ en utilisant P4 $= X$ en utilisant P9	$X \cdot 0 = (X \cdot 0) + 0$ en utilisant P3 $= (X \cdot 0) + (X \cdot \overline{X})$ en utilisant P9 $= X \cdot (0 + \overline{X})$ en utilisant P8 $= X \cdot \overline{X}$ en utilisant P3 $= 0$ en utilisant P9

ii Théorème 2: Théorème d'idempotence

$\forall X \in B$ alors $X + X = X$ et $X \cdot X = X$
Preuve: **"P" signifie Postulat**

$X + X = (X + X) \cdot 1$ en utilisant P4 $= (X + X) \cdot (X + \overline{X})$ en utilisant P9 $= X + X \cdot \overline{X}$ en utilisant P7 $= X + 0$ en utilisant P9 $= X$ en utilisant P3	$X \cdot X = (X \cdot X) + 0$ en utilisant P3 $= (X \cdot X) + (X \cdot \overline{X})$ en utilisant P9 $= X \cdot (X + \overline{X})$ en utilisant P8 $= X \cdot 1$ en utilisant P9 $= X$ en utilisant P4

iii Théorème 3: Théorème d'involution

$\forall X \in B$ alors $\overline{\overline{X}} = X$
Preuve: **"P" signifie Postulat**
On pose $Y = \overline{X}$, donc d'après P9 on a $Y + \overline{Y} = 1$ et $Y \cdot \overline{Y} = 0$ ce qui implique que $\overline{\overline{X}} = X$.

iv Théorème 4: Théorème d'absorption

$\forall X, Y \in B$ on a: $X + (X \cdot Y) = X$ et $X \cdot (X + Y) = X$
Preuve: **"P" signifie Postulat etT signifie Théorème**

$X + X \cdot Y = X \cdot 1 + X \cdot Y$ en utilisant P4 $= X \cdot (1 + Y)$ en utilisant P8 $= X \cdot 1$ en utilisant T1 $= X$ en utilisant P4	$X \cdot (X + Y) = (X + X) \cdot (X + Y)$ en utilisant T2 $= X + (X \cdot Y)$ en utilisant P7 $= X$ en utilisant T4.1

2. Lois de De Morgan

a) Principe

Deux théorèmes ont une importance particulière, il s'agit des lois de De Morganqui s'énoncent simplement:
$\forall X_0, X_1, \cdots, X_n \in B$, on a:
$\overline{X_0 \cdot X_1 \cdot X_2 \cdot \cdots \cdot X_n} = \overline{X_0} + \overline{X_1} + \overline{X_2} + \cdots + \overline{X_n}$
$\overline{X_0 + X_1 + X_2 + \cdots + X_n} = \overline{X_0} \cdot \overline{X_1} \cdot \overline{X_2} \cdots \overline{X_n}$

b) Démonstration

Pour cette démonstration on va considérer seulement le cas $n = 2$, c.à.d:
$\overline{A + B} = \overline{A} \cdot \overline{B}$
$\overline{AB} = \overline{A} + \overline{B}$
Preuve:
En se basant sur le postulat 9 on doit simplement démontrer que:
$(A + B) \cdot (\overline{A} \cdot \overline{B}) = 0$, $(A + B) + (\overline{A} \cdot \overline{B}) = 1$ pour la première
et $AB(\overline{A} + \overline{B}) = 0$, $AB + (\overline{A} + \overline{B}) = 1$ pour la deuxième.

$\overline{A+B} = \overline{A}\cdot\overline{B}$	$\overline{AB} = \overline{A} + \overline{B}$
• $(A+B)\cdot(\overline{A}\cdot\overline{B}) = (A\cdot\overline{A})\cdot\overline{B} + (B\cdot\overline{B})\cdot$ • $(A+B)+(\overline{A}\cdot\overline{B}) = (A+\overline{A}+B)(B+\overline{B}+$ $= (1+B)(1+A) = 1$	• $AB(\overline{A}+\overline{B}) = A\overline{A}B + AB\overline{B} = 0$ • $AB + (\overline{A}+\overline{B})$ $= (\overline{A}+\overline{B}+A)(\overline{B}+\overline{A}+B)$ $= (1+\overline{B})(1+\overline{A}) = 1$

c) Exemple d'utilisation

Dans cet exemple on va utiliser le théorème de De Morgan pour démontrer le théorème suivant:

Example

Théorème : $\forall X, Y, Z \in B$ on a $(X+Y)(\overline{X}+Z)(Y+Z) = (X+Y)(\overline{X}+Z)$

Method

$$(X+Y)(\overline{X}+Z)(Y+Z) = \overline{\overline{(X+Y)(\overline{X}+Z)(Y+Z)}}$$
$$= \overline{\overline{X+Y} + \overline{\overline{X}+Z} + \overline{Y+Z}}$$
$$= \overline{\overline{X}\cdot\overline{Y} + X\cdot\overline{Z} + \overline{Y}\cdot\overline{Z}}$$
$$= \overline{\overline{X}\cdot\overline{Y} + X\cdot\overline{Z} + \overline{Y}\cdot\overline{Z}(X+\overline{X})}$$
$$= \overline{\overline{X}\cdot\overline{Y} + X\cdot\overline{Z} + \overline{Y}\cdot\overline{Z}\cdot X + \overline{Y}\cdot\overline{Z}\cdot\overline{X}}$$
$$= \overline{\overline{X}\cdot\overline{Y}(1+\overline{Z}) + X\cdot\overline{Z}(1+\overline{Y})}$$
$$= \overline{\overline{X}\cdot\overline{Y} + X\cdot\overline{Z}}$$
$$= \overline{\overline{(X+Y)} + \overline{(\overline{X}+Z)}}$$
$$= (X+Y)(\overline{X}+Z)$$