Chapitre 0

Introduction

L'homme a besoin d'un système de codage pour identifier, quantifier, qualifier les objets, les lieux, les évènements…Ces codes lui permettent de mémoriser, traiter et communiquer les informations. Ils peuvent être : le langage, l'écriture, numérations, le graphisme, etc..
Chaque code respecte des règles bien définies. Par exemple, à l'écriture correspond:

- Une liste des symboles prédéfinis : L'alphabet
- Des règles d'utilisations des symboles :
  - Plusieurs symboles (lettres) → un mot
  - Plusieurs mots → une phrase
- Des règles de syntaxe pour ordonner les mots dans une phrase.

Pour un ordinateur, le codage de l'information permet d'établir une correspondance qui permet sans ambiguïté de passer d'une représentation d'une information dite externe (une image, un son, un texte..) à une autre représentation de la même information (dite interne: sous forme binaire, n'utilisant que des 0 et des 1) suivant un ensemble de règles précises.

La raison pour laquelle les ordinateurs manipulent des données binaires est liée au fonctionnement de leurs composants physiques. Les transistors et les condensateurs, qui sont les éléments de base d'un ordinateur, possèdent deux états stables : activé/désactivé ou chargé/déchargé. Ainsi, un transistor dans l'état activé va stocker l'information 1 (ou 0 s'il est dans l'état désactivé).

L'objectif de ce document est de prendre connaissance des notions de base du codage de l'information, ainsi que des connaissances sur la théorie formelle basée sur l'Algèbre de Boole pour la synthèse des circuits.

Ce travail est destiné aux étudiants de première année licence.

Chapitre 1

Codification et représentation des nombres

1 Objectifs

- Savoir définir la base d'un système de numération.
- Savoir définir le rang et le poids d'un chiffre.
- Savoir représenter un nombre sous forme polynomiale.
- Savoir déterminer la valeur décimale d'un nombre de base b quelconque et vice versa.
- Savoir convertir un nombre binaire en un nombre octal ou en hexadécimal et vice versa.
- Savoir effectuer les opérations arithmétiques directement dans le système binaire, octal et hexadécimal.
- Comment représenter les nombres négatifs dans la machine.
- Comment représenter les nombres réels dans la machine.

2 Introduction

Les systèmes numériques complexes tels que les calculateurs doivent traiter toute sorte d'informations. A cette fin, ces informations doivent être codées à l'aide des chaines binaires. Ce chapitre décrit les notions fondamentales du codage de l'information utilisé par les systèmes numériques ainsi que les opérations arithmétiques réalisées sur ces codes.

3 Les systèmes de numération

Le système de numération décrit la façon avec laquelle les nombres sont représentés ; et il est définit par :

- Un alphabet: ensemble de symboles ou chiffres,
- Des règles d'écritures des nombres: Juxtaposition de symboles.

Il existe plusieurs systèmes de numérations dont les plus connus sont [1]:

- Le système décimal (b=10) qui est utilisé et pratiqué dans notre vie quotidienne. Ce système utilise dix chiffres: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}.
- Le système binaire (b=2) qui est utilisé par les ordinateurs. Ce système utilise deux chiffres {0, 1}. Par convention on identifie 0 comme une absence de tension, et 1 comme une présence de tension. L'ordinateur comprend donc uniquement des nombres en base 2. Exemple : 101101 , 01100110, 11111111, 10000001. Ces 0 et 1 sont appelés bit qui est une abréviation de **bi**nary digi**t**.

- Le système octal ($b=8=2^3$) qui permet de coder trois bits par un seul symbole. Ce système utilise huit chiffres: {0, 1, 2, 3, 4, 5, 6, 7}.
- Le système hexadécimal (b=16) est utilisé pour réduire encore plus l'écriture des nombres binaire. La base hexadécimale est aussi une puissance de 2 ($16 = 2^4$). qui permet de coder quatre bits par un seul symbole. Ce système utilise size chiffres: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}.

4 Les entiers positifs

4.1 Forme polynomiale

On peut décomposer tout nombre N en fonction de puissances entières de la base de son système de numération. On notera en indice la base du système de numération dans lequel le nombre N envisagé est écrit (10 dans l'exemple ci-dessous) [2].
Considérons par exemple, le nombre décimal 1234, On aura :

$$(1234)_{(10)} = 1x10^3 + 2x10^2 + 3x10^1 + 4x10^0$$

Le chiffre de droite (4 dans l'exemple) s'appelle le chiffre de poids faible. Celui de gauche (1 dans l'exemple) s'appelle le chiffre de poids fort. L'exposant de la base, associé à un chiffre d'un nombre quelconque, s'appelle le **rang**. Par exemple 4 est de rang 0 tandis que 1 est de rang 3 dans l'exemple ci dessus. On peut généraliser cette notion et écrire sous forme polynomiale tout nombre N de base b quelconque.

On aura :

$$N = \sum_{i=0}^{i=n} a_i \times b^i \qquad (1)$$

Où a_i est un chiffre qui appartient à la base b tel que $0 \leq a_i < b$. i est le rang du chiffre a_i et n est l'exposant de b du chiffre de poids fort.

4.2 Changement de base (transcodage)

Le transcodage (ou conversion de base) est l'opération qui permet de passer de la représentation d'un nombre exprimé dans une base (b) à la représentation du même nombre mais exprimé dans une autre base (b').

4.2.1 Conversion d'un nombre de base quelconque en un nombre de base décimale

Elle s'obtient par la forme polynomiale vue au paragraphe précédent.

Exemple : donner la valeur décimale du nombre binaire N=1010
Solution :

$$(1010)_{(2)} = \sum_{i=0}^{i=3} a_i \times 2^i = 1x2^3 + 0x2^2 + 1x2^1 + 0x2^0 = 8 + 0 + 2 + 0 = 10_{(10)}$$

$a_3\ a_2\ a_1\ a_0$

4.2.2 *Conversion d'un nombre de base décimale en un nombre de base quelconque*

Soit N un nombre écrit en base décimale. Si on veut le convertir dans une autre base b, il faut donc que l'égalité donné par l'équation 1 soit vérifiée. Le problème revient donc à déterminer les valeurs de a_i. Pour cela deux méthodes sont possibles :

1ère méthode : la méthode des puissances qui consiste à chercher les différentes puissances entières de la base b. l'algorithme commence par chercher la plus grande puissance entière de b contenue dans N, retrancher ensuite cette quantité du nombre N, recommencer ce processus en considérant le reste obtenu.

Exemple : convertir le nombre $N=75_{(10)}$ en nombre Octal.

Solution

On aura successivement :

$75 - 64 \longrightarrow 1 \times 8^2$

$11 - 8 \longrightarrow 1 \times 8^1$

$3 \longrightarrow 3 \times 8^0$

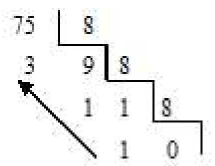
Donc $N=75_{(10)}= 1 \times 8^2 + 1 \times 8^1 + 3 \times 8^0$

On a donc : $N=75_{(10)} =113_{(8)}$

i	8^i
0	1
1	8
2	64

2ème méthode (L'algorithme d'Euclide): cette méthode est simple et plus rapide que la précédente. Elle consiste à faire des divisions successives par la base b jusqu'à ce que le quotient soit égal à zéro. On écrit ensuite tous les restes à partir de la fin et de gauche à droite, en les convertissant en lettres s'il y a lieu (dans le cas hexadécimale par exemple).

Pour le nombre précédant, on obtient :



On a donc : $N=75_{(10)} =113_{(8)}$

5 Les nombres fractionnaires

On écrira un nombre fractionnaire Nf inferieur à 1 sous la forme :

$$Nf = \sum_{i=1}^{i=n} a_i \times b^{-i}$$

$0 \leq a_i < b$, i est le rang du chiffre a_i et n est l'exposant du chiffre de poids faible.

Problème 1: convertir un nombre fractionnaire de base b en décimal.
Exemple : convertir Nf=0,1011$_{(2)}$ en décimal.

Solution : nous obtenons :
Nf=$1x2^{-1} + 0x2^{-2} + 1x2^{-3} + 1x2^{-4}$ =0.5+0.125+0.0625=0.6875$_{(10)}$

Problème 2 : convertir un nombre décimal fractionnaire en un nombre de base b.
Pour convertir un nombre décimal fractionnaire en un nombre de base b, il faut multiplier sa partie fractionnaire par la base. La partie entière du résultat est encadré, et la partie fractionnaire résultante est multipliée par la base. On recommence ce processus jusqu'à ce qu'un de ces critères soit vérifié :

1. Le résultat de la multiplication est un nombre entier
2. Après une certaine précision (nombre de chiffres après la virgule)

Exemple : convertir Nf=0,85$_{(10)}$ en binaire, prendre 04 chiffres après la virgule.

Solution : nous obtenons :
0.85 x 2 = 1,70
0.70 x 2 = 1.4
0.4 x 2 = 0.8
0.8 x 2= 1.6

On écrit de gauche à droite les nombres encadrés pris de haut en bas.
On aura donc Nf=0,85$_{(10)}$ = 0,1101$_{(2)}$

6 Les conversions directes (Conversions par paquets)

6.1 Conversion binaire-octal et vice versa

Correspondance	
Octal	Binaire
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Tableau 1 : Correspondance entre le système octal et le système binaire

Le système octal (b=8) permet de coder trois bits par un seul chiffre. Ce système utilise les huit chiffres suivants: {0, 1, 2, 3, 4, 5, 6, 7}.Pour convertir un nombre fractionnaire binaire en octal, il faut juste grouper les bits par blocs de trois à partir de la virgule, en allant vers la gauche pour la partie entière et vers la droite pour la partie fractionnaire. Convertir ensuite ces blocs en octal en se basant sur le tableau 1 suivant :

Exemple 1 :
Convertir le nombre binaire N=11010101,11(2) en Octal.

Solution :

N=	011	010	101	,	$110_{(2)}$
	3	2	5	,	$6_{(8)}$

Exemple 2 :
Convertir le nombre octal N=$657{,}12_{(8)}$ en binaire.

Solution :
Ecrire par blocs de trois bits, la valeur binaire des chiffres du nombre octal. On obtient :

6	5	7	,	1	$2_{(8)}$
110	101	111	,	001	$010_{(2)}$

6.2 Conversion binaire-hexadécimal et vice versa

Le système hexadécimal (b=16) permet de coder quatre bits par un seul symbole. Ce système utilise les dix chiffres de la base décimale: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}, plus, les six symboles suivants : {A, B, C, D, E, F}. Pour convertir un nombre fractionnaire binaire en Hexadécimal, il faut juste grouper les bits par blocs de quatre à partir de la virgule, en allant vers la gauche pour la partie entière et vers la droite pour la partie fractionnaire. Convertir ensuite ces blocs en Hexadécimal en se basant sur le tableau 2 :

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

Tableau 2: Correspondance entre le système Hexadécimal et le système binaire