Codage de l'information

Table des matières

Introduction

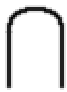
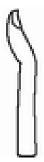
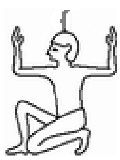
Le traitement automatique de l'information dans les systèmes nécessite un **codage** de cette information, de façon à pouvoir la transmettre, la traduire, l'afficher...

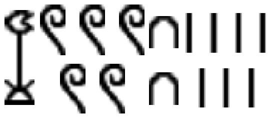
Systèmes de numération

1. Numération additive, puis positionnelle

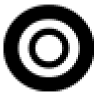
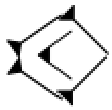
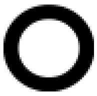
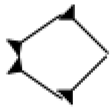
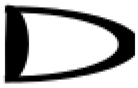
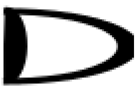
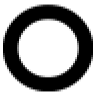
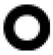
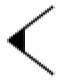
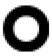
Historiquement, les premiers systèmes de numération ont été additifs : **plusieurs symboles** avaient chacun une valeur différente, et il suffisait de les **additionner** pour obtenir le résultat final. Chaque symbole était répété autant de fois que nécessaire, et organisé parfois par superposition pour des questions d'espace.

Système de numération de l'Égypte antique - base 10 ? *Exemple*

Valeur	Signe hiéroglyphique
1	
10	
100	
1000	
10000	
100000	
1000000	

1527 donne 

Système de numération en Mésopotamie (3000 av JC) - base 60 ***Exemple***

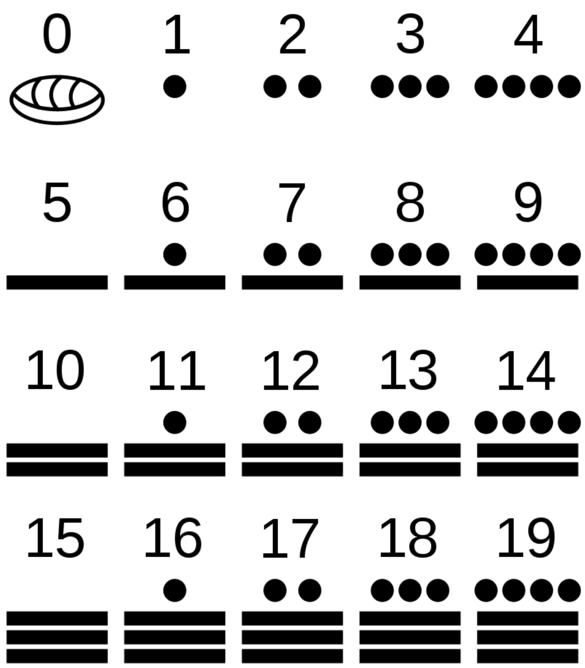
Valeur	sumérien pour objets	sumérien pour graines	Signe mésopotamien cunéiforme
36000			
3600			
1800			
600			
180			
60			
10			
6			
1			

Pour des questions de facilité de calcul et de lecture, la **numération positionnelle** s'est progressivement installée.

Numération positionnelle ***Définition***

Dans la numération positionnelle, chaque **position** d'un chiffre ou symbole est **reliée** à la position voisine par un **multiplicateur**, appelé **base du système de numération**.

Système de numération maya (2000 av JC) - base 20 *Exemple*



 correspond à $9018 = 1 \times 7200 + 5 \times 360 + 0 \times 20 + 18$

2. Bases

Chaque nombre entier N peut être représenté dans une base b :

$N_{(b)} = a_n\ b^n + a_{n-1}\ b^{n-1} + \ldots + a_0\ b^0$

- a_i sont les **symboles** (ou caractères, ou « *digits* ») de la base ; il y a autant de symboles que la valeur de la base
- b_i représentent les **poids** des symboles a_i, avec b qui est la **valeur** de la base.

On écrit habituellement la base en indice du nombre écrit entre parenthèses.

Bases courantes *Définition*

- Base 2 (base binaire) = {0,1}
- Base 10 (base décimale) = {0,1,2,3,4,5,6,7,8,9}
- Base 16 (base hexadécimale) = {0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F}

Remarque

Il existe d'autres bases moins utilisées dans l'industrie ou en sciences (base 8, base 60), voire d'intérêt historique (base 12, base 20, cf. partie précédente)

La base 2 est extrêmement utilisée en informatique où transitent des données logiques à deux valeurs possibles.

La manipulation de ces données nécessite fréquemment le passage en base 16 (où des paquets de 4 bits de la base 2 sont regroupés et exprimés par un seul symbole de la base 16). Les nombres binaires de huit chiffres sont aussi utilisés : les **octets**.

3. Passage d'une base à une autre

Transcodage de la base décimale vers les bases binaire ou hexadécimale *Exemple*

$(29)_{10} = 1.2^4 + 1.2^3 + 1.2^2 + 0.2^1 + 1.2^0 = (11101)_2$

$(29)_{10} = 1.16^1 + 13.16^0 = (1D)_{16}$

3.1. Passage d'une base quelconque à la base décimale

D'après la définition du nombre entier N représenté dans une base b, un nombre exprimé dans une base :

- binaire $(1011)_2$ sera égal à $1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 1 * 1^0$ donc $8 + 0 + 2 + 1 = 11$ de la base décimale
- hexadécimale $(A3B)_{16}$ sera égal à $10 * 16^2 + 3 * 16^1 + 11 * 16^0$ donc $2560 + 48 + 11 = 2619$ de la base décimale

3.2. Passage de la base décimale à une autre

Division successive *Méthode*

Puisqu'un nombre peut être écrit $N_{(b)} = a_n\ b^n + a_{n-1}\ b^{n-1} + \ldots + a_1\ b^1 + a_0$, la méthode consiste à obtenir les poids b_i des symboles a_i.

En divisant par la valeur b de la base, on obtient $N_{(b)} = b\left(a_n\ b^{n-1} + a_{n-1}\ b^{n-2} + \ldots + a_1\ b^0\right) + a_0$, où :

- $a_n\ b^{n-1} + a_{n-1}\ b^{n-2} + \ldots + a_1\ b^0$ est le **quotient**
- a_0 le **reste**.

Le reste donne donc la valeur de a_0, et il suffit de répéter l'opération pour obtenir les valeurs des a_i, jusqu'à obtenir un quotient de 0.

Attention

On obtient les symboles de poids les plus faibles en premier !

Écriture d'un nombre décimal en binaire *Exemple*

On cherche à écrire 19 en binaire. La valeur b de la base est donc 2.

- $19 = 2 * 9 + 1 \longrightarrow$ reste 1
- $9 = 2 * 4 + 1 \longrightarrow$ reste 1
- $4 = 2 * 2 + 0 \longrightarrow$ reste 0
- $2 = 2 * 1 + 0 \longrightarrow$ reste 0
- $1 = 2 * 0 + 1 \longrightarrow$ reste 1

Ainsi, 19 s'écrit en binaire 10011.

Codes

On peut distinguer les codes :

- **pondérés**, où chaque symbole possède un poids
- **non pondérés**, où une table de correspondance est nécessaire pour coder et décoder l'information.

1. Codes binaires

1.1. Code binaire naturel

C'est le code **pondéré** binaire **correspondant à la définition de la base 2**.

Il nécessite cependant une grande quantité de **bits** (**bi**nary digi**ts**) pour exprimer un nombre. Par exemple, un mot binaire de 4 bits ne pourra représenter qu'un nombre décimal compris entre 0 et 15, et un mot binaire de 8 bits (octet) ne pourra représenter qu'un nombre décimal compris entre 0 et 255.

De plus, plusieurs bits peuvent changer de valeur lorsqu'on passe d'une valeur décimale à une autre (exemple : $(3)_{10} = (011)_2$ et $(4)_{10} = (100)_2$).

Méthode

Pour remplir une tableau en binaire naturel, il suffit de :

- commencer à la première ligne avec uniquement des **0**
- alterner les **0** et **1** toutes 2^n lignes, pour chaque bit de poids n

Exemple

valeur décimale	2^3	2^2	2^1	2^0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1

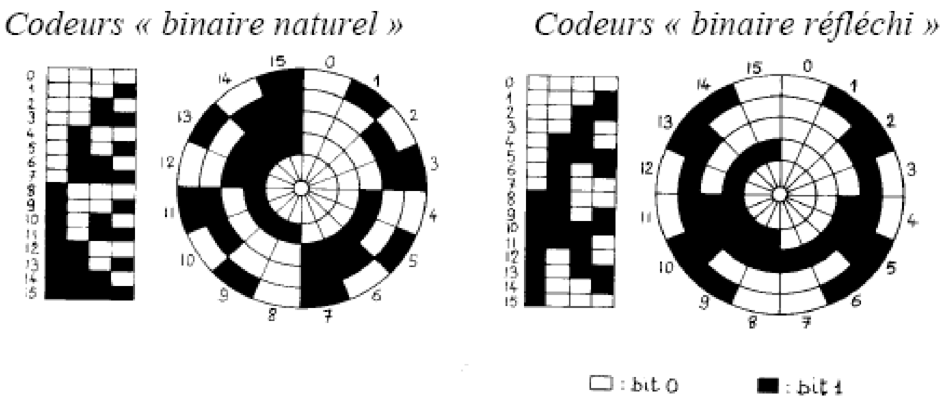
valeur décimale	2^3	2^2	2^1	2^0
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

1.2. Code binaire réfléchi (par exemple code Gray)

C'est un code **non pondéré** binaire qui utilise des "symétries" (ou "réflexions") visuelles dans son écriture.

Il est très utilisé dans les codeurs absolus (roue codeuse), et dans les tableaux de Karnaugh (afin de permettre la simplification visuelle des équations logiques).

L'un des avantages est qu'**un seul bit change de valeur entre deux valeurs décimales adjacentes** : cela garantit une erreur d'interprétation maximale d'un seul incrément.



Méthode

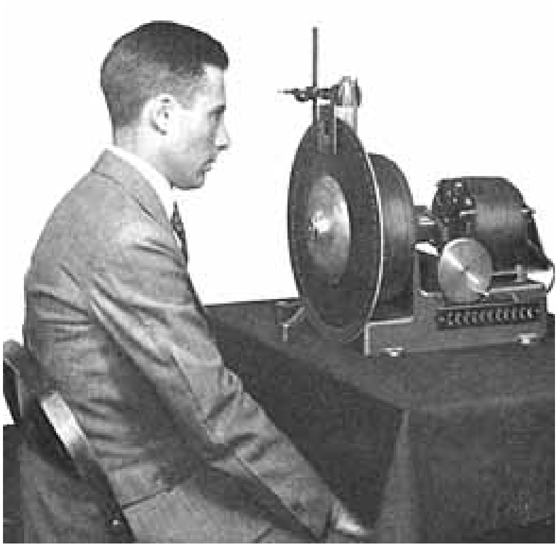
Pour obtenir le code Gray à partir d'une table de vérité en binaire naturel, il faut effectuer une symétrie toutes les $2\ (2^1),\ 4\ (2^2),\ 8\ (2^3),\ 16\ (2^4)$... lignes :

- en commençant par le bit de poids le plus faible
- en changeant le bit de poids le plus fort

Exemple

valeur décimale	2^3	2^2	2^1	2^0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	0	1	0
4	0	1	1	0
5	0	1	1	1
6	0	1	0	1
7	0	1	0	0
8	1	1	0	0
9	1	1	0	1
10	1	1	1	1
11	1	1	1	0
12	1	0	1	0
13	1	0	1	1
14	1	0	0	1
15	1	0	0	0

Complément



Franck Gray

Physicien et chercheur aux laboratoires Bell, il apporta plusieurs innovations aux systèmes de télévision, en utilisant des systèmes électro-optiques. Il développa un code binaire réfléchi qui a conservé son nom.

1.3. Code Décimal Codé en Binaire (DCB)

C'est un code **pondéré** basé sur le binaire naturel, mais qui est adapté à la représentation des nombres décimaux.

Le code binaire naturel n'associe pas de bit particulier aux unités, dizaines, centaines, etc.. En revanche, le DCB associe 4 bits pour chaque puissance de 10 de la base décimale.

Ce code est utilisé pour les afficheurs 7 segments : chaque afficheur reçoit sous la forme d'un nombre binaire à 4 bits le chiffre à afficher compris entre 0 et 9.

Exemple : 1664 s'écrit $0001.0110.0110.0100$ en DCB.

L'inconvénient est que le nombre de bits nécessaires est plus important qu'en binaire naturel : la taille mémoire nécessaire dans le système est plus grande.

1.4. Code p parmi n

C'est un code **non pondéré** qui consiste à choisir parmi n bits :

- p bits égaux à 1
- $n - p$ bits égaux à 0.

Un code p parmi n courant est une version du code 3 parmi 5.

En voici sa table de correspondance, où d_0 est le nombre décimal et $c_4c_3c_2c_1c_0$ est le mot en DCB :

Code 3 parmi 5

d_0	c_4	c_3	c_2	c_1	c_0
0	0	0	1	1	1
1	0	1	0	1	1
2	0	1	1	0	1
3	0	1	1	1	0
4	1	0	0	1	1
5	1	0	1	0	1
6	1	0	1	1	0
7	1	1	0	0	1
8	1	1	0	1	0
9	1	1	1	0	0

L'application courante utilisant ce code est le **tri postal automatisé**. Un code numérique est associé au pli et traduit en code à barres sur l'enveloppe.

Les bâtonnets foncés correspondent à 1 et les bâtonnets transparents à 0, et des espaces séparent les bâtonnets.

Remarque

Un intérêt majeur du code "p parmi n" est la détection d'erreur dans la transmission de l'information.

En effet, si l'on reçoit un nombre codé en "2 parmi 5" par exemple : pour détecter une éventuelle erreur il suffit de compter le nombre de 1 présent dans chaque groupe de 5 bits. Si un groupe ne présente pas deux 1, on peut conclure qu'il est erroné.

2. Codes-barres

Définition

Un code à barres ou "code-barres" est la représentation d'une donnée sous forme d'un symbole constitué de barres (ou carrés) et d'espaces dont l'épaisseur varie.

Histoire

Le premier brevet d'un code à barres date de 1952, et avait pour but d'automatiser l'enregistrement de produits manufacturés. L'utilisation massive dans les supermarchés s'est généralisée dans les années 1970.

Types

- **unidimensionnel** : série de lignes parallèles d'épaisseur variable. La lecture peur être faite de façon bidirectionnelle pour confirmer le décodage fait dans le premier sens.

  exemple : code-barres EAN retrouvé sur les produits de supermarchés
- **bidimensionnel** :
  - empilé : lignes unidimensionnelles empilées verticalement. La lecture peut être faite par balayage.
  - 2D : motif rectangulaire ou carré nécessitant une lecture par prise de photo.

    exemple : code QR

Code QR

5 901234 123457 >

Code EAN

Exemples de codeur, décodeur, transcodeur

III

1. Codeur

Un **codeur** transforme généralement des informations données par l'opérateur en informations compréhensibles par le système.

Codeur décimal

Exemple

Soit un codeur décimal transformant un chiffre décimal (de 0 à 9, c'est-à-dire entrée e_i) en mot binaire de quatre bits ($a_3\ a_2\ a_1\ a_0$)

Table de vérité du codeur

Entrées	Sorties			
	a_3	a_2	a_1	a_0
e_0	0	0	0	0
e_1	0	0	0	1
e_2	0	0	1	0
e_3	0	0	1	1
e_4	0	1	0	0
e_5	0	1	0	1
e_6	0	1	1	0
e_7	0	1	1	1
e_8	1	0	0	0
e_9	1	0	0	1

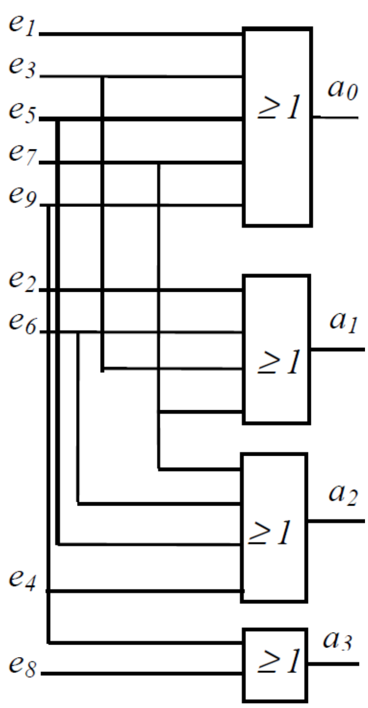
Équations logiques de chacune des sorties :

$a_0 = e_1 + e_3 + e_5 + e_7 + e_9$

$a_1 = e_2 + e_3 + e_6 + e_7$

$a_2 = e_4 + e_5 + e_6 + e_7$

$a_3 = e_8 + e_9$



Logigramme du codeur décimal

2. Décodeur

Après traitement de l'information par le système, il est souvent nécessaire de **décoder** le résultat afin de le rendre compréhensible par l'opérateur.

Décodeur décimal ***Exemple***

Soit un décodeur décimal transformant une information codée sur quatre bits en un chiffre décimal.

Table de vérité du décodeur

Entrées				Sorties
a_3	a_2	a_1	a_0	
0	0	0	0	s_0
0	0	0	1	s_1
0	0	1	0	s_2
0	0	1	1	s_3
0	1	0	0	s_4
0	1	0	1	s_5
0	1	1	0	s_6
0	1	1	1	s_7
1	0	0	0	s_8
1	0	0	1	s_9

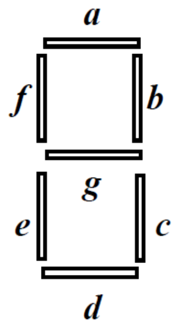
On peut obtenir une équation logique pour chacune des sorties, par exemple ici $s_7 = \overline{a_3}\ a_2\ a_1\ a_0$

3. Transcodeur

Il permet de passer d'un code à un autre.

Transcodeur pour afficheur 7 segments ? *Exemple*

Un calcul numérique étant fait sur quatre bits a_i, on veut présenter le résultat sous la forme d'un chiffre décimal apparaissant sur un afficheur à sept segments.



Afficheur 7 segments

Table de vérité

Nombre décimal	**Entrées**				**Sorties**						
	a_3	a_2	a_1	a_0	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1

a_3a_2 a_0a_1	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	X	X	X	X
10	1	1	X	X

Les 6 combinaisons d'entrée non utiles pour l'afficheur sont représentées par les "X" dans le tableau de Karnaugh.

La simplification visuelle peut utiliser indifféremment ces cases comme 0 ou 1.

$a = a_1 + a_3 + a_0\, a_2 + \overline{a_0}\, \overline{a_2}$

a_3a_2 a_0a_1	00	01	11	10
00	1	0	0	1
01	0	0	0	1
11	X	X	X	X
10	1	0	X	X

$e = a_1\, \overline{a_0} + \overline{a_0}\, \overline{a_2}$