

Informatique fondamentale

Binaire et opérations

R.Gosswiller

- 1 Introduction
- 2 Rappels d'algèbre booléenne
- 3 Encoder en binaire
- 4 Représentation des entiers
- 5 La compilation

Introduction

L'informatique

L'informatique n'est pas seulement le code !

Domaines

Mobile, Web, Intelligence artificielle, Robotique, Big data, Internet des objets, Informatique industrielle, ...

Domaines d'application

Spatial, Industrie, Banque, Commerce, Administration, Communication, ...

Qu'est-ce que la mémoire ?

Définition

La mémoire est un espace virtuel dans lequel l'on stocke des données. Cet espace a une taille finie.

Adresse

Chaque emplacement en mémoire possède un identifiant unique, nommé **adresse**.

Chaque adresse correspond à un emplacement de taille 1 octet.
Plan d'adressage indexé selon des octets.

Qu'est-ce que la mémoire ?

Dispositifs

Il existe différents composants chargés du stockage électronique de l'information

Exemples

Bande magnétique, flash, CD, disquette, ... : mémoire 'morte' Barettes,

Qu'est-ce que la mémoire ?

Indexation

La méthode d'organisation et de répartition des données en mémoire s'appelle le **plan d'adressage**.

Qu'est-ce que la mémoire ?

Il existe deux types de mémoire

Mémoire vive (RAM)

Composant électroniquement proche du processeur

Accès rapide

Temps de vie limité

Mémoire cache

Mémoire morte (ROM)

Zone de stockage étendue

Eloignée du processeur

Temps de vie illimité

Rappels d'algèbre booléenne

Tables de vérité

Opérations logiques fondamentales

ET

a	b	$a.b$
0	0	0
0	1	0
1	0	0
1	1	1

NON

a	$\neg a$
0	1
1	0
-	-
-	-

OU

a	b	$a + b$
0	0	0
0	1	1
1	0	1
1	1	1

XOR

a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

Tables de vérité

Opérations logiques fondamentales

NAND

a	b	$\neg(a.b)$
0	0	1
0	1	1
1	0	1
1	1	0

IMP

a	b	$a \implies b$
0	0	1
0	1	1
1	0	0
1	1	1

NOR

a	b	$\neg(a + b)$
0	0	1
0	1	0
1	0	0
1	1	0

XNOR

a	b	$\neg(a \oplus b)$
0	0	1
0	1	0
1	0	0
1	1	1

Lois de Boole

Idempotence

$$a + a + a + \dots + a = a$$

Associativité

$$(a.b).c = a.(b.c) = a.b.c$$

$$(a + b) + c = a + (b + c) = a + b + c$$

Commutativité

$$a + b = b + a$$

$$a.b = b.a$$

Distributivité

$$a + (b.c) = (a + b).(a + c)$$

$$a.(b + c) = (a.b) + (a.c)$$

Lois de Boole

Neutre

$$a + 0 = a$$

$$a.1 = a$$

Absorption

$$0.a = 0$$

$$1 + a = 1$$

Complémentarité

$$\neg(\neg(a)) = a$$

Simplification

$$a + \neg(a).b = a + b$$

$$a.(\neg(a) + b) = a.b$$

$$a.b + \neg(a).c + b.c = a.b + \neg(a).c$$

Lois de De Morgan

La négation de la somme est égale au produit des négations

$$\neg(a + b) = \neg(a) \cdot \neg(b)$$

La négation du produit est égale à la somme des négations

$$\neg(a \cdot b) = \neg(a) + \neg(b)$$

Encoder en binaire

Les bases

Représentation

Toute valeur numérique peut être exprimée dans différentes bases

Nom	Base	Set	Exemple
Binaire	2	0, 1	1010 0011
Octal	8	0, 1, 2, 3, 4, 5, 6, 7	243
Décimal	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	63
Hexadécimal	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F	A3

!

Ne pas confondre coder (développer) et le codage (représentation de l'information).

Du binaire au décimal

Méthode

Chaque bit a un poids associé

Lecture de droite à gauche

Exemple

Valeur	2^5	2^4	2^3	2^2	2^1	2^0
100101	1	0	0	1	0	1
37	32	0	0	4	0	1

Du décimal au binaire

Méthode

On retrouve les puissances de 2 qui vont former les composantes de la valeur

Méthode par division euclidiennes successives

Exemple

$$214 = 128 + 64 + 16 + 4 + 2 = 2^7 + 2^6 + 2^2 + 2^1 = 01000110.$$

Du décimal au binaire

Calcul de la taille

Méthode logarithmique

Exemple

$$S(bits) = \lfloor \frac{\ln(x)}{\ln(2)} \rfloor + 1$$

Représentation hexadécimale

Valeurs

Hexa	Décimal	Binaire
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001

Hexa	Décimal	Binaire
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

Conversion vers binaire

Hexadécimal : Méthode du groupement par 4

Etape	Valeur
Valeur de base	0011001001110101111
Groupement	001 1001 0011 1010 1111
Padding	0001 1001 0011 1010 1111
Conversion	193AF

Octal : Méthode du groupement par 3

Etape	Valeur
Valeur de base	0011001001110101111
Groupement	0 011 001 001 110 101 111
Padding	000 011 001 001 110 101 111
Conversion	0311657

L'octet

Définition

Un octet est un ensemble de 8 bits

1 octet = 1 byte Un octet permet donc de représenter $2^8 = 256$ valeurs

Représentations

On peut représenter un octet en binaire 101100111_b

Ou en hexadecimal : $0xB7$

Multiples de l'octet

Multiples en Système International

Nom	Notation	Valeur
Octet	o	1
Kilooctet	ko	1000
Mégaoctet	Mo	10^6 , soit 10^3 ko
Gigaoctet	Go	10^9 , soit 10^3 Mo
Téraoctet	To	10^{12} , soit 10^3 Go
Pétaoctet	Po	10^{15} , soit 10^3 To
Exaoctet	Eo	10^{18} , soit 10^3 Po
Zettaoctet	Zo	10^{21} , soit 10^3 Eo
Yottaoctet	Yo	10^{24} , soit 10^3 Zo

Préfixes binaires

Puissances de 2

Nom	Notation	Valeur
Octet	o	1
Kibiocet	kio	1024
Mébiocet	Mio	2^{20} , soit 2^{10} kio
Gibiocet	Gio	2^{30} , soit 2^{10} Mio
Tébiocet	Tio	2^{40} , soit 2^{10} Gio
Pébiocet	Pio	2^{50} , soit 2^{10} Tio
Exbiocet	Eio	2^{60} , soit 2^{10} Pio
Zébiocet	Zio	2^{70} , soit 2^{10} Eio
Yobiocet	Yio	2^{80} , soit 2^{10} Zio

Représentation des entiers

Bit de signe

Principe

Afin de représenter les entiers relatifs, on assigne le MSB (bit de poids fort) à 1.

Exemples

Sur 4 bits	-4	1100
Sur 4 bits	-7	1111
Sur 8 bits	-7	1000 0111

Problème

?

Avec la représentation classique, le zéro possède deux représentations
De plus, la représentation binaire directe ne permet pas d'addition simple en binaire

Exemples

Sur 8 bits : $00000000 = 10000000$

$-1 + 1 = 1001 + 0001 \neq 1010$

Le complément à 1

Principe

On inverse tous les bits d'un mot binaire

Exemple

0	0	1	1	0	1
1	1	0	0	1	0

Le complément à 2

Principe

On calcule le complément à 1, puis on rajoute 1

On transforme tous les 1 (à droite) en 0, jusqu'au premier 0 (transformé en 1)

Exemple

Opérateur	0	0	1	1	0	1
CPL1	1	1	0	0	1	0
CPL2	1	1	0	0	1	1

Conversion naturel vers relatif

Conversion

Afin de calculer la valeur binaire d'un entier relatif, on calcule le complément à 2 de son opposé positif.

Exemple

Opérateur	Valeur
Départ	-42
Naturel	42
Binaire	101010
Binaire 8 bits	0010 1010
CPL1	1101 0101
CPL2	1101 0110

Le dépassement

Principe

Le CPL2 permet une détection du dépassement

Le résultat reste correct si on ne tient pas compte du dépassement

Exemple

$$\begin{array}{rcccc|cl}
 . & 0 & 1 & 1 & 0 & 6 \\
 + & 1 & 1 & 1 & 0 & -2 \\
 \hline
 = (1) & 0 & 1 & 0 & 0 & 4
 \end{array}$$

Les types entiers

Représentation en complément à 2 (64 bits)

Type	Taille	Borne inférieure	Borne supérieure
char	8	-2^7	$2^7 - 1$
int	32	-2^{31}	$2^{31} - 1$
long	64	-2^{63}	$2^{63} - 1$
short	16	-2^{15}	$2^{15} - 1$
unsigned char	8	0	$2^8 - 1$
unsigned int	32	0	$2^{32} - 1$
unsigned long	64	0	$2^{64} - 1$
unsigned short	16	0	$2^{16} - 1$

La compilation

Le code source

Définition

Le code d'un programme est l'ensemble des instructions qu'il regroupe.
Le code d'un programme est appelé sa **source**.

La compilation

Définition

La compilation est l'étape consistant à traduire le code source en langage machine.

Principe

On génère un fichier binaire qui sera exécuté par la machine.

La compilation

Contrôle

Le compilateur vérifie la validité syntaxique et grammaticale du code.
Si la syntaxe est incorrecte, il renvoie des erreurs et stoppe la compilation.

Conclusion

- Représenter l'information en binaire
- Types fondamentaux variés
- Encodages différents pour les valeurs négatives
- Bases de représentation