

Technologie des ordinateurs et des réseaux

Pierre-Alain Goupille

Ancien professeur d'informatique
en BTS informatique de gestion à Angers

9^e édition

DUNOD

Toutes les marques citées dans cet ouvrage
sont des marques déposées par leurs propriétaires respectifs.

Cet ouvrage est une nouvelle présentation de la neuvième édition
parue initialement dans la collection Sciences Sup en 2010.

Illustration de couverture :
©madpixblue - fotolia.com

© Dunod, 1996, 1998, 2000, 2004, 2008,
2010, 2015, 2020 pour la nouvelle présentation

© Masson, 1990, 1992, 1994

11 rue Paul Bert, 92240 Malakoff
www.dunod.com

ISBN 978-2-10-082162-4

Le Code de la propriété intellectuelle n'autorisant, aux termes de l'article L. 122-5, 2° et 3° a), d'une part, que les « copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective » et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, « toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause est illicite » (art. L. 122-4).

Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contrefaçon sanctionnée par les articles L. 335-2 et suivants du Code de la propriété intellectuelle.

Table des matières

AVANT-PROPOS	XI
CHAPITRE 1 • INTRODUCTION À LA TECHNOLOGIE DES ORDINATEURS	1
1.1 Définition de l'informatique	1
1.2 Première approche du système informatique	1
CHAPITRE 2 • NUMÉRATION BINAIRE ET HEXADÉCIMALE	5
2.1 Pourquoi une logique binaire ?	5
2.2 Le langage binaire (binaire pur)	7
2.3 Le langage hexadécimal	12
CHAPITRE 3 • REPRÉSENTATION DES NOMBRES	19
3.1 Notion de mot	19
3.2 Nombres en virgule fixe	20
3.3 Nombres en virgule flottante	25
CHAPITRE 4 • REPRÉSENTATION DES DONNÉES	31
4.1 Le code ASCII	31
4.2 Codes ISO 8859	34
4.3 Le code EBCDIC	37
4.4 Unicode	38
4.5 Code base64 utilisé par le protocole MIME	42
CHAPITRE 5 • PROTECTION CONTRE LES ERREURS, ENCODAGES ET CODES	45
5.1 Le contrôle de parité	45
5.2 Les codes autovérificateurs ou autocorrecteurs	46
5.3 Étude de quelques encodages et codes courants	51

CHAPITRE 6 • CONCEPTION DES CIRCUITS INTÉGRÉS	59
6.1 Un peu d'histoire	59
6.2 Technique de conception des circuits intégrés	61
CHAPITRE 7 • UNITÉ CENTRALE DE TRAITEMENT	65
7.1 Approche des blocs fonctionnels	65
7.2 Étude de l'unité centrale de traitement	67
7.3 Fonctionnement de l'unité centrale	73
CHAPITRE 8 • MODES D'ADRESSAGE ET INTERRUPTIONS	81
8.1 Les modes d'adressage	81
8.2 Adressage du 8086	84
8.3 Interruptions	86
8.4 Application aux microprocesseurs Intel i86	88
CHAPITRE 9 • LES BUS	97
9.1 Généralités	97
9.2 Bus ISA ou PC-AT	100
9.3 Bus MCA	100
9.4 Bus EISA	100
9.5 Bus VESA Local Bus	101
9.6 Bus PCI	101
9.7 Bus PCI-X	102
9.8 PCI Express	102
9.9 InfiniBand	103
9.10 HyperTransport	104
9.11 Bus SCSI	105
9.12 SAS – SERIAL SCSI	107
9.13 Bus FireWire – IEEE 1394	108
9.14 Bus USB	110
9.15 Bus graphique AGP	112
9.16 Fibre Channel	113
CHAPITRE 10 • MICROPROCESSEURS	115
10.1 Le microprocesseur Intel 8086	115
10.2 Notions d'horloge, pipeline, superscalaire...	119
10.3 Le rôle des chipsets	123
10.4 Les sockets, socles ou slots	125
10.5 Étude détaillée d'un processeur : le Pentium Pro	126
10.6 Évolution de la gamme Intel	132
10.7 Les coprocesseurs	135
10.8 Les microprocesseurs RISC	136
10.9 Les autres constructeurs	137
10.10 Le multiprocessing	137
10.11 Mesure des performances	139

CHAPITRE 11 • MÉMOIRES – GÉNÉRALITÉS ET MÉMOIRE CENTRALE	143
11.1 Généralités sur les mémoires	143
11.2 Les mémoires vives	147
11.3 Les mémoires mortes	159
11.4 La mémoire Flash	162
CHAPITRE 12 • BANDES ET CARTOUCHES MAGNÉTIQUES	167
12.1 La bande magnétique	167
12.2 Les cartouches magnétiques	172
CHAPITRE 13 • DISQUES DURS ET CONTRÔLEURS	177
13.1 Principes technologiques	177
13.2 Les modes d'enregistrements	182
13.3 Les formats d'enregistrements	183
13.4 Capacités de stockage	183
13.5 Principales caractéristiques des disques	186
13.6 Les contrôleurs ou interfaces disques	188
13.7 Disques durs externes, Extractibles, amovibles	191
13.8 L'avenir des disques dur	192
13.9 Les technologies RAID	193
CHAPITRE 14 • DISQUES OPTIQUES	199
14.1 Le disque optique numérique	199
CHAPITRE 15 • DISQUETTES, DISQUES AMOVIBLES ET CARTES PCMCIA	213
15.1 Les disquettes	213
15.2 Les disques amovibles	218
15.3 La carte PCMCIA	219
CHAPITRE 16 • SYSTÈMES DE GESTION DE FICHIERS	221
16.1 Préambules	222
16.2 Étude du système FAT	225
16.3 Partitions LINUX	240
16.4 Le système NTFS	241
CHAPITRE 17 • GESTION DE L'ESPACE MÉMOIRE	251
17.1 Généralités	251
17.2 Gestion de la mémoire centrale	251
17.3 Gestion mémoire sous MS-DOS	256
17.4 Gestion mémoire avec Windows 9x, NT, 2000...	259
CHAPITRE 18 • IMPRIMANTES	263
18.1 Généralités	263
18.2 Les diverses technologies d'imprimantes	264
18.3 Interfaçage des imprimantes	271
18.4 Critères de choix	273

CHAPITRE 19 • PÉRIPHÉRIQUES D’AFFICHAGE	277
19.1 Les écrans classiques	277
19.2 Les écrans plats	285
CHAPITRE 20 • PÉRIPHÉRIQUES DE SAISIE	295
20.1 Les claviers	295
20.2 Les souris	301
20.3 Les tablettes graphiques	303
20.4 Les scanners	304
CHAPITRE 21 • TÉLÉINFORMATIQUE – THÉORIE DES TRANSMISSIONS	309
21.1 Les principes de transmission des informations	309
21.2 Les différentes méthodes de transmission	314
21.3 Les modes de transmission des signaux	319
21.4 Les techniques de multiplexage	321
21.5 Les différents types de relations	322
CHAPITRE 22 • TÉLÉINFORMATIQUE – STRUCTURES DES RÉSEAUX	327
22.1 Les configurations de réseaux	327
22.2 Techniques de commutation	331
22.3 Éléments constitutifs des réseaux	333
22.4 Les modèles architecturaux (OSI, DOD...)	360
CHAPITRE 23 • PROCÉDURES ET PROTOCOLES	369
23.1 Procédures	369
23.2 Protocoles	370
CHAPITRE 24 • PROTOCOLE TCP/IP	393
24.1 Pile de protocoles TCP/IP	393
24.2 IPSec	426
CHAPITRE 25 • RÉSEAUX LOCAUX – GÉNÉRALITÉS ET STANDARDS	431
25.1 Terminologie	431
25.2 Modes de transmission et modes d'accès	433
25.3 Ethernet	435
25.4 Token-Ring	447
CHAPITRE 26 • INTERCONNEXION DE POSTES ET DE RÉSEAUX – VLAN & VPN	449
26.1 Introduction	449
26.2 Les fonctions de l'interconnexion	451
26.3 Les dispositifs d'interconnexion	452
26.4 Les VLAN	470
26.5 Réseaux Privés Virtuels – VPN	474
26.6 MPLS – Multi Protocol Label Switching	478
26.7 Administration de réseaux	479

CHAPITRE 27 • NAS OU SAN – STOCKAGE EN RÉSEAU OU RÉSEAU DE STOCKAGE	489
27.1 Généralités	489
CHAPITRE 28 • L'OFFRE RÉSEAUX ÉTENDUS EN FRANCE	499
28.1 L'Offre Orange Business SERVICES	499
28.2 Opérateurs alternatifs	512
28.3 Réseaux radio, transmission sans fil	514
28.4 Internet	515
BIBLIOGRAPHIE ET WEBOGRAPHIE	519
INDEX	521

Avant-propos

Ce livre, directement inspiré de cours dispensés depuis plus de 20 ans en BTS Informatique de Gestion, est maintenant devenu une référence avec plus de 30 000 exemplaires vendus. Cette huitième édition actualise dans la mesure du possible, les données techniques des matériels décrits et prend en compte l'évolution des technologies. La plupart des chapitres ont été complétés ou remaniés afin d'améliorer la démarche pédagogique ou pour tenir compte de l'évolution fulgurante des technologies.

L'ouvrage est conçu autour de quatre grands pôles :

- les chapitres 1 à 5 traitent de la manière dont sont représentées les informations dans un système informatique ;
- les chapitres 6 à 11 présentent la manière dont travaille la partie proprement « active » de la machine (unité centrale, mémoire...) ;
- les chapitres 12 à 20 présentent une assez large gamme des composants et périphériques rattachés au système (disques, imprimantes, écrans...) ;
- enfin, les chapitres 21 à 28 traitent des réseaux locaux ou de transport.

Dans la plupart des cas, le chapitre est suivi d'exercices corrigés de manière à vérifier si les notions abordées sont bien acquises. N'hésitez pas à consulter d'autres ouvrages ou les très nombreux sites Internet (en recoupant toutefois les informations car certains sites offrent des informations parfois « farfelues » quand elles ne sont pas carrément fausses). Si vous avez des doutes ou des questions non résolues, n'oubliez jamais que le professeur est là pour vous aider à répondre à vos interrogations.

L'informatique étant une matière en perpétuelle et très rapide évolution, il se peut que des données techniques, à jour lors de la rédaction de ce livre, soient déjà dépass-

sées lors de son impression et/ou de sa lecture. Il est également possible que de nouvelles technologies aient vu le jour ou que d'autres aient disparues. C'est pourquoi les valeurs fournies sont essentiellement « indicatives » et peuvent bien entendu varier selon les constructeurs, les évolutions technologiques ou divers autres critères d'analyse... Rien de tel que la lecture de revues, ou la consultation des sites Internet, pour rester informé !

Ce livre pourra être utilisé avec profit par les élèves et les professeurs des classes de terminales, de BTS ou d'IUT des sections Informatique ou Comptabilité, en licence d'Informatique, en MIAGE... pour assurer ou compléter leurs cours et, d'une manière plus générale, par tous ceux qui sont curieux de comprendre « comment ça marche ».

Merci à tous ceux qui m'ont aidé à corriger et compléter cette édition et bonne lecture à tous.

Chapitre 1

Introduction à la technologie des ordinateurs

1.1 DÉFINITION DE L'INFORMATIQUE

L'informatique est la « Science du traitement rationnel et automatique de l'information ; l'ensemble des applications de cette science ». Les mots employés dans cette définition de l'informatique par l'Académie française ne sont pas anodins.

Tout d'abord, l'informatique est une **science**, c'est-à-dire qu'elle obéit à des règles et des lois précises, et les traitements qu'elle réalise se font de manière rationnelle. **Le traitement automatique** mentionné par la définition fait référence aux **ordinateurs** dont nous étudierons la technologie dans les chapitres suivants. **L'information** est la matière traitée par les ordinateurs. Ils doivent donc être capables de la « comprendre », tout comme l'homme comprend une langue et des concepts.

1.2 PREMIÈRE APPROCHE DU SYSTÈME INFORMATIQUE

Commençons par présenter rapidement ce qu'est un ordinateur et établissons un parallèle avec la façon dont travaille, par exemple, un employé de bureau. Pour travailler, l'employé qui est au centre des trai-



tements, et que l'on peut donc considérer comme **l'unité centrale de traitement** s'installe à son bureau où on lui remet dans un dossier le travail à faire (**informations en entrée**), tandis que dans un autre dossier il devra mettre le travail fait (**informations en sortie**).

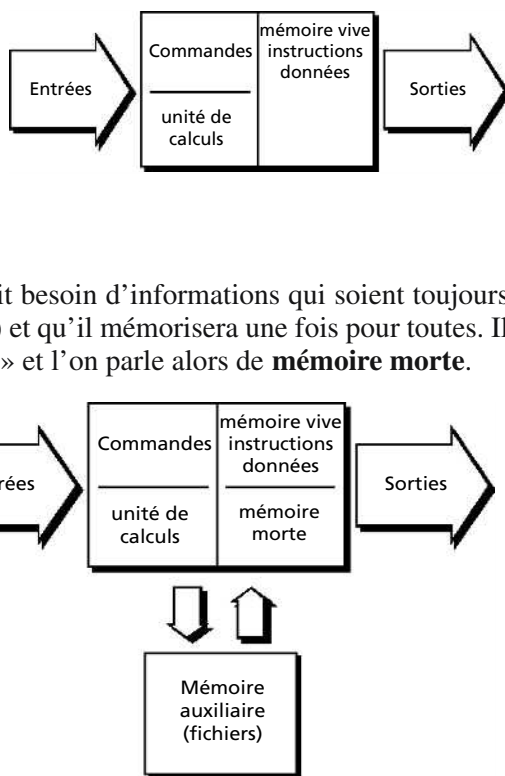
Nous avons là un **système de traitement** qui reçoit des informations en entrée, exécute un traitement sur ces informations et restitue des informations en sortie.

Pour exécuter son travail, l'employé peut avoir besoin de réaliser des calculs (opérations mathématiques, additions, comparaisons...), il dispose pour cela d'une calculatrice (ou **unité de calcul**). Pour ne pas oublier ce qu'on lui demande, il va noter sur un brouillon les instructions qu'il a reçues et qui constituent son **programme** de travail ; de même il va certainement être amené à noter quelques informations sur les données qu'il va traiter.

Il mémorise donc des **instructions** à exécuter et des **données** à traiter. Ces informations peuvent être actualisées et on peut ainsi considérer que l'on se trouve en présence d'une mémoire qui « vit » (où l'information naît, vit et meurt). On parle alors de **mémoire vive**.

De plus, il se peut que pour réaliser la tâche demandée, l'employé ait besoin d'informations qui soient toujours les mêmes (le taux de TVA par exemple) et qu'il mémorisera une fois pour toutes. Il s'agit là d'une mémoire qui ne « vit pas » et l'on parle alors de **mémoire morte**.

Certaines informations (telles que le catalogue des prix, les adresses des clients...) sont trop volumineuses pour être mémorisées « de tête », et l'employé aura alors à sa disposition des classeurs contenant des **fichiers** clients, tarifs... qui constituent une **mémorisation** externe ou **auxiliaire** de certaines données. Dans un ordinateur nous retrouvons également tous ces constituants.



L'ordinateur se présente le plus souvent dans la vie courante sous l'aspect du micro-ordinateur. Faisons preuve de curiosité et « levons le capot ! ». Nous pouvons d'ores et déjà distinguer un certain nombre d'éléments. Le plus volumineux est souvent le bloc d'alimentation électrique, accompagné d'un ventilateur de refroidissement.

Partant de ce bloc, un certain nombre de fils électriques alimentent les divers composants parmi lesquels on distingue un certain nombre de **cartes** ou **contrôleurs**, recouvertes de puces électroniques et enfichées sur une carte plus grande, dite « **carte mère** » située au fond de la machine.

Cette carte mère supporte l'unité de traitement, plus communément désignée sous le terme de **microprocesseur** ou **puce**. On utilise parfois le terme de « fonds de panier », pour désigner la série de connecteurs permettant d'enficher les diverses cartes. La carte mère présente divers connecteurs de formes et de couleurs différents. L'un d'eux porte sous forme de barrette(s) la mémoire centrale de l'ordinateur (**mémoire vive**).

Certaines cartes (contrôleurs) disposent de nappes de câbles qui renvoient vers des modules permettant de loger de manière interne un certain nombre d'unités de stockage (**mémoire auxiliaire**) : lecteurs de disquettes, disque dur, streamer, CD-ROM... Il est également possible de rencontrer ces divers modules à l'extérieur du boîtier principal et vous pouvez déjà constater que l'on relie à ce boîtier l'écran de visualisation (ou moniteur vidéo) ainsi que le clavier, qui permet d'entrer les informations nécessaires au traitement, ou encore la souris.

Dans les chapitres de cet ouvrage nous étudierons, plus précisément, ce que recouvrent tous ces termes, la manière dont sont constitués techniquement les ordinateurs et quels sont les divers périphériques qui leur sont associés (imprimantes, écrans...), mais il convient tout d'abord d'observer sous quelle forme doivent se présenter les informations dans la machine afin qu'elles puissent être assimilées par le système informatique.

EXERCICES

1.1 Reconstituez, sans vous aider du livre, le schéma synoptique d'un système informatique simplifié tel que nous l'avons présenté précédemment.

1.2 Nommez les divers composants ci-après.



Chapitre 2

Numération binaire et hexadécimale

2.1 POURQUOI UNE LOGIQUE BINAIRE ?

Le passage d'une information, d'un langage compréhensible par l'homme à un langage compréhensible par le système informatique, s'appelle **codage** ou **codification**. Nous verrons qu'il existe de nombreuses possibilités de codage de l'information, en BINAIRE, en HEXADÉCIMAL, en BCD, en ASCII... Mais étudions d'abord les éléments de base du langage binaire qui est le fondement même de la logique informatique telle qu'elle existe à l'heure actuelle. Il est en effet possible que d'ici quelques décennies la logique binaire soit remplacée par une logique « multi-états » bien que la logique binaire ait encore de beaux jours devant elle.

Les systèmes informatiques actuels étant construits à l'aide de circuits intégrés – composants électroniques qui rassemblent pour certains des dizaines voire des centaines de millions de transistors, ne fonctionnent actuellement que selon une **logique à deux états** telle que, de façon schématique, le courant passe ou ne passe pas dans le transistor. Ces deux états logiques, conventionnellement notés **0** et **1**, déterminent cette logique **binaire** correspondant (de manière un peu « réductrice ») à deux niveaux électriques.

Toute information à traiter devra donc être représentée sous une forme assimilable par la machine, c'est-à-dire sous une forme **binaire**, que ce soit en interne dans la machine, ainsi que nous venons de le signaler, mais également sur les « fils » permettant de faire circuler l'information entre les composants de l'ordinateur.

Dès lors qu'ils empruntent des voies filaires, les signaux binaires imposent classiquement l'usage de deux fils (l'un véhiculant le signal et l'autre servant de masse).

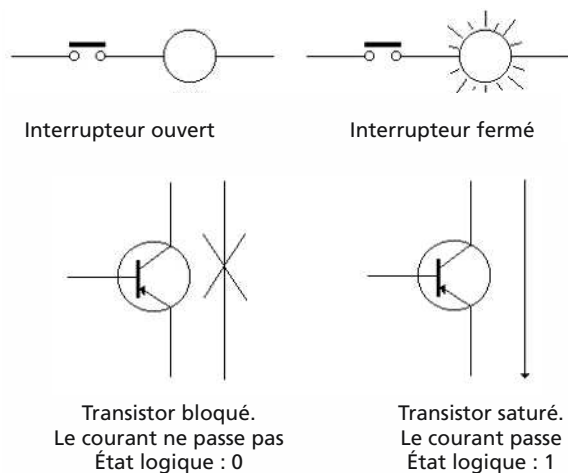


Figure 2.1 États binaires

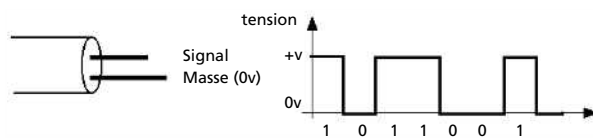


Figure 2.2 Transmission classique

Cette technique, dite **NRZ** (*No Return to Zero*) ou **bus asymétrique – SE** (*Single Ended*), bien que toujours très utilisée, présente cependant des inconvénients liés aux phénomènes électromagnétiques qui font que les signaux s'affaiblissent rapidement et deviennent illisibles, notamment lors de la transmission de composantes continues telles que de longues suites de 0 ou de 1. La méthode NRZ est donc peu à peu remplacée par une autre technique dite **mode différentiel, bus différentiel** ou **symétrique – LVD** (*Low Voltage Differential*), où chaque fil de la paire véhicule le signal et son inverse.

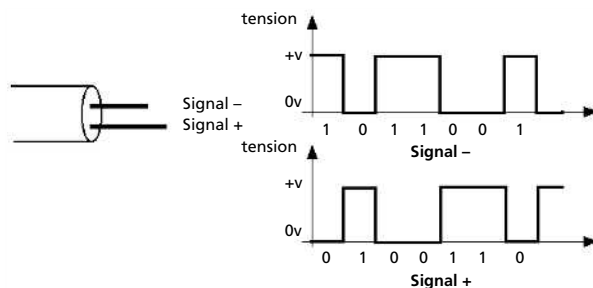


Figure 2.3 Transmission symétrique

2.2 LE LANGAGE BINAIRE (BINAIRE PUR)

En langage binaire, ou plus simplement « en binaire », on dispose d'un alphabet dont les **symboles** sont le **0** et le **1** qui, combinés, doivent permettre de définir toute information à traiter. La base de numération employée est ici de 2 (car on utilise 2 symboles) et les calculs se font alors en **base 2**. Un nombre en base 2 est conventionnellement noté n_2 .

On notera ainsi 101_2

Un nombre en base 10, tel que nous l'employons d'habitude, devrait donc être noté sous la forme n_{10} , par exemple, 135_{10} , mais généralement une telle notation est omise. Il convient cependant d'y prêter attention dans certains cas (c'est ainsi que 101_2 et 101_{10} ne correspondent pas du tout à la même valeur dans une base commune). Le **binaire pur** est à distinguer d'une autre forme de numération binaire dite **binaire réfléchi** qui consiste à faire progresser les valeurs binaires en ne faisant varier qu'un seul bit d'une valeur binaire à la valeur binaire suivante. Toutefois cette technique, employée en automatisme, ne nous concerne pas.

2.2.1 Conversions

Afin de simplifier la démarche d'apprentissage, nous ne traiterons dans un premier temps que le cas des nombres entiers positifs.

On passe d'un nombre en base 10 à un nombre en base 2 par divisions successives.

Soit 135_{10} à convertir en base 2.

135	/	2	=	67	reste	1	↑
67	/	2	=	33	reste	1	↑
33	/	2	=	16	reste	1	↑
16	/	2	=	8	reste	0	↑
8	/	2	=	4	reste	0	↑
4	/	2	=	2	reste	0	↑
2	/	2	=	1	reste	0	↑
1	/	2	=	0	reste	1	↑

Les flèches indiquent le sens de lecture. Le nombre 135_{10} équivaudra ainsi à 10000111_2 en binaire pur. Chaque élément binaire, pouvant prendre une valeur 0 ou 1, est appelé un **digit binaire** ou, plus couramment **bit** (abréviation de l'anglais **binary digit**). Une suite de quatre bits prendra le nom de **quartet** (ou **nibble**), une suite de huit bits le nom d'**octet**. Attention : en anglais **octet** se traduit par **byte** et il faut absolument éviter la confusion entre bit et **byte** !

0	1	1	0	0	0	0	1
---	---	---	---	---	---	---	---

Figure 2.4 Un octet

Le passage d'un nombre en base 2 à un nombre en base 10 peut se faire par multiplications successives.

Il suffit en effet de multiplier chaque élément du nombre binaire (ou **bit**) par le chiffre 2 élevé à une puissance, croissant par pas de 1, comptée à partir de zéro en partant de la droite, puis d'effectuer la somme des résultats obtenus.

Soit 10011_2 à convertir en décimal.

1	0	0	1	1	
×	×	×	×	×	
2^4	2^3	2^2	2^1	2^0	
(16)	(8)	(4)	(2)	(1)	
=	=	=	=	=	
16	0	0	2	1	dont la somme donne 19_{10}

On constate, à l'examen de cette méthode, que l'on multiplie chacun des bits examinés, par les valeurs successives (en partant de la droite du nombre binaire) 1, 2, 4, 8, 16, 32, 64, 128... qui sont en fait déterminées par le poids binaire du digit.

On peut alors en déduire une autre technique de conversion des nombres décimaux en binaire, qui consiste à retrancher du nombre initial la plus grande puissance de 2 possible, et ainsi de suite dans l'ordre décroissant des puissances. Si on peut retirer la puissance de 2 concernée on note 1 sinon on note 0 et on continue de la sorte jusqu'à la plus petite puissance de 2 possible soit 2^0 pour des entiers.

Reprenons le cas de notre premier nombre 135_{10}

de 135	on peut	(1)	retirer	128	reste	7
7	on ne peut pas	(0)	retirer	64	reste	7
7	on ne peut pas	(0)	retirer	32	reste	7
7	on ne peut pas	(0)	retirer	16	reste	7
7	on ne peut pas	(0)	retirer	8	reste	7
7	on peut	(1)	retirer	4	reste	3
3	on peut	(1)	retirer	2	reste	1
1	on peut	(1)	retirer	1	reste	0

Attention : on lit les valeurs binaires, de haut en bas ! Mais le résultat est encore (bien entendu !) 10000111_2 .

Une variante – rapide et pratique en termes de calcul mental... à utiliser un jour d'examen par exemple – consiste à procéder mentalement par addition successive des puissances de 2 décroissantes trouvées et à noter 1 si la somme est inférieure ou égale au résultat recherché et 0 dans le cas contraire.

Ainsi, en reprenant notre valeur 135, la plus grande puissance de 2 que l'on peut retenir est 128 (2^7) on note donc 1. Si on y ajoute la puissance de 2 suivante soit 64 (2^6) le total va donner 192 donc « trop grand » on note 0 (pour l'instant nous avons noté 1 et 0). La puissance de 2 suivante est 32 soit $128 + 32 = 160$ donc « trop grand » on note 0 (soit 100), la puissance de 2 suivante est 16 soit $128 + 16 = 144$ donc « trop grand », on note 0 (soit 1000). La puissance de 2 suivante est 8 soit $128 + 8 = 136$ donc « trop grand ». On note encore 0 (soit 10000). La puissance de 2 suivante est 4 soit $128 + 4 = 132$ donc « ça loge ». On note 1 (soit 100001). La puissance de 2 suivante est 2 soit $132 + 2 = 134$ donc « ça loge » et on note encore 1 (soit 1000011). Enfin la puissance de 2 suivante est 1 soit $134 + 1 = 135$ donc « ça

loge » on note alors 1 ce qui donne pour l'ensemble 10000111_2 ce qui correspond bien à ce que nous devons trouver !

Cette technique impose de connaître les valeurs décimales associées aux puissances de 2 ! Mais qui ne s'est pas amusé à ce petit « jeu »... 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048... ou « à l'envers »... 2048, 1024, 512, 256, 128...

TABLEAU 2.1 LES 10 PREMIERS NOMBRES BINAIRES

Base 10	Base 2	Base 10	Base 2
1	1	6	110
2	10	7	111
3	11	8	1000
4	100	9	1001
5	101	10	1010

a) Notion de poids binaire

Le **poids binaire** ou « poids du bit » correspond à la puissance à laquelle est élevé le bit lors de la conversion binaire-décimal. Ainsi, les bits situés à droite du nombre sont les bits de **poids faible** car leur puissance évolue à partir de la valeur 0 (2^0) pour aller vers 2^1 , 2^2 , etc. Les bits situés à gauche du nombre sont les bits de **poids fort** car leur puissance est la plus élevée et va en décroissant pour se rapprocher des bits de poids faible. Ainsi, sur un octet, le bit situé le plus à gauche est le bit de poids fort (2^7) le plus significatif ou **MSB** (*Most Significant Bit*) alors que le bit situé le plus à droite est le bit de poids faible (2^0) ou **LSB** (*Less Significant Bit*). À partir de quand les poids forts deviennent-ils des poids faibles et inversement... Tout est affaire de relativité !

2.2.2 Opérations binaires

Les opérations sur les nombres binaires s'effectuent de la même façon que sur les nombres décimaux. Toutefois, il ne faut pas oublier que les seuls symboles utilisés sont le **1** et le **0**, nous aurons ainsi les opérations fondamentales suivantes.

a) Addition

$$\begin{array}{rclcl}
 0 & + & 0 & = & 0 \\
 0 & + & 1 & = & 1 \\
 1 & + & 0 & = & 1 \\
 1 & + & 1 & = & 0 \quad \text{et « on retient 1 »} \quad \text{ou encore} \quad \begin{array}{r} 1 \\ + 1 \\ \hline \end{array} \\
 & & & & \text{retenue} \rightarrow 1 \quad 0
 \end{array}$$

b) Soustraction

$$\begin{array}{rclcl}
 0 & - & 0 & = & 0 \\
 0 & - & 1 & = & 1 \quad \text{et « on retient 1 »} \quad \text{OU ENCORE} \quad \begin{array}{r} 0 \\ - 1 \\ \hline \end{array} \\
 & & & & \text{RETENUE} \rightarrow 1 \quad 1 \\
 1 & - & 1 & = & 0 \\
 1 & - & 0 & = & 1
 \end{array}$$

c) *Multiplication*

Les multiplications binaires s'effectuent selon le principe des multiplications décimales. On multiplie donc le multiplicande par chacun des bits du multiplicateur. On décale les résultats intermédiaires obtenus et on effectue ensuite l'addition de ces résultats partiels.

$ \begin{array}{r} 1\ 0\ 0\ 0 \\ \times 1\ 0\ 1\ 0 \\ \hline 0\ 0\ 0\ 0 \\ 1\ 0\ 0\ 0 \\ 0\ 0\ 0\ 0 \\ 1\ 0\ 0\ 0 \\ \hline 1\ 0\ 1\ 0\ 0\ 0\ 0 \end{array} $	<i>Somme intermédiaire →</i> <i>Retenues générées par l'addition de la somme</i> <i>intermédiaire et de la retenue précédentes →</i> <i>Retenues générées par l'addition qui aboutit à</i> <i>la somme intermédiaire →</i>	$ \begin{array}{r} 1\ 0\ 1\ 1 \\ \times 1\ 1\ 1 \\ \hline 1\ 0\ 1\ 1 \\ 1\ 0\ 1\ 1 \\ 1\ 0\ 1\ 1 \\ \hline 1\ 1\ 0\ 0\ 0 \\ 1\ 1 \\ \hline 1\ 1\ 1 \\ \hline 1\ 0\ 0\ 1\ 1\ 0\ 1 \end{array} $
---	--	---

d) *Division*

Nous avons vu que la multiplication était basée sur une succession d'additions. Inversement la division va être basée sur une succession de soustractions et s'emploie de la même façon qu'une division décimale ordinaire.

Soit $1\ 1\ 0\ 0_2$ à diviser par $1\ 0\ 0_2$

$$\begin{array}{r|l}
 1\ 1\ 0\ 0 & 1\ 0\ 0 \\
 - 1\ 0\ 0 & 1 \\
 \hline
 0\ 1\ 0\ 0 & \\
 - 1\ 0\ 0 & 1 \\
 \hline
 0\ 0\ 0 &
 \end{array}$$

Le résultat qui se lit en « descendant » est donc $1\ 1_2$.

Soit $1\ 0\ 1\ 1\ 0\ 0_2$ à diviser par $1\ 0\ 0_2$

$$\begin{array}{r|l}
 1\ 0\ 1\ 1\ 0\ 0 & 1\ 0\ 0 \\
 - 1\ 0\ 0 & 1 \\
 \hline
 0\ 1 & \\
 1\ 1 & 0 \\
 1\ 1\ 0 & \\
 - 1\ 0\ 0 & 1 \\
 \hline
 1\ 0 & \\
 1\ 0\ 0 & \\
 - 1\ 0\ 0 & 1 \\
 \hline
 0 &
 \end{array}$$

Le résultat qui se lit en « descendant » est donc $1\ 0\ 1\ 1_2$.

2.2.3 Cas des nombres fractionnaires

Les nombres utilisés jusqu'à présent étaient des entiers positifs ou négatifs. Bien sûr il est également possible de rencontrer des nombres fractionnaires qu'il conviendra de pouvoir également coder en binaire.

Nous avons vu précédemment que la partie entière d'un nombre se traduisait en mettant en œuvre des puissances positives de 2. Sa partie décimale se traduira donc en fait en mettant en œuvre des puissances **négatives** de 2. Le nombre binaire obtenu se présente sous la forme d'une partie entière située à gauche de la marque décimale, et d'une partie fractionnaire située à droite.

Ainsi

$$(1 \times 2^2) + (0 \times 2^1) + (0 \times 2^0)$$
$$(1 \times 4) + (0 \times 2) + (0 \times 1)$$

$$100$$

$$.01_2$$

sera équivalent à

$$(0 \times 2^{-1}) + (1 \times 2^{-2})$$
$$(0 \times 1/2) + (1 \times 1/4)$$

soit 4.25_{10}

La conversion binaire/décimal ci-dessus se fait donc de manière aisée, il en est de même pour la conversion décimal/binaire. Nous ne reviendrons pas sur la conversion de la partie entière qui se fait par divisions successives par 2 telle que nous l'avons étudiée au début de ce chapitre.

En ce qui concerne la partie fractionnaire, il suffit de la multiplier par 2, la partie entière ainsi obtenue représentant le poids binaire (limité ici aux seules valeurs 1 ou 0). La partie fractionnaire restante est à nouveau multipliée par 2 et ainsi de suite jusqu'à ce qu'il n'y ait plus de partie fractionnaire ou que la précision obtenue soit jugée suffisante.

Soit 0.625_{10}

$0.625 \times 2 = 1.250$
 $0.250 \times 2 = 0.500$
 $0.500 \times 2 = 1.000$

Poids binaire 1×2^{-1}
Poids binaire 0×2^{-2}
Poids binaire 1×2^{-3}

Quand il ne reste plus de partie fractionnaire, on s'arrête.
Ainsi 0.625_{10} devra se traduire par $.101_2$.

À titre utilitaire, vous trouverez (tableau 2.2) un tableau de base des diverses puissances de 2 positives ou négatives.

TABEAU 2.2 LES 10 PREMIÈRES PUISSANCES DE 2 (POSITIVES ET NÉGATIVES)

2^{-n}	n	2^n
1	0	1
0.5	1	2
0.25	2	4
0.125	3	8
0.0625	4	16
0.03125	5	32
0.015625	6	64
0.0078125	7	128
0.00390625	8	256
0.001953125	9	512
0.0009765625	10	1024

© Dunod – La photocopie non autorisée est un délit.

Cette utilisation particulière de la numération binaire des fractionnaires ne présente à peu près qu'un intérêt d'école (quoique dans certains cas cela puisse servir). Dans la majorité des cas, l'informaticien sera amené à manipuler les conversions décimal-binaire ou binaire-décimal ainsi que les additions et soustractions ; les autres opérations étant d'un usage relativement moins courant.

2.3 LE LANGAGE HEXADÉCIMAL

Le langage binaire s'il est compréhensible par la machine (et peut donc être appelé à ce niveau **langage machine**) est difficilement « assimilable » par l'homme dès lors qu'on manipule de grandes séries binaires. On utilise donc un autre système de notation, le système **hexadécimal** – base 16, que nous étudierons ici. Précisons qu'un système **octal** (base 8) a également été utilisé mais est tombé en désuétude. En notation hexadécimale on utilise un **alphabet** comportant **16** symboles. Les chiffres ne pouvant représenter que 10 symboles (0 à 9 inclus), on doit alors utiliser 6 symboles supplémentaires pour compléter notre alphabet. Les symboles retenus ont donc été les 6 premières lettres de l'alphabet.

0 1 2 3 4 5 6 7 8 9 A B C D E F

Figure 2.5 Alphabet hexadécimal

2.3.1 Conversions

Il est possible de passer d'un nombre en base 10 à un nombre en base 16 par divisions successives. On note les restes de ces divisions qu'on lit ensuite en « remontant ».

Convertir en base 16 le nombre 728₁₀

728 ₁₀	:	16	=	45	Reste	8	
45 ₁₀	:	16	=	2	Reste	13	↑
2 ₁₀	:	16	=	0	Reste	2	

Les nombres supérieurs à 9 n'existent pas, en tant que tels, dans la notation hexadécimale où ils doivent être remplacés par des lettres. Ainsi notre « 13 » devra-t-il être remplacé par la lettre D qui est son **équivalent hexadécimal**.

Nombre décimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Équivalent hexadécimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Figure 2.6 Les symboles hexadécimaux

Nous pouvons donc enfin écrire le résultat sous sa forme définitive :
728₁₀ = 2D8₁₆

En informatique, on notera plus souvent la base 16 par la présence d'un 0x (comme heXadécimal) devant le nombre.

Inversement il est aisé de passer d'un nombre en notation hexadécimale à un nombre décimal par multiplications successives, en suivant les principes énoncés dans la conversion binaire. Cependant il ne s'agit plus, cette fois, de puissances de 2 mais de puissances de 16 puisqu'en hexadécimal la base est 16.

$$\begin{array}{r} 1 \times 16^2 + 3 \times 16^1 + 13 \times 16^0 \\ 1 \times 256 + 3 \times 16 + 13 \times 1 \\ 256 + 48 + 13 = 317 \end{array}$$

Si l'on représente les 16 symboles de l'alphabet hexadécimal en binaire on constate que l'on utilise pour chacun d'eux un maximum de 4 bits.

0	0
1	1
2	10
3	11
4	100

5	101
6	110
7	111
8	1000
9	1001

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

Ainsi 732_{10} donne en binaire 1011011100_2

On décompose ce nombre binaire en blocs de 4 bits, à partir de la droite, soit :

	0 0 1 0	1 1 0 1	1 1 0 0
Soit en décimal	2	13	12
Soit en hexadécimal	2	D	C

Le passage de l'hexadécimal en binaire peut donc se faire de la manière inverse, c'est-à-dire en convertissant les chiffres qui composent le nombre hexadécimal en leur équivalent binaire.

Ainsi : 1AB_{16}

1	A	B	donne en binaire
0001	1010	1011	

2.3.3 Représentation des valeurs hexadécimales

La représentation des valeurs sous un format hexadécimal ne se traduit pas visuellement par la présence de l'indice (16) derrière la valeur. On utilise parfois la lettre H ou plus généralement dans les représentations issues des ordinateurs, on fait précéder cette valeur des deux symboles 0x.

Ainsi, $2A03_{16}$, 2A03H ou 0x2A03 sont des représentations valides d'une même valeur hexadécimale. On rencontrera le plus souvent la forme 0x2A03

EXERCICES

2.1 Convertir en binaire les nombres 397_{10} , 133_{10} et 110_{10} puis en décimal les nombres 101_2 , 0101_2 et 1101110_2 et vérifier en convertissant pour revenir à la base d'origine.

2.2 Effectuer les opérations suivantes et vérifier les résultats en procédant aux conversions nécessaires.

a) $1100 + 1000 =$

b) $1001 + 1011 =$

c) $1100 - 1000 =$

d) $1000 - 101 =$

e) $1 + 1 + 1 + 1 =$

2.3 Réaliser les opérations suivantes et vérifier les résultats en procédant aux conversions nécessaires.

a) $1011 \times 11 =$

b) $1100 \times 101 =$

c) $100111 \times 0110 =$

2.4 Réaliser les opérations suivantes et vérifier les résultats en procédant aux conversions nécessaires.

a) $100100 / 11 =$

b) $110000 / 110 =$

2.5 Convertir en binaire 127.75_{10} puis 307.18_{10} . Vous pourrez constater, à la réalisation de ce dernier exercice, que la conversion du .18 peut vous entraîner « assez loin ». C'est tout le problème de ce type de conversion et la longueur accordée à la partie fractionnaire dépendra de la précision souhaitée.

2.6 Convertir en hexadécimal *a)* 3167_{10} *b)* 219_{10} *c)* 6560_{10}

2.7 Convertir en décimal *a)* $0x3AE$ *b)* $0xFF$ *c)* $0x6AF$

2.8 Convertir en base 16 *a)* 128_{10} *b)* 101_{10} *c)* 256_{10}
 d) 110_2 *e)* 1001011_2

2.9 Convertir en base 10 *a)* $0xC20$ *b)* $0xA2E$

2.10 Convertir en base 2 *a)* $0xF0A$ *b)* $0xC01$

SOLUTIONS

2.1 *a)* Convertir en binaire le nombre 397_{10}

$397 : 2 = 198$	reste 1
$198 : 2 = 99$	reste 0
$99 : 2 = 49$	reste 1
$49 : 2 = 24$	reste 1
$24 : 2 = 12$	reste 0
$12 : 2 = 6$	reste 0
$6 : 2 = 3$	reste 0
$3 : 2 = 1$	reste 1
$1 : 2 = 0$	reste 1

Le résultat se lit en « remontant » et nous donne donc 110001101_2 .

Vérification :

1	1	0	0	0	1	1	0	1_2
×	×	×	×	×	×	×	×	×
2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
(256)	(128)	(64)	(32)	(16)	(8)	(4)	(2)	(1)
=	=	=	=	=	=	=	=	=
256	128				8	4	1	→ soit 397 ₁₀

b) Convertir en binaire le nombre $133_{10} = 10000101_2$

c) Convertir en binaire le nombre $110_{10} = 1101110_2$

d) Convertir en décimal le nombre $101_2 = 1 \times 22 + 0 \times 21 + 1 \times 20$

soit $1 \times 4 + 0 \times 2 + 1 \times 1 = 5_{10}$

e) Convertir en décimal le nombre $0101_2 = 5_{10}$. Un zéro devant un nombre n'est pas significatif, que ce soit en décimal ou en binaire.

f) Convertir en décimal le nombre $1101110_2 = 110_{10}$

2.2 Effectuer les opérations.a) addition de $1\ 1\ 0\ 0 + 1\ 0\ 0\ 0$ b) $1\ 0\ 0\ 1 + 1\ 0\ 1\ 1 = 1\ 0\ 1\ 0\ 0$ c) soustraction de $1\ 1\ 0\ 0 - 1\ 0\ 0\ 0$

$$\begin{array}{r}
 1\ 1\ 0\ 0 \\
 - 1\ 0\ 0\ 0 \\
 \hline
 0\ 1\ 0\ 0
 \end{array}$$

d) $1\ 0\ 0\ 0 - 1\ 0\ 1$

$$\begin{array}{r}
 1\ 0\ 0\ 0 \\
 - \quad 1\ 0\ 1 \\
 \hline
 1\ 1\ 0\ 1 \quad \text{« soustraction » intermédiaire} \\
 1\ 1\ 1 \quad \text{retenues intermédiaires} \\
 \hline
 0\ 0\ 1\ 1
 \end{array}$$

e) $1 + 1 + 1 + 1 = 100$ **2.3** Réaliser les opérations.a) $1\ 0\ 1\ 1 \times 1\ 1$

$$\begin{array}{r}
 1\ 0\ 1\ 1 \\
 \times 1\ 1 \\
 \hline
 1\ 0\ 1\ 1 \\
 1\ 0\ 1\ 1 \\
 \hline
 1\ 1\ 1\ 1\ 1\ 1
 \end{array}$$

b) $1\ 1\ 0\ 0 \times 1\ 0\ 1$

$$\begin{array}{r}
 1\ 1\ 0\ 0 \\
 \times 1\ 0\ 1 \\
 \hline
 1\ 1\ 0\ 0 \\
 0\ 0\ 0\ 0 \\
 1\ 1\ 0\ 0 \\
 \hline
 1\ 1\ 1\ 1\ 0\ 0
 \end{array}$$

c) $1\ 0\ 0\ 1\ 1\ 1 \times 0\ 1\ 1\ 0 = 1\ 1\ 1\ 0\ 1\ 0\ 1\ 0$ **2.4** Réaliser les opérations.a) $1\ 0\ 0\ 1\ 0\ 0 / 1\ 1 =$

$$\begin{array}{r|l}
 1\ 0\ 0\ 1\ 0\ 0 & 1\ 1 \\
 - 1\ 1 & 1 \\
 \hline
 0\ 0\ 1\ 1 & \\
 - 1\ 1 & 1 \\
 \hline
 0\ 0\ 0 & 0 \\
 0\ 0\ 0\ 0 & 0
 \end{array}$$

b) $1\ 1\ 0\ 0\ 0\ 0 / 1\ 1\ 0 = 1\ 0\ 0\ 0$