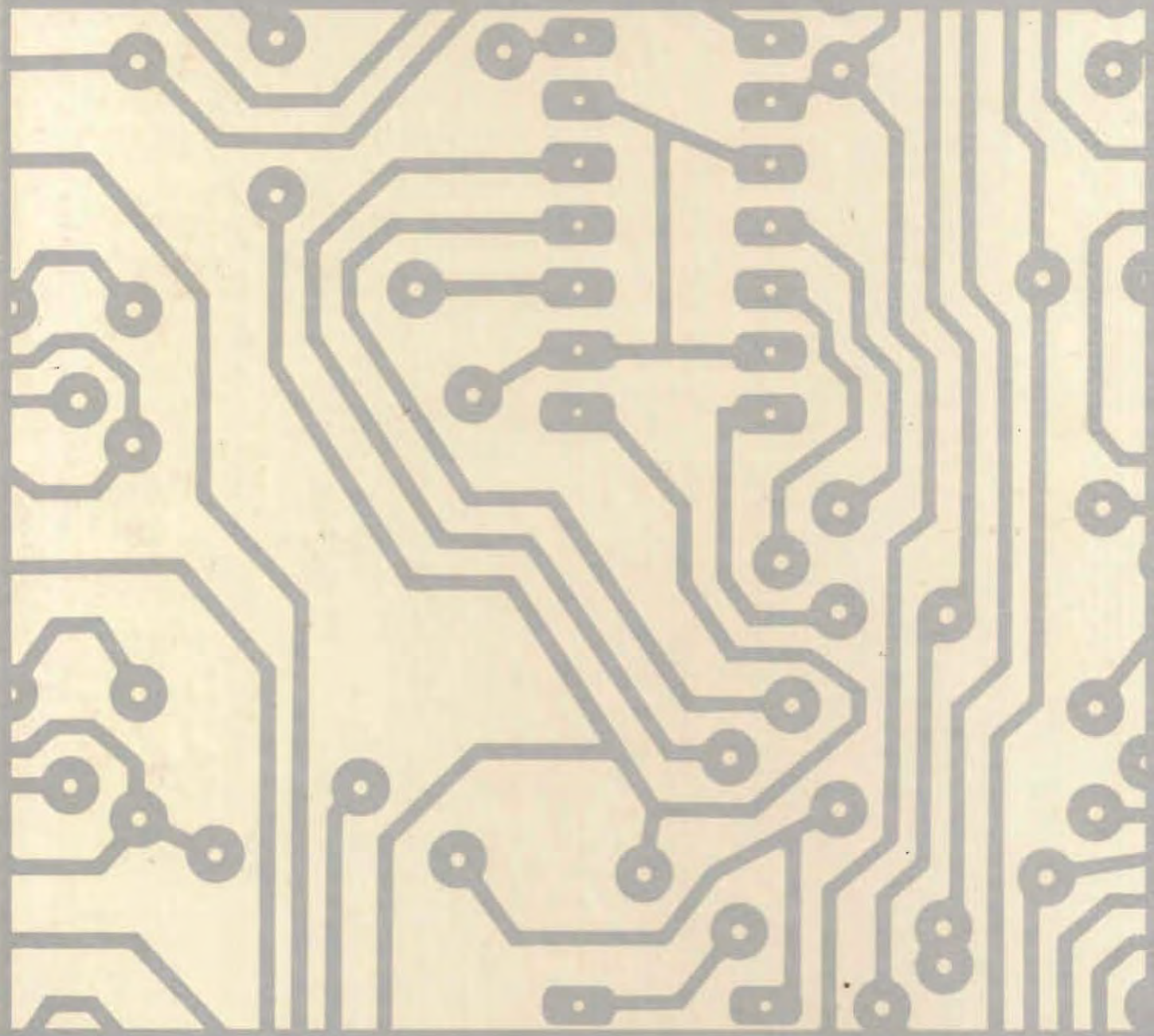
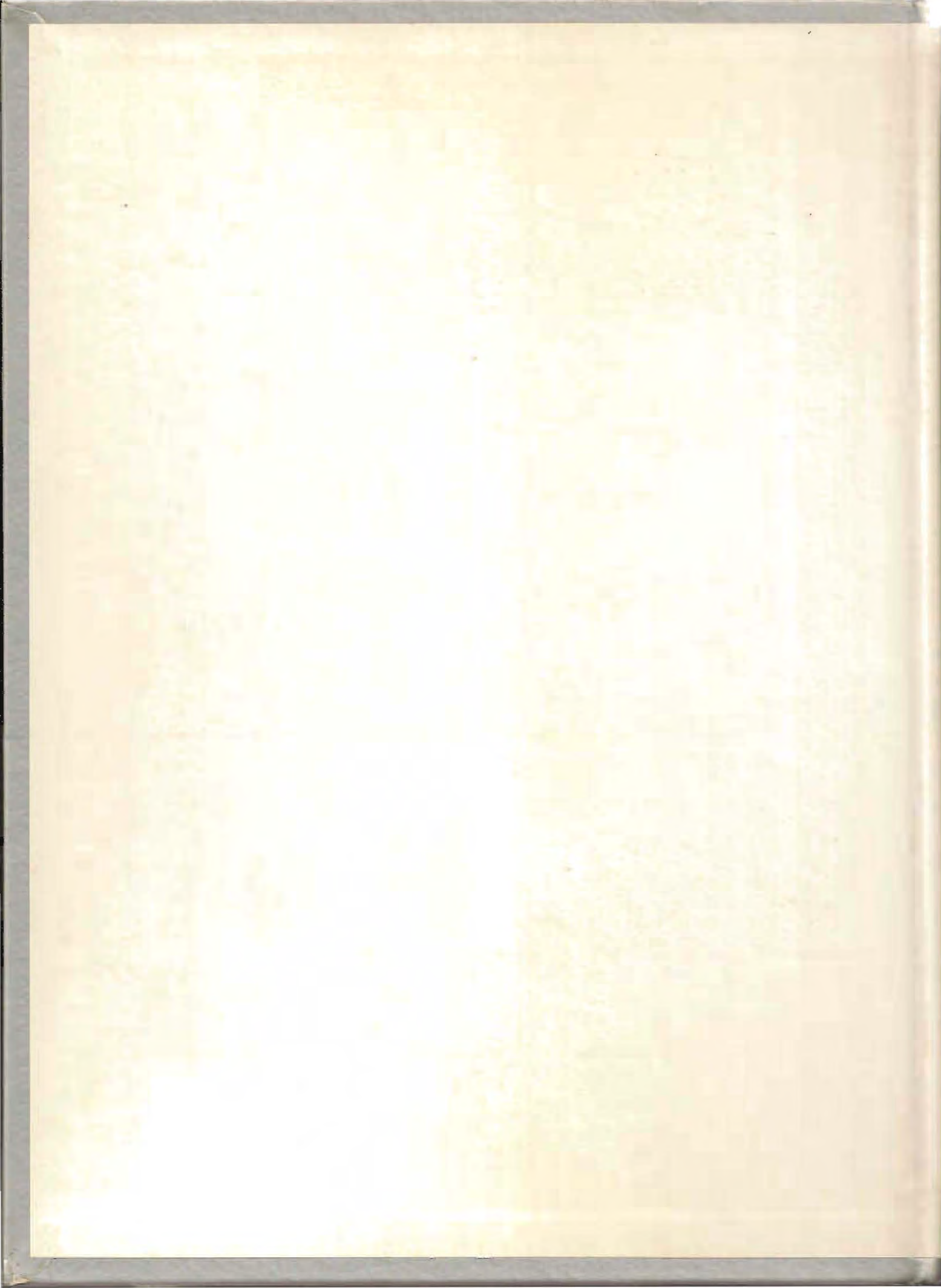


# BASIC PLUS

ENCYCLOPEDIE D'INITIATION  
A L'INFORMATIQUE ET AUX ORDINATEURS



EDITIONS DU HENNIN



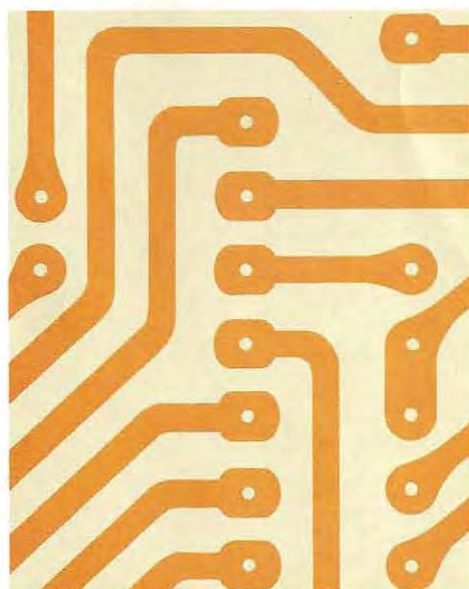
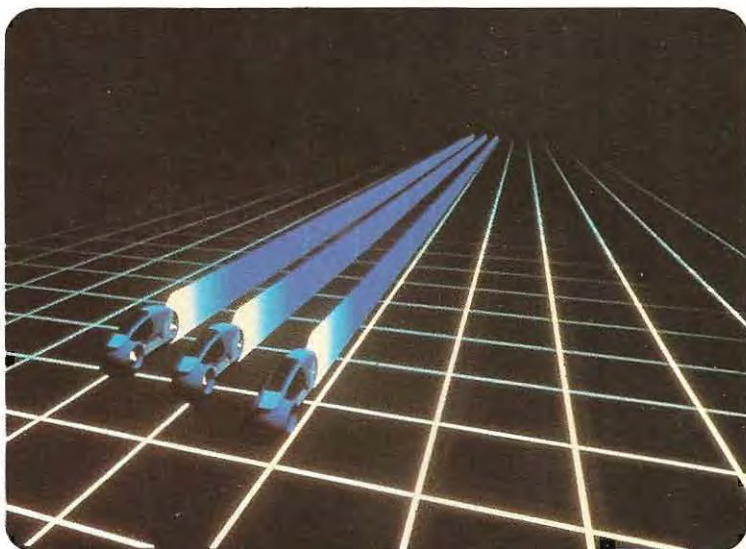
# BASIC PLUS

ENCYCLOPEDIE D'INITIATION A L'INFORMATIQUE ET AUX ORDINATEURS



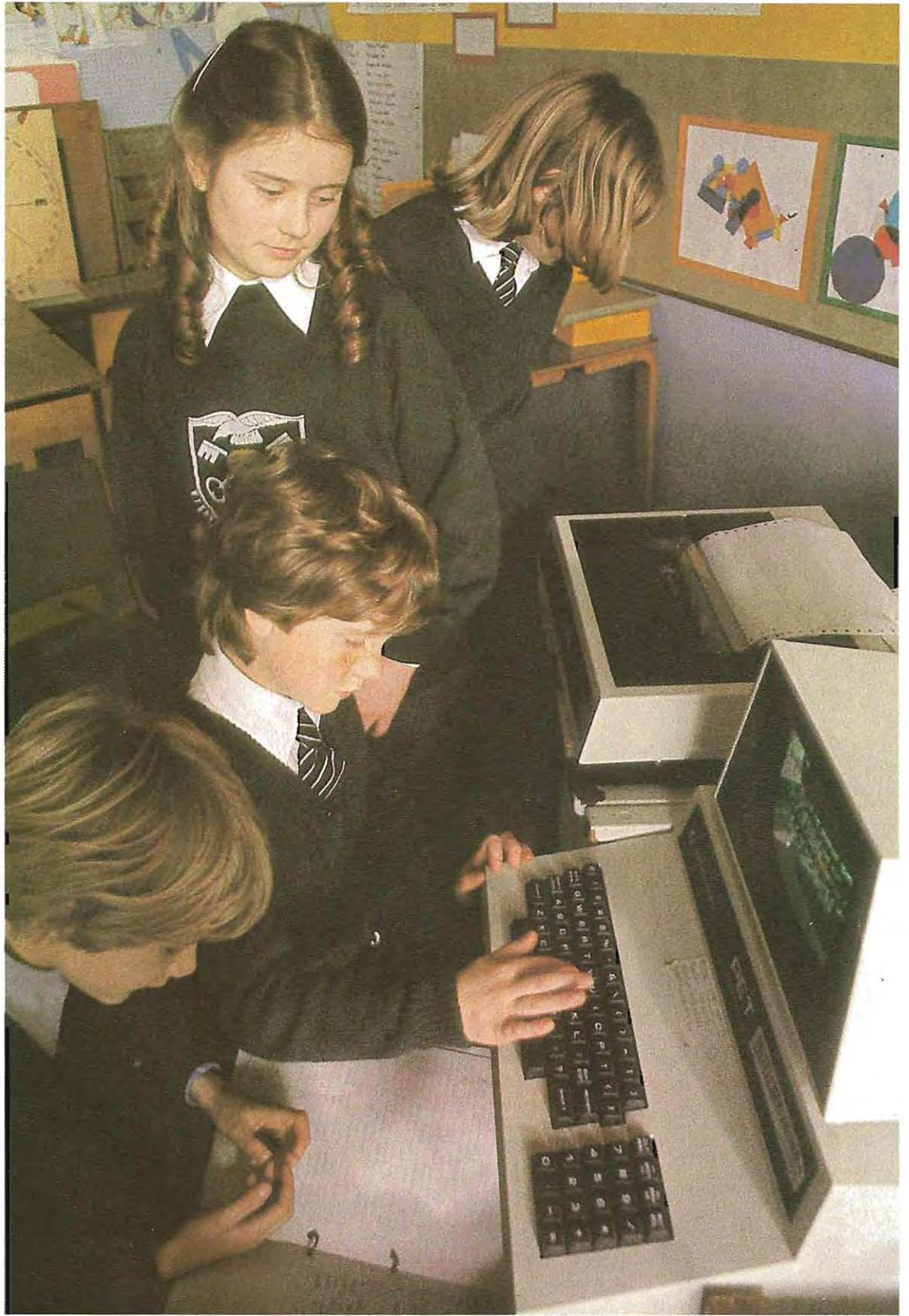
EDITIONS  
DU HENNIN

# ENCYCLOPEDIE D'INITIATION A L'INFORMATIQUE ET AUX ORDINATEURS



# BASIC PLUS





Stuart Franklin-Sygm

## PRESENTATION

Jusqu'à ces dernières années, l'informatique a été le domaine réservé de quelques initiés qui, pour mener à bien des travaux très complexes, avaient besoin d'être secondés par des machines.

Imposantes, tant par leur taille que par l'aspect sérieux des hommes en blouses blanches qui les entouraient, ces machines n'équipaient que les centres d'industrie et de recherche les plus importants.

Aujourd'hui, les ordinateurs font partie de notre vie et sont en train de modifier notre culture. On les retrouve partout, dans les écoles, dans les usines, dans les bureaux et jusque dans les foyers, où ils sont parfois monopolisés par les enfants. Ils dirigent les trains, pilotent les avions et les engins spatiaux, transmettent nos communications téléphoniques. Ils remplissent aussi bien les fonctions de bibliothécaire et d'archiviste que de traducteur. Ils sont présents dans les blocs opératoires des hôpitaux, mais ils savent également jouer aux échecs, et tentent de prévoir la météo.

Nous sommes indiscutablement au seuil d'une ère nouvelle, dans laquelle l'ordinateur se banalisera au point d'être à la portée de tout le monde.

Il est facile de prévoir que d'ici peu un ordinateur familial, doté d'un écran et d'une imprimante ne coûtera pas plus cher qu'un magnétoscope et sera, par conséquent, très largement diffusé.

Avec l'ordinateur domestique nous pouvons, non seulement avoir des activités professionnelles, étudier, organiser des jeux et programmer la vie quotidienne de la famille, mais également aborder les sujets de l'information et de l'enseignement d'une manière tout à fait différente.

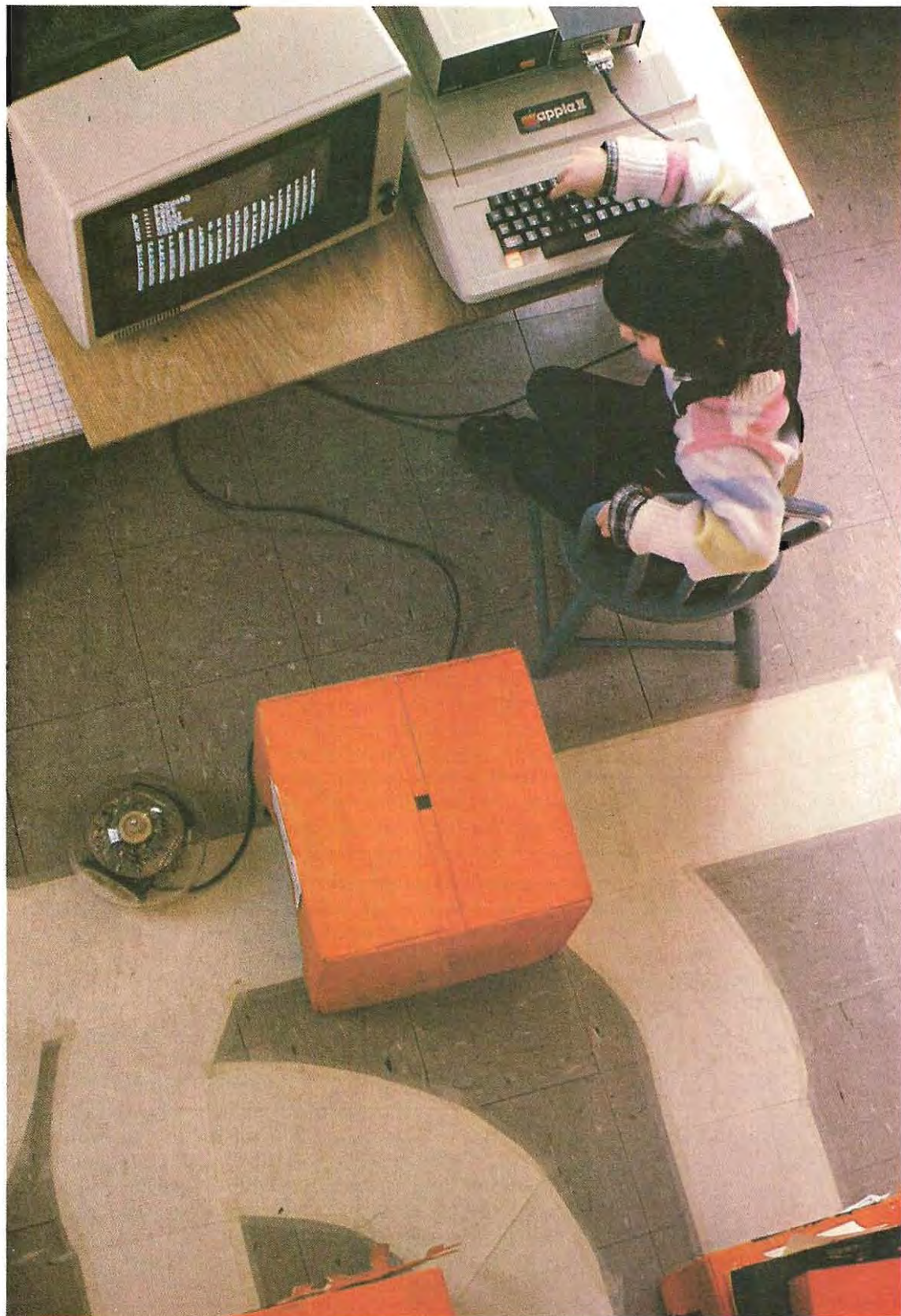
Mais se servir d'un ordinateur nécessite, comme toute technique nouvelle, un apprentissage. Il faut savoir dialoguer avec la machine, lui poser des questions intelligibles et pouvoir comprendre les informations qu'elle donne. Une encyclopédie pratique de l'informatique répond à ces besoins.

Si vous avez décidé d'acquérir un ordinateur personnel, la lecture de BASIC PLUS orientera votre choix, mais avant tout, elle vous permettra d'apprendre le langage de l'ordinateur et ses applications.

Volontairement construite comme un cours et illustrée de schémas et de nombreuses photographies, l'encyclopédie BASIC PLUS guide le lecteur pas à pas dans sa découverte des structures et du fonctionnement d'un ordinateur, des services qu'il peut en attendre et jusqu'à la façon de concevoir et d'écrire un programme.

BASIC PLUS constitue un véritable cours adapté à l'usage de chacun.

Il est probable que d'ici quelques années celui qui ne sera pas capable d'utiliser un ordinateur se sente un marginal, comme celui qui, aujourd'hui, ne sait pas se servir d'un téléphone. Il était donc temps qu'un ouvrage comme BASIC PLUS voie le jour, et initie à l'informatique tous ceux qui, de plus en plus nombreux, comprennent l'intérêt de cette science dont les applications entrent tous les jours davantage dans notre vie quotidienne et dont l'influence sur notre civilisation ne manquera pas d'être décisive.



Grazia Neri/Wheeler Pictures



# SOMMAIRE

## Comment est fait un ordinateur

*Pour juger de la diversité et de la puissance des systèmes informatiques offerts par un marché en continuelle expansion, il est nécessaire de connaître la logique d'un ordinateur, ses fonctions, la manière dont elles sont mises en œuvre et leurs applications.*

### L'ordinateur, machine intelligente ?

Qu'est-ce qu'un programme et quel est l'intérêt d'une machine programmable. Les données : définition et utilisation. Le passé et l'avenir du calcul automatique : du boulier aux premiers essais pour concevoir une machine à calculer. Les calculatrices. L'idée d'une « programmation » : naissance du premier ordinateur. Développement de l'ordinateur : la miniaturisation mènera-t-elle aux machines « pensantes » ? Les principales applications professionnelles de l'ordinateur.

### La logique de l'ordinateur

Comment « raisonne » un ordinateur. Machines analogiques et machines numériques. Un premier exemple d'application : la balance électronique. Classification des ordinateurs. Réseaux de transmission des données, liaisons entre ordinateurs. Organisation d'un centre de calcul.

### Les systèmes de numération

Les impulsions : définition et exemples. Systèmes de numération dans le monde des ordinateurs. Traduction des nombres dans les diverses bases. Opérations arithmétiques en base 2. Tables de conversion des nombres en systèmes binaire, octal et hexadécimal.

### Éléments de logique

Les opérateurs logiques : quels sont-ils et comment s'en sert-on. Les ensembles. Applications des opérateurs logiques : recherche et sélection dans les mémoires. Les circuits intégrés et leurs applications. Exemples de circuits intégrés qui développent des fonctions logiques. La protection des données : circuits pour la création des codes avec les opérateurs logiques.

### Codes de transmission des données

Les codes : traduction de nos symboles alphabétiques et numériques en langage

machine. Description du code ASCII. Les codes transparents, la codification des lettres et des nombres. Quelques codes de sécurité. Systèmes de transmission des données. Protocoles synchrones et asynchrones.

### Microsystèmes

Structure d'un système à base de micro-ordinateur. Les principaux organes. Les Bus de liaison. Périphériques, méthodes et protocoles de transmission des données. La fonction de l'unité centrale (UC ou CPU) et les calculs qu'elle peut exécuter. Fonctionnement d'un système à microprocesseur : les cycles et les instructions. Compteurs de programme, registres, mémoires. Utilisation de la mémoire : cycles d'une instruction. Les sauts à l'intérieur d'un programme. Interaction avec le monde extérieur et gestion des interruptions. L'état de Hold. Technique DMA de transfert des données.



## Traduction d'un problème pratique en un schéma logique

*Qu'est-ce qu'un programme. Méthodes pour rationaliser et résoudre un problème pratique. Présentation des principaux langages symboliques. Les méthodes de stockage des données. Structuration des données. La logique à suivre pour la saisie des données.*

### La programmation

Les langages de programmation. Langages compilés et langages interprétés. La compilation. Langages évolués. Langages machine. Caractéristiques, champs d'utilisation et critères d'évaluation.

### Analyse et organigrammes

Etablissement des programmes: analyse du problème et des données disponibles. Schématisation du problème et méthodes de structuration. Applications: définition des procédures de gestion d'un magasin. Organigrammes: définitions et symboliques. Emploi des symboles. Les index et les indicateurs d'état. Boucles et compteurs: méthodes d'utilisation, exemples d'applications. Informatique scientifique et informatique de gestion. Différentes méthodes de programmation.

### Stockage des données

Principaux types de mémoire de masse. L'archivage pour l'emploi de micro et de mini-ordinateurs. Fichiers et enregistrements. Les divers types de fichiers: principales caractéristiques et champs d'utilisation. Structuration et organisation des données: quelques exemples. Archives complexes: méthodes de stockage. Les banques de données. Quelques exemples de procédures pour la gestion des banques de données.

### Etablissement des programmes

Présentation des principaux ordinateurs personnels et mini-ordinateurs. Tableaux comparatifs et critères d'évaluation. Segmentation des programmes en fonction de la complexité du problème. Critères de choix d'une machine.

## Le langage de la programmation Basic

*Théories et méthodes pour programmer en Basic; le fonctionnement en mode conversationnel; les fonctions mathématiques, ce qu'elles sont et leur développement en Basic; méthodologies pour un usage correct des instructions Basic. Méthodes de division et de segmentation des programmes. Mise au point par segments de programme.*

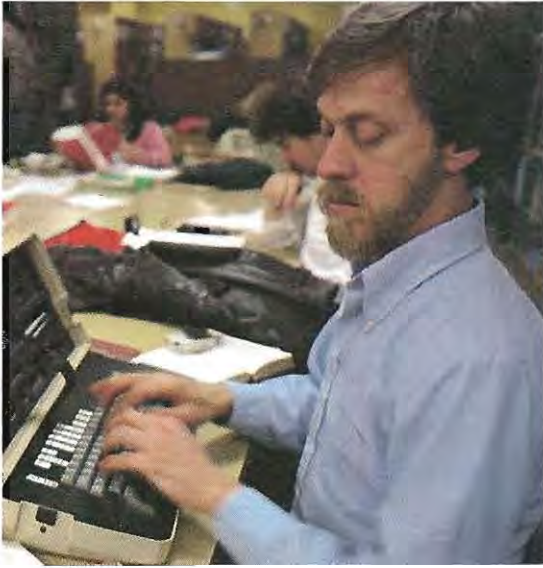
### Généralités

Emploi du Basic interprété pour le déroulement des calculs: le Basic dans le mode conversationnel. Constantes et variables, définitions; précision dans le calcul. Représentation des nombres réels en format exponentiel. Représentation des nombres entiers et réels à l'intérieur de la machine. Les zones mémoire: mémoire programme et mémoire des données. Les calculs: priorités, précision et arrondis des calculs. Les variables de chaîne; ce qu'elles sont et comment on les utilise. Opérations avec les variables de chaîne. Différences de précision obtenue par les divers types d'ordinateurs personnels.

### Les commandes

Les commandes du Basic. Ecriture et sauvegarde des programmes. Edition. Les instructions TRON et TROFF; la recherche des erreurs. Format des instructions Basic.





### Les instructions

Instructions d'affectation pour les variables. Les boucles : utilisation et syntaxe. Les sauts : sauts d'instructions, appels de sous-programmes. Utilisation des sauts et des sous-programmes. Les opérateurs logiques dans la programmation. Les instructions conditionnelles. Usage de ces instructions pour le diagnostic. Applications : établissement et développement d'un programme pour le calcul des amortissements.

### Les fonctions

Notion de fonction. Représentation graphique. Fonctions numériques en Basic : emploi, précision et limites. Fonctions numériques définies par l'utilisateur. Fonctions de chaîne en Basic. Fonctions de chaîne définies par l'utilisateur. Aperçus de statistique : génération des nombres aléatoires. Applications : établissement et écriture d'un programme pour résoudre des équations ; établissement et écriture de programmes pour la gestion des chaînes de caractères.

### Les tableaux

Mémorisation des données. Définition des tableaux mono et multidimensionnels. Tableaux numériques et de chaîne. Les instructions d'attribution dans les tableaux. Application : établissement et écriture d'un programme pour effectuer des calculs avec un nombre quelconque de chiffres ; utilisation des tableaux pour la génération des messa-

ges d'erreurs ; établissement et écriture d'un programme pour le calcul des zones.

### Fonctions d'entrée et de sortie des données

Instructions de lecture et d'impression des données. Instructions d'impression selon un format. Le curseur : commande de gestion de l'écran. Utilisation de l'imprimante, codes pour les fonctions particulières. Les tâches programmables : les instructions de programmation et leur utilisation. Applications : établissement et écriture d'un programme pour la mise en ordre des données. Les menus, systèmes de présentation et de lancement des programmes d'une procédure. Les instructions de manipulation des bits. Les instructions pour l'accès direct en mémoire.

### Gestion des archives

Les formats de mémorisation des données sur disquette. Codification et transformation des données. Les instructions de lecture et d'écriture sur disquette. Fonctions numériques et de chaîne en corrélation avec la disquette. Méthodes de gestion des fichiers multivolumes. La gestion des fichiers complexes : description de quelques programmes pour la gestion des données. L'organisation de l'archivage ; problèmes connexes à cette fonction. Programmes utilitaires pour l'organisation des archives. Recherche et sélection des données ; les sous-programmes de comparaison. Applications : établissement et écriture du programme de gestion d'un fichier d'adresses, de rubriques ou d'échéances ; établissement et écriture du programme de gestion d'un magasin ; établissement et écriture du sous-programme permettant d'arrondir les calculs.

### Structuration des programmes

Principes de structuration des programmes. Technique d'approche Top-down. Généralisation des sous-programmes. Segmentation des programmes. Transfert des données entre programmes ou entre sous-programmes ; emploi de la disquette comme support. Les masques écran, exemples d'applications.

### Conclusions

Méthodes de validation des programmes. Recherche des erreurs. Précis des instructions Basic. Applications : recours à plusieurs fichiers dans un même programme. Etablissement et préparation des organigrammes de premier niveau pour une procédure de facturation.

## La programmation avancée

*Trois autres langages importants du calculateur: l'Assembleur, le Cobol, le Fortran. Examen approfondi des instructions en Fortran, comparées à celles du Basic, et leurs applications scientifiques. Techniques de programmation en langages mixtes. L'emploi de l'ordinateur dans la réalisation des graphiques et des dessins. Les applications possibles, en particulier dans le domaine du contrôle automatique des outillages et des appareillages complexes.*

### L'Assembleur

Le langage Assembleur, structure et caractéristiques essentielles. Format des principaux types d'instructions en Assembleur. Quelques exemples. Le Macro-assembleur. Emploi de l'Assembleur; quand et pourquoi il peut être utile. Applications: procédure de facturation, préparation des organigrammes.

### Le Cobol

Le langage Cobol; structure et caractéristiques essentielles. Quelques exemples d'instructions.

### Le Fortran

Structure du langage Fortran. Champ d'utilisation. Les instructions Fortran. Comparaison entre les instructions Fortran et les instructions Basic. Précis des instructions Fortran. Applications: procédure de facturation; écriture des programmes. Traduction en Fortran de quelques programmes Basic.

### Programmes mixtes

Systèmes de programmation en plusieurs langages. Utilisation de programmes mixtes.

Basic compilé: différences et avantages par rapport au Basic interprété. Systèmes d'exploitation: structure et fonctions. Principaux programmes utilitaires dans les systèmes d'exploitation. Systèmes d'exploitation en multiprogrammation. Application: programme de calcul des salaires.

### L'infographie

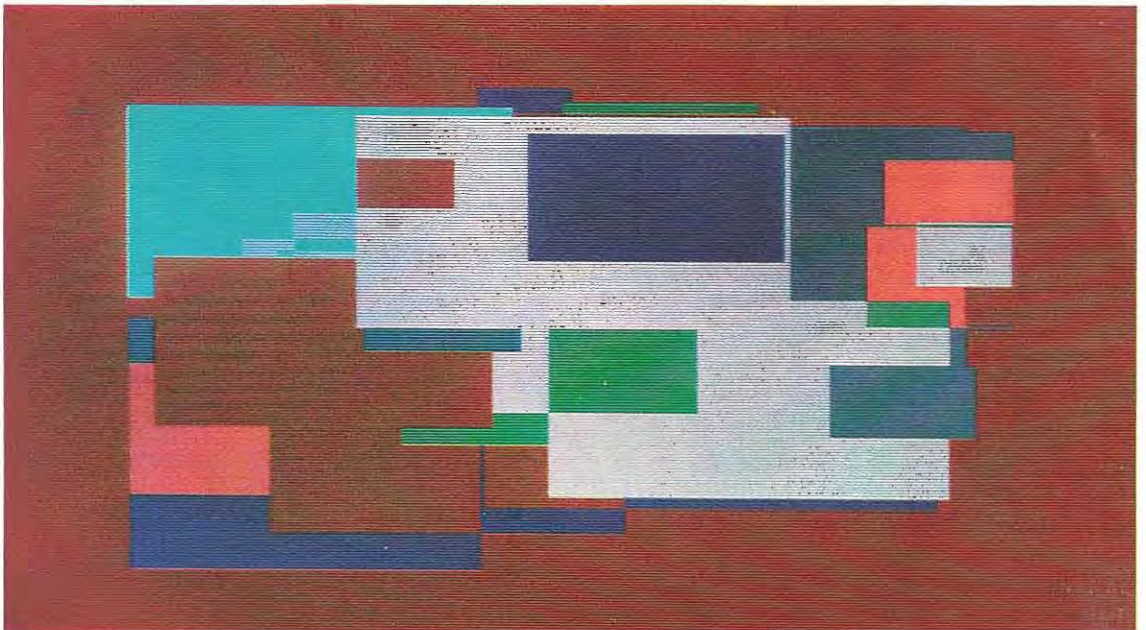
Graphiques, histogrammes, échelles de représentation. Sous-programmes pour la préparation des histogrammes et des graphiques. Le vidéographique. Le traceur de courbes. Emploi de l'imprimante pour la préparation des histogrammes. Applications: écriture des sous-programmes graphiques; écriture des sous-programmes pour la préparation des histogrammes. Représentation graphique de quelques fonctions.

### Les applications spécifiques

L'avenir: développement des systèmes à microprocesseur. Schéma d'emploi des microordinateurs dans les contrôles automatiques.

### Glossaire général

### Index analytique



## PETIT GLOSSAIRE

**ASCII** (American Standard Code for Information Interchange). Code principal utilisé pour l'échange de données entre ordinateurs.

**Bit** (de l'angl. binary digit). Quantité minimale d'information : 0 ou 1.

**Boucle** (angl. loop). Partie de programme exécutée de manière répétitive, soit un nombre de fois prédéterminé, soit jusqu'à l'obtention du résultat attendu. Exemple : partie de programme lisant sans arrêt l'état du clavier dans l'attente de l'enfoncement d'une touche.

**Bus**. Ensembles de conducteurs électriques (pistes du circuit imprimé) permettant l'échange des informations.

**Byte**. Mot américain désignant un regroupement de 8 bits et utilisé dans le sens plus limité d'octet (nombre binaire de 8 chiffres).

**Champ**. Partie d'un enregistrement contenant une donnée ou une catégorie de données. Un champ est numérique s'il ne contient que des chiffres, alphanumérique s'il contient aussi des lettres.

**Circuit intégré**. Circuit électronique rassemblant une grande quantité de composants sous un très petit volume et en mesure d'exécuter des fonctions très complexes.

**Clé**. Information alphanumérique permettant de retrouver un ensemble de données. On distingue les clés de recherche et les clés de protection.

**Codes transparents**. Les codes ASCII auxquels ne correspondent pas des caractères. On les utilise pour envoyer des commandes, ou des réponses à des commandes.

**Compilateur**. Programme utilitaire permettant de transformer un programme écrit dans un langage évolué (Basic généralement) en un programme dans un autre langage (machine) directement exécutable par le CPU (important gain de temps).

**CPU** (Central Processing Unit). En français : unité centrale (UC). Elle assure le traitement des informations. C'est le circuit qui gouverne tout le fonctionnement d'un micro-ordinateur.

**Curseur**. Repère lumineux de l'écran, généralement matérialisé par un petit carré, à partir duquel la machine commence à émettre les messages et à accepter les données d'entrée. Il marque l'emplacement dans lequel s'insère le caractère suivant.

**Display**. Affichage. Moyen visuel de communication avec la machine, généralement un écran, ou moniteur vidéo, mais aussi, par exemple, des afficheurs à cristaux liquides (micro-ordinateurs portables).

**Disquette** (angl. floppy disk). Nom courant du disque souple. Sa capacité de stockage est de l'ordre de 100 000 à 1 200 000 caractères selon le type.

**DMA** (Direct Memory Access). Méthode de transfert des données avec accès direct à la mémoire.

**Drapeau** (angl. flag). Variable dont la valeur dépend

de la vérification de situations déterminées. Elle sert à mémoriser une situation.

**Enregistrement** (angl. record). Caractérisé par son champ, un enregistrement est une partie d'un fichier. Par exemple, dans un fichier d'adresses, le code postal est un enregistrement numérique.

**Fichier** (angl. file). Zone de la mémoire de masse dans laquelle sont mémorisées des informations homogènes (archives, etc.)

**Hardware**. Ensemble du matériel employé pour le traitement des informations.

**Hold**. Etat du CPU (UC) en attente de l'achèvement d'une opération.

**Impulsion**. Signal électrique de brève durée. Une impulsion peut correspondre à un bit.

**Interrupt**. Interruption de programme provoquée par une cause externe reconnue et traitée par le CPU.

**I/O** (Input/Output). Fonction d'échange des informations entre un ordinateur et ses périphériques. Les termes français sont entrée et sortie (E/S).

**Organigramme** (angl. flowchart). Représentation graphique des opérations qui doivent s'effectuer dans un programme et de leur résolution.

**Programme objet**. Programme source ayant été assemblé, et difficilement modifiable, mais dont l'exécution par le CPU est immédiate.

**Programme source**. Programme écrit en langage symbolique qui ne sera utilisable par la machine que lorsqu'il aura été « assemblé » (traduit), devenant ainsi un programme objet, directement exécutable.

**Puce** (angl. chip). Petite pastille de matériau semi-conducteur (silicium le plus souvent) et portant un ou plusieurs circuits intégrés.

**Saut** (angl. jump). Instruction qui permet d'isoler, en la sautant, une partie de programme. Synonyme de rupture de séquence.

**Segmentation**. Technique avec laquelle sont divisés les programmes de grande masse en parties (segments) plus petites, plus faciles et plus rapides à utiliser et à mettre au point.

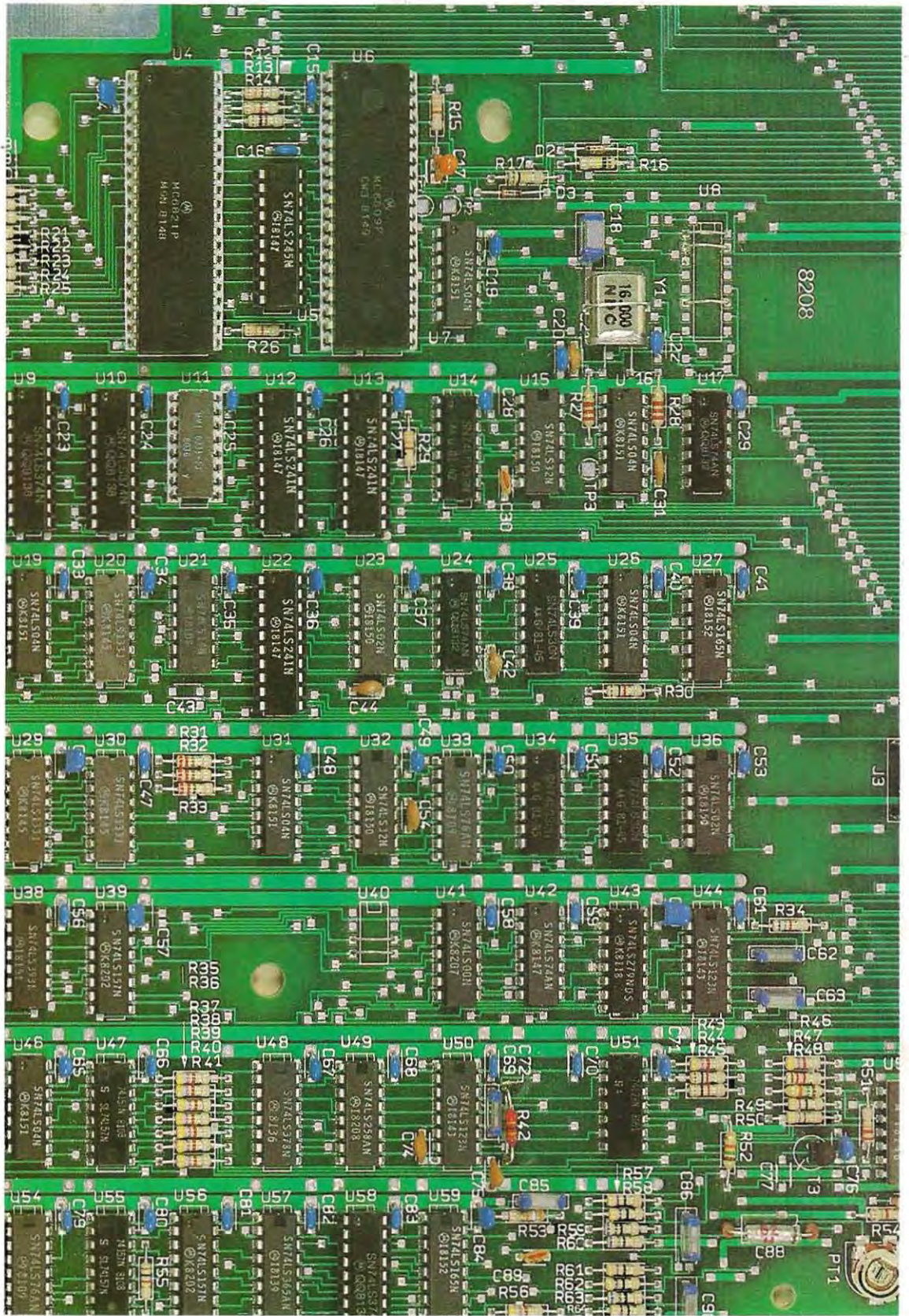
**Software**. Ensemble des programmes destinés au traitement par ordinateur. En français on parle de logiciel, par opposition à matériel (hardware).

**Sous-programme** (angl. subroutine). Partie de programme qui exécute des fonctions spécifiques. Il peut être utilisé à partir de n'importe quel point du programme principal.

**Top-down**. Analyse descendante. Technique de programmation qui consiste à développer l'analyse et à codifier des instructions, en subdivisant le problème en sous-problèmes plus simples et eux-mêmes décomposables.

**Tri** (angl. sorting). Programme particulier qui sert au classement des données. Les méthodes de tri sont nombreuses et présentent des difficultés et des intérêts divers.

Carte principale d'un micro-ordinateur T07 (D.R.)



# L'ordinateur, machine intelligente ?

Bien que de création relativement récente, les ordinateurs nous sont désormais assez familiers et il est peut-être bon de rappeler comment sont nés les premiers d'entre eux. Dérivant des calculatrices, les ancêtres de l'ordinateur ont d'abord été des appareils capables d'effectuer rapidement de très nombreux calculs. Mais ils n'étaient utilisés que par des chercheurs et restaient ignorés du grand public.

L'évolution rapide de l'électronique et la nécessité, pour l'industrie, de disposer de

machines auxquelles on pouvait confier des opérations à caractère répétitif, ont donné un élan décisif au développement et à la commercialisation des ordinateurs.

Le rôle principal de ces premières machines était d'effectuer toutes les activités qui ne requièrent pas l'intervention de l'intelligence, mais qui présentent cependant une complexité nettement supérieure à l'exécution de calculs. C'étaient par conséquent des machines capables de traiter des informations.

En outre, l'ordinateur se différencie de la calculatrice par le fait qu'il est aussi capable d'« apprendre ». On peut en effet lui enseigner comment exécuter certaines tâches, c'est-à-dire qu'il peut être programmé. Cet « apprentissage » des premiers ordinateurs a, cependant, un caractère passif : la machine elle-même n'y contribue en aucune manière.

Ce type d'enseignement, la **programmation**, est assez proche des réflexes conditionnés.

**L'intérieur d'un tel magasin fascinera toujours les passionnés d'informatique. Il leur offre, en effet, un large choix de documentation et de matériel.**



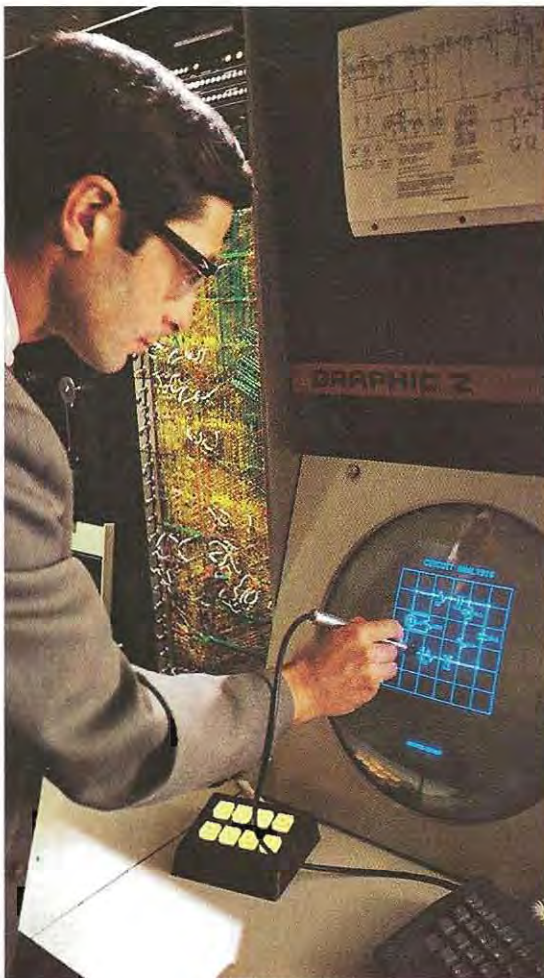
Grazia Neri/Wheeler Pictures

Prenons un exemple. Nous pouvons dresser notre chien à aller chez le marchand de journaux, à s'y faire donner le journal et à nous l'apporter. Ces actions sont déterminées par des réflexes conditionnés acquis durant le dressage, et elles n'impliquent aucune contribution de l'intelligence. Si le chien trouve le kiosque fermé, ou si le journal n'a pas paru, il ne prendra pas de décision.

Le même concept s'applique au « dressage » (programmation) des ordinateurs. Ces derniers peuvent exécuter toutes les fonctions pour lesquelles ils ont été programmés, mais dès qu'apparaît un événement imprévu, l'intervention de l'homme est nécessaire.

Pourtant, certaines recherches en cours pourraient permettre de construire des appareils dotés d'une forme de « conscience ». Au-

**Le crayon optique sert à enregistrer directement des informations en les traçant sur l'écran.**



Marka

jourd'hui, cette possibilité appartient encore à la science-fiction, bien que certains ordinateurs plus évolués soient déjà capables de détecter les erreurs, voire de diagnostiquer des pannes et d'isoler les circuits défectueux. Les capacités actuelles des ordinateurs et leur développement prévisible nous les font apparaître comme des machines extrêmement sophistiquées. Pourtant les ordinateurs peuvent être utilisés par tous. C'est surtout vrai pour les petits systèmes comme les ordinateurs personnels qui, malgré leur complexité, remplacent de plus en plus les calculatrices.

L'emploi de ces dernières est désormais aussi indispensable aux ingénieurs et aux commerçants qu'aux élèves, qui trouvent normal de confier les calculs à la machine, réservant le temps ainsi libéré aux activités qui font appel à leur capacité de raisonner.

Pour faire un devoir de mathématique, l'élève doit d'abord poser le problème, puis exécuter les calculs. Cette dernière partie du travail n'a rien à voir avec la rationalisation du problème et la solution apportée ; elle peut donc être confiée à la machine.

Si, par exemple, un élève veut calculer l'hypoténuse d'un triangle rectangle dont il connaît les côtés, il doit recourir au théorème de Pythagore qu'il posera ainsi :

$$a) \text{ hypoténuse} = \sqrt{(\text{premier côté})^2 + (\text{second côté})^2}.$$

Si les côtés valent 8 (premier côté) et 4 (second côté), la formule, obtenue à partir de la précédente en remplaçant les lettres par les valeurs numériques, est :

$$b) \text{ hypoténuse} = \sqrt{8^2 + 4^2} = \sqrt{64 + 16} = \sqrt{80} \approx 8,9.$$

Pour trouver la valeur de l'hypoténuse et terminer l'exercice, l'élève devra extraire la racine carrée d'un nombre. La génération précédente lisait le résultat sur les tables. Aujourd'hui ce résultat est fourni par la calculatrice, et l'élève de demain le demandera à l'ordinateur. Voyons pourquoi. La solution du problème est dans la formule a), le stade suivant b) étant nécessaire pour trouver la valeur numérique. L'élève qui a effectué la première partie a prouvé qu'il est capable de poser le problème et donc de le résoudre ; la phase suivante est seulement un achèvement ennuyeux et peu significatif : on peut le déléguer à la machine. S'il existait un appareil





Marka

**L'enseignement assisté par l'ordinateur permet à chaque étudiant de travailler à son rythme.**

auquel on puisse donner ces cinq instructions : « prends la valeur du premier côté et mets-la au carré », « prends la valeur du second côté et mets-la au carré », « additionne les deux carrés », « extrais la racine de la somme obtenue », « donne-moi le résultat », l'élève aurait, non seulement fini son travail après la première phrase, mais il aurait aussi préparé la réponse pour tous les autres problèmes appliquant le théorème de Pythagore.

### Les données

Une telle machine existe : c'est l'ordinateur, auquel nous pouvons « enseigner », en le programmant, comment résoudre le théorème de Pythagore. Notre rôle consiste à lui fournir, la première fois, toutes les indications nécessaires au calcul. Les fois suivantes nous n'aurons plus qu'à lui donner les nouvelles valeurs des côtés, et la machine, connaissant déjà les opérations à accomplir, pourra fournir rapidement le résultat.

Quand nous effectuons une addition, nous connaissons les deux termes et nous cherchons le résultat. Les deux nombres à addi-

tionner sont les données du calcul.

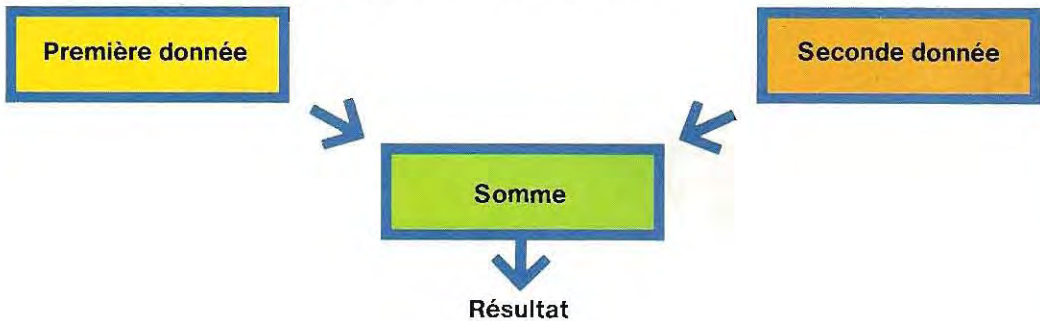
Si nous cherchons un numéro de téléphone sur l'annuaire, nous devons connaître un nom, un prénom et une adresse. Nous utilisons ces informations comme données pour retrouver le numéro de téléphone. Dans ce cas, l'action (recherche du numéro) se termine avec la lecture du numéro de téléphone lui-même ; celui-ci aussi est une donnée, puisqu'il est déjà contenu dans l'annuaire.

Dans le premier cas (calcul) on obtient une valeur (résultat) qui n'existait pas ; mais dans le second (recherche) on extrait une valeur déjà existante (numéro de téléphone). Ces deux méthodes d'utilisation des données sont schématisées page 16. Avec les ordinateurs, il convient de suivre la même logique. Pour exécuter un calcul il faut enregistrer toutes les valeurs numériques des données du problème ; dans une recherche il faut également connaître l'ensemble des données sur lequel s'effectue la sélection.

Lorsque nous parlons avec quelqu'un, nous procédons à un échange d'informations. Ces données (informations) proviennent de notre

## EMPLOI DES DONNEES DE L'ORDINATEUR

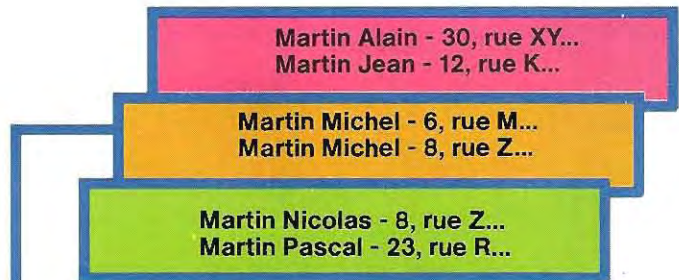
**Les données sont utilisées pour fournir un résultat**



**Les données sont utilisées pour opérer des choix (sélections)**

**Martin**

La première sélection extrait les données ayant le même premier mot que celui recherché (nom)



**Martin Michel**

La deuxième sélection extrait les données qui coïncident aussi avec le deuxième mot (prénom)



**Martin Michel - 6, rue M...**

La dernière sélection extrait les données qui coïncident aussi avec, le nom, le prénom et l'adresse



Cette dernière sélection permet d'extraire la donnée recherchée (le numéro de téléphone)

mémoire, quand, par exemple, nous relatons un fait ; elles peuvent être le résultat d'une élaboration, si nous exprimons une opinion.

Pour relater un fait ou exprimer un avis, il faut en connaître les circonstances. De même, **l'ordinateur, pour fournir une réponse, doit posséder toutes les données.**

Par exemple, en regardant de l'eau qui bout, nous concluons que, si nous la touchions, elle nous brûlerait. Ce processus mental est possible si on nous a d'abord prévenus du danger et si nous avons emmagasiné l'information. Le raisonnement suivi est simple. Il associe deux données : eau qui bout et risque de brûlure.

Le même processus s'applique à l'ordinateur : pour commencer, il faut lui fournir (en « entrée ») toutes les données et les éventuelles règles d'association ; par la suite, au moment de faire une vérification, la machine sélectionnera parmi toutes les réponses possibles la bonne réponse et elle la présentera en « sortie ».

Cependant, nos possibilités d'emmagasiner des données dans notre cerveau ne se limitent pas à des échanges : nous apprenons aussi par nos expériences. Revenons à l'exemple de l'eau qui bout. Si aucun avertissement ne nous avait été donné, l'information aurait pu provenir d'une situation vécue. La première fois, nous aurions touché le récipient et nos récepteurs (nos terminaisons nerveuses) auraient envoyé des sensations de douleur à notre cerveau, qui aurait catalogué le danger. De cette mémorisation aurait résulté notre comportement futur en présence d'une situation analogue. Cette méthode d'apprentissage, qui nous enrichit constamment de nouvelles données, a été également employée avec les ordinateurs.

C'est une application qui donne d'excellents résultats et laisse entrevoir de remarquables perspectives.

Dans les exemples présentés sont contenues les deux formes possibles de raisonnement :

- **conception déterministe**
- **conception probabiliste**

La conception déterministe consiste à associer cause et effet au moyen d'une grande quantité de données, préalablement introduites dans l'ordinateur. Autrement dit, toutes les situations possibles doivent être prévues.



Ericsson

### **Ordinateur personnel haut de gamme doté d'un clavier ergonomique et d'un écran couleur.**

Dans la conception probabiliste, la logique est plus complexe et sera mieux expliquée avec un exemple.

Supposons que nous voulons apprendre à l'ordinateur à jouer aux échecs.

Selon la méthode déterministe, nous devons stocker dans sa mémoire une réponse à chaque situation possible. Un processus de ce genre, outre qu'il serait fort laborieux, serait très certainement incomplet.

L'alternative probabiliste consiste à lui faire mémoriser les règles selon lesquelles les figures peuvent être déplacées et à lui faire jouer un grand nombre de parties.

Durant chaque partie, si un coup (d'abord joué par hasard) donne de bons résultats, l'ordinateur mémorise la situation, et le coup réalisé est classé comme valide. Si, au contraire, le résultat est négatif, la situation est classée comme à éviter.

Naturellement, la machine perdra toutes les premières parties, mais augmentera peu à peu son expérience et donc le nombre de situations qu'elle peut identifier. De ce fait, il deviendra de plus en plus difficile de la battre. C'est ainsi que l'ordinateur peut « apprendre ».

## L'ordinateur peut-il penser ?

Le savant américain Robert Jastrow, faisant un rapprochement entre les différentes caractéristiques du cerveau humain et de l'ordinateur, dans son livre *The Enchanted Loom*, a formulé quelques théories sur la capacité de « penser » de l'ordinateur, et de fascinantes hypothèses sur ses développements futurs.

*Peu de personnes seraient disposées à nier la supériorité de l'ordinateur en ce qui concerne l'arithmétique et les calculs mathématiques en général, mais elles admettraient difficilement qu'un ordinateur puisse faire plus. Comment une machine peut-elle penser ou raisonner ? La chose paraît impossible, mais un intéressant programme d'ordinateur, exposé par A.L. Samuel d'IBM, illustre comment cela peut arriver. Le programme qui enseigne à un ordinateur le jeu d'échecs démontre que la machine est capable d'exprimer des raisonnements étonnamment proches des raisonnements humains, si elle a reçu la formation appropriée.*

*Samuel commença à dresser son ordinateur en lui enseignant les règles des échecs, qu'il emmagasinait en mémoire. La formule qu'il proposa analysait la position des pièces séparément, puis elle en additionnait les points forts et les points faibles, tels que le nombre d'occasions de prendre un pion de l'adversaire et les pièges éventuels dans lesquels les siens pouvaient tomber. La formule comprenait aussi des jugements plus subtils ; elle conseillait les coups qui permettaient l'échange d'un pion, dans le cas, bien sûr, où la machine avait l'avantage. Si, au contraire, elle était en difficulté, la formule déconseillait l'échange. N'importe quel joueur humain suit la même stratégie. Enfin, la formule exprimait la force de la position de la machine à l'aide d'un nombre : plus élevé était le nombre, plus forte était la position. La formule était pour la machine l'équivalent de l'expérience d'un joueur humain qui, en général, fonde sa stratégie sur un ensemble de règles empiriques issues de son expérience. N'importe qui, tenu de faire des choix entre diverses options – le conseiller financier ou le chef d'entreprise – affronte le problème de la même manière.*

*Après avoir enseigné à la machine les bases de la stratégie du jeu d'échecs, Samuel la programma pour qu'elle améliore progressivement sa tactique au cours du jeu. Il lui conféra ainsi la capacité d'apprendre à partir de l'expérience. Pour l'ordinateur cela impliquait une mise à jour permanente des données de la formule après chaque coup. Comment la machine pouvait-elle dépasser son maître en sagesse ? En comparant la valeur qu'on attendait d'un coup à celle effectivement totalisée après la réponse de l'adversaire. La différence consistait naturellement dans le fait que l'adversaire pouvait aussi ne pas jouer le coup que la machine lui avait déjà attribué. La comparaison disait à quels facteurs de la formule on devait attribuer un poids plus ou moins grand. La machine remplaçait ces « poids » dans la formule immédiatement après chaque coup, changeant chaque fois la formule et améliorant ainsi sa stratégie de jeu. Exactement comme un être humain, la machine apprenait sur le terrain.*

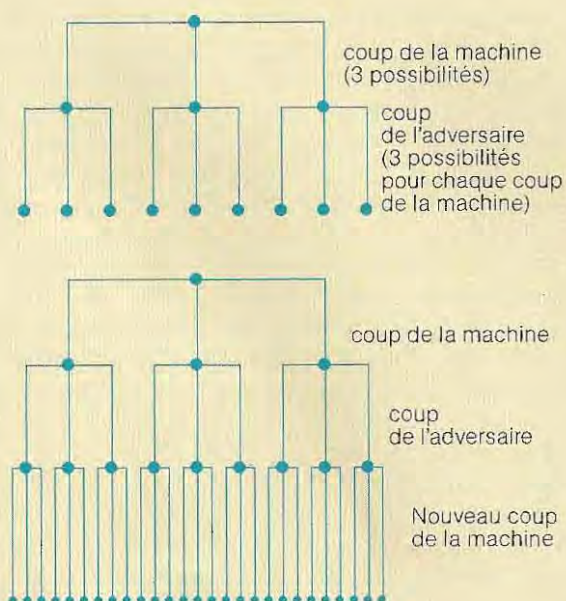
*Comment jouer contre un ordinateur ? Au début de la partie, on dispose l'échiquier avec les pièces du jeu, qu'un assistant déplacera à chaque coup, ou bien échiquier et pièces apparaissent sur un écran de télévision. Tout cela est au bénéfice du joueur humain, car la machine n'a pas besoin de regarder l'échiquier puisque sa mémoire électronique a enregistré toutes les positions et tous les échanges du jeu.*

*Les deux adversaires jouent maintenant l'un après l'autre. Le joueur humain communique ses propres coups à la machine en les tapant sur un clavier. Quand la machine décide de la réponse, elle fait clignoter le sigle correspondant sur l'écran de télévision ou l'imprime sur du papier. Se référant constamment à sa formule, la machine élabore chaque coup à la manière d'un être humain ; elle cherche à prévoir les développements du jeu aussi loin que possible, ainsi que les mérites relatifs de chaque coup éventuel, considérant que l'adversaire répondra avec le meilleur coup possible pour lui. Ce qui est aussi ce que ferait un être humain.*

*Le style de l'ordinateur dans le jeu d'échecs est en fait très semblable à celui d'un débutant qui calcule ses coups : « Si je bouge cette pièce, il bougera cette autre. Si je bouge cette autre, il bougera... ». La différence entre le*

novice et le joueur expérimenté est que ce dernier est familiarisé avec beaucoup de situations de jeu et avec les réactions probables de l'adversaire : il peut prévoir le résultat d'un coup très à l'avance, sans élaborer consciemment les coups intermédiaires. A ce point, un observateur extérieur dirait que le bon joueur a un sens intuitif des positions sur l'échiquier. En réalité, il élabore les coups de la même façon que l'ordinateur, mais la pensée se déroule au niveau inconscient.

Comment «pense» l'ordinateur? Supposons qu'à un certain point du jeu la machine ait 3 coups possibles. Elle explore chaque possibilité, calcule la valeur d'un coup particulier à l'aide de la formule, puis cherche à établir la réponse la plus probable de l'adversaire. Supposons que l'adversaire aussi ait 3 réponses possibles pour chaque coup de l'ordinateur. L'ordinateur devra donc considérer un total de 9 coups. Le schéma suivant illustre les alternatives qui, à ce stade de la partie, s'offrent à l'ordinateur. Chaque bifurcation correspond à une alternative. Le diagramme porte le nom d'«arbre des décisions». Anticipant les coups, la machine voit maintenant neuf possibilités. Elle en tient compte grâce à sa mémoire électronique, et élabore pour chacune le coup suivant. Ce qui revient à ajouter une autre ramification à l'arbre. Il présente alors 3 possibilités pour chacune des 9 précédentes, soit 27 branches.



La phase ultérieure de ce processus porterait à 81 le nombre des possibilités, ce qui signifie que la machine devrait avoir en mémoire 81 échiquiers. Cela commence à devenir difficile, même pour un ordinateur, et surtout pour les anciens modèles dont disposait Samuel dans les années 50. Le programme de Samuel s'arrêtait là.

Aujourd'hui les machines ont une rapidité mille fois supérieure, et sont capables de prévoir un bien plus grand nombre de coups ; elles jouent donc beaucoup mieux aux échecs.

La subtilité de la stratégie du jeu confère à l'ordinateur une qualité pratiquement «biologique». Après avoir fait une partie d'échecs contre un ordinateur considéré comme le meilleur joueur «artificiel» mondial, le champion écossais David Levy – maître international et l'un des 500 meilleurs joueurs du monde – déclara : «on a tendance à considérer ces ordinateurs comme des personnes, surtout quand on voit qu'ils ont compris ce que vous faites et qu'ils cherchent à faire quelque chose d'intelligent». Certains spécialistes pensent d'ailleurs que le champion du monde des échecs 1990 ne sera pas un être humain.

Bien entendu, le jeu d'échecs est un jeu très abstrait. La machine pourvue d'une logique adaptée aux échecs ne possède pas l'inspiration créatrice qui caractérise les plus hautes sphères de la pensée humaine. A l'égal d'un jeune diplômé, elle a de prodigieuses capacités intellectuelles, mais il lui manque peut-être encore la sagesse. Et pourtant, cette supériorité du cerveau sur l'ordinateur est déjà remise en question par les progrès techniques les plus récents.

Ces nouveaux développements dépendent aussi de l'augmentation prodigieuse du nombre des composants que l'on peut regrouper dans un circuit intégré. Les circuits intégrés sont comme les neurones du cerveau humain ; très nombreux et tous reliés entre eux, ils constituent le «cerveau électronique moderne». Vers 1970, une «puce» (chip) ne pouvait recevoir qu'un nombre restreint de transistors et de composants électroniques. Par conséquent, elle pouvait penser ou se souvenir, mais pas les deux à la fois. Aujourd'hui les meilleurs fabricants américains et japonais de circuits intégrés savent



**L'ordinateur personnel joue aux échecs, mais le niveau du jeu est réglable.**

regrouper tant de composants électroniques dans un circuit que, pour la première fois, il est possible de réunir sur une seule puce des circuits pensants et des unités de mémoire. Il en résulte qu'un tel circuit peut à la fois se souvenir, raisonner et prendre des décisions.

Cette combinaison de fonctions est extrêmement importante dans la mesure où, par exemple, un circuit de mémoire capable de penser peut être relié à des circuits voisins auxquels il pourra envoyer des instructions et qui lui en enverront. C'est pourquoi, quand on envoie une instruction aux unités de mémoire de l'ordinateur pour en extraire une information – le nom d'une personne par exemple – stockée dans un circuit déterminé, les circuits de mémoire directement concernés peuvent transmettre aux circuits voisins des informations connexes et offrir à l'utilisateur une plus grande quantité d'informations que ce qu'il en attendait.

On se rapproche ici du type de mémoire associative formée dans le cortex cérébral et qui constitue un puissant élément du raison-

nement humain. Le cerveau répond à une simple demande d'information par une vaste quantité d'éléments, tous liés à l'information demandée par des associations qui puisent dans l'expérience passée. La pensée humaine est grandement facilitée par la richesse de la réponse du cerveau à de telles demandes, réponses qui se développent au niveau des milliers de connexions des cellules cérébrales. Un ordinateur qui présenterait les mêmes circuits, chacun relié à un grand nombre d'autres circuits, serait un vrai cerveau au silicium. Comme le cerveau humain, il fonctionnerait à la manière d'une unité indépendante, opérant au niveau inconscient. Les dernières améliorations techniques apportées aux ordinateurs rendent possible un autre type de pensée créatrice. La pensée de l'ordinateur avait tendance à être de type mécanique et dénuée d'imagination. C'est que les circuits d'un ordinateur sont fort simples par rapport à ceux du cerveau humain. Chaque circuit d'ordinateur n'est lié qu'à deux ou trois fils d'arrivée des autres parties de la



machine, tandis qu'une cellule microscopique du cerveau reçoit du reste du cerveau des dizaines de milliers de fibres nerveuses. Les innombrables liaisons d'une cellule à l'autre, combinées aux caractéristiques subtiles des circuits du cerveau, expliquent en grande partie l'extraordinaire puissance de cet organe.

En théorie, les ordinateurs auraient pu être construits depuis longtemps avec des circuits à entrées variables, exactement comme le cerveau humain. Cependant, même petit, un ordinateur de ce type aurait nécessité des centaines de milliers de fils distincts pour les connexions de circuit à circuit, ce qui serait impossible à construire.

Avec les nouveaux circuits intégrés, toutefois, le discours change. Les fils sont absents et les connexions – microscopiques – font partie de la structure même du circuit. Cette innovation, qui a l'apparence d'un simple progrès technologique, est au contraire une phase fondamentale dans l'évolution de l'ordinateur. Elle permet en effet la construction de machi-

nes dotées d'un fonctionnement semblable à celui du cerveau humain. Si l'on en croit les prévisions, de tels ordinateurs verront le jour dans les années 90. A maints égards, ils rivaliseront avec l'esprit humain et auront beaucoup des attributs d'une vie intelligente ; une réceptivité au monde extérieur, la capacité d'apprendre par l'expérience, une prompte compréhension des idées nouvelles.

La plupart des gens diront qu'un ordinateur ne sera jamais un organisme vivant, parce qu'il n'a ni sentiments, ni émotions. Il ne mange pas, ne bouge pas, ne grandit pas ; et il est fait de métal et de plastique, au lieu de chair et d'os. On pourrait pourtant lui ajouter des roues et un moteur. Et cet ordinateur sur roulettes pourrait être programmé de façon à se diriger vers une prise de courant et à s'y relier pour avaler une « gorgée » d'électricité dans le cas où ses batteries seraient déchargées et où ses voltmètres signaleraient les tiraillements de la faim. Il est vrai qu'il faudrait quelqu'un pour lui procurer l'électricité, mais la plupart des gens n'assurent pas eux-mêmes leur alimentation. D'ailleurs, si c'était nécessaire, il ne serait pas difficile de construire un ordinateur capable de dépister l'électricité et de se l'approprier de lui-même.

Même les sentiments et les émotions peuvent être, si nécessaire, intégrés à un ordinateur, exactement comme la nature les a intégrés dans les parties les plus antiques du cerveau humain pour assurer la survie de l'homme. Samuel fit cette découverte alors qu'il cherchait à pousser son ordinateur à apprendre plus rapidement et à se montrer meilleur élève. La machine jouait assez bien pour un débutant, mais elle ne se battait pas assez. Au lieu de pousser à une rapide victoire quand elle avait le dessus, comme aurait fait un joueur humain, elle s'attardait sur toutes les alternatives possibles. Samuel décida d'inverser sa psychologie. Il modifia le programme de façon que, lorsque la machine avait l'avantage, elle devenait agressive, choisissant le coup qui devait lui donner la victoire le plus rapidement possible. Quand elle était dans une mauvaise position, elle adoptait des tactiques dilatoires, et choisissait le coup qui faisait durer le jeu le plus longtemps possible.

(D'après THE ENCHANTED LOOM, de Robert Jastrow. © 1981 by Reader's Library, Inc. ; 1982 Editions Mazarine pour l'édition française, AU-DELA DU CERVEAU.)

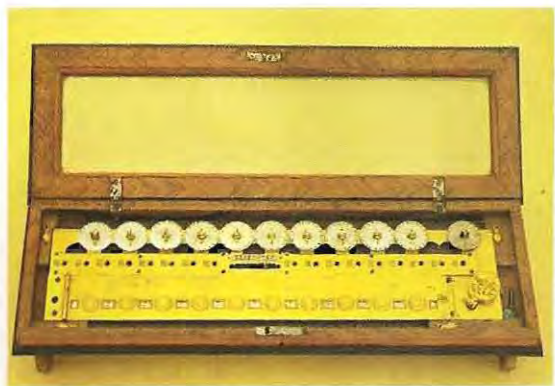
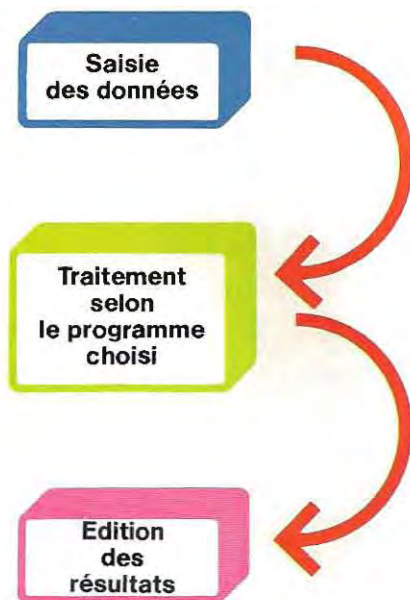
## De la calculatrice à l'ordinateur : le programme

La première machine construite pour effectuer des calculs fut d'abord une calculatrice mécanique qui exécutait des additions et des soustractions. Elle a ensuite été perfectionnée, devenant électromécanique et acquérant la capacité de faire les quatre opérations. La limite essentielle de cette machine est qu'elle contraint à poser chaque fois l'opération à effectuer et qu'on ne peut pas lui donner d'« instructions » quant au travail à réaliser. Au contraire, les ordinateurs peuvent être « formés », c'est-à-dire programmés pour exécuter certaines tâches.

L'exemple qui vient immédiatement à l'esprit est celui, bien connu, de la machine à laver. Dans cet appareil électroménager, est monté un système qui, en fonction du temps passé et de la température de l'eau, provoque les différentes phases du lavage. L'utilisateur peut choisir entre divers types de lavage (plus ou moins longs, à température plus ou moins élevée), c'est-à-dire entre divers types de programmes.

Cette machine a donc, elle aussi, reçu des données (le temps et la température de l'eau) ; selon le programme choisi, c'est le traitement de ces données qui déclenche les cycles de lavage.

Le fonctionnement d'une machine programmable peut être ainsi schématisé :



Musée d'Histoire des Sciences, Florence

**Machine arithmétique inventée en 1666 par Samuel Morland.**

## Passé et présent

Le désir de se débarrasser des calculs, surtout s'ils sont longs et ennuyeux, pour consacrer plus de temps à la réflexion, préoccupe l'homme depuis l'Antiquité.

Les premiers instruments d'aide au calcul – les bouliers – apparaissent en Chine vers le VIII<sup>e</sup> siècle avant J.-C. Les Romains, en revanche, utilisaient l'abaque, planchette portant des rainures horizontales dans lesquelles on pouvait déplacer des sortes de rondelles. La première rainure représentait les unités, la deuxième les dizaines, etc. ; c'était là une autre forme de boulier. Le boulier est encore en usage aujourd'hui et des opérateurs particulièrement habiles sont capables de s'en servir pour extraire des racines carrées. Ce fut d'ailleurs pendant longtemps le seul moyen disponible.

En 1642, Pascal invente une machine arithmétique à engrenages. Le perfectionnement de cet appareil permettra de construire la première machine à faire les multiplications. Cette machine multipliait en procédant par additions successives.

Toutes ces machines restèrent toutefois des curiosités jusqu'en 1850 environ. On construisit alors des machines à additionner plus simples et plus fiables, qui eurent enfin une utilisation pratique.

A la fin du siècle (1889) fut fabriquée une machine à faire les divisions. Elle contenait une table de Pythagore et effectuait la division en sélectionnant le résultat dans la table.

Toutes ces machines, qui constituaient pourtant les fondements du calcul mécanique,



n'exécutaient qu'une seule opération à la fois et n'avaient aucune forme de mémoire. La première machine capable de faire plusieurs opérations, et dotée d'une mémoire, fut conçue en Angleterre en 1821 par le mathématicien Charles Babbage.

Sur le principe de fonctionnement conçu par Babbage, on fabriqua par la suite des machines plus évoluées, parfois même dotées d'imprimantes.

En 1929, la première application pour le calcul scientifique fut conçue aux Etats-Unis. Le fonctionnement mécanique, lent et lourd à manipuler, fut ensuite remplacé par des mécanismes électromagnétiques actionnés par des dispositifs électriques.

Cette innovation conduisit à la fabrication, vers 1944, du Mark 1, un appareil qui, pour l'époque, avait une potentialité énorme : dix opérations à la seconde. Ses dimensions étaient remarquables aussi : 18 mètres de long sur 2,50 mètres de hauteur ! A peine trois ans plus tard, fut construite la première machine à tubes électroniques. Les opérations étaient effectuées par des circuits électroniques, mais le principe de fonctionnement était toujours celui de Babbage.

La notion de programme apparut vers 1950 : on fournissait à la machine toutes les indications nécessaires aux calculs et elle les effectuait ensuite sans autre intervention de l'opérateur. Le premier ordinateur était né.

Depuis, ce secteur a connu un développe-

ment vertigineux, qu'on n'a observé dans aucun autre domaine scientifique.

Les tubes électroniques ont été remplacés par des *transistors*, capables de remplir les mêmes fonctions avec un encombrement des dizaines de fois moindre et avec peu de dissipation de chaleur. Le progrès suivant a été l'introduction des *circuits intégrés* (entre 1965 et 1970). Cette nouvelle technologie a permis la construction d'ordinateurs aux dimensions très réduites et dotés d'une grande vitesse de calcul.

Les calculateurs modernes peuvent effectuer un million d'opérations à la seconde contre une dizaine seulement pour Mark 1 !

Depuis 1970, d'énormes progrès ont été réalisés dans le domaine de la miniaturisation des composants et des circuits. Ils ont été essentiels pour la naissance des micro-ordinateurs. Le tableau suivant résume les principales étapes de cette évolution.

<b>VIII<sup>e</sup> s. av. J.-C.</b>	Bouliers
<b>1642</b>	Mach. à additionner
<b>1850</b>	Mach. à additionner dotées de mémoire
<b>1889</b>	Mach. à diviser
<b>1929</b>	Calcul scientifique Imprimantes
<b>1944</b>	Mark 1. Dispositifs électromagnétiques
<b>1950</b>	Programmation
<b>1955</b>	Transistors
<b>1965-1970</b>	Circuits intégrés
<b>1970-1980</b>	Grands progrès de la miniaturisation

**Centre de calcul moderne organisé autour d'un ordinateur de grande puissance.**



## Le tisseur de nombres : histoire d'une invention

Le XVIII<sup>e</sup> siècle, riche en bouleversements économiques, avait été relativement pauvre en découvertes scientifiques. Dans la dernière décennie, toutefois, naquit (en 1792) Charles Babbage, généralement considéré comme le père de l'ordinateur moderne.

Babbage se penchait depuis longtemps sur des problèmes de calculs et sur la compilation de données. L'une des tâches les plus ardues et sujettes à erreurs était la compilation des tables de logarithmes : Babbage continuait d'y trouver des erreurs banales. Pensant justement à la nature répétitive de ces calculs et au gaspillage de temps, Babbage comprit qu'une machine devrait être capable d'effectuer ce type d'opérations et le schéma de base d'un tel appareil se forma très vite dans son esprit.

En 1821 il se sentit assez sûr de son fait pour annoncer à la Société Royale Astronomique qu'il avait construit un prototype. Lors d'une démonstration, il expliqua que sa machine fonctionnait selon « la méthode des différences » et il fournit des détails très convaincants sur son fonctionnement.

Nous n'allons pas expliquer comment marchait la machine : qu'il suffise de dire qu'elle était en mesure de résoudre des équations de polynômes en calculant les différences successives entre ensembles de nombres. Il la montra à une assemblée de la Société en 1822 ; la présentation fut si bien acceptée que la médaille d'or de la Société fut attribuée à son écrit « Observations sur les applications des machines au calcul des tables mathématiques ».

Désormais enthousiaste, Babbage se mit à construire la version finale de la machine. Même le coup d'œil le plus superficiel aux plans de Babbage révèle combien ils étaient audacieux. Les principes fondamentaux du calcul, les petites roues et les mécanismes n'avaient pourtant rien de particulièrement nouveau. La machine était cependant beaucoup plus grosse et beaucoup plus compliquée que le prototype ; en outre, avec son optimisme, Babbage avait prévu qu'elle ne devait pas seulement calculer des tables, mais aussi les imprimer sur papier. Il

demanda une subvention au gouvernement britannique qui lui accorda 1 500 livres sterling. Il fit construire une usine sur les terrains des Babbage, il embaucha d'habiles ouvriers et les travaux commencèrent.

La tâche principale était de créer, sur des tours spéciaux, avec la tolérance la plus étroite possible, les centaines d'arbres, de roues et d'engrenages qui devaient constituer les parties fonctionnantes de la machine à différences. Et c'est là qu'apparurent les premiers grands obstacles. Le modèle construit pour la Société Astronomique était seulement un prototype. Les très petites irrégularités de ses composants provoquaient bien une certaine instabilité du système, mais pas au point, cependant, de causer des arrêts ou de bloquer le fonctionnement global. Au contraire, dans la machine à différences, des séries de petites imperfections tendaient à s'additionner, provoquant de grandes vibrations et des grippages.

Babbage poussa ses ouvriers à travailler avec une meilleure précision. Une amélioration s'ensuivit, mais pas suffisamment appropriée à la complexité globale du système. Insensible aux premiers échecs, Babbage s'entêta, sans réaliser pourtant que c'était demander aux machines de l'époque des performances qu'elles ne pouvaient pas fournir.

Il demanda de plus en plus d'argent au gouvernement. On lui accorda jusqu'à 17 000 livres, puis il fut décidé que c'en était trop et le projet fut suspendu en 1833.

Si Babbage avait été une personne raisonnable, ce qu'il n'était pas, cela aurait été le moment de s'arrêter. Au contraire, son esprit infatigable commença à se consacrer à un schéma encore plus ambitieux : c'est à ce moment que naquit le concept d'ordinateur. Bien que la machine à différences ait constitué un net progrès sur celles qui avaient été construites auparavant, elle n'était capable d'accomplir qu'une seule tâche : résoudre des équations de polynômes. Le résultat était obtenu à travers une série de mouvements convenus : un ensemble de petites roues en faisait tourner un autre, un levier en levait un autre et ainsi de suite, en une séquence plus ou moins prévisible.

En d'autres termes, la machine n'était capable d'exécuter que l'unique type de fonction pour laquelle elle avait été fabriquée.

En y réfléchissant bien, Babbage se rendit compte qu'il faisait fausse route. Il se dit qu'une machine capable d'effectuer un type de calculs devrait être aussi capable d'effectuer tous les autres calculs. Le bien-fondé de cette idée allait être mathématiquement démontré pratiquement un siècle plus tard par un autre génie anglais : Alan Turing.

Pourquoi ne pas concevoir une machine en la déterminant à l'avance, non pas pour une tâche définie, mais pour plusieurs tâches, afin que son propriétaire, au moment où il le voudrait, puisse lui en demander l'exécution. Une telle machine risquait d'être plus compliquée, et de coûter encore plus cher à construire, mais elle serait aussi beaucoup plus utile. Sans doute, il pouvait paraître évident de voir la nouvelle machine comme un ensemble de plusieurs autres, possédant un seul mécanisme d'entrée et une connexion commune à l'imprimante ou à tout autre type de sortie.

Babbage refusa ce qui venait d'abord à l'idée de tout le monde, et il aboutit à une intuition extraordinaire. Son projet devait permettre d'utiliser les organes internes de la machine de multiples façons. De cette manière, devant une tâche particulière, une seule séquence aurait été exécutée, et chaque tâche différente aurait été liée à des modèles uniques d'activité interne.

Il manquait cependant encore l'idée intelli-

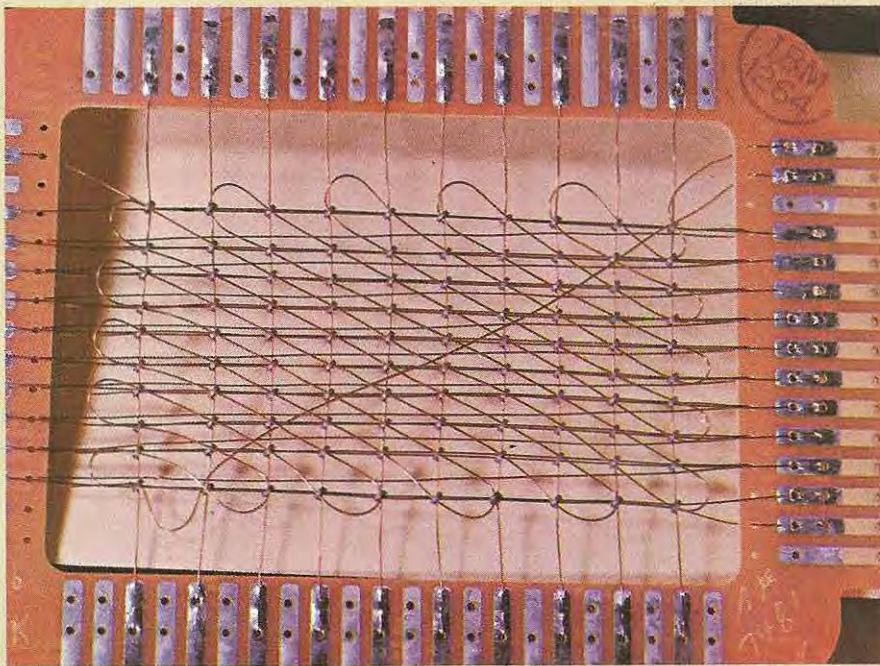
gente de savoir comment on pourrait « dire » à la machine quelle action elle avait à choisir parmi tout cet « ensemble d'actions » et à quel moment précis elle aurait à la déclencher.

Babbage appela ce dispositif « machine analytique ». Il convient de souligner ici qu'il parlait déjà d'un « calculateur programmable ».

En premier lieu, cette machine comprenait un ensemble de dispositifs d'entrée, qui permettaient d'y introduire des nombres ou des instructions. Puis il y avait une unité arithmétique, ou processeur, c'est-à-dire la partie de la machine qui exécutait vraiment les calculs. Babbage l'appela « la machine ». En troisième lieu, on trouvait une unité de contrôle assurant que le calculateur exécutait bien une tâche plutôt qu'une autre, et qu'il accomplissait bien tous les calculs dans la bonne séquence. C'est un magasin, ou mémoire, qui retenait les nombres en instance de traitement. Enfin, il y avait le dispositif de sortie proprement dit.

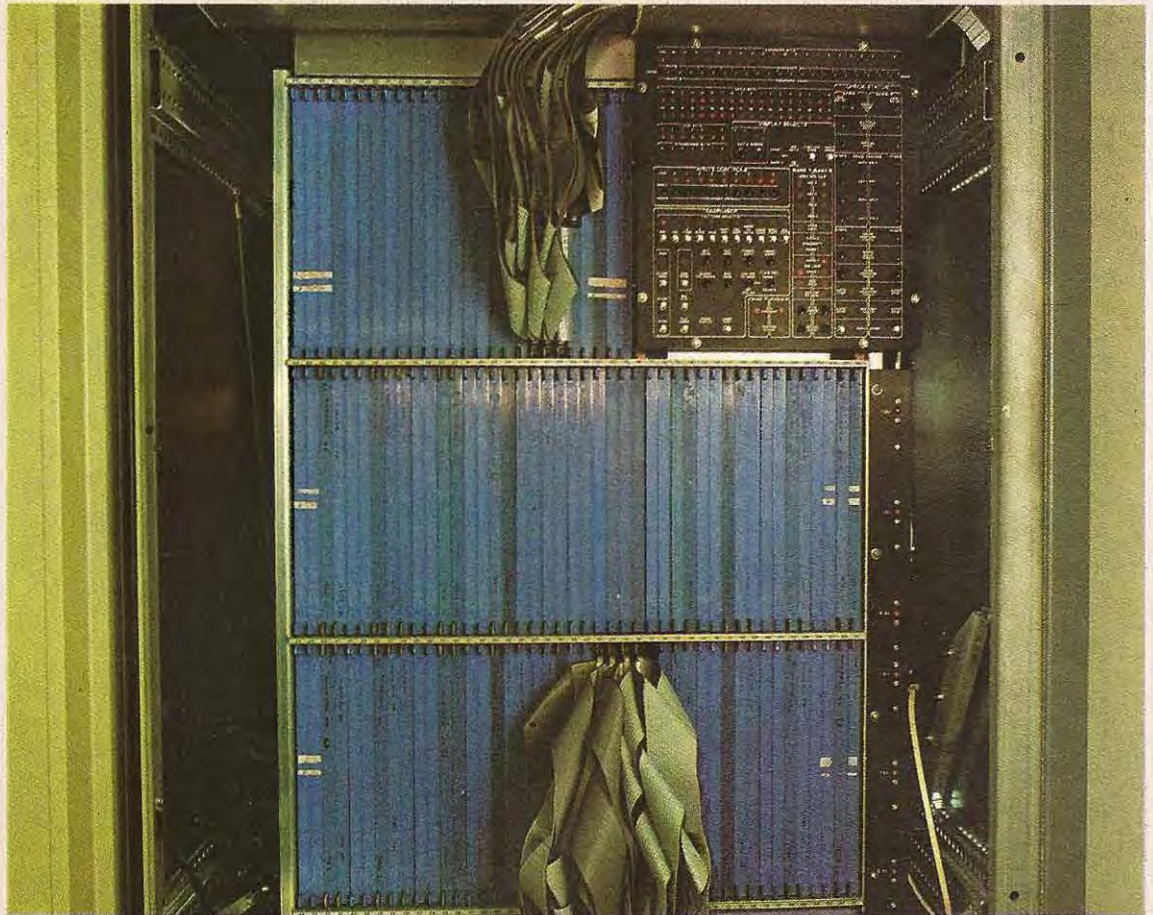
Bien qu'encore schématique, ces cinq parties sont les organes essentiels d'un ordinateur ancien ou moderne. Les dispositifs de calcul et de mémoire de Babbage étaient des colonnes de roues à dix dents, réunies par une incroyable forêt d'arbres et de joints.

Le prototype de la machine à différences était manœuvré à la main. On actionnait un levier pour faire bouger les engrenages ; une cloche sonnait quand le stade de calcul



Archives Curcio/IBM

**La première forme de mémoire électronique était constituée de noyaux magnétiques, où l'élément de mémoire est un petit anneau (tore) de ferrite et où l'entrelacement des fils sert à sélectionner les "adresses", ainsi qu'à lire et écrire la donnée au moyen d'impulsions électroniques.**



**Les mémoires des ordinateurs évoluent. Ici : intérieur d'un ordinateur moderne constitué d'un assemblage de cartes mémoire reliées par des câbles plats.**

recherché était atteint. On introduisait alors ce qu'on voulait, on actionnait de nouveau le levier jusqu'à la sonnerie de la cloche et ainsi de suite. Babbage, qui avait conscience de l'absurdité de ces manœuvres, décida d'adopter l'alimentation à la vapeur pour sa dernière version, la machine analytique. Pour charger et pour programmer les instructions de l'unité de contrôle, Babbage s'inspira d'une invention du Français Joseph Jacquard. Ce dernier s'était rendu compte que les tisserands accomplissaient devant leur métier une tâche délicate mais répétitive et qui aurait donc pu être automatisée. Il avait alors inventé un carton rigide perforé. Pendant le tissage, une série de tiges guidaient les fils de chaîne, et la fonction du carton perforé était d'en bloquer certaines tandis que les autres passaient à travers les trous du carton, continuant le tissage. A chaque passage de la navette, un carton perforé d'une série donnée

de trous apparaissait sur le passage des tiges, contrôlant le dessin du tissu. En quelque sorte, il contrôlait aussi le métier à tisser. Babbage comprit que ce système aurait été autrement utile pour contrôler la séquence des calculs dans sa machine.

Le rapport avec le tissage fut également découvert par Ada, comtesse de Lovelace qui, dans un commentaire sur la machine de Babbage, écrivit : « la machine analytique tisse ensemble les algébriques comme le métier de Jacquard tisse les feuilles et les fleurs ». Cette femme était une mathématicienne exceptionnelle, et qui plus est, d'une beauté hors du commun. Un croquis de l'époque la montre ressemblant légèrement à Elizabeth Taylor, avec quelques traces des beaux traits de son père, Lord Byron. Quand elle fit connaissance de Babbage et se rendit compte de l'importance de ses tentatives, elle se mit à étudier à fond les plans de la

machine analytique et se fit donner des explications de vive voix. Elle avait de l'argent et son enthousiasme, mais il fallut plusieurs années avant d'arriver à quelque chose.

Finalement elle publia une longue série de notes intitulée « Observations sur la machine analytique de Monsieur Babbage ». C'est une excellente lecture pour qui veut connaître l'entreprise dans le détail, et on y voit que lady Lovelace avait conscience de quelques-uns des problèmes philosophiques qui se posent dans la construction d'une machine de ce type, problèmes dont certains n'ont toujours pas été résolus aujourd'hui. Par exemple, sur la question de savoir si la machine était créative ou non, elle écrivait : « la machine analytique n'a nullement la prétention de créer quoi que ce soit. Elle peut faire tout ce que nous sommes en mesure de programmer. Elle peut faire des analyses, mais elle ne peut avancer aucune vérité ni aucune relation analytique. Son but, c'est de rendre accessible ce que nous connaissons déjà ». Ce commentaire très intelligent semble avoir été la première déclaration sur les capacités intellectuelles d'un ordinateur. Un ordinateur peut faire seulement ce pour quoi il est programmé. Comme on le verra par la suite, la question est intéressante, bien que peut-être un peu secondaire. Le mérite de l'avoir formulée en revient entièrement à lady Lovelace.

Le plus important est l'effet psychologique que les écrits de cette dame eurent sur Babbage. Jusque-là, à part lors de son succès auprès de la Société Astronomique, il n'avait jamais été approuvé de personne, ni même compris. Le plus satisfaisant pour lui était que la comtesse ait pris la peine d'étudier ses théories sous leur aspect mathématique et qu'elle n'y ait trouvé aucune imperfection. Lui, il savait bien que la machine analytique devait fonctionner et voici que cette femme le croyait aussi. Alors il ne restait plus qu'à construire l'appareil en question.

Cependant, en Angleterre le gouvernement avait changé. Le ministre du Trésor avait renoncé à financer un projet qui paraissait désormais irréalisable. Le temps passait, la machine à différences restait un ensemble incomplet de roues et de cames ; la machine analytique n'était qu'une série de croquis, d'idées et de notes de lady Lovelace. Les choses allaient de mal en pis. Ada Lovelace mou-

rut, âgée de trente-six ans, et Babbage continua seul, obtenant peu de résultats. Les gouvernements se succédaient, aucun ne sympathisant avec son idée. Disraeli écrivit même, faisant preuve d'un humour plutôt sarcastique, que, selon lui, le seul emploi qu'on pouvait faire de la machine était de s'en servir pour calculer combien elle avait coûté...

Pourtant, pendant ce temps, d'autres mathématiciens et ingénieurs feuilletaient avec grand intérêt les publications de Babbage et les notes de lady Lovelace. L'un d'eux, l'ingénieur suédois George Scheutz, fit construire sa propre version de la machine à différences. Contrairement à Babbage, il obtint un succès si retentissant qu'il construisit aussitôt un prototype qu'il montra comme le premier « modèle pour la production » à une Exposition d'Ingénierie en 1855. Dans la foule qui se pressait autour de ce curieux dispositif, se tenait Babbage, âgé de soixante-treize ans et fort découragé. Quand on lui demanda un commentaire sur la machine de Scheutz, il le fit courtoisement et ils se congratulèrent. Néanmoins, on peut imaginer les pensées qui lui ont alors traversé l'esprit.

Babbage devait mourir en 1871, à l'âge de soixante-dix-neuf ans. Il faut bien admettre qu'il est mort déçu, après avoir pourtant conçu un projet si révolutionnaire qu'il devait un jour changer le monde. Ses contemporains le considéraient comme un génie rêveur et sans jugement, mais en tout cas comme un génie. Ils étaient même tellement convaincus de la supériorité de son intelligence, qu'après sa mort ils soumirent son cerveau à un examen attentif pour voir s'il présentait des caractéristiques physiques différentes d'un cerveau commun. Victor Horsely, l'un des plus célèbres chirurgiens de l'époque, s'occupa de l'examen. Après avoir tourmenté des millions de neurones éteints, il déclara que ce cerveau ne lui semblait pas différent de tant d'autres qu'il avait déjà vus. Toutefois, personne ne se décida à détruire le cerveau de Babbage, et il est encore conservé dans deux bocaux, un pour chaque hémisphère, au musée Hunter du Collège Royal des Chirurgiens, où on vous le montrera si vous le désirez.

(D'après THE MIGHTY MICRO de Christopher Evans. © Christopher Evans 1979.)

## Evolution des ordinateurs : applications

Dans les années 80, l'évolution de l'ordinateur n'a été marquée par aucun événement spectaculaire. Pourtant, les progrès de la miniaturisation allaient permettre de diminuer la taille et le prix des ordinateurs, les rendant accessibles à un plus grand nombre d'amateurs.

Pendant ce temps-là, un profond changement intervenait dans la société.

Dans un monde industrialisé, la nécessité première est l'information. L'échange d'informations au moment opportun peut signifier l'aboutissement d'une recherche scientifique, le juste choix d'une décision économique ou, plus simplement, un précieux gain de temps. L'ordinateur est l'instrument idéal pour ces applications. Sa capacité de mémoriser autorise à stocker de grandes quantités d'informations auxquelles l'utilisateur peut accéder très rapidement.

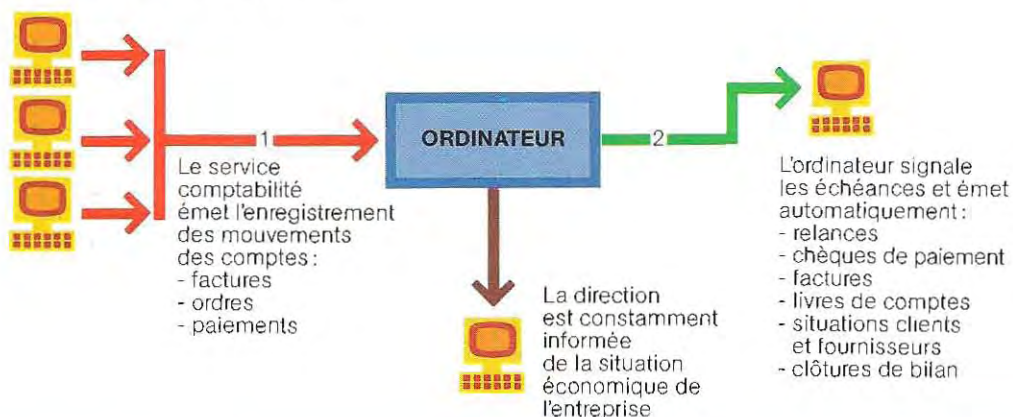
Les exemples sont multiples. Prenons d'abord celui de la gestion d'une bibliothèque. La recherche des volumes qui traitent d'un sujet déterminé est longue et ennuyeuse si elle est faite à la main. Au contraire, si la bibliothèque est gérée par un ordinateur, les informations sont acquises très rapidement, et sans contraindre l'utilisateur à une activité manuelle qui, par ailleurs, vole du temps à sa créativité.

Citons un autre exemple. Dans toute entreprise, les différents départements doivent connaître la situation générale de la société pour pouvoir programmer leurs propres activités. Ainsi, le stock de pièces finies doit être connu aussi bien du service de production que de celui des expéditions. Cette complexité d'informations ne peut être gérée avec succès que par l'intermédiaire d'un ordinateur. Les schémas reproduits dans cette page et la suivante donnent une idée des services que peut rendre l'ordinateur dans une entreprise.

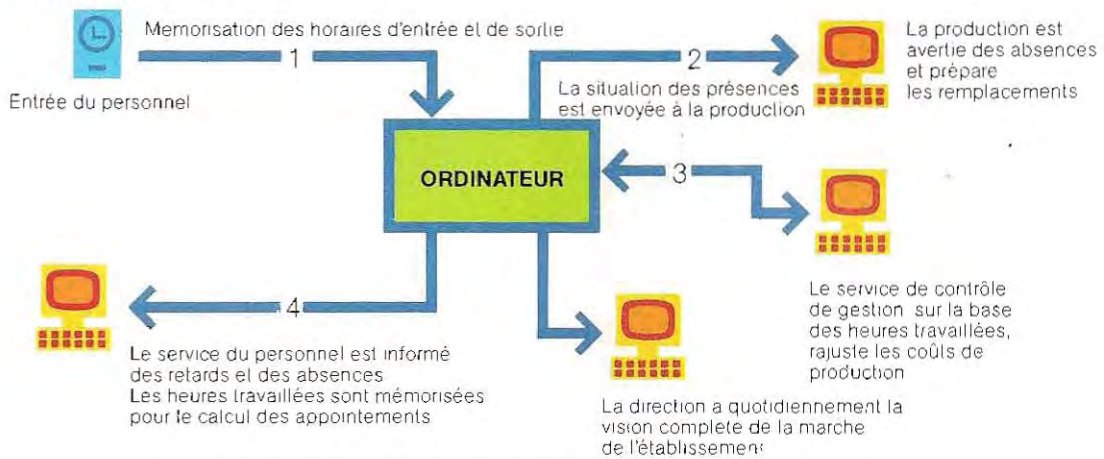
### PRINCIPAUX CHAMPS D'APPLICATION DE L'ORDINATEUR DANS UNE ENTREPRISE



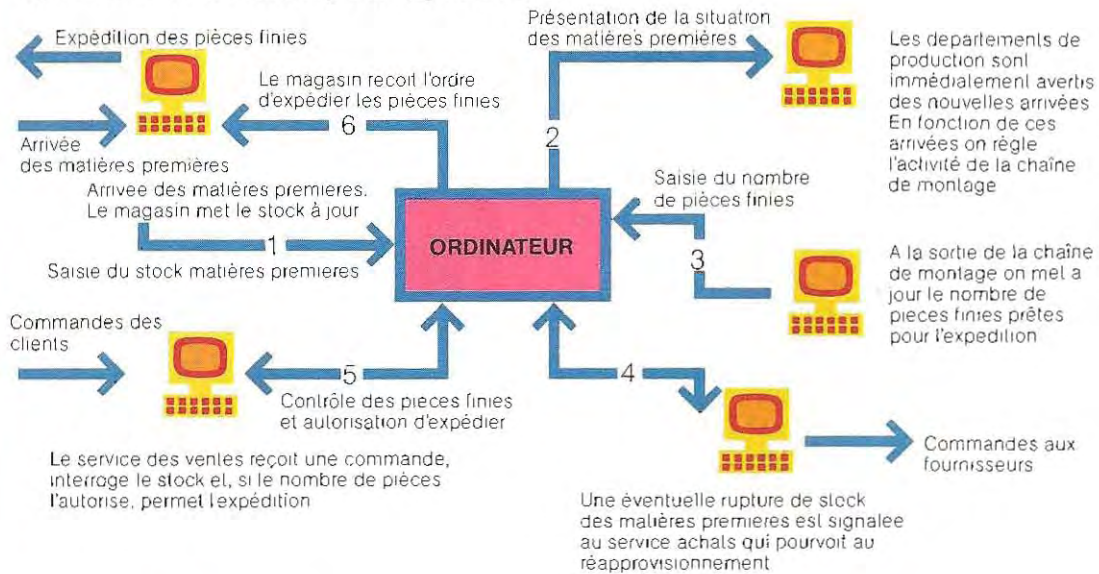
#### Gestion de la comptabilité



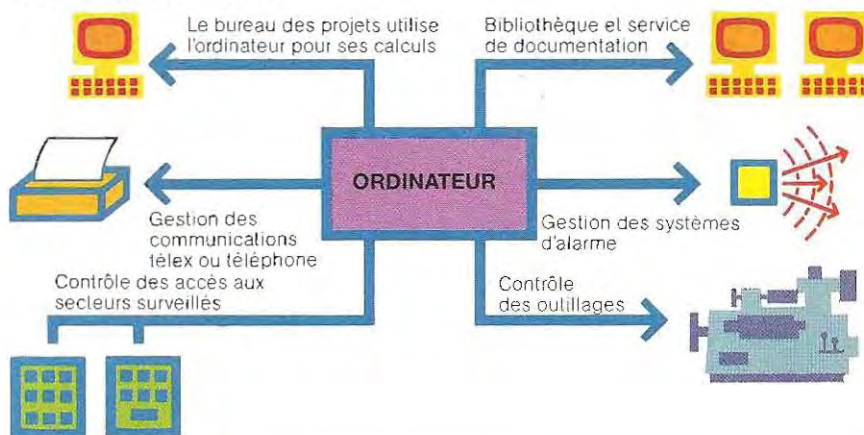
## Gestion du personnel



## Système d'information pour la production



## Applications particulières

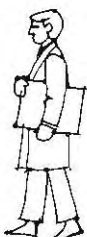


## STRUCTURE D'UN CENTRE DE CALCUL

### Utilisateur



### Ingénieur-système



Analyse le problème global et indique quel type de machine et quelle puissance de calcul sont nécessaires à sa solution.

### Analyste



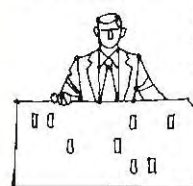
Analyse le problème en terme d'opération. Fournit les indications de base pour l'écriture du programme.

### Programmeur



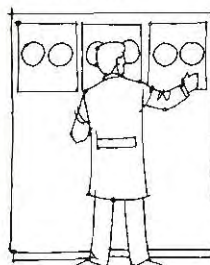
Traduit les indications de l'analyste sous forme de programme.

### Opérateur de saisie



Prépare les données sous la forme adaptée à leur introduction dans l'ordinateur.

### Pupitreur



Suit le fonctionnement de la machine.

La science particulière qui étudie les formes et les modes de traitement automatique de l'information est l'**Informatique**.

Cette science a des aspects très complexes, et elle transformera, dans un proche futur, les procédures de la gestion d'entreprise.

L'informatique n'a pas seulement cet aspect rigoureusement scientifique applicable à une industrie complexe : elle peut également être une aide de la gestion familiale.

Combien d'entre nous connaissent, à tout moment, le montant de leur compte en banque ? La majorité des gens a en tête un chiffre approximatif, parfois assez éloigné de la réalité. Or il existe déjà des services bancaires à domicile qui permettent, à partir d'un terminal,

d'interroger son propre compte, ou de faire des transferts d'un compte à l'autre, entre autres.

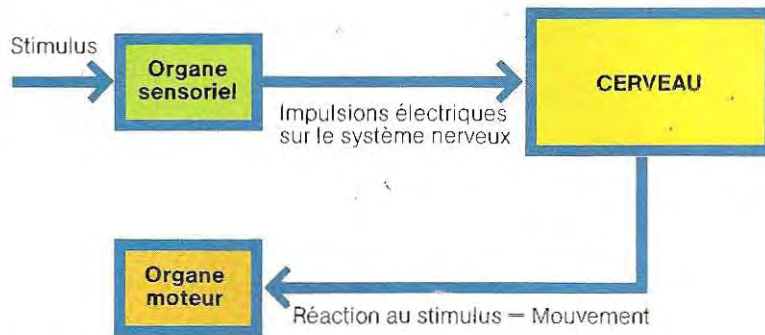
Il y a encore peu de temps, les ordinateurs étaient des instruments coûteux et compliqués, dont l'usage exigeait la compétence de spécialistes ayant chacun des fonctions bien définies (voir le schéma de cette page). La structure nécessaire au fonctionnement de ces machines constituait une charge économique considérable et, par conséquent, les ordinateurs ne pouvaient être vraiment utilisés que dans de grandes entreprises industrielles. L'entreprise de petite ou moyenne dimension n'avait aucun intérêt économique à adopter des procédures automatisées.

Ces dernières années, les développements



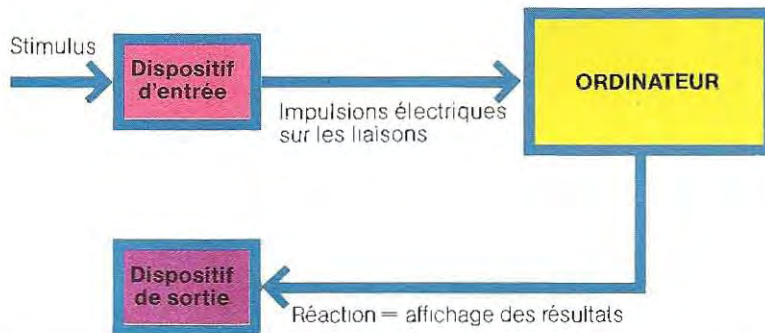
## SCHEMA LOGIQUE DU PROCESSUS ACTION-REACTION

### Processus action-réaction dans le cerveau



Un stimulus, la piqûre d'un insecte par exemple, produit une série d'impulsions électriques. Le cerveau les traite et commande, en réaction, une action motrice.

### Processus action-réaction dans l'ordinateur

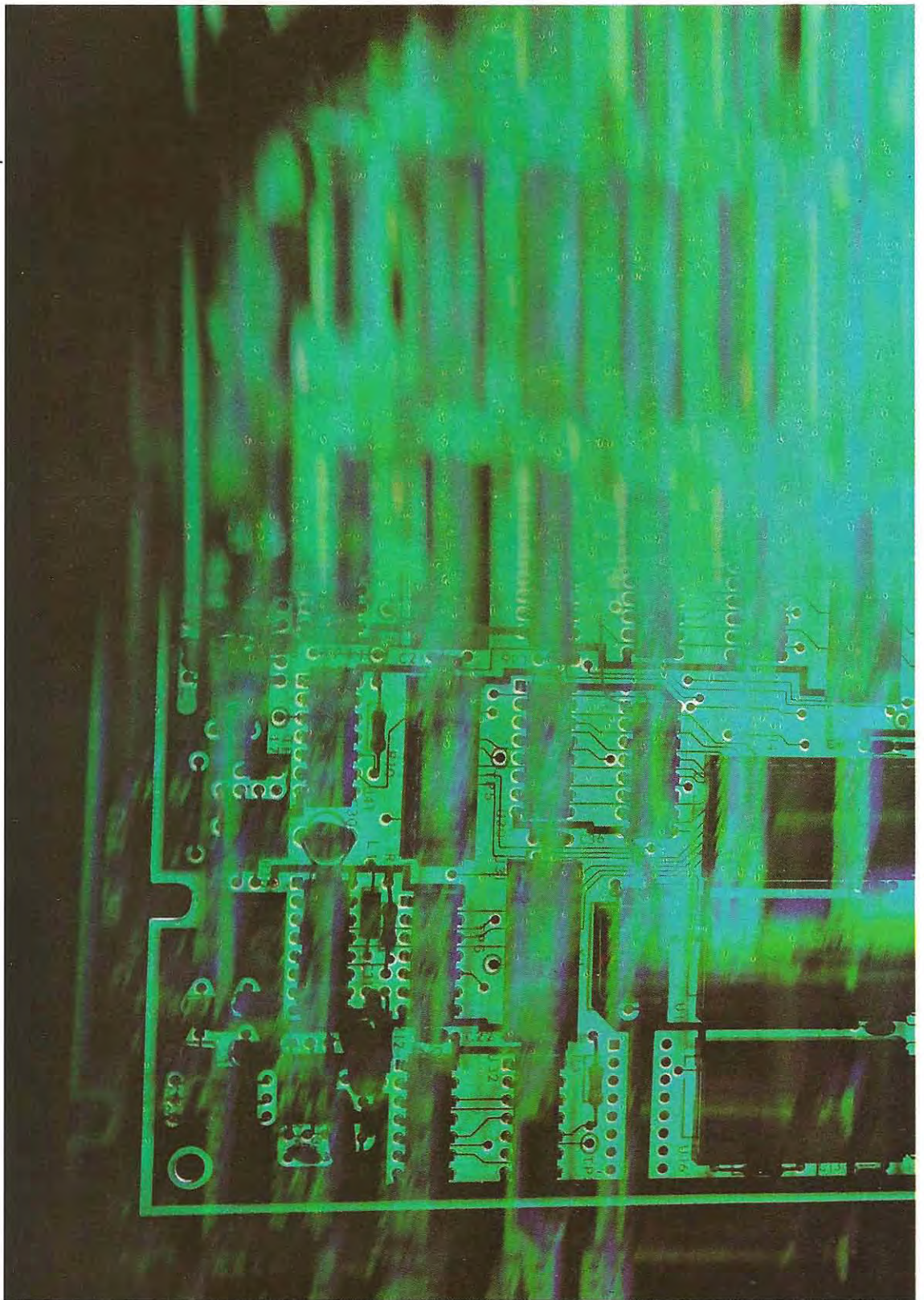


Dans l'ordinateur, le stimulus est l'ordre d'exécuter une certaine transaction. Cet ordre est converti en signaux électriques dans le dispositif d'entrée et envoyé, sur les liaisons appropriées, à l'ordinateur. Les signaux sont interprétés et provoquent l'affichage des résultats (réaction).

technologiques ont au contraire rendu possible la production d'ordinateurs de dimensions très réduites, d'utilisation plus facile et d'un coût assez bas. On imagine facilement ce que pourra être leur développement. Déjà, certains ordinateurs parlent, sont en mesure de lire des textes, d'exécuter des dessins, de jouer aux échecs, de faire des traductions, de jouer de la musique. Quelques prototypes automobiles sont même équipés de microprocesseurs qui obéissent à la parole: ils savent reconnaître la voix de leur propriétaire. Le développement le plus intéressant interviendra peut-être dans le domaine de la médecine en raison des remarquables analogies entre les mécanismes de la perception

des stimuli par le corps humain et la méthode d'acquisition des données dans un ordinateur (voir schéma de cette page). Le neurone (unité fondamentale du système nerveux) peut se trouver seulement dans deux états; excité ou non excité. Ces deux seuls états possibles sont à la base du fonctionnement des ordinateurs numériques.

Le développement des ordinateurs et les analogies avec le corps humain ont conduit à la naissance d'une nouvelle science: la **Robotique**. Les découvertes en ce domaine ont permis de produire des machines capables d'effectuer pratiquement tous les travaux humains et qui peuvent être programmées pour avoir des réactions semblables aux nôtres.



Mario Falconi

**Le circuit imprimé est l'un des éléments fondamentaux de toutes les machines électroniques.**

# La logique de l'ordinateur

Un ordinateur est constitué de composants électroniques simples ou de sous-ensembles permettant :

- la saisie des informations
- leur traitement
- leur mémorisation (stockage)
- leur diffusion (transmission, édition, etc.).

Le traitement est l'affaire du programme, lequel est exécuté par l'unité centrale, qui doit elle-même disposer d'une unité de calcul, d'une unité logique (qui assure les comparaisons entre données) et d'une mémoire centrale où est logé le programme.

Le déroulement d'un programme peut être séquentiel ou logique. Dans une série séquentielle, chaque fonction est toujours effectuée après la précédente et avant la suivante.

Dans une série logique, l'ordre normal peut être modifié selon des événements déterminés. Prenons des exemples hors de l'ordinateur :

- Des feux de circulation tricolores fonctionnent de façon séquentielle : les couleurs se succèdent toujours dans le même ordre.
- L'emploi du changement de vitesses d'une voiture est une séquence logique,

puisque le passage d'un état (vitesse) à l'autre est réglé par des causes non liées au temps (voir schéma en bas de page).

Toutes les fonctions que l'ordinateur devra développer sont prédéfinies et mises en œuvre par le programme. Dans le programme il conviendra de prévoir toutes les voies possibles résultant d'éventuels choix logiques.

Les ordinateurs peuvent être classés en deux familles fondamentales :

- **Ordinateurs analogiques**
- **Ordinateurs numériques**

## Les ordinateurs analogiques et les ordinateurs numériques

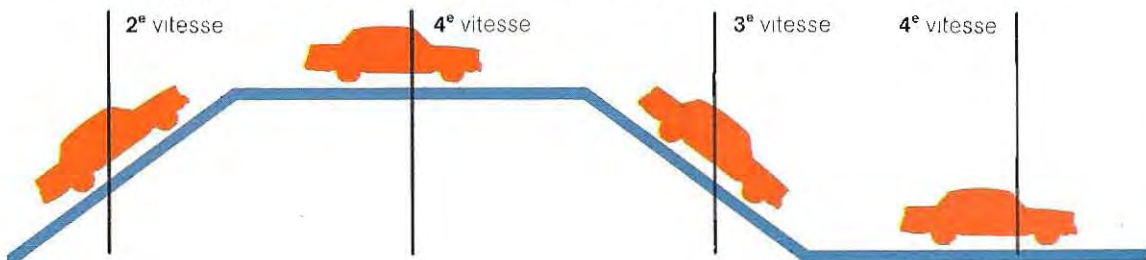
Le terme analogique dérive de leur principe de fonctionnement. Le calcul mathématique à résoudre est effectué au moyen d'une analogie avec des grandeurs physiques. La règle à calcul, par exemple, est un dispositif analogique puisqu'elle permet de trouver la solution d'opérations arithmétiques grâce à une analogie avec des longueurs (voir figure p. 34, en haut). Dans ce genre de machines, le résultat d'une opération est obtenu en exploitant des lois physiques adéquates. Pour chaque type d'opération il faut donc le circuit approprié, et ce sont des machines limitées à des emplois spécifiques de type essentiellement scientifique.

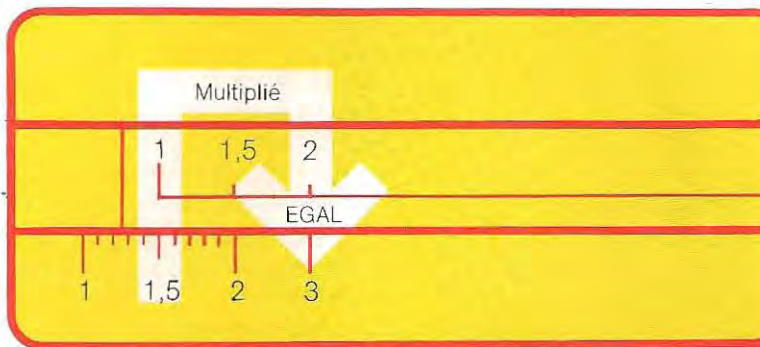
Nous nous limiterons à parler des ordinateurs électroniques actuels, même si notre démonstration peut s'appliquer à d'autres machines d'un type plus ancien (ordinateurs à fluide ou ordinateurs mécaniques).

### Mode séquentiel : les feux de signalisation



### Mode logique : l'emploi du changement de vitesses





### REGLE A CALCUL

La règle à calcul est un instrument communément utilisé pour exécuter des calculs parfois complexes, exception faite des additions et des soustractions. C'est un exemple d'instrument analogique : en effet elle effectue des calculs par analogie avec les longueurs des graduations qu'elle porte.

Les circuits électroniques de l'ordinateur peuvent être de deux sortes :

- circuits numériques
- circuits analogiques

Un circuit numérique ne sait reconnaître que deux situations :

**Présence de signal = ON**

**Absence de signal = OFF**

En revanche, le circuit analogique est capable de mesurer la présence ou l'absence de signal, et également sa valeur.

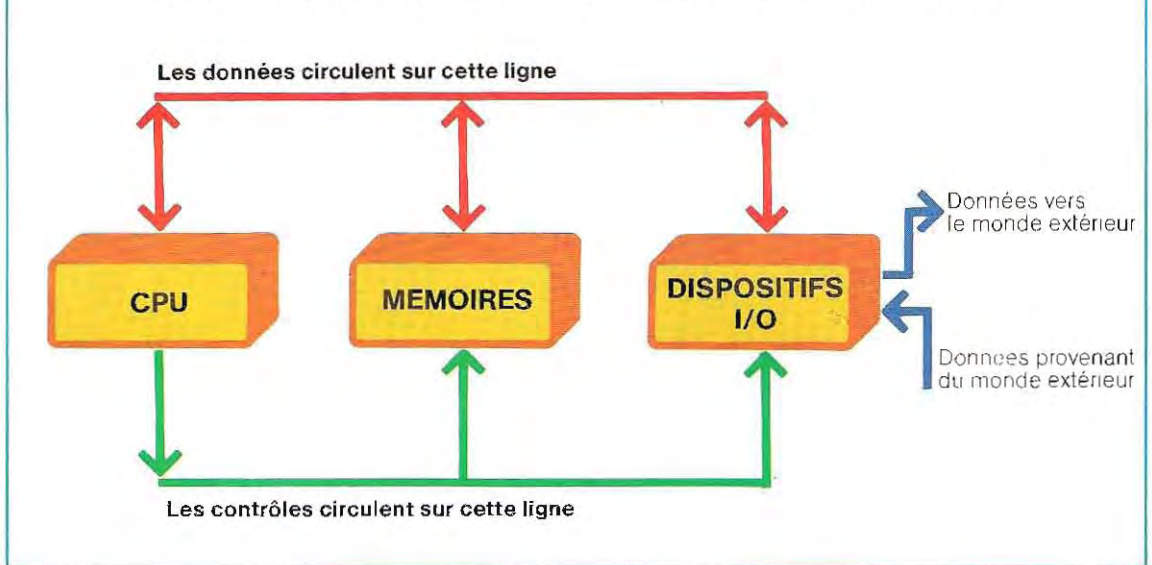
La plaque photographique est un dispositif analogique dans la mesure où elle produit une donnée (photo) proportionnellement au signal (lumière).

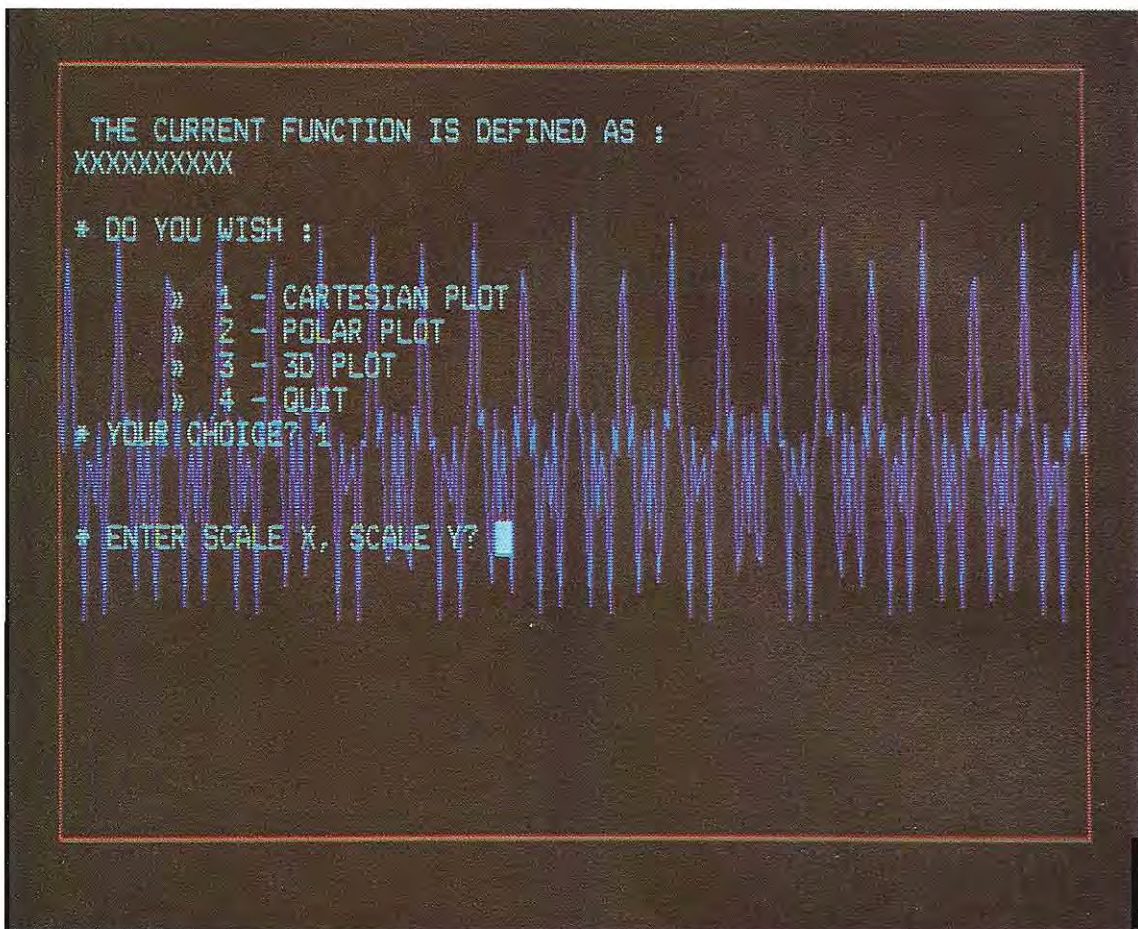
Un interrupteur est un dispositif numérique

qui n'a que deux possibilités : ouvert ou fermé. Un poste radio, un téléviseur, une installation stéréo sont des appareils analogiques qui fournissent des images et des sons proportionnels aux signaux reçus.

**Un ordinateur numérique est une machine capable d'exécuter des fonctions logiques à travers une série d'états :  
ON/OFF  
(Présence/Absence)**

### STRUCTURE PRINCIPALE D'UN SYSTEME A MICRO-ORDINATEUR





**Type d'affichage écran (display) représentant le menu d'un programme particulier permettant l'affichage graphique d'une fonction choisie.**

Un ordinateur peut exécuter les fonctions suivantes :

- Opérations arithmétiques entre deux données
- Comparaison entre deux données et choix consécutifs
- Transfert et mémorisation des données.

Bien que le nombre de ces fonctions principales soit très restreint, on peut tout de même, grâce à certaines combinaisons, effectuer n'importe quel calcul.

Le schéma de la page 34 (en bas) résume la structure d'un ordinateur, qui regroupe :

- Un dispositif capable d'exécuter les opérations arithmétiques et les comparaisons, c'est le CPU (unité centrale de traitement, ou UC).
- Un (ou plusieurs) dispositif capable de mémoriser les données, c'est la mémoire.

- Un dispositif capable de recevoir et de faire entrer les informations, c'est l'entrée (Input) et de les sortir pour les communiquer à l'homme, c'est la sortie (Output), le tout désigné par I/O, ou E/S en français.

L'unité centrale (CPU) assure l'exécution des calculs et envoie les ordres nécessaires au fonctionnement des autres unités.

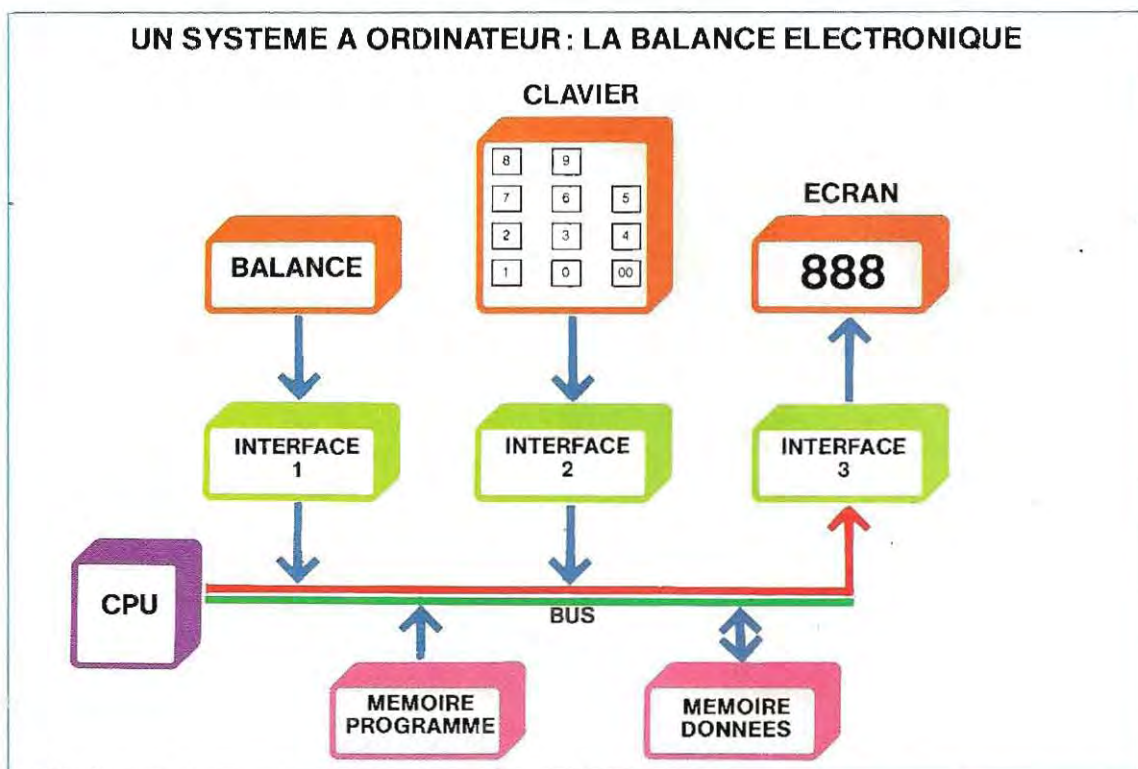
La mémoire est chargée de recevoir et de stocker les données, puis de les restituer quand on les lui demande.

Les dispositifs d'entrée et de sortie transforment les données pour les rendre compréhensibles à la machine et à l'homme.

Reportons-nous au schéma de la page 36 illustrant le cas de la balance électronique qu'utilisent beaucoup de commerçants.

Pour simplifier, la ligne de transmission des données et celle du contrôle sont représentées par une liaison unique, le Bus.

## UN SYSTEME A ORDINATEUR : LA BALANCE ELECTRONIQUE



Les fonctions que doit exécuter le système sont :

**A / Pesée d'un objet et saisie par la machine de la valeur de ce poids.**

La pesée est assurée par la balance. Cette donnée est transformée en information numérique dans l'interface 1. L'interface 1 est un dispositif d'entrée (Input).

**B / Saisie du prix (à l'unité) de l'objet.**

La fonction est remplie par le clavier et l'interface 2. L'interface 2 est un autre dispositif d'entrée.

**C / Calcul du prix total.**

La fonction est exécutée à l'intérieur du CPU (ou UC).

**D / Présentation du résultat.**

La fonction est effectuée par l'ensemble interface 3 et display\*. L'interface 3 est un dispositif de sortie (Output).

Dans le schéma, on voit deux autres blocs :

■ **Mémoire Programme**

■ **Mémoire Données**

Voyons maintenant pourquoi il faut utiliser deux mémoires.

La fonction de la Mémoire Programme est évidente. Cette dernière contient une série d'informations qui, correctement interprétées, signalent au CPU la séquence (A-B-C-D) de fonctions à mettre en action, et qui constituent donc le programme. La Mémoire Données permet d'emmagasiner les données au moment de leur entrée et de les restituer quand le CPU le demande. La séquence des opérations est la suivante :

- 1 / La balance signale qu'un objet est posé sur son plateau.
- 2 / Le CPU demande à la Mémoire Programme l'opération à accomplir.
- 3 / La Mémoire Programme signale qu'elle reçoit la donnée (poids) et qu'elle la transmet à la Mémoire Données.
- 4 / Une fois le transfert accompli, le CPU interroge la Mémoire Programme.
- 5 / La Mémoire Programme signale de mettre en œuvre le clavier et d'acquérir la donnée (prix).
- 6 / Le CPU actionne l'interface 2 et saisit la donnée du clavier.
- 7 / Le CPU interroge la Mémoire Programme.
- 8 / La Mémoire Programme répond qu'il faut prélever la donnée acquise au stade 3 et la multiplier par celle reçue au stade 5.
- 9 / Pour finir, après un nouveau cycle d'inter-

\* Par le terme display, ou écran (ou visuel), on désigne le dispositif de visualisation des données. Nous reviendrons sur ce sujet par la suite.

rogations du CPU à la Mémoire Programme, le résultat de l'étape 8 est envoyé à l'interface 3 qui le fait apparaître sur l'écran de la balance.

La suite d'opérations qui vient d'être décrite, une fois traduite dans un langage accessible à la machine, constitue un **programme**.

### Où et comment utiliser un ordinateur

L'emploi d'un ordinateur est lié à une condition essentielle :

#### IL FAUT POUVOIR LUI FOURNIR

#### TOUTES LES DONNEES NECESSAIRES

Cela n'est pas aussi évident que l'on peut croire. On entend parfois des questions étranges de la part de ceux qui connaissent mal l'ordinateur et qui lui attribuent des capacités tout à fait exagérées.

Il faut se rappeler en effet qu'un ordinateur ne peut traiter des informations (données) que si ces dernières lui ont d'abord été fournies.

Le second point à examiner est :

#### QUAND L'ORDINATEUR SERT-IL ?

On ne peut pas dire, a priori, que l'acquisition d'un ordinateur soit toujours une bonne opération financière. En général, pour qu'une telle machine soit rentable, il faut qu'au moins une partie des conditions suivantes soit réunie.

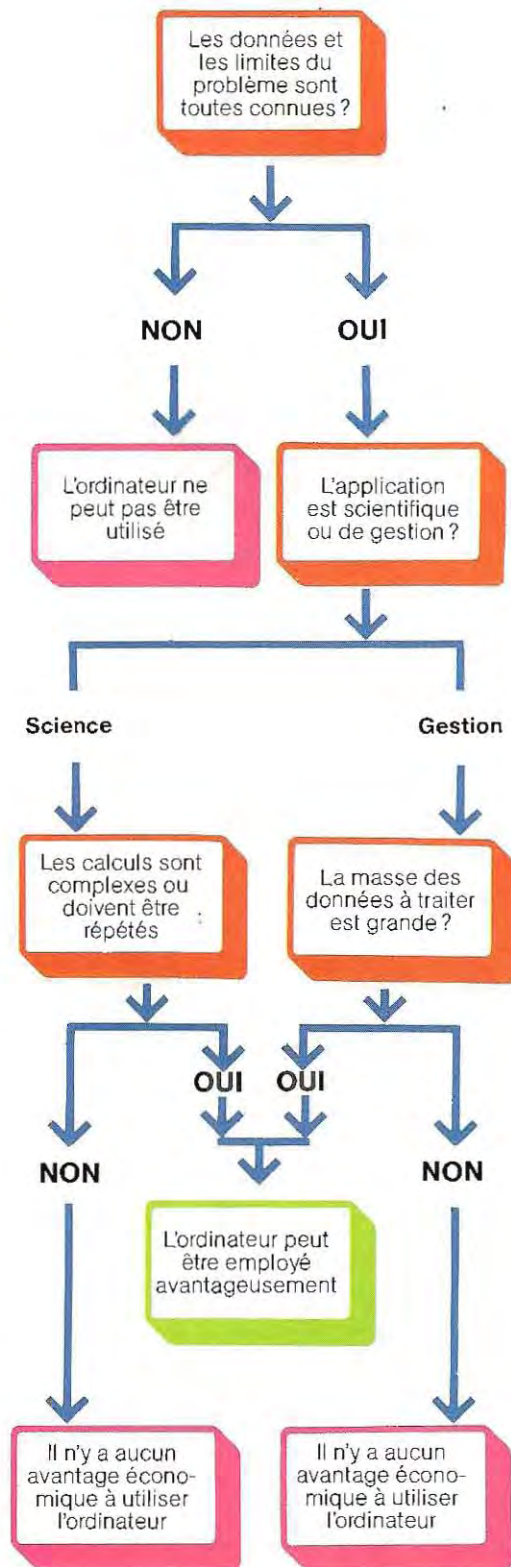
- 1 / On connaît bien toutes les données.
- 2 / On aura à traiter une grande quantité de données de façon répétitive.
- 3 / On aura une grande quantité de calculs à faire.
- 4 / On aura à contrôler constamment la même chose.

Le point 1 est la condition de base.

L'ordinateur ne peut pas donner plus que ce qui lui a été fourni. Il ne sortira pas de résultats si le programme ne sait pas de quoi il s'agit. Le point 2 indique dans quelles conditions un ordinateur est utile. Il faut se rappeler que si l'on installe ce type d'équipement, la dépense initiale sera suivie d'autres dépenses dues au développement des programmes (logiciel) qui imposent une mise de fonds importante. Ces dépenses ne seront amorties que si elles sont compensées par un fort allègement du travail humain, ou par l'acquisition d'un grand nombre d'informations permettant l'amélioration des décisions de gestion.

Le point 3 concerne les problèmes scientifi-

## ANALYSE DU PROBLEME



ques en particulier. Très souvent, la mise au point d'un projet nécessite des calculs répétés pour obtenir le meilleur résultat possible. Le dernier point (4) concerne l'emploi de la machine pour contrôler un processus quelconque. Dans ce cas, l'acquisition d'un ordinateur est presque toujours un investissement rentable.

Le schéma de la page 37 résume bien l'ensemble des conditions qu'il faudra vérifier avant de décider l'achat d'un ordinateur.

### Classement des ordinateurs

Jusqu'à maintenant nous avons utilisé le terme d'ordinateur pour désigner des machines qui peuvent, en réalité, être très différentes. Essayons donc de les classer selon les performances dont elles sont capables. Elles peuvent avoir les caractéristiques essentielles suivantes :

**A / Rapidité de calcul.**

**B / Rapidité de transfert des données.**

**C / Capacité plus ou moins grande de mémorisation des données.**

**D / Nombre plus ou moins grand d'utilisateurs qui peuvent s'en servir en même temps.**

La valeur de chacun de ces points permet de définir les grandes lignes du champ d'appli-

cation de la machine, et donc la catégorie à laquelle elle appartient. On distingue :

- **GROS ORDINATEURS**
- **MINI-ORDINATEURS**
- **MICRO-ORDINATEURS et ORDINATEURS PERSONNELS**

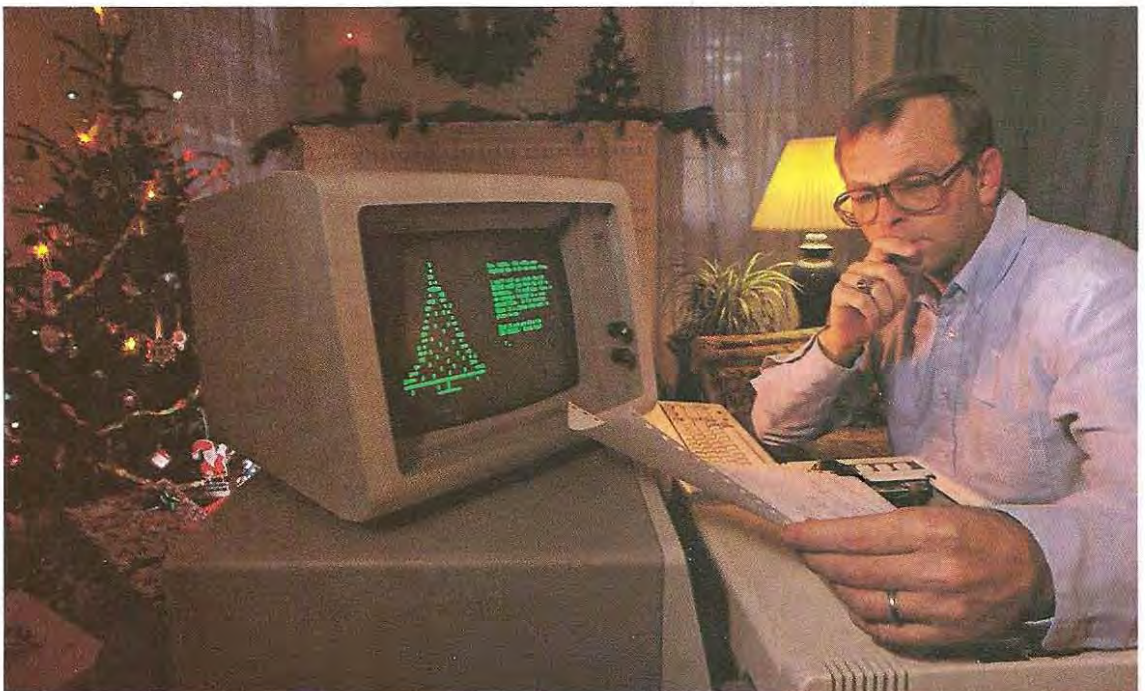
Le GROS ORDINATEUR est, comme son nom l'indique, un ordinateur de grandes dimensions, et qui peut être utilisé par plusieurs personnes en même temps.

Il faut une équipe d'informaticiens pour le faire fonctionner. Pour être performant il doit avoir d'énormes capacités de mémoire et une grande rapidité d'exécution. Ce type de machine est réservé à la gestion et à l'administration des grandes industries ou à la recherche dans d'importantes unités scientifiques.

C'est sur les gros ordinateurs qu'on calcule les orbites des satellites ou les trajectoires des fusées ; qu'on effectue les calculs et les vérifications d'ingénierie civile pour les ensembles complexes, et, enfin, qu'on élabore les projets de circuits électroniques.

Dans le domaine de la gestion, ils sont utilisés pour des archives de grand volume, comme celles des tribunaux. Ils sont tout à fait adaptés, aussi, à la gestion des registres de l'état-civil et du fisc, des stocks de marchandises ou des cycles de production.

### L'ordinateur personnel à la maison.



Grazia Neri/Contact





Ericsson

**Reliés à un gros ordinateur, les terminaux permettent à tous les utilisateurs d'accéder à la banque de données de l'entreprise.**

Il faut également mentionner d'autres emplois très spécifiques, comme la simulation d'une guerre. Ces ordinateurs sont en effet capables d'étudier toutes les phases d'une guerre hypothétique, ce qui permet de mettre au point un certain nombre de stratégies.

Le MINI-ORDINATEUR est au contraire un appareil de prix relativement limité, plus facile à utiliser et qui, en général, sert à plusieurs personnes en même temps. Les structures du mini-ordinateur sont assez extensibles dans la mesure où on peut leur ajouter des unités additionnelles qui en augmentent les capacités.

La tendance des fabricants est de fournir, pour commencer, des machines dont les possibilités sont limitées aux besoins immédiats de leurs clients, et d'y apporter ensuite les extensions qui leur sont devenues nécessaires. La puissance d'un système en termes de capacité de mémoire, de dispositifs d'entrée et de sortie s'appelle **configuration**.

Le champ d'utilisation de ces machines est très varié et dépend de leur configuration. L'utilisateur du mini-ordinateur n'a pas des besoins assez grands pour un gros ordinateur mais ne serait pas satisfait par les services d'un micro-ordinateur, insuffisant pour lui.

Le MICRO-ORDINATEUR et l'ORDINATEUR PERSONNEL sont en principe des machines destinées à un seul utilisateur, et dont l'emploi

est très simple. Comme nous le verrons mieux par la suite, la frontière entre micro et ordinateur personnel est parfois assez vague, d'autant plus que de nouvelles technologies permettent d'augmenter la puissance des micro-ordinateurs.

Si l'on a quelquefois tendance à faire des micro-ordinateurs et des ordinateurs personnels deux types distincts, c'est plutôt en raison de la différence de leurs capacités de mémorisation des données et des langages utilisés (nombre plus ou moins grand et variété des instructions disponibles), alors que toutes leurs autres fonctions sont semblables. En revanche, leur utilisation simplifiée et leur prix nettement plus bas permettent à beaucoup de personnes de les acquérir pour des usages très divers.

Voici la liste des principales utilisations de ces petits ordinateurs :

- Préparation des enveloppes de paie dans les petites entreprises
- Comptabilité du petit commerce ou du grand magasin
- Facturation
- Gestion d'immeubles
- Calculs d'ingénierie civile
- Gestion des petites bibliothèques
- Gestion des fichiers d'adresses
- Analyses financières
- Gestion de stocks.

## L'ordinateur à la maison

L'entrée dans les foyers de ce que l'on nomme terminal domestique ou micro-ordinateur familial, caractérise l'une des applications les plus révolutionnaires de la cybernétique moderne.

Les perfectionnements de la technologie et l'abaissement du prix des machines permettent désormais aux particuliers d'acquérir des micro-ordinateurs et de bénéficier d'un type tout nouveau d'informations et de services à domicile.

On peut déjà entrevoir les opérations bancaires faites avec le terminal domestique; de nombreux projets de ce type sont déjà en phase d'expérimentation aux Etats-Unis. Il est facile également d'envisager le «shop at home», c'est-à-dire «faites vos courses à domicile». Les services d'information apparaissent très diversifiés et certains, déjà opérationnels aux Etats-Unis, puisent leurs données dans plus d'un millier de bases de données. Parmi ces services nommons ceux qui fournissent des informations professionnelles de caractère médical ou agricole, pour ne citer que ceux-là.

Toutefois, les secteurs d'application qui intéresseront le plus le lecteur européen sont ceux de type interactif. Selon ce mode, l'utilisateur peut obtenir chez lui des informations qui le concernent particulièrement: sur la sécurité par exemple, aussi bien dans le cas d'une urgence médicale que pour la protection contre intrus et voleurs, ou encore contre les incendies et les inondations. L'utilisateur peut également participer en «temps réel» à des sondages d'opinion: on diffuse le résultat du sondage au fur et à mesure qu'on recueille les réponses des personnes interrogées.

Parmi les chaînes américaines privées, celles qui ont été créées par les groupes ethniques, les petites communautés ou les associations religieuses jouissent d'un grand succès dans ce type de transmission. Le terminal domestique est très utile en mode interactif surtout pour les associations de bienfaisance qui cherchent à collecter des fonds en direct chez le particulier.

D'autres applications, sans doute moins originales, mais assurées de réussir, sont les études à domicile et la version moderne de l'autodidacte ou du rat de bibliothèque, qui pourra désormais rester chez lui.

**L'ordinateur familial vous aide à tenir votre compte bancaire, à établir la liste des achats de la semaine, et peut même vous donner des conseils de santé.**



Dominique Fradin



Thermorégulation des pièces. Organisation et gestion des services de la maison

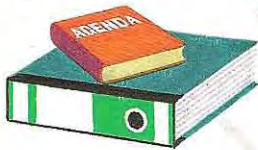


Liaison avec des banques de données



Transmission et réception de documents

Contrôle de la fermeture des portes et des fenêtres. Alarme incendie et fuite de gaz.



Agenda. Archives de la famille. Comptabilité et correspondance



Paiement et transfert de fonds



Logiciel de formation



Achats au supermarché



Jeux



Annuaire téléphoniques



Faire de la musique et l'apprendre



Gérer la bibliothèque et les informations culturelles

Poste de travail à domicile en liaison avec le bureau





### L'ordinateur à la maison permet de mieux suivre la vie de tous les membres de votre famille.

*Citons enfin les programmes de loisirs comme les émissions sportives, que l'utilisateur sélectionnera à l'avance.*

*Tous ces services, que l'on peut compter désormais au nombre de nos futures habitudes de vie, nous laissent imaginer à quel point cette agression cybernétique sera plus contraignante encore que celle que nous a imposé la télévision. Cette dernière implique en effet un programme rigide égal pour tous, parfois abêtissant, et concentré en des horaires souvent paralysants pour la vie sociale. Le micro-ordinateur, au contraire, est, à première vue, l'instrument du choix, de la participation, de la personnalisation et de la liberté.*

*Pourtant, cela n'empêche pas d'entrevoir les querelles de famille qui risquent de naître autour de cet instrument. En effet, c'est un peu comme si on concentrait dans un seul appareil un téléphone et une encyclopédie, un téléviseur et un projecteur de cinéma, un agenda et la liste des courses, le livre de comptes et un échiquier... Il ne faudra donc pas s'étonner qu'un seul clavier et qu'un seul écran soient vite insuffisants.*

*Mais la réponse naturelle sera là encore dans les progrès de la technologie: à très court terme il sera possible à chaque foyer d'acquérir plusieurs claviers et plusieurs écrans, permettant ainsi à tous les membres de la famille de bénéficier de multiprogrammes.*

*Ce sera l'apparition de petits réseaux d'ordinateurs familiaux.*

*Ces constatations, ces réflexions, partent des données observées par quelqu'un qui a vécu la réalité américaine. Cependant, il n'est pas inutile de présenter le problème au lecteur européen tant il est vrai que l'élévation de son niveau de vie aussi bien que les perfectionnements de la technologie mettront très prochainement à sa portée une situation semblable. L'informatique pénètre dans tous les pays industrialisés de la même manière: proportionnellement aux revenus des habitants. Si cette situation nouvelle ne se matérialise pas immédiatement chez nous par l'acquisition massive de ces micro-ordinateurs, cela ne veut pas dire que l'évolution ne se fait pas dans le même sens. Elle est moins rapide en Europe qu'aux Etats-Unis, mais, ce qui sera un phénomène de masse aux Etats-Unis vers 1985, le sera en Europe en 1990. Néanmoins, nous pourrons bientôt constater l'impact de ces expériences sur une partie limitée de la population. Ce résultat sera-t-il à la gloire des marchands d'ordinateurs ou à la gloire de l'évolution culturelle? C'est là un des aspects du défi que comporte aussi la révolution cybernétique.*

(D'après L'INFORMATICA A DOMICILIO de Mauro Langfelder, Feltrinelli 1983.)

Les plus petits de ces modèles entrent de plus en plus dans les foyers. Ils peuvent y servir de répertoire d'adresses, ou aider à étudier et à gérer le budget familial. Certains modèles coûtent déjà moins cher qu'un téléviseur et ils font désormais partie de nos appareils électroménagers.

## Réseaux d'ordinateurs

Tous les ordinateurs doivent communiquer avec leur utilisateur. Ils disposent donc de circuits capables de recevoir et de transmettre des données. Le plus souvent, la transmission aboutit à une imprimante ou à un écran et provient d'un clavier. Mais rien n'empêche d'y ajouter un autre ordinateur.

C'est ce qu'on appelle un réseau d'ordinateurs. Il permet essentiellement la transmission et l'utilisation d'un plus grand nombre d'informations. Or, comme nous l'avons déjà vu, l'intérêt principal d'un ordinateur est la gestion de grandes quantités de données. Mais plus leur nombre augmente, plus il devient difficile de les gérer ensemble. Le réseau d'ordinateurs permet justement de centraliser ces données tout en en rendant l'accès possible à plusieurs utilisateurs.

La structure type d'un réseau d'ordinateurs est représentée dans le schéma ci-dessous.

Ce schéma met en évidence ses fonctions principales :

- Le satellite (\*) 1 est directement relié à l'ordinateur principal, c'est ce que l'on appelle un réseau étoilé.
- Les satellites 2 et 3, reliés de la même façon étoilée, ont un concentrateur.

Ce système est adopté pour faciliter la transmission des données sur de longues distances.

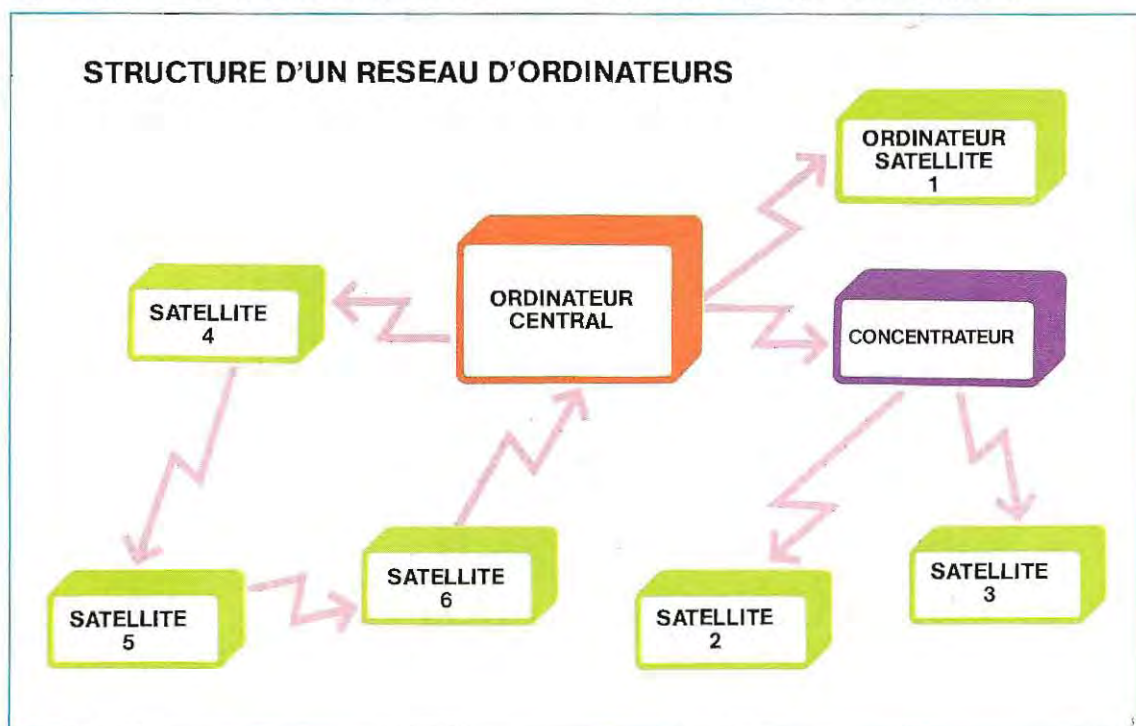
Dans notre exemple, les deux satellites (2 et 3) et le concentrateur sont dans un site très distant de l'ordinateur principal.

Grâce au concentrateur on peut réunir un certain nombre de lignes locales en une seule ligne étoilée. Le concentrateur a précisément la tâche de diriger l'échange des données sur la ligne éloignée et de les répartir entre les satellites. De cette manière, la ligne de liaison à longue distance est unique. Ensuite elle se divise en un certain nombre de lignes locales (une par satellite).

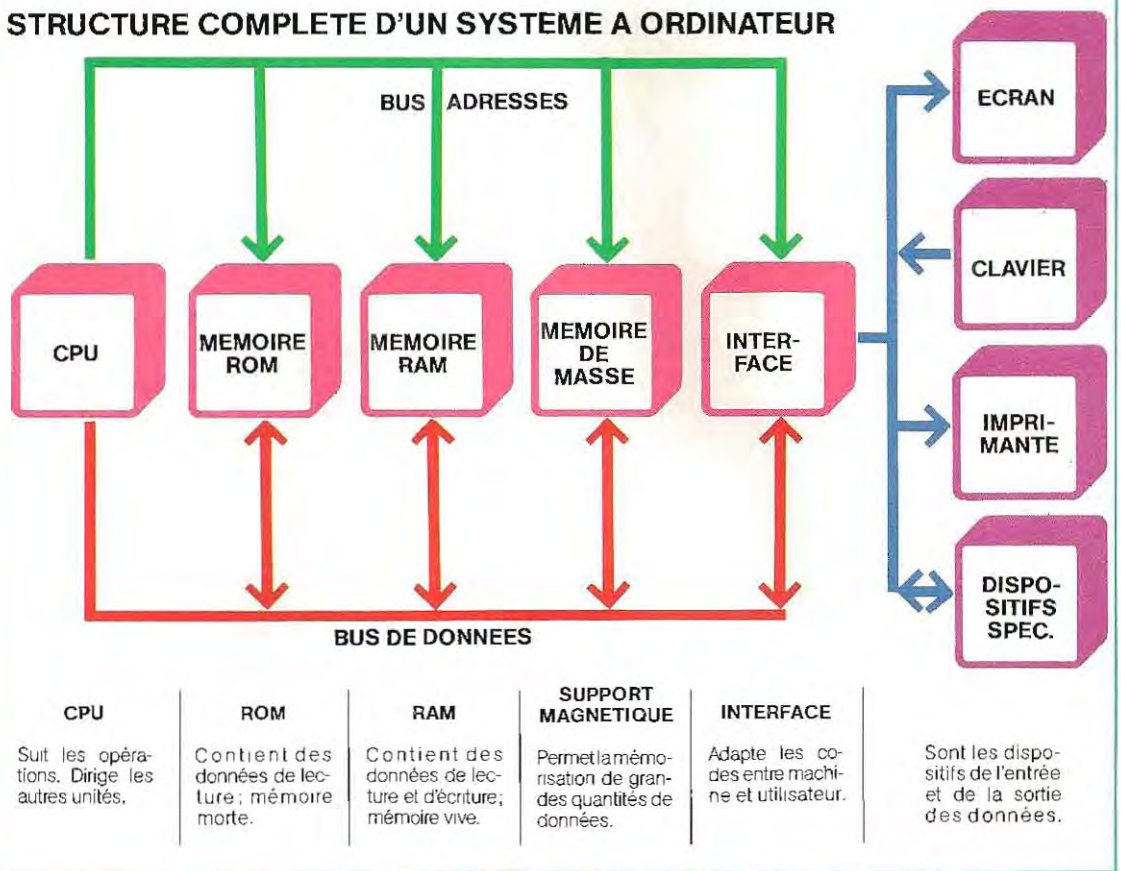
Toujours dans le même schéma, les satellites 4, 5 et 6 constituent un réseau maillé.

Dans ce système on se passe du concentrateur, mais ce sera la panne si l'un des satellites ou l'une des lignes tombe en panne.

\* Par satellite on entend un ordinateur ou n'importe quel autre dispositif qui exécute des fonctions, en partie ou totalement dirigées par une autre machine.



## STRUCTURE COMPLETE D'UN SYSTEME A ORDINATEUR



Le schéma de la page 43 donne un aperçu des liaisons en réseaux. On distingue différents *modes de liaisons*: les liaisons télégraphiques (télex), les lignes téléphoniques (réseau commuté), les lignes spécialisées (deux ou quatre fils), les réseaux à valeur ajoutée (Transpac, Euronet) et les liaisons par satellite (Télécom 1).

Il faut distinguer également les *modes de transmissions*, qui peuvent être parallèles ou en série, synchrones ou asynchrones; ou encore ce que l'on nomme half ou full duplex. Citons enfin les *équipements* disponibles: modems et convertisseurs, multiplexeurs et concentrateurs, ainsi que commutateurs de circuits, de paquets, de messages, etc.

La télématique française est très avancée grâce, notamment, à la conversion progressive du réseau de télécommunications en technologique numérique et à son expérience des réseaux publics de commutation par paquets (Transpac), sujets sur lesquels nous aurons l'occasion de revenir plus tard dans cet ouvrage.

**Un dispositif numérique reconnaît seulement deux états:**

**ON et OFF. (Allumé et Eteint).**

**L'ordinateur numérique peut exécuter quatre fonctions principales:**

**OPÉRATIONS ARITHMETIQUES  
COMPARAISONS**

**TRANSFERT DES DONNEES  
MEMORISATION DES DONNEES.**

**Un programme est une série logique de fonctions élémentaires.**

**L'emploi de l'ordinateur est possible seulement si les termes du problème sont cernés avec précision.**

Le schéma en haut de la page résume la structure complète d'un système à ordinateur. Les différentes fonctions de chacun des composants seront analysées par la suite.

# Les systèmes de numération

Dans ce chapitre nous exposerons les différentes manières de représenter nos symboles (chiffres et lettres) sous une forme accessible à l'ordinateur.

Nous donnerons à la fin du chapitre les méthodes de calculs à utiliser avec ces symboliques. Le sujet peut paraître un peu rebutant parce qu'il oblige le lecteur à oublier sa logique personnelle faite de nuances, de réflexions et d'intuitions, pour entrer dans la logique de la machine qui exclut toute forme d'interprétation.

L'aspect conceptuel de cette question devra retenir toute votre attention, alors que l'aspect technique du développement d'un calcul dans sa version adaptée à l'ordinateur est beaucoup plus facile à assimiler, même si les exemples donnés dans ces pages paraissent compliqués.

## Les impulsions

N'importe quel dispositif numérique reconnaît seulement deux états :

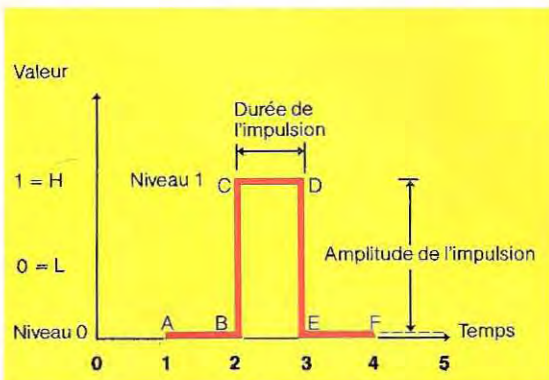
**ON = Présence de signal**

**OFF = Absence de signal**

La condition OFF = 0 est le repère et prend le nom de NIVEAU 0.

La condition ON = 1 est représentée proportionnellement à la valeur du signal et prend le nom de NIVEAU 1.

Ces deux conditions peuvent être représentées graphiquement de la façon suivante :



Nous commençons à observer le phénomène à l'instant A. Entre A et B (disons : 1

seconde, soit de la seconde 1 à la seconde 2), le niveau reste bas = 0 (le niveau 0 peut aussi être indiqué par la lettre L).

Entre B et C (très court) on passe du niveau 0 au niveau 1 (le niveau 1 est en général indiqué par la lettre H).

Le niveau H se maintient pour le temps C-D. Dans l'intervalle (très court) D-E on passe du niveau H au niveau L.

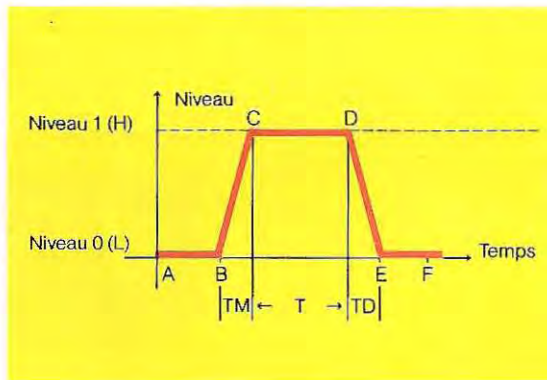
L'état de niveau 0 peut durer indéfiniment. Le cycle :

- Niveau L
- Niveau H
- Niveau L

prend le nom d'**impulsion**.

En fait, la forme d'une impulsion ne peut pas être celle que nous venons de lui donner.

Sur la figure, on ne tient pas compte du temps nécessaire au passage du niveau 0 (L) au niveau 1 (H). De même, pour le temps nécessaire au passage inverse H-L (points D et E). Car aucun circuit électronique ne peut passer d'un état à l'autre en un temps nul. Donc, la course réelle d'une impulsion est celle-ci :



Pour faciliter la représentation, les temps ont été augmentés et on a fait coïncider le point A avec le temps 0.

Le temps nécessaire au passage du niveau 0 au niveau 1 (points B et C) prend le nom de **temps de montée** (il est indiqué sur la figure par TM).

La durée de l'impulsion, au niveau 1, se réduit au segment C-D (indiqué par T).

Le temps de passage du niveau 1 au niveau 0 (points D et E) prend le nom de **temps de descente** (indiqué par TD sur la figure).

La ligne B-C, qui matérialise le temps de montée (TM), est le **front de montée**.

De même, le segment D-E, qui matérialise le temps de descente, est le **front de descente**.

En résumé, les principaux paramètres d'une impulsion sont :

- **Durée = Temps T (segment C-D)**
- **Temps de montée = Temps TM (seg. B-C = Front de montée)**
- **Temps de descente = Temps TD (seg. D-E = Front de descente)**

### Commutation

Pour ses calculs, un ordinateur n'utilise que les deux symboles ON et OFF.

On indique l'état ON par 1 (ou avec la lettre H), et l'état OFF par 0 (ou la lettre L).

Supposons maintenant un dispositif numérique en mesure de nous montrer son état (par exemple avec un témoin lumineux) et doté d'une entrée, comme dans le schéma A. Imaginons qu'au début l'état de sortie soit 0 (lampe éteinte), avec l'entrée en 0 également. En mettant l'entrée en 1, on changera l'état du dispositif numérique et la sortie sera portée à 1 (lampe allumée).

L'observateur dira qu'une impulsion a été produite. Ce changement d'état porte le nom de **commutation**.

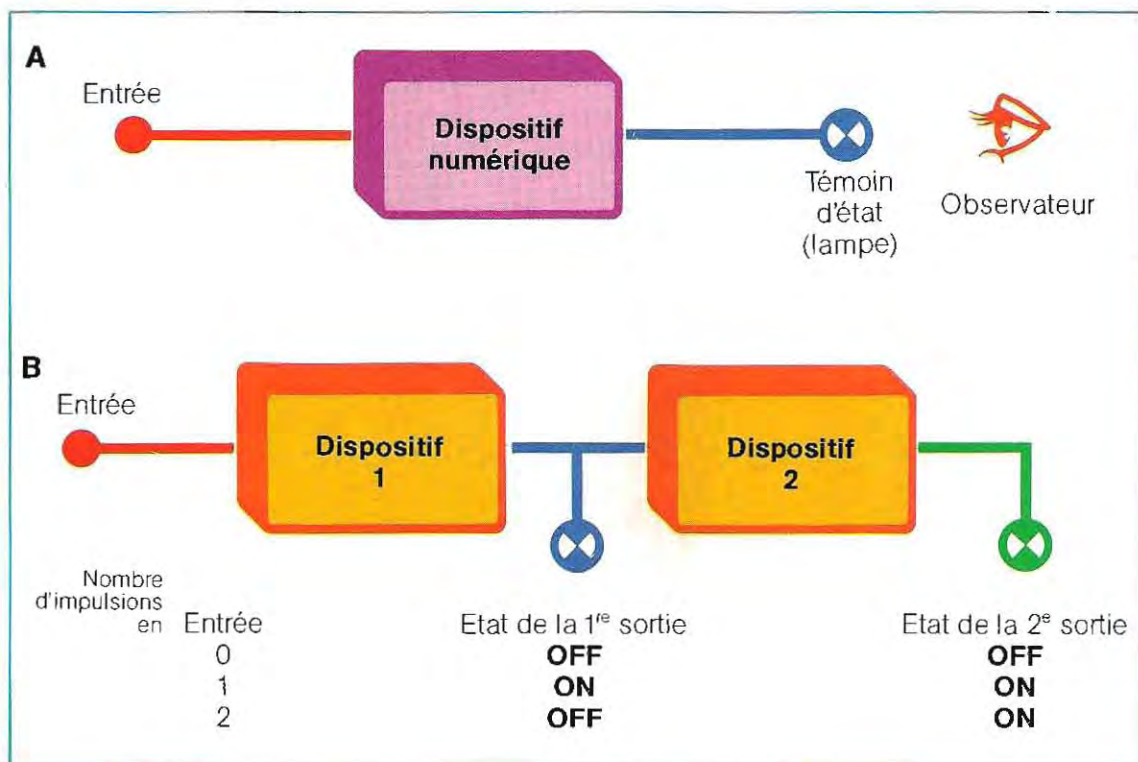
Le dispositif a commuté une fois.

Si l'on remet l'entrée à 0, puis si on la porte de nouveau à 1, le dispositif aura une nouvelle commutation et la sortie se portera de 1 à 0. Puisque l'observateur se souvient d'avoir vu d'abord l'état 1, en voyant l'état 0 il dira qu'une nouvelle commutation s'est produite. Le dispositif a donc commuté deux fois, c'est-à-dire qu'il a compté 2.

Supposons maintenant que l'observateur commence son expérience seulement au moment de la seconde impulsion d'entrée.

Dans ce cas il verra la sortie passer de 1 à 0 et il ne pourra pas savoir s'il s'agit d'une seule commutation ou de deux. Cette incertitude disparaît si l'on dispose à la sortie du système, en plus du témoin lumineux, un second dispositif identique, comme on peut le voir sur le schéma B.

La sortie du dispositif 1 provoque la commutation du dispositif 2, il devient donc possible de déterminer quel a été le nombre d'impulsions en entrée (0, 1, 2), puisque le second dispositif mémorise et met en évidence les changements qui affectent le premier.



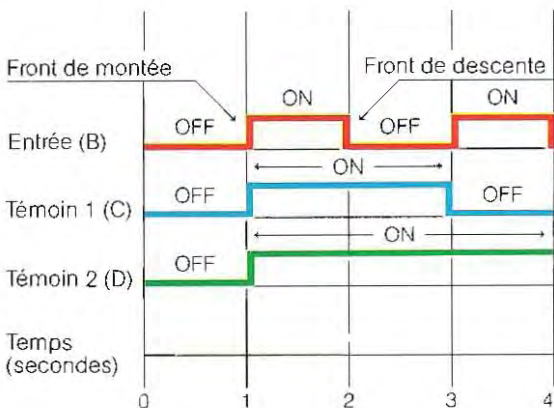


Les états des Entrée, Témoin 1, Témoin 2, peuvent être résumés en une table.

A	B	C	D
Nombre d'impulsions	Entrée	Témoin 1	Témoin 2
0	0	0	0
1	1	1	1
2	1	0	1

Ce tableau illustre le fonctionnement du système du point de vue logique et se nomme **table de vérité**. Chaque circuit numérique a une table plus ou moins compliquée selon les fonctions qu'il doit exécuter.

Supposons qu'à l'entrée les impulsions arrivent au rythme d'une toutes les deux secondes, et construisons le diagramme suivant:



- Le témoin 2 (D) s'éteint au second allumage du témoin 1 (C); nous avons donc réalisé un compteur par 2.
- Le dispositif qui produit les impulsions équidistantes dans le temps, porte le nom d'**horloge** (clock).

Notons que, dans un circuit numérique électronique, **LE CHANGEMENT D'ETAT INTERVIENT QUAND IL SE PRODUIT UN PASSAGE OFF-ON A L'ENTREE.**

### Les bases de numération

Il y a différents systèmes de numération, mais le plus adapté à notre logique est le système positionnel.

Dans ce système, chaque chiffre a un poids ou valeur, dépendant de sa position.

En écrivant le numéro 5, nous entendons 5 unités; si nous changeons la position du symbole 5 nous pouvons entendre 5 dizaines (50), 5 centaines (500), etc. Le symbole 5 prend donc une

valeur différente selon sa position. Chaque position a une valeur correspondant à la valeur de base du symbole (0, 1, 2, ... 5, ... 9) multiplié par 10 un nombre de fois correspondant à sa position, comme nous le voyons sur cette table:

	Centaines	Dizaines	Unités
Poids	100	10	1
Exposant	$5 \times 10^2$	$5 \times 10^1$	$5 \times 10^0$
Valeur position.	$5 \times 100$	$5 \times 10$	$5 \times 1$
Valeur	500	50	5

A chaque changement de position le nombre est multiplié par 10. La valeur 10 est la base de notre système de numération, qui à cause de cela se nomme **système positionnel en base 10** (numération arabe).

La numération romaine est un exemple de système non positionnel.

Dans ce système, la valeur 5 est représentée par le symbole V, tandis que la valeur 50 a le symbole L. Pour passer de 5 à 50, il ne suffit pas de changer la position du symbole 5 (V), mais il faut en introduire un nouveau (L).

Ce type de numération, peu pratique à interpréter et peu adapté à l'exécution des calculs, a été abandonné. A sa place, on a universellement adopté le système positionnel.

Le système positionnel le plus courant est le système décimal: les symboles sont dix et la base est 10.

Mais dans les ordinateurs on ne peut pas utiliser dix symboles différents. Comme nous l'avons déjà dit, la machine ne sait reconnaître que deux conditions: ON (niveau haut) et OFF (niveau bas). Elle ne dispose donc que de deux symboles 1 (ON) et 0 (OFF).

C'est pour cette raison qu'il faut utiliser un système de numération qui prévoit l'usage de seuls symboles 1 et 0, c'est pour cela que ce système s'appelle le **système binaire**. Ce type de numération, contrairement à ce que l'on pourrait penser, n'est pas né avec l'ordinateur. Il était déjà utilisé en Chine vers le XII<sup>e</sup> siècle. Le système binaire est, lui aussi, un système positionnel car chacun des deux symboles possibles a une valeur dépendant de sa position.

## A l'école avec l'ordinateur

On peut considérer l'ordinateur comme une machine extraordinaire pour enseigner, mais pour enseigner quoi ? N'importe quoi : pas seulement l'informatique et toutes les matières scientifiques, comme les mathématiques, la chimie, la biologie, mais les matières qui, traditionnellement, n'appartiennent pas au groupe scientifique, comme les langues et la musique. Les enfants, d'ailleurs, semblent avoir une prédisposition naturelle à l'usage d'un ordinateur personnel. Prenons l'exemple d'un programme très simple pour enfants de l'école élémentaire, le « programme pour l'enseignement des soustractions ». Dans le dialogue qui suit, les élèves obtiennent une réaction immédiate à leurs réponses.

Ordinateur :  $3 - 2 = ?$

Enfant : 1

Ordinateur : Très bien.  $7 - 2 = ?$

Enfant : 5

Ordinateur : Formidable. Et  $23 - 16 ?$

Enfant : 6

Ordinateur : Oh, oh ! Tu veux essayer de nouveau ?

Enfant : 7

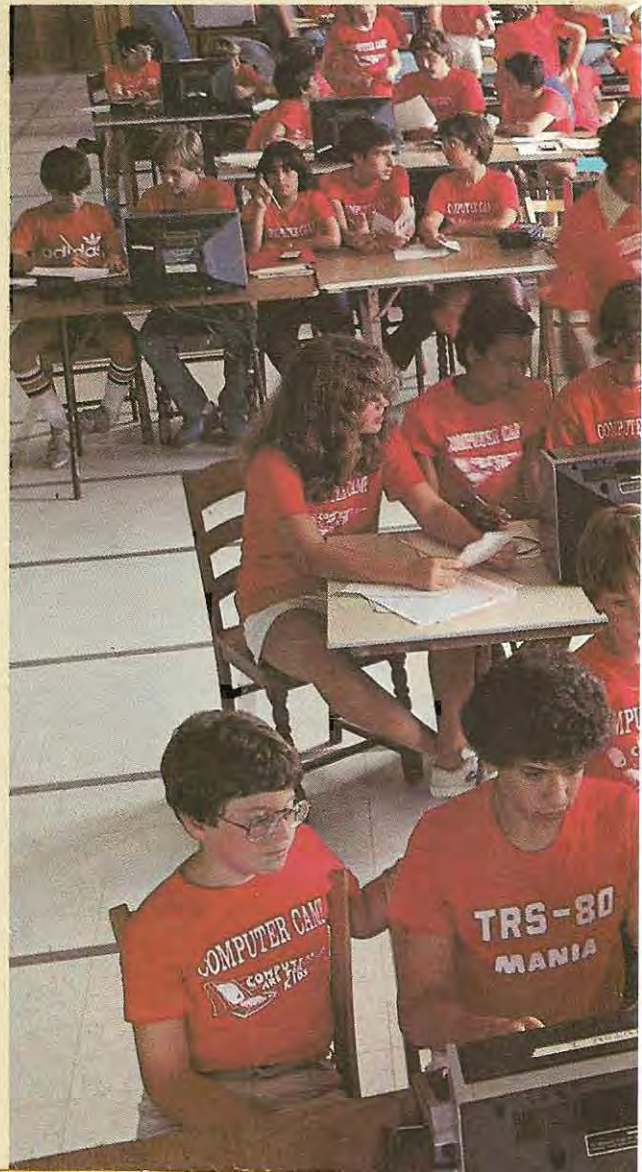
Ordinateur : Cette fois c'est bien. Et  $44 - 38 ?$

A chaque réponse exacte, l'élève est félicité, et il est arrêté si la réponse est fausse. Les expressions « Très bien », « Formidable », etc., utilisées dans l'exemple de la soustraction, ne sont que des propositions schématiques. Une réaction plus pittoresque, comme un visage souriant ou courroucé, ou des dessins animés accompagnés d'effets sonores, tels qu'une sonnerie de trompette pour les bonnes réponses, et un crissement de freins, ou le bruit d'un choc violent pour les réponses fausses, serait évidemment plus efficace et plus amusante, à condition cependant que les réactions à une réponse inexacte ne soient pas trop bouleversantes, sinon l'élève se découragera rapidement.

Le type de présentation choisie pour le programme peut varier à l'infini. Dans notre exemple, le programme propose des problèmes de plus en plus difficiles, dès qu'il est sûr que l'élève est capable de résoudre les plus faciles. C'est pourquoi le programme modèle passe rapidement au-delà des problèmes de soustractions à un chiffre, comme  $3 - 2$ ,

$7 - 2$ , qui sont faciles à résoudre. Les réactions de l'ordinateur ne sont pas exclusivement déterminées par deux bonnes réponses successives de l'élève. L'horloge contenue dans l'ordinateur peut, par exemple, avoir signalé aussi le temps pris pour répondre aux deux questions. En outre, les données de la leçon précédente peuvent avoir été conservées pour que la machine puisse rappeler à l'élève (qui l'aurait alors oublié) qu'il a déjà fait ce genre de problèmes. En tout cas, le calcul à deux chiffres plus difficile,  $(23 - 16)$  lui a été présenté ensuite, suivi de  $44 - 38$ , et ainsi de suite. A part des calculs de difficulté croissante, le programme peut aussi offrir à l'élève la possibilité de changer de sujet (et de passer par exemple à la lecture de poésies, à la composition de musique rock, ou à une partie de football), ou même de mettre fin à la leçon, si le temps d'une réponse à l'autre et la qualité de cette dernière indiquent que l'élève commence à se fatiguer ou à s'ennuyer.

Grazia Neri/E. Ferorelli



Bien qu'il soit élémentaire, cet exemple applique à l'enseignement de la soustraction une méthode très supérieure à celle des manuels classiques. L'élève disposant d'un livre ne peut résoudre les problèmes que dans l'ordre où ils lui sont présentés, qu'ils soient trop faciles ou trop difficiles. En outre, l'étudiant ne reçoit aucune indication sur l'exactitude de ses réponses. Si le corrigé du problème lui est fourni au bout de plusieurs jours, il risque d'avoir tout oublié de la question.

L'intérêt et le niveau d'apprentissage sont beaucoup plus élevés si la réponse est fournie au fur et à mesure. Cela est vrai aussi dans les lycées où, trop souvent, le professeur fait son cours sans tenir compte des questions posées par ses étudiants. Ceux-ci notent scrupuleusement ses paroles sans oublier la liste des devoirs à faire. Les étudiants attendent parfois plusieurs jours après la remise de leur travail et, au moment des corrections, ne sont plus toujours en mesure de comprendre

pourquoi ils ont commis des erreurs. On se rend compte à quel point il est important que l'élève puisse dialoguer davantage et c'est pourquoi l'enseignement assisté par l'ordinateur pourrait être très utilement adopté pour d'autres cours que des cours de mathématiques. C'est toute une pédagogie nouvelle et constructive qui voit le jour avec l'emploi des ordinateurs individuels dans les écoles. L'expérience se développe déjà, aux États-Unis et dans de nombreux pays d'Europe comme la France ou la Grande-Bretagne.

Dans ce domaine, l'ordinateur est le nouvel outil didactique qui s'intègre dans un autre rapport entre le professeur et l'élève.

(D'après THE ELECTRONIC COTTAGE. © Joseph Deken, 1981.)

**L'engouement des moins de 20 ans pour l'ordinateur favorisera autant l'implantation dans les écoles de l'enseignement assisté par ordinateur que la création de clubs d'amateurs.**



Nous allons à présent examiner en détail les mécanismes du système positionnel. Nous avons vu que dans la notation décimale habituelle, chaque nombre est représenté par une suite ordonnée des chiffres 0, 1, 2, 3... 9. La valeur d'un chiffre, le 5 par exemple, est différente selon la position qu'il occupe dans le nombre ou, en d'autres termes, selon la quantité de chiffres qui le suivent : c'est ainsi que 50 diffère de 500.

Ecrire un nombre en base 10 est une opération abstraite qui nous est dictée par l'habitude. En réalité nous utilisons les chiffres disponibles comme multiplicateurs des puissances successives de la base 10. Rappelons que les puissances d'une base sont les valeurs obtenues en calculant le produit d'autant de facteurs égaux à la base qu'indiqué par l'exposant placé à droite et un peu au-dessus. Par exemple :

$10^0 = 1^*$	puissance zéro (unité)
$10^1 = 10$	première puissance (dizaine)
$10^2 = 10 \times 10 = 100$	deuxième puissance (centaine)
$10^3 = 10 \times 10 \times 10 = 1000$	troisième puissance (millier)

Dans la numération décimale, chaque chiffre, en partant de la droite et en progressant vers la gauche, est un multiplicateur des puissances croissantes de la base 10 (on parle couramment d'unités, de dizaines, de centaines, de milliers, etc., mais ces termes perdent leur signification quand la base n'est plus 10). La procédure rigoureuse qui permet d'interpréter un nombre du système décimal et d'en obtenir la valeur est décrite par le tableau ci-dessous, qui se rapporte au nombre entier 1024 :

Colonne	d	c	b	a
Nombre (chiffres)	1	0	2	4
Puissance de la base	$10^3$	$10^2$	$10^1$	$10^0$
Poids des positions	1000	100	10	1
Valeur des chiffres (chiffre x poids)	1000	0	20	4
Valeur du nombre	$1000 + 0 + 20 + 4 = 1024$			

Si nous utilisons couramment la notation décimale c'est une pure coïncidence, attribuée par certains au fait que notre capacité de visualiser immédiatement un comptage est limitée à 10, c'est-à-dire au nombre de nos doigts. Une quantité peut être exprimée par n'importe quelle autre notation, en choisissant

\* Un nombre quel qu'il soit élevé à la puissance 0 donne pour résultat 1. Par conséquent,  $2^0 = 1$

une valeur quelconque pour la base et en répétant la procédure de construction du nombre. Aristote avait remarqué que la plupart des peuples comptaient par dizaines. On note cependant des restes de base 5 dans plusieurs idiomes (le grec par exemple) et des vestiges d'une base 20, en français notamment. C'est le choix de la base qui détermine le nombre des symboles utilisables.

#### **BASE 10, 10 symboles :**

**0, 1, 2, 3, 4, 5, 6, 7, 8, 9.**

#### **BASE 2, 2 symboles :**

**0, 1 (ou encore L, H).**

#### **BASE 8, 8 symboles :**

**0, 1, 2, 3, 4, 5, 6, 7.**

#### **BASE 16, 16 symboles :**

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

Nous reviendrons en détail sur les bases 8 et 16. Voyons pour l'instant le système de numération en base 2.

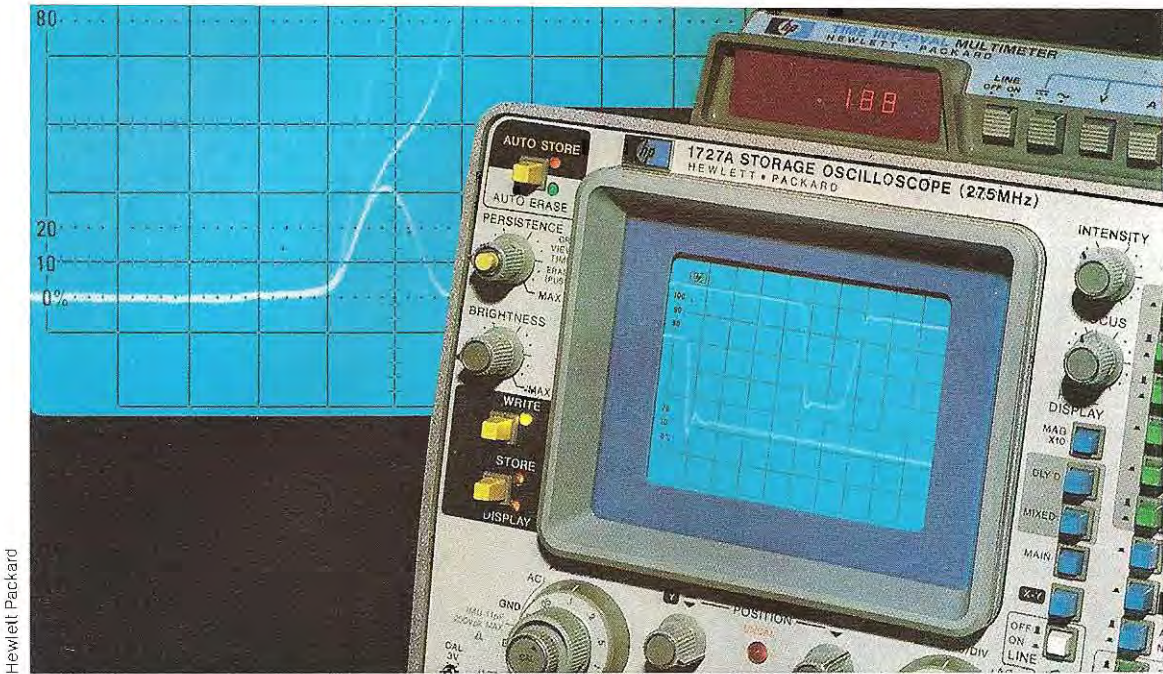
#### **Notation binaire**

La notation binaire (à base 2) est positionnelle, comme la notation décimale : si nous accomplissons machinalement les opérations nécessaires à l'écriture et à l'interprétation d'un nombre décimal, pour le système binaire, nous sommes obligés de les développer toutes, mais, par ailleurs, les règles restent les mêmes.

Cette question sera par la suite, approfondie bien au-delà des connaissances de base nécessaires à la programmation, et présente même un intérêt purement informatif : le lecteur acquerra la pratique indispensable grâce aux nombreux exemples qui lui sont présentés dans cet ouvrage.

La notation binaire n'utilise que deux symboles : 0 et 1. Les chiffres occupant les différentes positions vont multiplier les puissances successives de la base 2. La procédure d'interprétation d'un nombre binaire (ici le nombre 1011) est décrite par le tableau suivant, analogue à celui que nous venons d'utiliser pour la base 10 :

Colonne	d	c	b	a
Nombre (chiffres)	1	0	1	1
Puissances de la base	$2^3$	$2^2$	$2^1$	$2^0$
Poids des positions	8	4	2	1
Valeur des chiffres (chiffre x poids)	8	0	2	1
Valeur du nombre	$8 + 0 + 2 + 1 = 11$ (décimal)			



Oscilloscope Hewlett Packard montrant la forme des ondes de signaux numériques.

Prenons un autre exemple, avec interprétation du nombre binaire de 8 chiffres 11111111 :

Colonne	h	g	f	e	d	c	b	a
Nombre (chiffres)	1	1	1	1	1	1	1	1
Puissances de la base	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
Poids des positions	128	64	32	16	8	4	2	1
Valeur des chiffres (chiffre x poids)	128	64	32	16	8	4	2	1
Valeur du nombre	$128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255$ (décimal)							

Comme nous l'expliquerons plus loin, le groupe de 8 chiffres binaires (dont la valeur décimale maximum est 255) est l'unité de base (octet) de certains ordinateurs. Tout nombre est ici traduit par une série de 1 et de 0. Le système binaire exige toutefois une quantité importante de symboles pour indiquer une valeur numérique ; il est donc peu pratique pour l'utilisateur de la machine. Aussi a-t-on introduit deux autres systèmes positionnels : l'**octal** et l'**hexadécimal**.

Ces systèmes permettent d'utiliser une représentation adaptée au mode de fonctionnement de l'ordinateur tout en étant compacte, et donc plus commode que le binaire. Ils donnent la possibilité d'écrire un nombre en évitant de recourir à une longue série de 1 et de 0. Pour montrer les différences qui existent entre les systèmes binaire, octal et hexadécimal, on peut dire que le rapport est le même qu'entre

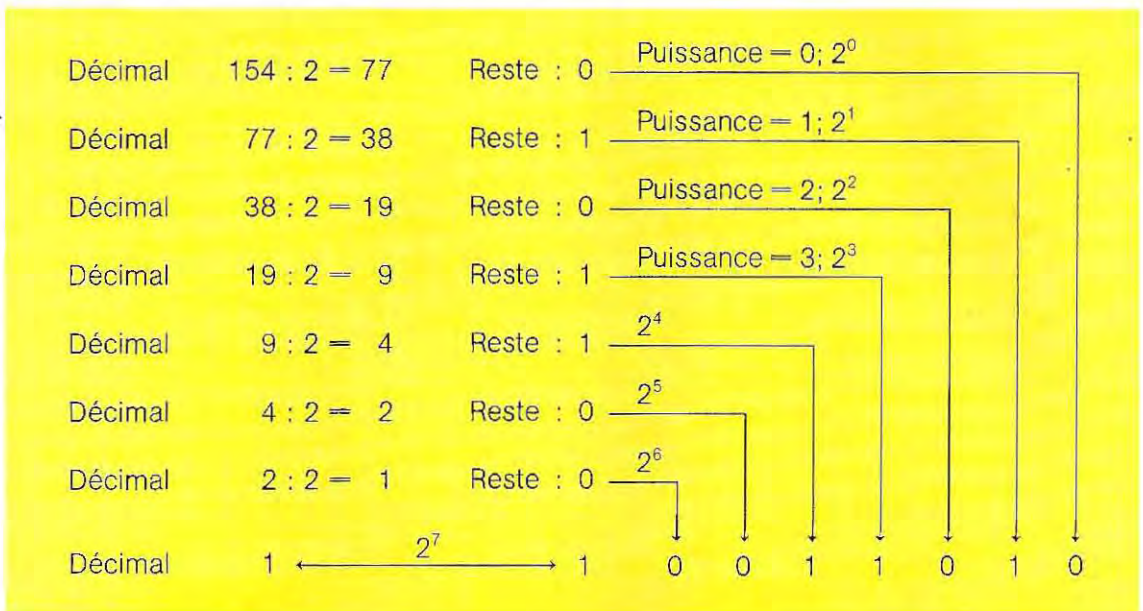
l'expression du temps en minutes et en heures. 8 heures ou 480 minutes sont la même chose ; la différence est une expression plus ou moins compacte et plus ou moins pratique. De même, la valeur d'un nombre ne change pas, qu'on l'exprime en binaire, en octal ou en hexadécimal.

L'ordinateur, quant à lui, utilise toujours et uniquement le système binaire, appelé aussi langage machine. N'oublions pas que :

**UN DISPOSITIF NUMERIQUE EXECUTE SES CALCULS EN BASE 2.**

Nous avons établi comment calculer la valeur décimale d'un nombre binaire. Voyons à présent l'opération inverse : comment obtenir un nombre binaire en partant du décimal. On divise le nombre décimal par 2 jusqu'à obtenir 1. Pour chaque division, on inscrit 0 s'il n'y a pas de reste, 1 s'il y a un reste.

Considérons par exemple le nombre décimal 154 :

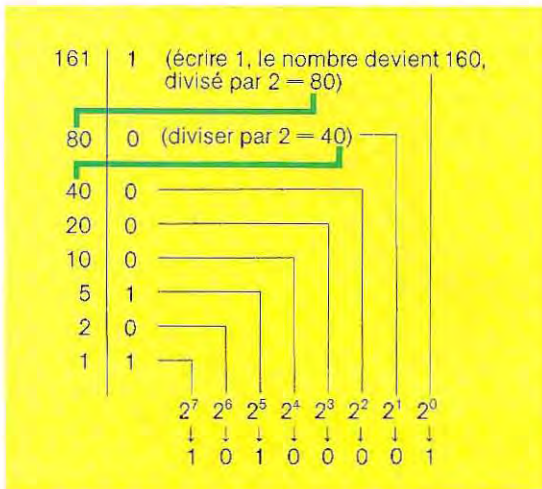


En reportant les restes, nous obtenons 10011010, soit l'équivalent binaire de 154 en décimal.

On peut accélérer l'opération en procédant comme suit :

- A/** si le nombre est pair, écrire 0
- B/** Si le nombre est impair, écrire 1 et déduire 1 du nombre
- C/** Diviser par 2
- D/** Si le résultat n'est pas 1, repartir de A avec la valeur obtenue.

Par exemple, en prenant le nombre décimal 161, nous avons :



Pour vérifier le résultat, transformons le

nombre binaire 10100001 en décimal :

$$1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1$$

$$2^7 + 2^5 + 2^0$$

$$128 + 32 + 1$$

Le total nous donne 161.

### Notation hexadécimale

Nous avons dit un peu plus haut que les ordinateurs utilisent couramment des nombres binaires composés de 8 chiffres ou multiples de 8 (16, 32). Pour indiquer ces nombres, il faut 8 symboles (ou respectivement 16, 32) pouvant valoir chacun 0 ou 1 ; la notation devient trop longue.

Pour plus de commodité, on a donc introduit une forme de numération plus compacte : la numération **hexadécimale**, c'est-à-dire à base 16. A l'intérieur de la machine, la représentation reste binaire. Le système hexadécimal n'est qu'un système de numération plus commode pour nous.

Avec ce système, les unités valent de 0 à 15 (comme les unités décimales valent de 0 à 9) ; le groupe supérieur commence à 16, c'est-à-dire à la valeur de la base (comme les dizaines du système décimal commencent à 10). Considérons par exemple le nombre binaire 10100001 (qui correspond, nous l'avons vu, à 161 décimal) ; nous savons que chaque symbole (0 ou 1) représente une puissance de 2.

Reportons le nombre binaire sur un tableau indiquant les puissances de 2 correspondantes :

- Exposant
- Valeur de la puissance
- Nombre binaire

7	6	5	4	3	2	1	0
128	64	32	16	8	4	2	1
1	0	1	0	0	0	0	1

X

Y

Considérons la moitié droite du nombre binaire 10100001, c'est-à-dire 0001 (il comporte les exposants 3, 2, 1, 0 et les valeurs 8, 4, 2, 1). Avec ce nombre binaire, on peut écrire au maximum (en donnant la valeur de tous les chiffres qui le composent) :

- $1111$  binaire  $= 8 + 4 + 2 + 1 = 15$  décimal. Pour écrire le nombre décimal 16, il faut passer au groupe suivant. En résumé :

**Un nombre binaire de 4 chiffres permet d'écrire au maximum 15. Pour écrire la valeur supérieure 16, il faut passer au groupe de 4 immédiatement supérieur (groupe X du tableau).**

On peut donc en conclure qu'en regroupant les nombres binaires par 4, on obtient un système de numération à base 16, appelé hexadécimal. En fait :

- Le groupe (Y) des unités hexadécimales permet d'écrire jus-

qu'à 15 (comme les unités décimales permettent d'écrire jusqu'à 9, c'est-à-dire une unité de moins que la base 10).

- Le groupe immédiatement supérieur (X) commence par la valeur de la base 16.

En fait, la valeur minimale (en dehors de 0) que l'on peut écrire dans le groupe X est 0001, qui vaut justement 16, de même que la valeur minimale du groupe des dizaines est 10.

En notation hexadécimale, chaque groupe de 4 chiffres binaires a des valeurs comprises entre 0 et 15.

Les exposants de 2 sont 0, 1, 2, 3 et les puissances sont 1, 2, 4, 8, qui au total donnent 15. Ecrivons à présent le nombre binaire 10100001 sous cette nouvelle forme :

- Exposant
- Valeur de la puissance
- Nombre binaire

second chiffre hexadécimal			
3	2	1	0
8	4	2	1
1	0	1	0

X

premier chiffre hexadécimal			
3	2	1	0
8	4	2	1
0	0	0	1

Y





Préparons à présent un tableau de traduction en hexadécimal.

Sachant qu'un chiffre hexadécimal regroupe 4 chiffres binaires, pour pouvoir écrire le nombre 11010111 il nous faut deux positions hexadécimales :

	2 <sup>e</sup> chiffre hexadécimal				1 <sup>er</sup> chiffre hexadécimal			
Valeur de la puissance	8	4	2	1	8	4	2	1
Nombre binaire	1	1	0	1	0	1	1	1
	8 + 4 + 1 = 13, c.-à-d. = <b>D</b>				4 + 2 + 1 = 7, c.-à-d. = <b>7</b>			

Nous avons donc :

- décimal : 215
- hexadécimal : **D7**
- binaire : 11010111

En guise de vérification, procédons en sens inverse. Nous allons tirer de la valeur hexadécimale **D7** son correspondant décimal :

- La valeur **7** représente les unités, le chiffre reste donc inchangé.
- La valeur **D** = 13 est le premier multiple de

la base, qu'il faut donc multiplier par la base 16.

Soit  $(13 \times 16) + 7 = 208 + 7 = 215$  décimal.

### Notation octale

C'est un procédé très comparable à l'hexadécimal. On divise le nombre binaire en groupes de 3 chiffres ayant les valeurs 4, 2, 1 (exposants 2, 1, 0). Pour le nombre binaire 10100001 = 161 décimal, on obtient :

	3 <sup>e</sup> chiffre octal			2 <sup>e</sup> chiffre octal			1 <sup>er</sup> chiffre octal		
Exposant	2	1	0	2	1	0	2	1	0
Valeur de la puissance	4	2	1	4	2	1	4	2	1
Nombre binaire	1	1	0	1	0	0	0	0	1
Valeur octale	<b>2</b>			<b>4</b>			<b>1</b>		

On obtient les correspondances suivantes :

Décimal : 161  
 Binaire : 10100001  
 Hexadécimal : **A1**  
 Octal : **241**

Comme nous l'avons vu dans le cas de la notation hexadécimale, pour traduire le nombre octal **241** en décimal il faut appliquer les règles suivantes :

**L'unité reste inchangée.**

**Le chiffre suivant doit être multiplié par 8<sup>1</sup>.**

**Le suivant par 8<sup>2</sup>, et ainsi de suite.**

On obtient :

Octal	Transformation	Décimal
1	—	1
4	x 8	32
2	x64	128
<b>Résultat</b>		<b>161</b>

### En résumé :

- Les circuits numériques fonctionnent en arithmétique à base 2.
- Un nombre binaire est représenté par une série de 1 et de 0.
- Chaque symbole représente le multiplicateur de la puissance correspondante de 2.
- La valeur décimale d'un chiffre hexadécimal s'obtient en le multipliant par la puissance correspondante de 16.
- En notation octale, la base est 8.
- La valeur décimale d'un chiffre octal s'obtient en le multipliant par la puissance correspondante de 8.

### Opérations en notation binaire

Nous avons dit que pour plus de commodité, on a introduit les notations octale (base 8) et hexadécimale (base 16). Mais l'ordinateur n'utilise pas directement ces notations : tous ses calculs se font en binaire.

Lorsqu'on écrit un programme, il faut parfois contrôler « à la main » les résultats de certaines instructions données à la machine, il est donc essentiel de connaître la méthode qui

permet d'effectuer des additions et des soustractions en base 2.

Nous verrons que ces contrôles n'ont pas pour objet de vérifier l'exactitude des calculs mathématiques (la machine les exécute beaucoup plus rapidement et plus précisément que nous), mais bien d'obtenir des résultats particuliers dans les opérations logiques. Nous traiterons aussi de la signification et des fonctions des opérations logiques. Pour l'instant, nous allons simplement illustrer ce concept.

Pour pouvoir dialoguer avec nous, un ordinateur doit posséder un tableau de conversion entre notre alphabet et les symboles qu'il utilise à la place de nos lettres et de nos chiffres. Il est facile d'en déduire que n'importe quel nombre est représenté à l'intérieur de l'ordinateur par une série de symboles 1 et 0.

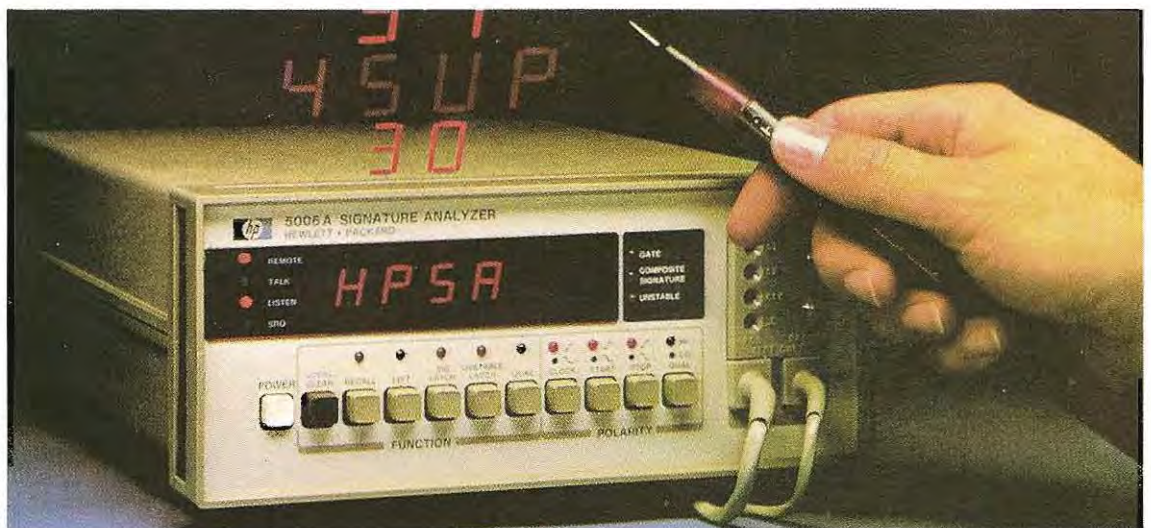
La machine ignorant tout autre symbole, il est également indispensable de traduire les lettres de l'alphabet.

Autrement dit, tous les symboles graphiques que nous utilisons sont représentés dans l'ordinateur en binaire. Si l'utilisateur veut entrer dans la mémoire de la machine la lettre A, cette lettre sera convertie en un nombre binaire.

Pour obtenir le code de la lettre B il suffira d'ajouter 1 au code de la lettre A, et ainsi de suite pour tout l'alphabet.

Cette méthode, nous le comprendrons mieux par la suite, permet de représenter la totalité de nos symboles.

### Analyseur de signature HP 5006 A : il teste le fonctionnement d'un circuit.



# TEST 1



Chacun des sujets traités sera suivi d'un test de contrôle. Le lecteur pourra vérifier l'exactitude de ses réponses par comparaison avec les solutions fournies plus loin. Si la moitié au moins de vos réponses est exacte, vous pouvez poursuivre sans difficulté la suite de votre lecture. Si le résultat est inférieur, nous vous conseillons de relire le chapitre entier, sauf s'il s'agit de questions particulières, qui ne sont pas indispensables à la compréhension de la suite (cette indication sera donnée avec la solution).

1 / Lesquelles de ces affirmations sont vraies :

- a - L'ordinateur utilise la notation binaire qui permet une interprétation facile par l'opérateur.
- b - La notation octale a été introduite pour donner une représentation plus compacte des nombres binaires.
- c - Un chiffre hexadécimal peut exprimer au maximum la valeur 15.
- d - Le chiffre 8 n'existe pas en notation octale (comment l'écrit-on en octal ?)

2 / Traduire les nombres binaires suivants en valeurs décimales correspondantes :  
1 0 0 1 - 1 1 - 1 - 1 0 1 0 1 - 1 1 0 0 1 - 1 1 1

3 / Traduire les nombres binaires suivants en valeurs octales correspondantes :  
1 1 0 1 1 1 - 1 1 - 1 0 1 0 1 0 - 1 1 1 1 1 1

4 / Traduire les nombres binaires suivants en val. hexadécimales correspondantes :  
1 1 0 1 1 1 1 1 - 1 1 0 1 1 0 1 1 - 1 1 1 1 1 1 1 1 - 1 1 1 1 1 1

5 / Traduire en binaire les nombres décimaux suivants :  
1 0 - 1 0 0 - 1 0 0 0

6 / Les quatre nombres binaires :

a - 0 0 0 1    b - 0 0 1 0    c - 0 1 0 0    d - 1 0 0 0

s'obtiennent en déplaçant le symbole 1 d'un cran vers la gauche.

- Quelle est, pour chacun d'eux, la valeur décimale correspondante ?
- En observant les traductions en valeur décimale, à quelle opération arithmétique peut-on assimiler le déplacement du symbole 1 vers la gauche ?
- Quelle opération arithmétique fait-on en déplaçant le symbole 1 vers la droite ?

7 / Lesquelles de ces affirmations sont vraies :

- a - En notation hexadécimale, les nombres supérieurs à 9 sont indiqués par des lettres de l'alphabet.
- b - En notation hexadécimale, on ne peut indiquer de nombre supérieur à 9.
- c - Un circuit numérique ne peut reconnaître d'autre état que la présence ou l'absence de signal.
- d - Le front de montée d'une impulsion correspond à la zone dans laquelle le signal passe du niveau bas au niveau haut.

8 / Traduire en décimal les nombres octaux suivants :  
7 - 7 0 - 7 0 0 - 1 1 1

9 / Traduire en décimal les nombres hexadécimaux suivants :  
7 - 7 0 - A - AA - AAA - 3FB

10 / Quelles sont les valeurs décimales les plus fortes pouvant être représentées par un nombre binaire de :  
2 chiffres (11) - 4 chiffres (1111) - 8 chiffres (11111111)

Les solutions se trouvent pages 76 et 77.

## La révolution informatique

*L'information, base de la plus grande partie de toute l'activité intellectuelle de l'homme, et l'information traitée, c'est-à-dire l'informatique, sont destinées à jouer un rôle de plus en plus grand dans notre société.*

*Du moins, pour autant que nous puissions en juger aujourd'hui.*

*En effet, la certitude en ce domaine n'est pas absolue, mais elle est pourtant déjà assez bien étayée pour servir de base à notre raisonnement. C'est un peu comme face à un iceberg : malgré les dimensions relativement réduites de la partie visible par rapport à celles de la partie cachée, l'iceberg nous paraît suffisamment impressionnant pour que nous puissions imaginer l'importance de ce que nous ne voyons pas. De même, nous pouvons affirmer que l'ordinateur et la micro-électronique seront les protagonistes et les éléments moteurs de la transformation des années à venir. Les plus impor-*

**Réactualisées en permanence, les informations sont à la disposition de tous les particuliers possédant un poste de télévision et un téléphone, ainsi qu'un décodeur s'il n'est pas déjà intégré dans le téléviseur. (Télématique, Vélizy.)**



Carlos Santos / Rapho

*tantes de ces nouvelles sciences ont été expérimentées, étudiées et portent déjà des noms. On parle en effet de télématique, de robotique et de bureautique pour désigner les principaux domaines d'application de l'informatique.*

*L'explosion de l'électronique concernera tous les secteurs de la vie quotidienne à la maison, au bureau et à l'usine. Le changement ne sera ni brutal, ni automatique ; il s'effectuera à travers une évolution culturelle portant à la connaissance généralisée, à l'informatique de masse.*

\* \* \*

Le terme **télématique** est issu de la fusion de deux autres mots : télécommunication et informatique. Il s'agit donc de la technologie de la communication reliée à l'informatique, ce qui représente le premier grand stade de la transformation qui nous intéresse.

Le téléphone, c'est-à-dire le terminal le plus répandu – on comptait 500 millions de postes installés dans le monde à la fin de 1980, et 20 millions d'abonnés en France à la fin de 1983 –, le téléviseur et l'ordinateur individuel qui peut les relier l'un à l'autre, tels sont les appareils par lesquels se fera le changement. Ce dernier sera possible grâce à l'amélioration des communications : par câbles, avec l'introduction de matériaux nouveaux comme les fibres optiques ; par voie hertzienne, avec l'utilisation des satellites à l'échelle planétaire. Encore une fois c'est l'information numérisée, c'est-à-dire traduite en code binaire, qu'elle soit graphique ou alphanumérique, qui sera l'instrument de la révolution à laquelle nous pouvons nous préparer. Le progrès se mesurera par la vitesse de transmission de l'information, la capacité de stockage de cette information dans les banques de données, ainsi que la possibilité de la récupérer et de la transmettre à grande distance.

Le terme télématique désigne toute une série de services supportés par les réseaux de télécommunication classiques ou nouveaux, qui mettront à la disposition des utilisateurs les informations les plus diverses contenues dans les banques de données (ensembles d'informations organisées). Cela laisse entrevoir la puissance dynamique et de développement de la télématique à l'égard des divers secteurs et des activités industrielles. Bon nombre de systèmes traditionnels sont appe-

lés à changer, tandis que d'autres naîtront sous la poussée des technologies nouvelles. La télématique agit d'ores et déjà comme facteur de développement pour les organisations de télévision, de téléphone et de communication qui préparent les structures nécessaires pour accueillir et transporter, par câbles ou par voie hertzienne, les informations vers les utilisateurs.

Par ailleurs, le développement né de la télématique intéressera toutes les structures de création, d'organisation et de fourniture d'informations. Les maisons d'édition et de presse, qui assument traditionnellement ce rôle, verront s'ouvrir un autre débouché pour leurs informations; bien d'autres organisations économiques (instituts de crédits ou bourses, par exemple) bénéficieront de la télématique qui leur permettra de mieux distribuer l'information qu'elles possèdent. La publicité ne restera pas étrangère à cette nouvelle méthode de communication, et la télématique lui offrira sans aucun doute un nouveau moyen d'atteindre le destinataire du message publicitaire. Tout cela nous conduit enfin à une vaste gamme de machines nouvelles: téléviseurs améliorés, ordinateurs individuels, téléphones, instruments et appareils de télécommunications, etc., avec encore d'autres possibilités de développement pour la production industrielle.

Les premières applications de la télématique commencent déjà à être diffusées dans le monde.

\* \* \*

La **robotique** n'appartient plus à la science-fiction. Les robots sont parmi nous. Dans les secteurs industriels de pointe, la recherche, l'espace et le fond des océans, ils remplacent l'homme pour toutes les tâches répétitives, fatigantes ou dangereuses. Leur emploi permet de réduire les coûts de production, rend accessibles des activités inimaginables jusque-là et augmente la compétitivité des procédés. Dans l'avenir, nous verrons les robots mécaniques, déjà dotés aujourd'hui d'organes sensoriels et dirigés par des systèmes électroniques et des microprocesseurs, se multiplier et remplir les fonctions les plus diverses.

Ces machines comportent en général un bras mécanique accomplissant des mouvements



**Le Télex utilise le réseau téléphonique; il comprend un clavier et une imprimante.**

dans toutes les positions et une main servant à prendre et à porter des objets. Souvent, les robots sont également dotés d'organes sensoriels qui leur confèrent la vue et le toucher, voire la parole. Une autre caractéristique des robots industriels est qu'ils s'adaptent à de nombreux emplois: ils mémorisent les tâches à accomplir et en apprennent de nouvelles, ce qui présuppose, aux côtés d'une structure mécanique, une structure électronique, c'est-à-dire un véritable mini-ordinateur. Les champs d'applications de la robotique industrielle sont innombrables. Le grand public connaît surtout ceux de l'industrie automobile mais ils sont présents dans de multiples autres secteurs où ils prennent la place de l'homme.

\* \* \*

Une fois de plus, l'histoire de l'information voit naître un nouveau terme par fusion de deux mots et de deux concepts, pour désigner un groupement empirique d'origine récente. Ce terme d'origine française désigne des techniques et des technologies relatives à l'automatisation des travaux de bureau: bureau plus informatique ont donné **bureautique**, dont l'usage se répand partout aujourd'hui.

(D'après A SCUOLA CON IL COMPUTER, Egidio Pentiraro. Laterza, 1983).

## Addition binaire

Pour réaliser une addition en arithmétique binaire, il suffit de ne pas oublier que la base de cette numération est 2 : quand le résultat de l'addition donne 2, il faut écrire 0 et retenir 1, exactement comme on le fait dans une addition en décimal quand le résultat est égal ou supérieur à 10.

Additionnons par exemple le chiffre binaire 1 à lui-même :

$$\begin{array}{r} 1 + \\ 1 = \\ \hline 10 \end{array}$$

Le résultat de  $1 + 1$  est 2 ; on écrit donc 0 et on retient 1.

Pour vérifier le résultat, transformons chacun de ces chiffres en décimal :

$$\begin{array}{r} 1 \text{ binaire} = 1 \text{ décimal} \\ + \quad \quad \quad + \\ 1 \text{ binaire} = 1 \text{ décimal} \end{array}$$

Résultat : 10 binaire = 2 décimal

Additionnons à présent les deux nombres binaires :

$$\begin{array}{r} 101101 \\ 10110 \end{array}$$

Avant d'effectuer cette addition, transformons les deux nombres binaires par leurs équivalents décimaux (nous utiliserons ces valeurs pour vérifier le résultat) :

$$101101 = 2^0 + 2^2 + 2^3 + 2^5 = 1 + 4 + 8 + 32 = 45$$

$$10110 = 2^1 + 2^2 + 2^4 = 2 + 4 + 16 = 22 \text{ décimal}$$

Le résultat de l'addition binaire doit donc être l'équivalent de  $45 + 22 = 67$  décimal.

Pour examiner cette addition binaire, mettons les deux nombres en colonnes à partir de la droite :

Colonne	g	f	e	d	c	b	a
1 <sup>er</sup> terme		1	0	1	1	0	1
2 <sup>d</sup> terme			1	0	1	1	0
	1	0	0	0	0	1	1

Retenues

A partir de la **colonne A** : les deux termes sont 1 et 0, le résultat est 1, il n'y a pas de retenue.

**Colonne B** : les deux termes sont 0 et 1, résultat = 1, retenue = 0.

**Colonne C** : les deux termes sont 1 et 1, résultat = 0, retenue = 1.

$1 + 1 = 0$  avec retenue de 1.

De même, en décimal on obtient :

$8 + 2 = 10$  c'est-à-dire 0 avec retenue de 1.

La retenue s'ajoute au résultat de la colonne D.

**Colonne D** : les termes sont 1 et 0, résultat = 1. Ajoutons la retenue précédente (1) :  $1 + 1 =$  nouv. résultat = 0. Nouvelle retenue = 1.

**Colonne E** : les termes sont 0 et 1, résultat = 1. Ajoutons la retenue précédente (1) :  $1 + 1 =$  nouv. résultat : 0. Nouvelle retenue = 1.

**Colonne F** : les termes sont 1 et 0, résultat = 1. Ajoutons la retenue précédente (1) :  $1 + 1 =$  nouv. résultat = 0. Retenue = 1.

**Colonne G** : elle ne comporte aucun terme, on y inscrit donc la dernière retenue (1). Le résultat est :

$$1000011 = 2^0 + 2^1 + 2^6 = 1 + 2 + 64 = 67 \text{ décimal.}$$

On peut effectuer de la même façon les additions à base 8 et à base 16.

Dans ces notations, il faut se souvenir que la retenue existe quand la somme est respectivement égale (ou supérieure) à 8 et à 16.

Prenons pour exemple la somme des deux mêmes nombres en notation octale.

Transformons d'abord les deux termes en notation octale :

Valeur	4	2	1	4	2	1
Nombre binaire à transformer	1	0	1	1	0	1
Correspondant octal	5			5		
	(4 + 1)			(4 + 1)		

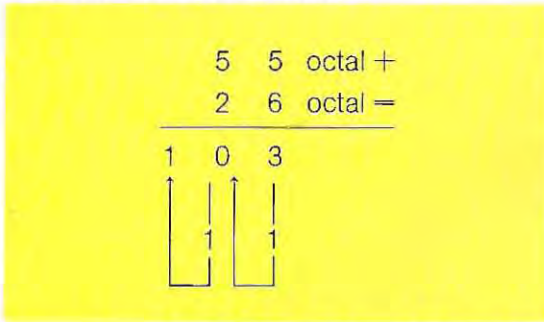
Valeur	4	2	1	4	2	1
Nombre binaire à transformer	*	1	0	1	1	0
Correspondant octal	2			6		

$$101101 = 55 \text{ octal}$$

$$10110 = 26 \text{ octal}$$

\*L'absence de symbole en première position équivaut à un 0. Le nombre binaire 10110 équivaut donc à 010110.

La somme en octal devient :



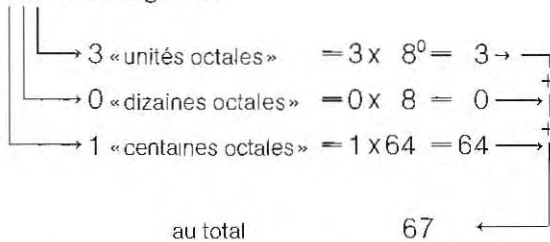
Le résultat (103 octal) s'obtient comme suit :

- En partant de la droite,  $5 + 6 = 11$ , de ce fait (en retirant la base on obtient 3 et une retenue = 1) dans la seconde colonne nous avons :  $5 + 2 + \text{retenue } (1) = 8$  en retirant la base, on obtient 0 et retenue = 1.

Dans la dernière colonne, il suffit d'écrire la retenue puisqu'il n'existe aucun terme.

En guise de vérification, transformons le résultat en décimal :

Le nombre 103 octal signifie :



ou de manière plus ramassée :

$$103 \text{ octal} = 3 \times 8^0 + 1 \times 8^2 = 3 + 64 = 67$$

### Soustraction binaire

La différence de deux nombres en notation binaire s'effectue selon les mêmes règles que la soustraction décimale mais, pour des raisons que nous verrons mieux par la suite, nous allons utiliser la « méthode du complément ».

Le complément d'un nombre binaire s'obtient en remplaçant le symbole 1 par le symbole 0 et vice versa. Le complément de 100101 est donc 011010.

La somme du nombre de départ et du complément est un nombre binaire composé uniquement de symboles 1 :

$$\begin{array}{r}
 \text{nombre} \quad 100101 + \\
 \text{complément} \quad 011010 \\
 \hline
 \text{somme} \quad 111111
 \end{array}$$

En ajoutant 1 à la somme on obtient une série continue de 0 :  $111111 + 1 = 1000000$ .

En appliquant la méthode du complément, la différence entre deux nombres binaires s'obtient comme suit :

**A /** Compléter le plus petit nombre pour parvenir au même nombre de chiffres que le premier terme.

**B /** Additionner le premier terme et le complément du plus petit, en excluant du résultat l'éventuel chiffre 1 qui dépasse à gauche la longueur du premier terme.

**C /** Ajouter 1\*.

Effectuons l'opération suivante :

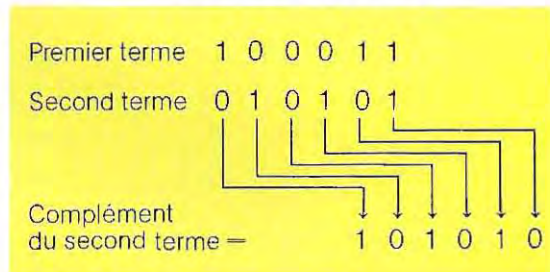
Premier terme    Second terme

$$100011 - 10101$$

En décimal, cette différence est :  $35 - 21 = 14$ .

### Etape A

Complétons le second terme, plus petit que le premier, à l'aide de symboles 0 :



### Etape B

On effectue la somme du premier terme (100011) et du complément du second terme (011010) :

$$\begin{array}{r}
 \text{Premier terme} \quad 100011 \\
 \text{Complément du second terme} \quad 011010 \\
 \hline
 \text{Somme} \quad 1001101
 \end{array}$$

Nous excluons de la somme le symbole 1 qui dépasse la longueur du premier terme.

### Etape C

On ajoute 1 à la somme :

$$\begin{array}{r}
 \text{somme} \quad 001101 + \\
 \quad \quad \quad 1 \\
 \hline
 \text{Résultat} \quad 001110 = 14 \text{ décimal}
 \end{array}$$

La procédure ainsi décrite peut paraître beaucoup plus lourde qu'elle ne l'est en réalité. En

\*Exclure les chiffres qui dépassent comme B.

fait, avec un minimum de pratique, la plupart des étapes que nous venons d'expliquer se font de tête.

Voici un autre exemple, exposé de manière abrégée.

Soit l'opération :

$$226 - 174 = 152 \text{ décimal} =$$

$$= 11100010 - 10101110.$$

En faisant le complément du second terme (10101110) puis la somme, on a :

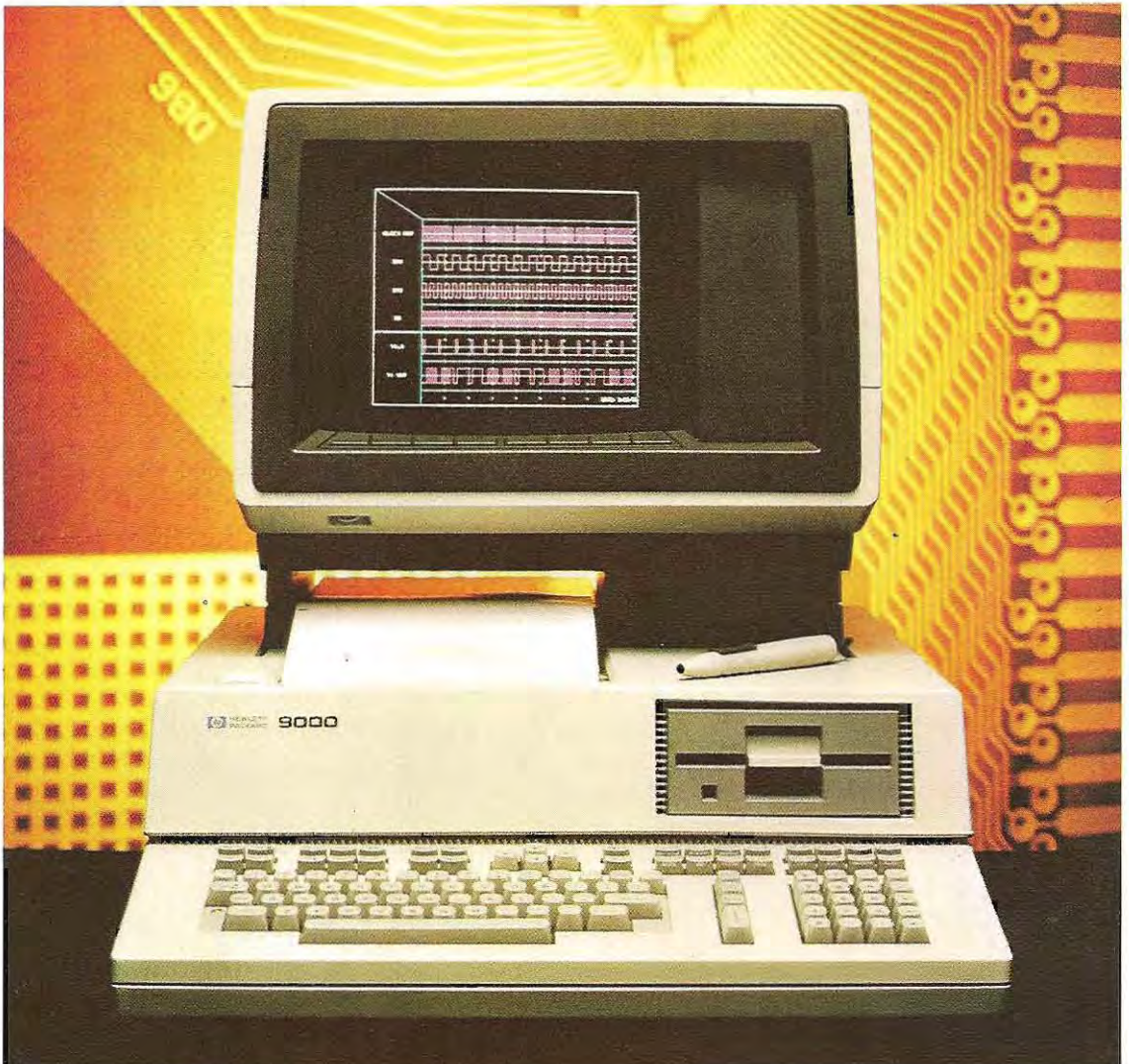
Premier terme	11100010	
Complément du 2 <sup>d</sup> terme	01010001	
Résultat de l'addition	00110011	+
		1
	00110100	=
		= 52 décimal

Pour compléter cet exposé, le listing (ou liste) 1 de la page 63 à la page 66 donne en partie les équivalences binaires, octales (OCT), et hexadécimales (HEX) des nombres décimaux de 1 à 255.

Le listing 2, pages 66 et 67, donne les mêmes équivalences pour les nombres compris entre 31100 et 31150.

Les données des colonnes OCT (octal) et HEX (hexadécimal) ont été tournées par l'ordinateur avec une instruction unique qui permet d'obtenir le résultat en quelques millièmes de seconde. Essayez donc de traduire le nombre 31142: même avec beaucoup de rapidité et d'attention, il vous faudra plusieurs minutes.

**Ordinateur HP 9000 montrant le comportement des signaux numériques d'un circuit logique.**













31116	0	1	1	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	74614	798C
31117	0	1	1	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	1	74615	798D
31118	0	1	1	1	1	0	0	1	1	0	0	1	1	1	0	1	1	0	1	74616	798E
31119	0	1	1	1	1	0	0	1	1	0	0	1	1	1	1	1	1	1	1	74617	798F
31120	0	1	1	1	1	0	0	1	1	0	0	1	1	0	0	1	0	0	0	74620	7990
31121	0	1	1	1	1	0	0	1	1	0	0	1	1	0	0	1	0	0	1	74621	7991
31122	0	1	1	1	1	0	0	1	1	0	0	1	1	0	0	1	0	0	1	74622	7992
31123	0	1	1	1	1	0	0	1	1	0	0	1	1	0	0	1	0	0	1	74623	7993
31124	0	1	1	1	1	0	0	1	1	0	0	1	1	0	0	1	0	0	1	74624	7994
31125	0	1	1	1	1	0	0	1	1	0	0	1	1	0	0	1	0	0	1	74625	7995
31126	0	1	1	1	1	0	0	1	1	0	0	1	1	0	0	1	0	0	1	74626	7996
31127	0	1	1	1	1	0	0	1	1	0	0	1	1	0	0	1	0	0	1	74627	7997
31128	0	1	1	1	1	0	0	1	1	0	0	1	1	0	0	1	0	0	0	74630	7998
31129	0	1	1	1	1	0	0	1	1	0	0	1	1	0	0	1	0	0	1	74631	7999
31130	0	1	1	1	1	0	0	1	1	0	0	1	1	0	0	1	0	0	0	74632	799A
31131	0	1	1	1	1	0	0	1	1	0	0	1	1	0	0	1	0	0	1	74633	799B
31132	0	1	1	1	1	0	0	1	1	0	0	1	1	0	0	1	0	0	0	74634	799C
31133	0	1	1	1	1	0	0	1	1	0	0	1	1	0	0	1	0	0	1	74635	799D
31134	0	1	1	1	1	0	0	1	1	0	0	1	1	0	0	1	0	0	0	74636	799E
31135	0	1	1	1	1	0	0	1	1	0	0	1	1	0	0	1	0	0	0	74637	799F
31136	0	1	1	1	1	0	0	1	1	0	0	1	1	0	0	0	0	0	0	74640	79A0
31137	0	1	1	1	1	0	0	1	1	0	0	1	1	0	0	0	0	0	1	74641	79A1
31138	0	1	1	1	1	0	0	1	1	0	0	1	1	0	0	0	0	0	0	74642	79A2
31139	0	1	1	1	1	0	0	1	1	0	0	1	1	0	0	0	0	0	0	74643	79A3
31140	0	1	1	1	1	0	0	1	1	0	0	1	1	0	0	0	0	0	0	74644	79A4
31141	0	1	1	1	1	0	0	1	1	0	0	1	1	0	0	0	0	0	0	74645	79A5
31142	0	1	1	1	1	0	0	1	1	0	0	1	1	0	0	0	0	0	0	74646	79A6
31143	0	1	1	1	1	0	0	1	1	0	0	1	1	0	0	0	0	0	0	74647	79A7
31144	0	1	1	1	1	0	0	1	1	0	0	1	1	0	0	0	0	0	0	74650	79A8
31145	0	1	1	1	1	0	0	1	1	0	0	1	1	0	0	0	0	0	0	74651	79A9
31146	0	1	1	1	1	0	0	1	1	0	0	1	1	0	0	0	0	0	0	74652	79AA
31147	0	1	1	1	1	0	0	1	1	0	0	1	1	0	0	0	0	0	0	74653	79AB
31148	0	1	1	1	1	0	0	1	1	0	0	1	1	0	0	0	0	0	0	74654	79AC
31149	0	1	1	1	1	0	0	1	1	0	0	1	1	0	0	0	0	0	0	74655	79AD
31150	0	1	1	1	1	0	0	1	1	0	0	1	1	0	0	0	0	0	0	74656	79AE

## Un terminal dans le téléviseur

Un simple écran de télévision peut aujourd'hui servir de terminal vidéo à un micro-ordinateur et permettre à tout utilisateur d'accéder à un réseau de transmission de données.

Depuis déjà quelques années nous avons accès à une série limitée d'informations de caractère général, par exemple les conditions et les prévisions météorologiques, l'heure exacte, les conditions de circulation, en composant tout simplement un numéro de téléphone. Mais c'est seulement depuis peu que la qualité du service a fait un bond en avant avec la mise en place des systèmes combinés de traitement et de diffusion de l'information. Aujourd'hui, beaucoup d'utilisateurs de la télévision, en appuyant sur certaines touches de leur appareil de télécommande, peuvent faire apparaître sur l'écran (en surimpression ou en écran complet) une vaste gamme de fiches

d'informations touchant les domaines les plus variés. Deux systèmes offrant ce type de services ont été réalisés : le Vidéotex diffusé ou Télétexte, et le Vidéotex interactif ou Télétel. Le premier, réalisé vers 1970, était diffusé par le réseau normal de télévision, le second est transmis par le réseau téléphonique; dans les deux cas, il faut un décodeur spécial pour visualiser les informations sur l'écran.

L'utilisateur, en même temps qu'il regarde un programme télévisé quelconque, peut, en appuyant sur l'une des touches de sa télécommande, faire apparaître la première page des informations Télétexte disponible à cet instant. Cette première page contient l'index général de la série et il suffit de composer sur le clavier de télécommande le numéro de la page intéressante pour qu'elle apparaisse à l'écran et y demeure jusqu'à ce qu'on la remplace en faisant un nouveau numéro.

**Le système Prestel diffuse des informations variées par l'intermédiaire du téléphone.**



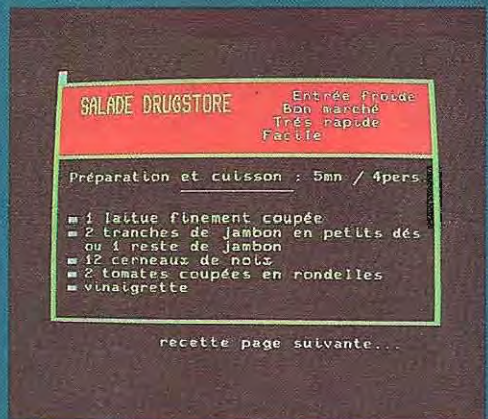
Les images de Télétexte sont préparées par des groupes de rédacteurs spécialisés qui entrent régulièrement dans l'ordinateur central (appelé « serveur ») les informations nécessaires. Chaque page de transmission se compose de vingt-quatre lignes de texte, de quarante caractères chacune. La composition du texte peut faire appel à sept couleurs (blanc, rouge, bleu, vert, magenta, bleu clair et jaune). Chaque caractère est transmis à l'aide d'un code numérique qui indique aux circuits électroniques chargés de la formation de l'image dans l'appareil récepteur lesquels des soixante-dix points élémentaires constituant sur l'écran l'ensemble destiné à représenter un caractère doivent être éclairés. Une seconde information codée précise la couleur à donner à chaque groupe de points. Un technicien décrirait plus exactement ce mécanisme en remplaçant avec indignation le terme de groupe par celui de « zone » destinée sur l'écran à chacun des caractères mis en œuvre sur une matrice de 7 x 10 pts. Certaines possibilités graphiques permettent en outre d'unifier les zones disponibles pour plusieurs caractères afin de former des lettres et des titres plus grands, et de colorer sélectivement certaines zones pour créer des images simples en deux dimensions.

Pendant l'horaire de transmission qui coïncide en général avec celui des émissions normales des programmes de télévision, toute la série des pages de textes est transmise sans interruption. La transmission d'une page de vingt-quatre lignes imprimées demande un peu moins d'un quart de seconde : intervalle négligeable. Si l'on pense qu'un service Télétexte normal ne comprend jamais moins d'une centaine de pages et que le service tout entier doit être transmis cycliquement, ligne par ligne et page par page, plusieurs minutes peuvent s'écouler entre l'instant où l'utilisateur demande une page déterminée et celui où elle apparaît à l'écran. Pour éviter la disparition de l'image, un circuit de mémoire maintient sur l'écran la page sélectionnée précédemment jusqu'à l'apparition de la nouvelle page. Ce temps mort constitue l'un des principaux inconvénients du système Télétexte, et l'on y a cherché remède. Les perfectionnements qui sont en phase expérimentale permettent déjà de prévoir un large développement des applications du système. Le premier

projet est connu sous le nom de Telesoftware ou Télélogiciel. D'après les techniciens, l'addition d'un microprocesseur aux circuits électroniques récepteurs permettra à l'avenir d'accéder directement à l'ordinateur central ou à un réseau d'ordinateurs interconnectés. On pourra de la sorte utiliser la télévision comme un « terminal intelligent » capable d'exécuter des programmes de calcul répondant à nos exigences particulières, par exemple pour la comptabilité et la gestion domestiques. Le second projet concerne un autre service réalisable dans le cadre du système Télétexte, c'est-à-dire l'addition, sur demande expresse du téléspectateur, de sous-titres au programme télévisé normal. D'une part, cela permettrait d'éliminer les difficultés dont souffrent les malentendants, et, de l'autre, cela donnerait à tous la possibilité de suivre les émissions en langue étrangère dans leur version originale.

En France, le Télétexte A2-Antiope (acquisition numérique et télévisualisation d'images organisées en pages d'écriture) est exploité depuis 1978, et fonctionne depuis novembre 1983 avec décodeur.

Viewdata est un système comparable à Télétexte, mais il offre de plus grandes possibilités à long terme. Diversement baptisé selon les pays (Vidéotex en France, et ailleurs Videotel, Prestel, Televideo), il rattache la multitude de ce que l'on peut à présent dénommer « terminaux télévisuels » à l'ordinateur central par l'intermédiaire des lignes téléphoniques normales. Prestel par exemple, le service Viewdata offert par l'administration britannique de télécommunications, fut le premier introduit en Europe. Pour accéder à ce service, l'utilisateur appuie sur l'une des touches de son boîtier de télécommande de télévision, comme pour le Télétexte ; le téléviseur se branche automatiquement sur l'ordinateur central par la ligne téléphonique locale. Dans d'autres versions plus simples (l'annuaire électronique), il faut effectuer manuellement l'opération de mise en relation avec la centrale de transmission, ce qui se fait tout simplement en soulevant le combiné du téléphone et en composant le numéro de la centrale. On peut alors appeler instantanément l'index général des informations diffusées par le service, choisir le sujet qui vous intéresse et composer sur le boîtier de télécommande le



A. Hatloppé/GFI

### Ces écrans montrent l'extrême diversité des informations transmises par Vidéotex.

mot-clé permettant l'accès à cette page. Un second index analytique et plus détaillé apparaît alors sur l'écran, et le choix se poursuit par une succession de sélections en arbre logique. Le système Vidéotex offre d'énormes possibilités. A la différence de ce qui se produit pour le Télétexte, les pages ne sont transmises à l'utilisateur qu'à sa demande sur le mode du dialogue. Les limites d'un système de ce type sont celles de la capacité de mémoire de l'ordinateur central.

Le système Prestel reçoit les données et les mises à jour d'environ cent cinquante organismes spécialisés dans différents secteurs « producteurs d'informations ». Il peut aujourd'hui renseigner l'utilisateur sur les questions les plus diverses : l'emploi de son temps libre, les voyages et les moyens de transport, les prix de détail, les statistiques, etc. L'utilisateur paie les pages de transmission qu'il

demande, le prix de chaque page apparaît sur l'écran en même temps que celle-ci et l'ordinateur débite automatiquement les frais sur le compte de l'utilisateur. Ce dernier peut à tout moment demander un état de son compte, qui apparaît directement à l'écran.

Le système Vidéotex permet déjà d'accéder directement aux banques de données. Certaines grosses sociétés comme Lockheed et des universités comme le Massachusetts Institute of Technology (MIT) aux Etats-Unis, gèrent déjà des banques de données considérables et ouvertes au secteur privé. En France, La Redoute, entre autres, offre un service opérationnel de prise de commandes à domicile par Vidéotex (expérience Télétel de Vélizy et annuaire électronique).

Un tel service est cependant destiné en priorité au monde professionnel et commercial. Son but est d'assurer la communication rapide



et sélective de nouvelles à caractère professionnel. Un certain degré de discrétion peut être obtenu en attribuant à chaque utilisateur une clé d'accès personnalisée fonctionnant comme un mot d'ordre pour l'ordinateur central, lequel peut alors accorder ou refuser à l'utilisateur l'accès à une section déterminée du fichier contenant les informations à diffusion limitée.

On peut échapper au risque d'interception de ces informations par des tiers au cours de leur passage sur la ligne de l'utilisateur habilité en adoptant des codes de transmission qui les rendent incompréhensibles à des personnes étrangères, mais cela exige l'intégration dans le système récepteur d'un microprocesseur programmé pour décoder le message à l'arrivée.

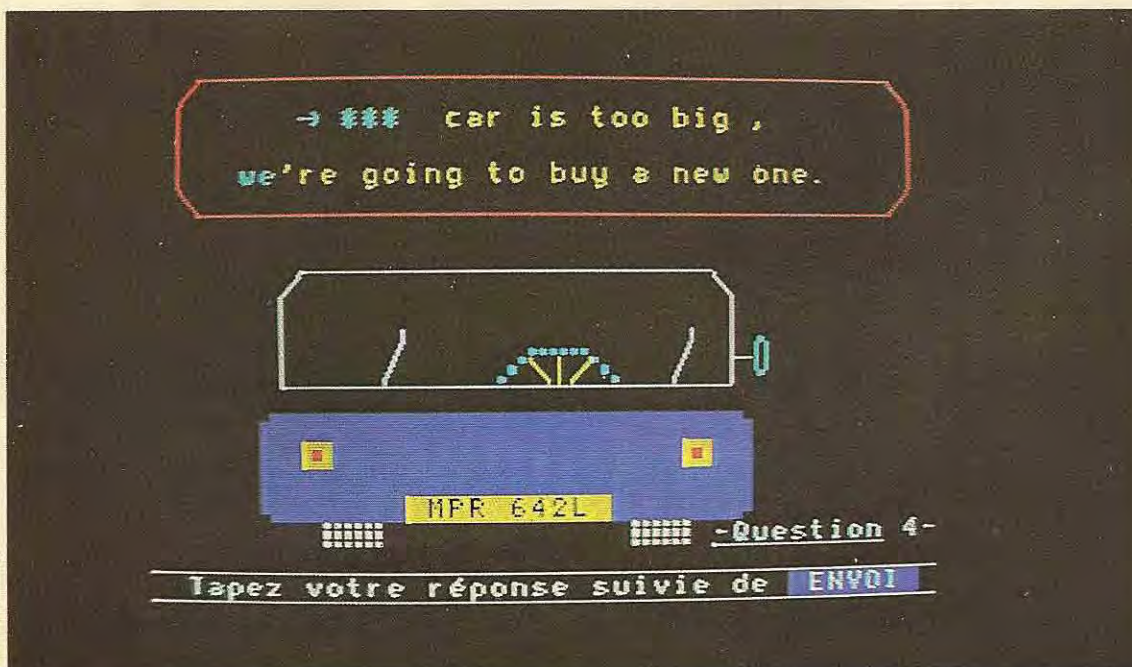
Autre caractéristique particulière au système Vidéotex : l'utilisateur peut communiquer de manière interactive avec l'ordinateur central. Les champs d'applications sont multiples. Le diagnostic médical assisté par ordinateur, par exemple, est un service qui exige un contact interactif entre l'utilisateur et l'ordinateur. Les programmes de diagnostic déjà utilisés par les plus grands hôpitaux permettent aux médecins d'économiser un temps précieux

dans l'analyse de l'état du patient à l'arrivée, et la mise en place d'un service analogue à domicile pourrait faire gagner beaucoup de temps aux médecins généralistes.

Pour améliorer encore l'interactivité (échange), il faudrait doter le système récepteur d'un microprocesseur et d'un clavier alphanumérique, ce qui ouvrirait à l'utilisateur l'accès non seulement à la mémoire de masse de l'ordinateur mais aussi à l'unité centrale de traitement, par exemple, pour faire exécuter des programmes personnels et en obtenir les résultats sur son écran.

Le coût d'un service de ce type, surtout s'il est utilisé rarement et sur des programmes simples, pourrait bien se révéler inférieur à celui d'un ordinateur individuel. L'adoption du microprocesseur offre aussi la possibilité de se relier à sa banque, en utilisant comme « passerelle » l'ordinateur qui gère le système de diffusion, pour accomplir toutes les opérations bancaires, y compris le transfert de fonds sur son compte ou à partir de celui-ci. Le téléviseur individuel fonctionne dans ce cas comme l'un des terminaux vidéo de la succursale bancaire, et permet d'accomplir les opérations ordinaires en toute commodité, chez soi et en temps réel.

**Grâce au système Vidéotex, les programmes de formation (ici cours d'anglais) donnent lieu à un « dialogue » entre ordinateur et utilisateur.**



## Eléments de logique

Certaines des fonctions accomplies par l'ordinateur peuvent être obtenues avec d'autres types de circuits. Nous traiterons dans ce chapitre des aspects fondamentaux de cette technique appelée **logique câblée**.

La question n'est pas directement rattachée aux ordinateurs mais constitue un premier pas indispensable pour comprendre la structure et le fonctionnement de ces machines. Nous allons commencer par donner quelques principes de logique qui seront repris et approfondis dans la partie consacrée aux langages de programmation.

### Les opérateurs

Le micro-ordinateur, ou plus exactement l'un de ses composants, l'unité centrale (UC), peut accomplir aussi bien des opérations arithmétiques que des **opérations logiques**. Les opérations sont surtout utilisées pour comparer des données entre elles et déclencher, à partir du résultat, une transaction particulière.

Par exemple, on peut insérer dans un programme calculant la feuille de paie un système de contrôle du nom qui se présente ; si ce nom est FIN, la machine doit en aviser l'opérateur et mettre fin au programme.

Dans ce cas, l'opération logique est la comparaison entre le nom et le mot FIN. Si le résultat est négatif (c'est-à-dire si le nom présenté n'est pas FIN) la machine poursuit son travail ; si le résultat est positif (le nom est FIN), la machine émet un signal et s'arrête.

Par certains aspects, les opérateurs logiques peuvent être considérés comme des opérateurs arithmétiques normaux. Nous commencerons par exploiter cette analogie.

Dans l'exécution d'un calcul arithmétique normal nous pouvons distinguer trois entités fondamentales :

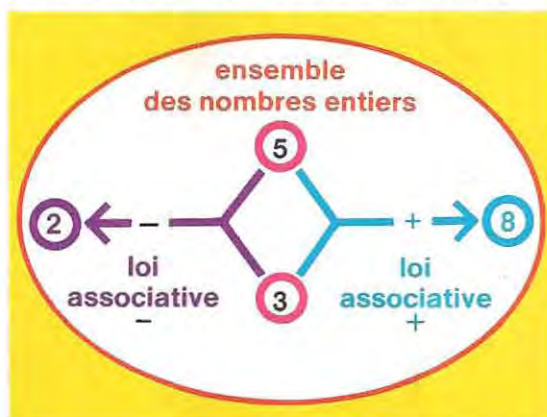
- Les opérands
- L'opérateur
- Le résultat

Les opérands sont les nombres sur lesquels nous effectuons l'opération ; l'opérateur est le symbole indiquant l'opération à accomplir ; le

résultat est le nombre associé par l'opérateur aux opérands.

Considérons l'expression arithmétique  $5 + 3 = 8$ . Le résultat (8) est associé aux opérands 5 et 3 par l'opérateur + (addition).

Si nous changeons l'opérateur (par ex.  $-$ , soustraction), les symboles 5 et 3 seront associés (par ce nouvel opérateur) au résultat (2). Les diverses opérations (addition, soustraction, multiplication, etc.) sont des **lois associatives**, c'est-à-dire des lois qui associent des groupes de nombres (opérands) à d'autres nombres (résultats). Les modalités de ces associations sont exprimées par le symbole (+, -, etc.) de l'opérateur. D'où la règle fondamentale suivante : une loi associative ne peut être appliquée qu'à des éléments homogènes c'est-à-dire appartenant au même ensemble (voir schéma). Cherchons à rendre plus clair le concept d'**ensemble**.



Au niveau de l'intuition, on peut considérer comme ensemble un groupe d'éléments (nombre, lettres, objets) répondant à des lois déterminées. La définition de ces lois et les propriétés des éléments incombent à l'algèbre abstraite. Ici, nous appellerons ensemble un groupement d'éléments homogènes.

Ce concept d'ensemble, inexact du point de vue mathématique, est cependant d'emploi courant dans les applications de l'ordinateur. Bon nombre d'erreurs dans les calculs ont pour origine l'utilisation d'opérands non homogènes (nous reprendrons un peu plus loin cette question en détails).

Nous venons d'évoquer des opérateurs correspondant aux opérations arithmétiques ordinaires, mais on peut inventer d'autres lois associatives, leur affecter un nom et les utiliser comme de nouveaux opérateurs.

En réalité, le processus n'est pas si simple. Il s'agit essentiellement d'introduire une nouvelle série d'opérateurs : les **opérateurs logiques**. Ces opérateurs obéissent à des règles spécifiques, parfois très complexes. Nous ne nous occuperons ici que de celles qui ont trait à la programmation des ordinateurs.

### Les opérateurs logiques

Reprenons l'exemple des feux de circulation. Il y a quatre situations possibles :

- 1 - feux éteints
- 2 - feu rouge
- 3 - feu orange
- 4 - feu vert.

A la question : « Quel est l'état des feux de circulation ? » la réponse est nécessairement l'une des quatre situations énumérées plus haut. Soit : éteint, OU rouge, OU orange, OU vert.

Le mot OU est un opérateur logique (en anglais OR). Le symbole est OU. On peut donc exprimer les quatre situations possibles de la façon suivante (en conservant feux éteints = 1, feu rouge = 2, etc.) : état des feux de circulation = 1 OU 2 OU 3 OU 4.

Question : « Quand une voiture a-t-elle le droit de passer ? »

Réponse : « Feux éteints ou feu vert ».

On peut traduire symboliquement la réponse par :

PASSEZ = 1-OU 4 (1 = éteint, 4 = vert).

En réalité, l'expression 1 OU 4 n'est pas suffisante puisque si les feux sont éteints, il faut encore s'assurer que la voie est libre. La réponse complète sera donc :

PASSEZ = feu vert ou feux éteints ET voie libre.

Considérons cette dernière condition. Le mot ET signifie qu'on doit avoir simultanément les deux situations : feux éteints et voie libre. C'est un nouvel opérateur logique que l'on symbolisera par ET (en anglais AND).

En attribuant à la condition « voie libre » le chiffre 5, on aura :

PASSEZ = 4 OU 1 ET 5.

Mais cette expression est susceptible de deux interprétations :

- 1 - PASSEZ = (4 OU 1) ET 5.
- 2 - PASSEZ = 4 OU (1 ET 5).

L'expression n° 1 signifie que le passage est autorisé si l'on a 4 ou 1 (4 OU 1) et en même temps 5 (ET 5). C'est une interprétation erronée.

L'expression n° 2 signifie que le passage est autorisé si l'on a la condition 4 qui se suffit à elle-même ou (OU) à la fois 1 et 5 (1 ET 5). Interprétation correcte.

Cet exemple fait apparaître l'intérêt qu'il y a d'utiliser correctement les parenthèses, exactement comme dans les formules algébriques. Les opérateurs ET et OU s'appliquent à deux éléments d'un ensemble quelconque. Considérons un exemple à deux chiffres binaires A et B.

Les deux éléments étant des chiffres binaires, ils peuvent prendre la valeur 0 ou 1. Il s'ensuit que les diverses combinaisons entre les valeurs A et B sont les suivantes :

A	B
1	1
1	0
0	1
0	0

En faisant appel aux opérateurs ET et OU, nous aurons :

I	II	III	IV
A	B	A ET B	A OU B
1	1	1	1
1	0	0	1
0	1	0	1
0	0	0	0

Les colonnes I et II (A, B) sont les opérands. La colonne III est le résultat de l'opération A ET B. Ce résultat est égal à 1 quand A et B valent 1.

La colonne IV est le résultat de l'opération A OU B. Ce résultat est égal à 1 quand au moins l'un des deux éléments A ou B vaut 1.

Introduisons maintenant un nouvel opérateur : OU eXclusif (XOR en anglais), que l'on représente par OUX (ou X OU).

Cet opérateur est équivalent à OU, sauf dans le cas où A = 1 et B = 1. Dans ce cas, il donne pour résultat 0, alors que OU donne 1.

En d'autres termes, OUX (OU exclusif) n'équivaut à 1 que si l'un seulement, de A ou de B, a pour valeur 1.

La table de vérité de cet opérateur est par conséquent :

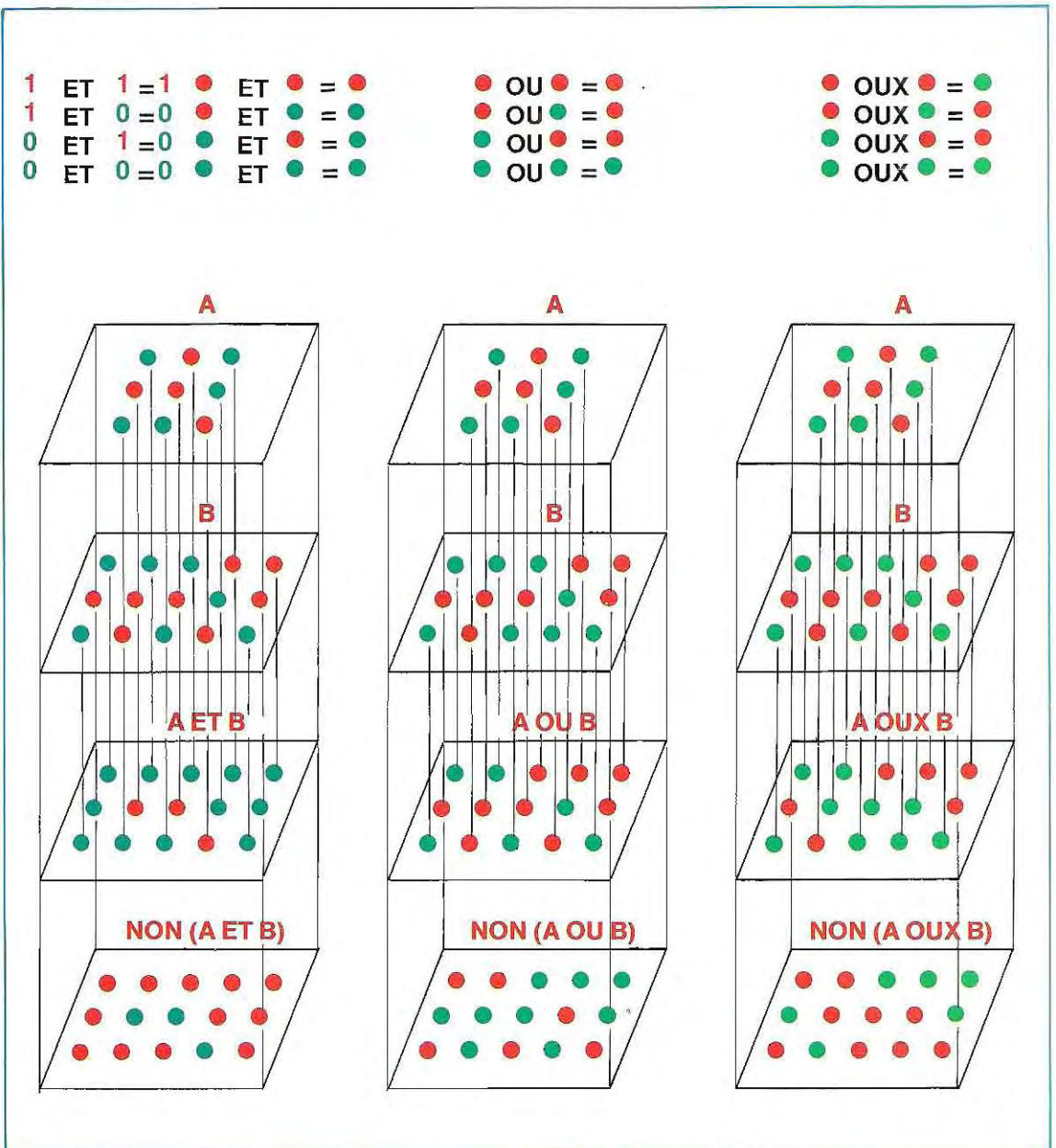
A	B	A OUX B
1	1	0
1	0	1
0	1	1
0	0	0

rande, avec un sens complémentaire, c'est-à-dire qu'il remplace le symbole 1 par le symbole 0, et inversement :

A	NON A
1	0
0	1

Le dernier de nos opérateurs est NON (en anglais NOT). Il s'applique à un seul opé-

Dans le graphique ci-dessous (point rouge = symbole 1 ; point vert = symbole 0), on verra l'application des opérateurs ET, OU, OUX, NON, à deux ensembles binaires.



Pour nous résumer, les principaux opérateurs logiques sont : ET, OU, OUX, NON. Leur table de vérité est celle-ci :

A	B	ET	OU	OUX	NON A	NON B
1	1	1	1	0	0	0
1	0	0	1	1	0	1
0	1	0	1	1	1	0
0	0	0	0	0	1	1

Le traitement des opérateurs logiques suit le même principe que celui des opérateurs arithmétiques.

En outre, les opérations logiques ne s'appliquent qu'à des éléments homogènes.

Jusqu'à maintenant, nous avons considéré les opérateurs logiques comme des nombres binaires à un seul chiffre (A = 1/0, B = 1/0). Mais on peut également les appliquer à des nombres binaires à plusieurs chiffres. Si nous prenons :

A = 01101  
B = 10011

Nous aurons la représentation suivante :

NON A = 10010  
NON B = 01100

A ET B

A	0	1	1	0	1
B	1	0	0	1	1
A ET B	0	0	0	0	1

Les deux nombres binaires A et B, écrits en système décimal, ont respectivement pour valeur :

A = 01101 = 13 décimal ( $2^3 + 2^2 + 2^0$ )

B = 10011 = 19 décimal ( $2^4 + 2^1 + 2^0$ )

et puisque A ET B = 1 (exemple précédent), le résultat doit rester valable si l'on considère, pour A et B, les valeurs décimales :

01101 ET 10011 = 13 ET 19 = 1

A OU B

A	0	1	1	0	1
B	1	0	0	1	1
A OU B	1	1	1	1	1

En décimal 13 OU 19 = 31 ( $11111 = 2^4 + 2^3 + 2^2 + 2^1 + 2^0 = 31$ )

A OUX B

A	0	1	1	0	1
B	1	0	0	1	1
A OUX B	1	1	1	1	0

En décimal 13 OUX 19 = 30 ( $11110 = 30$ ).

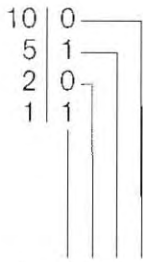
Opératrice travaillant sur un ordinateur à clavier séparé.



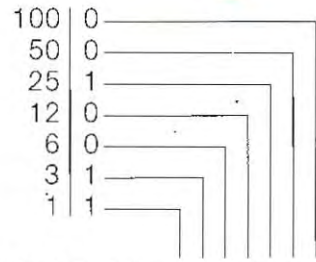
Marka



5 / La conversion s'opère en reportant les restes de la division par 2 :



10 décimal = 1 0 1 0



100 décimal = 1 1 0 0 1 0 0

de la même manière 1000 décimal = 1111101000 binaire = 3E8 hexadécimal.

6 / Binaire	Décimal
0 0 0 1	1
0 0 1 0	2
0 1 0 0	4
1 0 0 0	8

Comme on le voit, dans un nombre binaire, le déplacement d'un chiffre vers la gauche équivaut à une multiplication par deux. De même, un déplacement vers la droite équivaut à une division par deux. On arrive à la même conclusion en observant que, quel que soit le système de numération, un déplacement vers la gauche équivaut à une multiplication par le nombre qui sert de base. Dans le langage de la programmation, un déplacement prend le nom de ROTATION.

7 / a - vrai b - vrai c - faux d - vrai

8 / La conversion d'un nombre octal en décimal consiste à multiplier chaque chiffre octal par la puissance de 8 correspondante. Prenons pour exemple la conversion du nombre octal 700 :

$$700 \text{ octal} = 7 \times 8^2 + 0 \times 8^1 + 0 \times 8^0 = 7 \times 64 + 0 + 0 = 448$$

9 / La conversion d'un nombre hexadécimal est plus simple si l'on passe par la notation binaire.

Par exemple, le nombre hexadécimal 3FB correspond à

$$\begin{aligned} 3 &= 0 1 1 \\ F &= 1 1 1 1 \\ B &= 1 0 1 1 \end{aligned}$$

par conséquent :  $3FB = 1 1 1 1 1 1 1 0 1 1 = 1019$  décimal

On obtiendra le même résultat en multipliant chaque nombre hexadécimal par la puissance de 16 correspondante :

$$3FB = 3 \times 16^2 + 15 \times 16^1 + 11 \times 16^0 = 3 \times 256 + 15 \times 16 + 11 = 1019$$

10 /

	2 chiffres = 1 1 = décimal 3
	4 chiffres = 1 1 1 1 = décimal 15
	8 chiffres = 1 1 1 1 1 1 1 1 = décimal 255

### LISTING 3

I	J	ET	OU	OUX	J + UNON(I) + 1
11	65	1	75	74	54
11	66	2	75	73	55
11	67	3	75	72	56
11	68	0	79	79	57
11	69	1	79	78	58
11	70	2	79	77	59
11	71	3	79	76	60
11	72	8	75	67	61
11	73	9	75	66	62
11	74	10	75	65	63
11	75	11	75	64	64
11	76	8	79	71	65
11	77	9	79	70	66
11	78	10	79	69	67
11	79	11	79	68	68
11	80	0	91	91	69
11	81	1	91	90	70
11	82	2	91	89	71
11	83	3	91	88	72
11	84	0	95	95	73
11	85	1	95	94	74
11	86	2	95	93	75
11	87	3	95	92	76
11	88	8	91	83	77
11	89	9	91	82	78
11	90	10	91	81	79

### LISTING 4

I	J	ET	OU	OUX	J + UNON(I) + 1
32	65	0	97	97	33
32	66	0	98	98	34
32	67	0	99	99	35
32	68	0	100	100	36
32	69	0	101	101	37
32	70	0	102	102	38
32	71	0	103	103	39
32	72	0	104	104	40
32	73	0	105	105	41
32	74	0	106	106	42
32	75	0	107	107	43
32	76	0	108	108	44
32	77	0	109	109	45
32	78	0	110	110	46
32	79	0	111	111	47
32	80	0	112	112	48
32	81	0	113	113	49
32	82	0	114	114	50
32	83	0	115	115	51
32	84	0	116	116	52
32	85	0	117	117	53
32	86	0	118	118	54
32	87	0	119	119	55
32	88	0	120	120	56
32	89	0	121	121	57
32	90	0	122	122	58



Dans les listings 3 et 4 (p. 78), on retrouve les applications des opérateurs ET, OU, OUX et de l'opérateur composé  $J + (\text{NON } I) + 1$ , aux deux nombres  $I = 11$  et  $J$  de 65 à 90 (listing 3), et  $I = 32$ ,  $J$  de 65 à 90 (listing 4).

La série des nombres situés entre 65 et 90 peut, dans un ordinateur, prendre une signification très particulière. Dans les listings 3 et 4, la colonne  $J + (\text{NON } I) + 1$  représente la différence entre les deux nombres  $J$  et  $I$ .

L'expression  $J + (\text{NON } I) + 1$  constitue en fait la différence entre  $J$  et  $I$ , calculée selon la méthode du complément ( $\text{NON } I$  est le complément de  $I$ ).

### Logique câblée

Les opérateurs logiques peuvent s'appliquer aux signaux électriques de la même manière qu'ils s'appliquent aux symboles **1** et **0**.

Les signaux électriques numériques (voir le paragraphe consacré aux impulsions, page 45 et suivantes) sont la représentation physique des symboles **1** et **0**. En utilisant la même analogie :

**signal électrique haut = 1**

**signal électrique bas = 0**

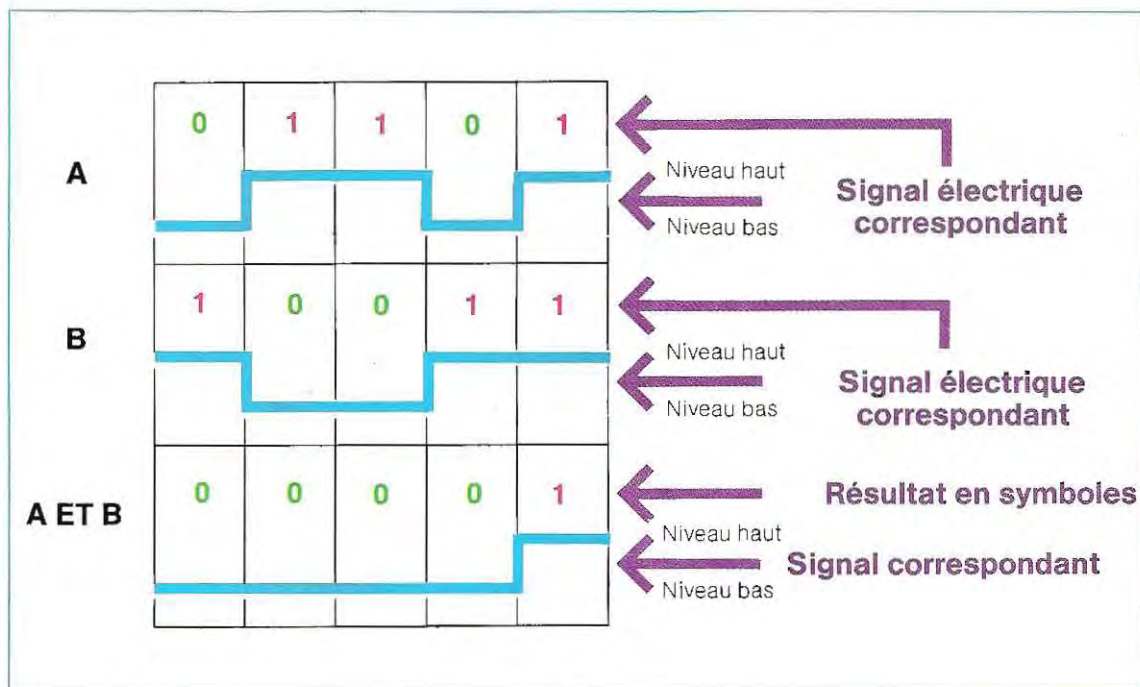
les valeurs binaires  $A = 01101$ ,  $B = 10011$  sont représentées par des signaux de niveau

**haut** pour les symboles **1** et de niveau **bas** pour les symboles **0** (voir le schéma en bas de cette page).

L'opération qui permet d'obtenir le signal  $A \text{ ET } B$  (par traitement des deux signaux  $A$  et  $B$ ) est effectuée par des circuits numériques spécialisés que l'on appelle les **circuits logiques**.

Chacun de ces circuits a une fonction qui lui est propre. A chaque opérateur logique correspond un circuit spécifique. Donc, en théorie, il nous faut construire un appareil particulier, renfermant un circuit pour chacune des fonctions logiques dont nous avons besoin. Concevoir un circuit, c'est relier électriquement entre eux un certain nombre de composants. On obtient ainsi un circuit intégré. Cette opération de connexion s'appelle un câblage. Et le circuit prend alors le nom de **logique câblée**.

On voit immédiatement l'inconvénient de ce type de logique : il faut autant de circuits que de fonctions. A l'inverse, l'ordinateur électronique se présente comme une machine à **logique programmable**. Il dispose en effet de circuits capables d'accomplir diverses opérations selon des ordres (**instructions**) dictés par le programmeur. Toutefois, de nombreuses fonctions annexes sont confiées à des logiques câblées.



## Circuits logiques

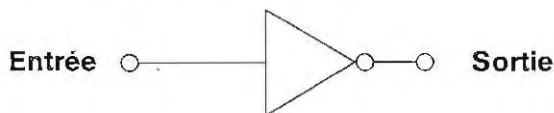
Les principaux opérateurs logiques sont :

- **NON**
- **ET**
- **OU**
- **OUX**

Ces fonctions sont réalisées par des circuits spécifiques. A chaque opérateur correspond un circuit électronique capable de réaliser la fonction définie.

### Opérateur NON

L'opérateur NON est celui qui inverse une donnée : la valeur **1** devient **0**, et vice versa. Le circuit capable de réaliser la fonction NON comporte donc une sortie pouvant prendre la valeur **0** (ou **L**) quand l'entrée est **1** (ou **H**). Si l'entrée varie, dans un temps donné, de l'état **1 (H)** à l'état **0 (L)**, la sortie varie en sens opposé, de l'état **0 (L)** à l'état **1 (H)**. Ce circuit est symbolisé de la façon suivante :

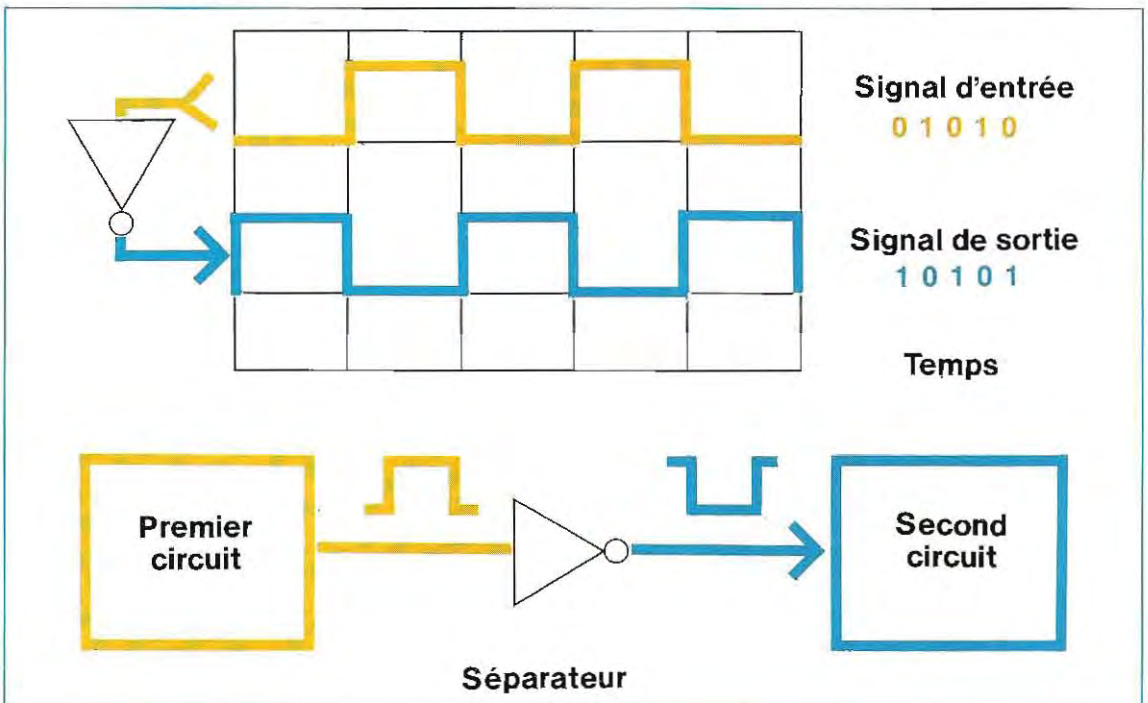


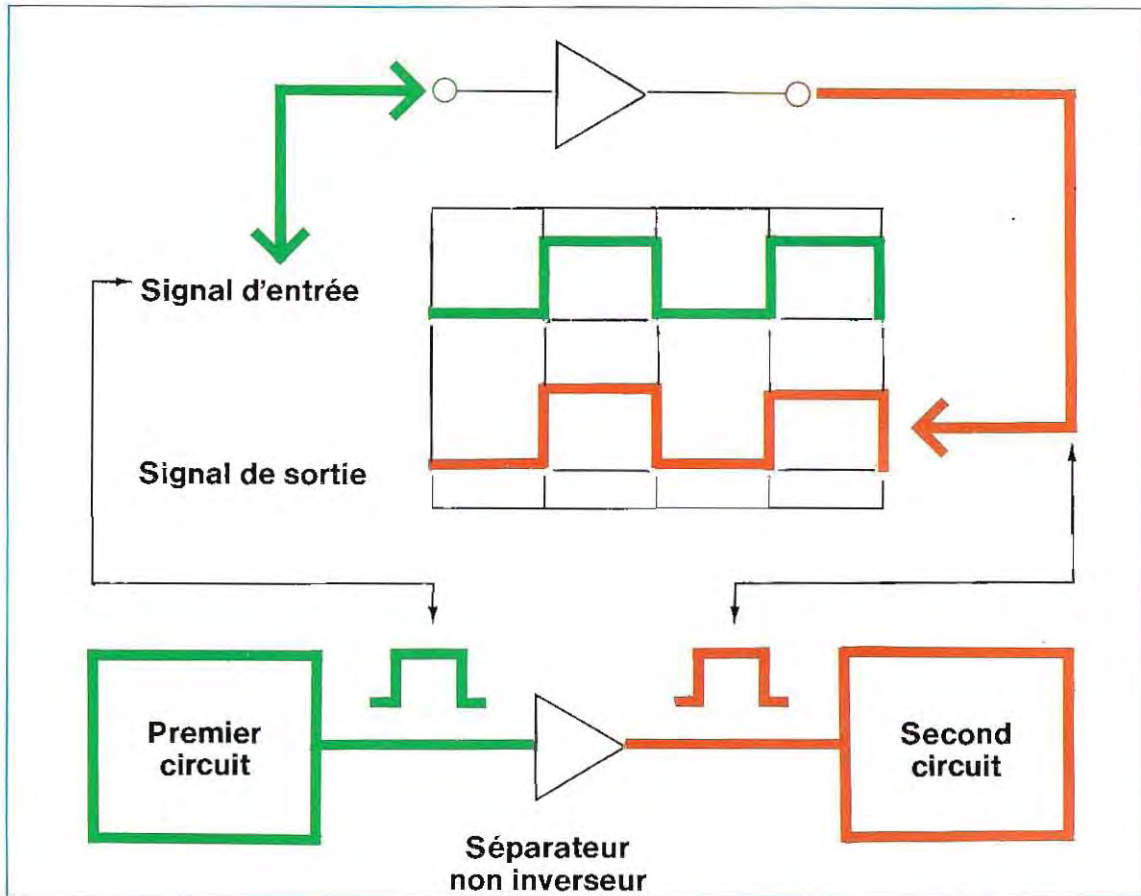
Introduisons en entrée une horloge (voir p. 47) : la sortie fournira l'opposé du signal d'entrée.

On trouvera dans le schéma au bas de la page le parcours des signaux d'entrée et de sortie. Ce circuit a pour fonction, non seulement d'inverser le signal, mais aussi de séparer des circuits d'un autre type. Il faut expliquer cette fonction : dans les ordinateurs électroniques, chaque élément a des fonctions complexes et comporte des circuits extrêmement sophistiqués qui renferment des dizaines de milliers de composants sur quelques millimètres carrés. On parvient à cette forte densité de composants (intégration) grâce à une technologie de pointe.

Les circuits sont conçus en vue d'une certaine fonction, mais ils sont incapables de générer de la puissance. Aussi, lorsqu'il s'agit de faire fonctionner un circuit qui requiert de la puissance, il faut interposer un troisième composant, appelé amplificateur, qui a pour but d'isoler l'un de l'autre les deux premiers et de donner de la puissance à un signal de sortie. Le schéma logique est représenté dans le graphique en bas de la page. Dans ce graphique, le séparateur est à la fois un inverseur remplissant la fonction NON (il représente à la sortie un signal inversé par rapport à celui d'entrée) et un amplificateur.

Dans certains cas, l'inversion est gênante, c'est pourquoi on a construit des circuits qui séparent, sans inverser le signal.





Dans le graphique ci-dessus, on voit le symbole, le parcours des signaux et le schéma logique d'application du circuit en question. Quant à l'amplificateur, c'est un circuit qui délivre un signal de sortie proportionné à celui d'entrée, mais d'intensité plus élevée.

On parle d'amplificateurs

- de signal : l'amplification s'applique à la valeur du signal ;
- de puissance : l'amplification s'applique à la puissance.

Prenons un exemple.

Imaginons un ordinateur qui commande une imprimante placée dans une pièce éloignée. Pour transmettre les informations de l'ordinateur à l'imprimante, on a recours à une liaison par câble.

En raison de la résistance du câble et des perturbations électriques provenant de l'extérieur, les signaux parviendront à l'imprimante déformés, et à un niveau trop bas. Un amplificateur est alors nécessaire. On le place immédiatement avant l'imprimante, et il a pour tâche de "régénérer" le niveau et la forme des signaux.

Par ailleurs, pour des raisons de sécurité, comme les circuits de l'ordinateur ne peuvent pas être reliés directement à une ligne de transmission (risques de court-circuit), il faut prévoir un deuxième amplificateur placé entre l'ordinateur et la ligne. Le schéma au bas de la page 82 illustre ce système et les formes d'onde utilisées.

Pour une application de ce type (transmission d'information par ligne), il existe des amplificateurs (line drivers) appropriés.

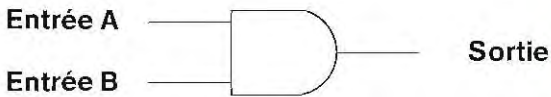
En réalité, les techniques de reconstitution des signaux sont beaucoup plus complexes qu'il n'apparaît sur le schéma.

Pour en terminer avec ce sujet, notons que ces circuits (buffer) s'emploient beaucoup, tant à l'intérieur des ordinateurs (pour isoler les uns des autres les différents composants) qu'à l'extérieur (pour faire la liaison entre l'ordinateur et l'extérieur).

### Opérateur ET

Ici, nous avons affaire à un circuit à double entrée, puisque l'opérateur ET exige deux

opérandes (A ET B). Le symbole graphique de ce circuit est le suivant :

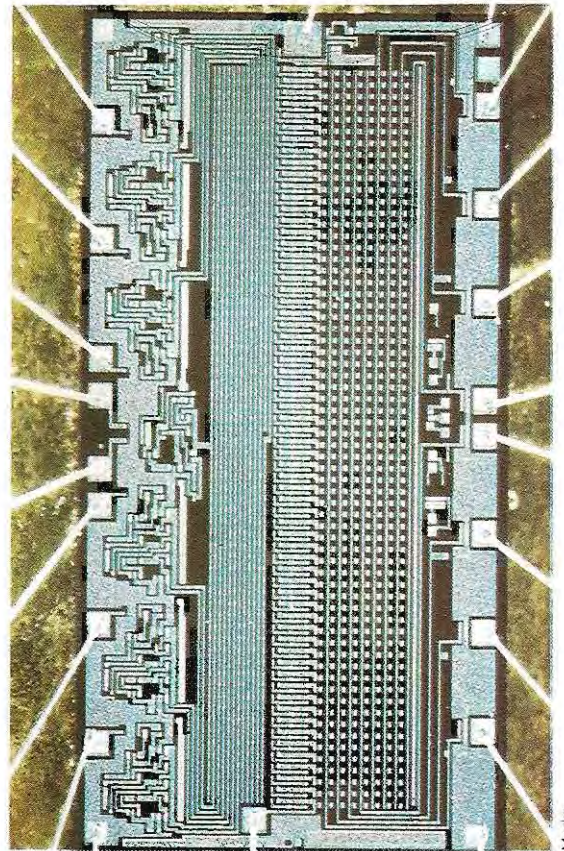


Il fonctionne exactement comme l'opérateur ET. Il faut se souvenir que le symbole **1** représente le signal à haut niveau = **H** et que le symbole **0** représente le signal à bas niveau = **L**. La table de vérité immédiatement déduite de la table de l'opérateur ET (en remplaçant **1** par **H** et **0** par **L**) est par conséquent :

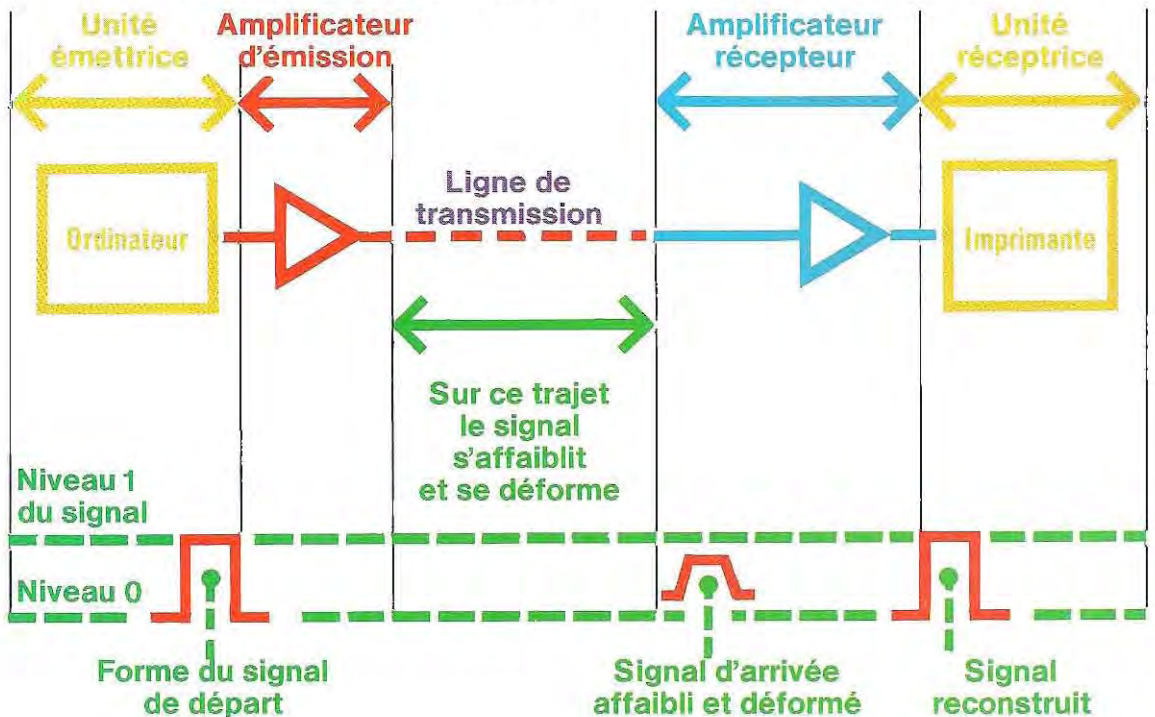
A	B	Sortie = A ET B
H	H	H
H	L	L
L	H	L
L	L	L

Le schéma reste inchangé si on inverse les symboles (on remplace **1** par **L** et **0** par **H**). Dans ce cas, on a un circuit dont la logique est inversée, mais le fonctionnement reste le même.

Intérieur d'un circuit intégré.



### SCHEMA DE LA TRANSMISSION DE SIGNAUX PAR LIGNE



Dans le schéma au bas de cette page, on a figuré les formes des signaux.

Dans le même schéma, on trouve également la version «inverseur», que l'on appellera NON ET (en anglais NAND).

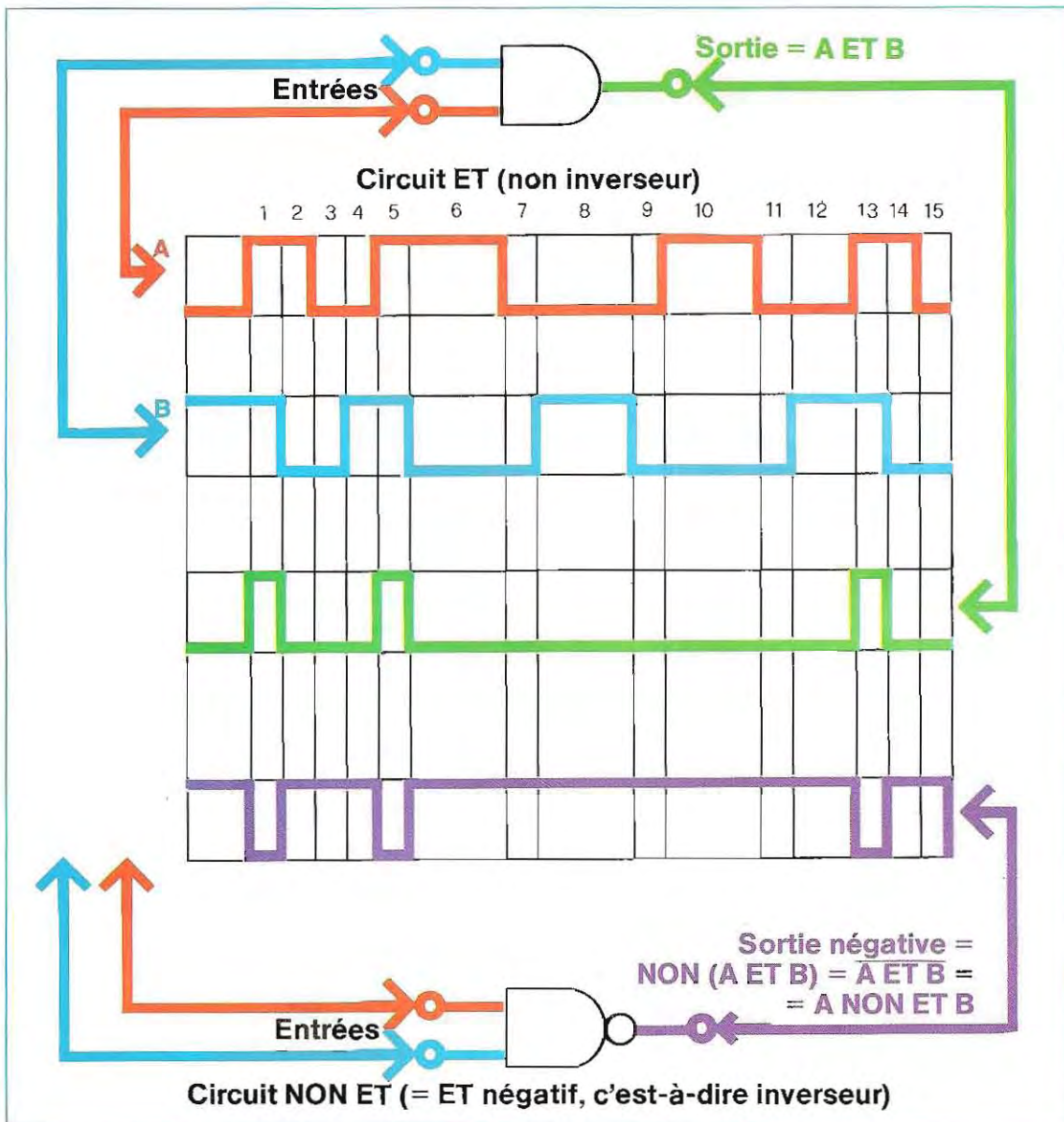
Pour le circuit NON ET, la table de vérité devient:

C'est l'opposé de la table du circuit ET.

Enfin, si l'une des deux entrées (A par exemple), se trouve bloquée à l'état **H**, le circuit NON ET se transforme en inverseur. Si l'on retranche de la table de vérité les situations correspondant à l'état  $A = L$  (qui n'existe pas, puisque A est bloqué en **H**), on a :

A	B	NON ET
H	H	L
H	L	H
L	H	H
L	L	H

A	B	NON ET
H	H	L
H	L	H
—	—	—
—	—	—



## Comment consulter un ordinateur

Nous avons à plusieurs reprises fait allusion aux applications possibles des ordinateurs à des tâches de gestion. Nous allons approfondir cette question et étudier, en particulier, comment procéder pour obtenir l'accès à un ensemble de données stockées en mémoire.

On a souvent besoin de consulter de grands fichiers informatisés selon une organisation très complexe. Comme nous l'avons vu, c'est justement l'intérêt de l'informatisation.

Mais dans la vie courante aussi, des ordinateurs plus simples et plus personnalisés se révèlent également utiles. Dans la vie moderne en général, ce sont des fichiers informatisés qui épargnent de longues et fastidieuses recherches. Il peut s'agir aussi bien d'une bibliographie portant sur un sujet spécifique, que l'utilisateur met en mémoire dans un ordinateur, que d'un répertoire téléphonique à usage domestique.

Il faut dans tous les cas un programme permettant d'insérer au fur et à mesure des données nouvelles et permettant de les consulter simplement et rapidement. Il serait trop long et encore un peu difficile à comprendre pour le lecteur d'expliquer ici le fonctionnement de ces programmes, mais nous pouvons néanmoins exposer les principes logiques.

Nous avons vu par exemple (p. 16) comment obtenir un numéro de téléphone figurant dans

une liste mise en mémoire. On peut dire que c'est une recherche qui procède par sélections successives à plusieurs niveaux. Ces sélections sont de plus en plus « affinées » et portent sur des champs de plus en plus restreints, qui aboutissent à une réponse unique. Prenons un exemple et suivons cette recherche.

Nous voulons trouver le numéro de téléphone de Michel Martin, qui habite au n° 6 de la rue M.

### Nom = Martin

Notre ordinateur va comparer les noms figurant dans sa mémoire, et sélectionner ceux qui satisfont à la condition demandée : nom = Martin. Ce premier niveau est très peu sélectif, et nous allons obtenir un grand nombre de réponses. Il faut donc procéder à une nouvelle sélection, dite de deuxième niveau, et introduire une donnée supplémentaire : le prénom.

### Prénom = Michel

Si nous posons les deux questions de façon indépendante, nous obtiendrions deux listes : celle des abonnés au téléphone de nom patronymique Martin, et celle des abonnés de prénom Michel. Ce n'est pas ce que nous voulons. Il faut donc lier les deux opérations au moyen de l'opérateur logique ET, afin que la

E.B.S. Procédure de recherche par sujets  
Nom de l'utilisateur ? SYLVIANE GUYARD\_

E.B.S. Procédure de recherche par sujets  
Nom de l'utilisateur ? SYLVIANE GUYARD.

1 - HISTOIRE	8 - ASTRONOMIE
2 - GÉOGRAPHIE	9 - BIOLOGIE
3 - BEAUX-ARTS	10 - PHYSIQUE
4 - SCIENCE	11 - ARCHÉOLOGIE
5 - TECHNOLOGIE	
6 - MATHÉMATIQUES	
7 - LITTÉRATURE	

ENTRER LES RUBRIQUES A SÉLECTIONNER

Sujet - Opérateur - Sujet

réponse (ou les réponses) satisfasse à la condition :

**Nom = Martin (premier niveau)**  
**ET**  
**Prénom = Michel (deuxième niveau)**

Mais nous avons besoin d'introduire un troisième critère de sélection, l'adresse, qui sera, elle aussi, vérifiée par rapport à d'autres adresses. La condition globale à fournir à l'ordinateur sera :

**Nom = Martin**  
**(premier niveau)**  
**ET**  
**Prénom = Michel**  
**(deuxième niveau)**  
**ET**  
**Adresse = n° 6 rue M.**  
**(troisième niveau)**

L'ordinateur doit alors nous donner le numéro de téléphone désiré, sauf s'il existe deux Michel Martin à l'adresse indiquée, mais on peut aussi prévoir ce cas.

Supposons maintenant que nous connaissons le nom et l'adresse de la personne, mais que nous ayons un doute quant au prénom. Monsieur Martin se prénomme peut-être Michel, ou Frédéric.

Nous allons cependant pouvoir faire notre recherche. Notre question sera, par consé-

quent, exprimée en ces termes :

- 1) **Nom = Martin**
- 2) **Prénom = Michel Ou Frédéric**
- 3) **Adresse = n° 6 rue M.**

Ces conditions doivent être simultanément remplies. La condition 2 se transcrit dans les termes que nous connaissons déjà :

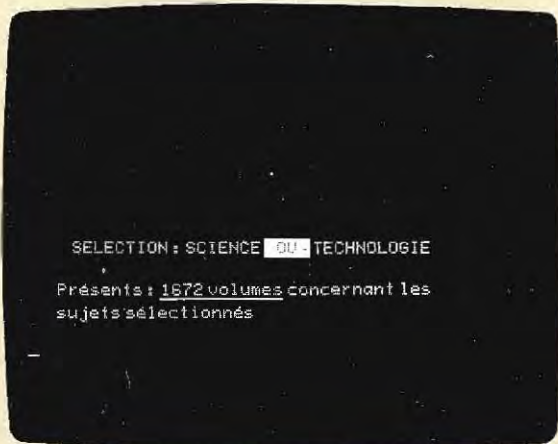
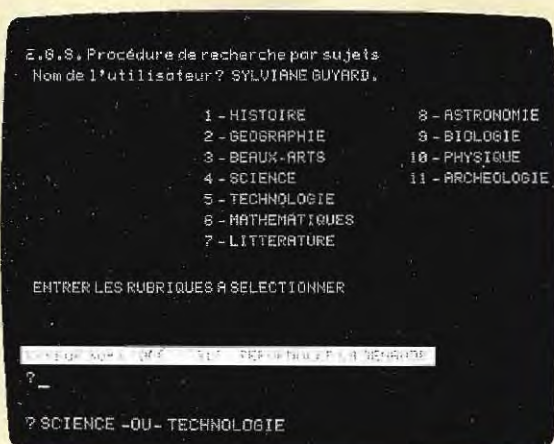
**Nom = Michel OU Frédéric**

La question globale est donc celle-ci :

**Nom = Martin-ET-Prénom =**  
**Michel-OU-Frédéric-ET-adresse =**  
**n° 6 rue M.**

Le doute que nous avons quant au prénom ne nous empêche pas d'obtenir un renseignement fiable, puisque la condition 3 viendra compléter la condition 2. En effet, l'ordinateur va explorer la zone du fichier qui contient les Martin. Il s'arrêtera dès qu'il rencontrera un Michel ou un Frédéric, et vérifiera (ou sélectionnera) en fonction de l'adresse fournie dans la question.

Comme dans le cas précédent, nous avons recherché une information caractérisée de manière univoque par la condition contenue dans la question.





**En stockant dans l'ordinateur les données bibliographiques, on facilite les recherches de la bibliothécaire.**

La situation serait différente si nous n'avions que le nom, sans prénom ni adresse. L'ordinateur nous fournirait en sortie les numéros de téléphone de tous les Martin figurant sur la liste. Liste très longue sans doute, puisque nous aurions demandé une information sans fournir les paramètres permettant de la cerner de manière univoque.

Ces programmes ont des applications multiples. Prenons pour exemple les grandes bibliothèques dans lesquelles chaque texte du fond documentaire est enregistré sous le

nom de l'auteur, et selon le titre, le thème, la langue dans laquelle il est écrit, la localisation sur les rayonnages de la bibliothèque, la disponibilité à la date de la demande, et bien d'autres renseignements encore. Cette gestion informatisée évite de longues et fastidieuses recherches, souvent génératrices d'erreurs et d'oublis.

Grâce aux fichiers informatisés, ce travail est aujourd'hui simplifié. Les opérateurs permettent d'organiser les informations archivées en mémoire, de façon à fournir aisément le ren-

Les sujets SCIENCE - TECHNOLOGIE contiennent les rubriques :

- 1 - PHYSIQUE
- 2 - OPTIQUE
- 3 - ELECTRONIQUE
- 4 - MECANIQUE
- 5 - CONSTRUCTION
- 6 - ELECTROMECHANIQUE
- 7 - INFORMATIQUE

ENTRER LES RUBRIQUES A SELECTIONNER

Sujet - Opérateur - Sujet

? ELECTRONIQUE - ET - INFORMATIQUE \_

SELECTION : ELECTRONIQUE ET INFORMATIQUE

Présents : 251 volumes sur les sujets sélectionnés



seignement demandé. On verra par exemple dans nos illustrations comment utiliser un ordinateur pour établir une bibliographie sur un sujet donné – ici le thème concerne les « mémoires informatiques ». Le bibliothécaire moderne demande au clavier la mise en mémoire du programme de recherche à utiliser. En réponse, l'ordinateur demande le nom de l'utilisateur. Cette demande peut avoir des motifs variés. Il peut s'agir d'interdire la consultation à des personnes non agréées ; ou bien de limiter la recherche de certains textes réservés ; ou tout simplement d'inscrire le nom du chercheur en tête de la liste qui constituera le produit final, et d'éviter ainsi toute confusion entre plusieurs recherches consacrées à des sujets proches. La machine fait alors apparaître sur l'écran un menu : les différentes rubriques où se retrouve le sujet de la recherche. Les sections qui nous intéressent peuvent être Science et Technologie. Nous aurons donc à fournir les conditions :

### Science - OU - Technologie

L'ordinateur nous indique les commandes à donner, et fait apparaître sur l'écran :

### Sujet - Opérateur - Sujet

Il faut respecter la syntaxe correcte pour que l'ordinateur reconnaisse l'instruction. Une erreur de formulation entraînerait l'annulation de l'instruction et ferait apparaître un message d'erreur. Il faudrait reformuler l'instruction.

Une fois reçue la première condition de sélection, la machine répond en donnant le nombre des textes relatifs au sujet choisi. Si ces textes ne sont pas trop nombreux, l'utilisateur peut demander la liste des ouvrages et leurs caractéristiques. Ici, notre liste est trop longue. Il faut donc affiner notre investigation. Nous allons en informer la machine en lui donnant une nouvelle instruction.

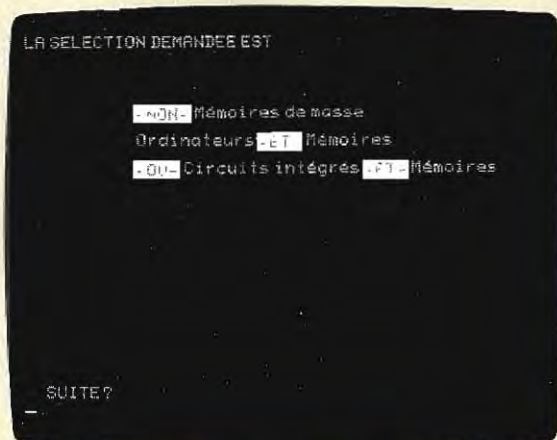
L'ordinateur va nous présenter sur l'écran les sous-sections qui composent la section générale Science - Technologie. Si le sujet qui nous intéresse peut rentrer dans la sous-section Electronique et Informatique, nous allons donner l'instruction :

### Electronique - ET - Informatique

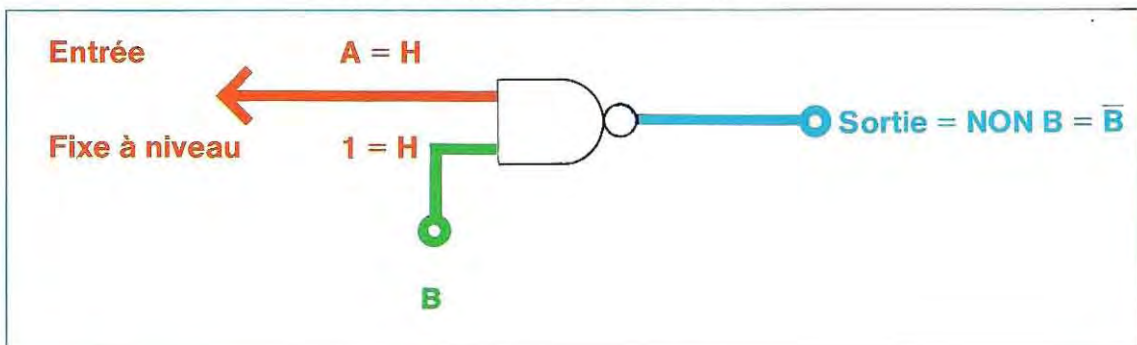
Ce nouveau tri va permettre une réduction du nombre des titres. Nous allons opérer la sélection définitive en donnant l'instruction :

### Circuits intégrés - ET - Mémoires

Si nous avons donné pour seule instruction Mémoires, nous aurions trouvé en sortie tous les textes qui ont trait aux mémoires, parmi lesquelles les bandes magnétiques et les disques. La caractéristique supplémentaire de «Circuits intégrés» nous permet de sortir seulement les textes relatifs aux mémoires électroniques et aux circuits intégrés. L'ordinateur a maintenant tous les éléments nécessaires pour mener à bien la recherche.



La sortie est l'opposé de l'entrée B, ou, en d'autres termes, la négation de B (NON B). En langage symbolique, si l'entrée est B, la sortie fournie par le circuit, c'est-à-dire l'opposé de B (négation de B), se transcrita par  $\bar{B}$ . C'est ce qu'indique le schéma suivant:



On voit dans ce schéma deux entrées générales (A et B) en forme d'ondes non périodiques. Pour en déduire la sortie, il faut observer que le signal de haut niveau est équivalent à **1**, et le signal de bas niveau est équivalent à **0**. De cette façon, comme le montrait le schéma page 83, nous avons :

Segment 1 :

les niveaux de A et B sont hauts : **A = 1, B = 1**.  
Donc A ET B = **1**, c'est-à-dire haut : **H**.

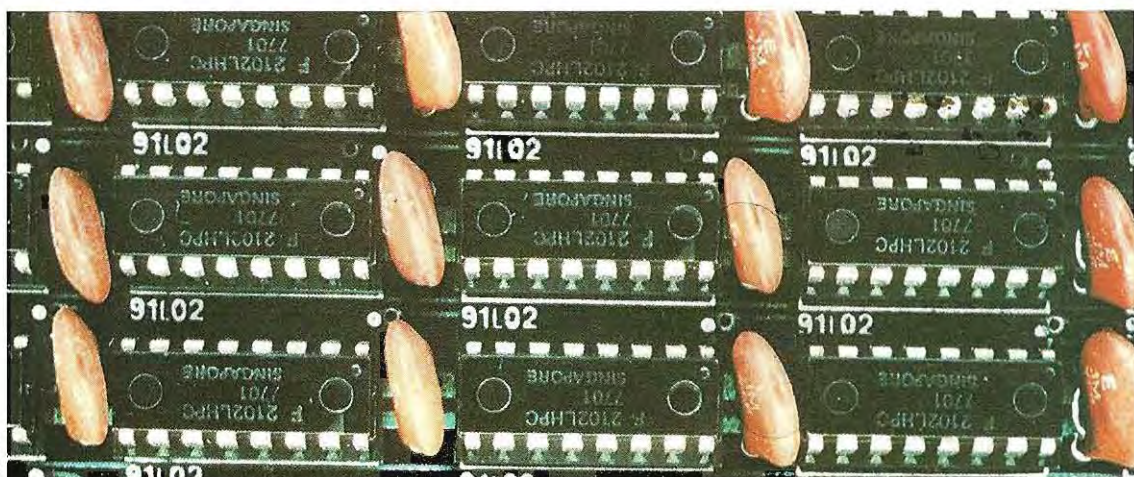
Segment 2 :

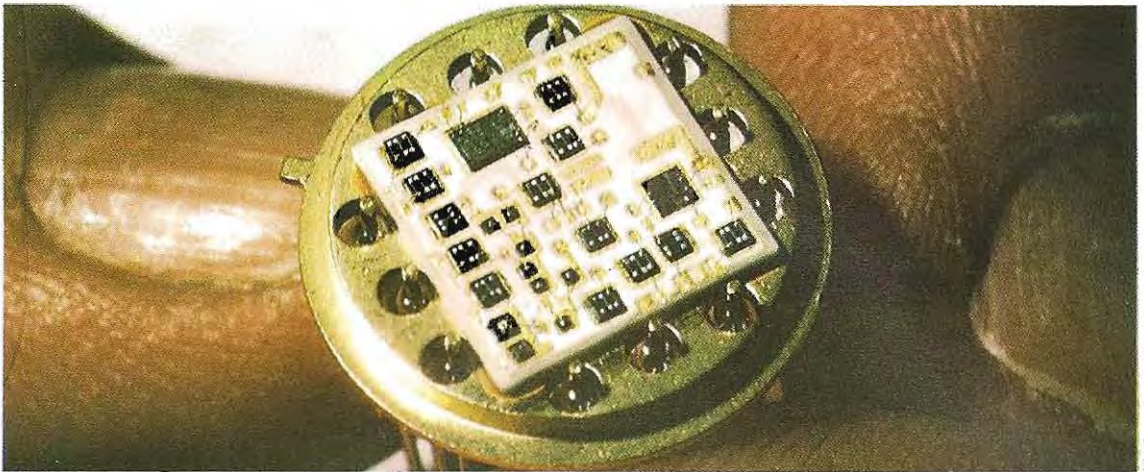
A est haut, A = **1** ; B est bas, B = **0**. Nous avons donc : A ET B = **0**, c'est-à-dire bas : **L**.

Dans le tableau à droite, nous avons réuni les 15 segments du schéma de la page 83.

Segment	A	B	A ET B	$\bar{A}$ ET $\bar{B}$
1	1	1	1	0
2	1	0	0	1
3	0	0	0	1
4	0	1	0	1
5	1	1	1	0
6	1	0	0	1
7	0	0	0	1
8	0	1	0	1
9	0	0	0	1
10	1	0	0	1
11	0	0	0	1
12	0	1	0	1
13	1	1	1	0
14	1	0	0	1
15	0	0	0	1

### Circuits intégrés regroupés sur une carte.





Marka

Une puce de silicium-saphir, très agrandie.

### Opérateur OU

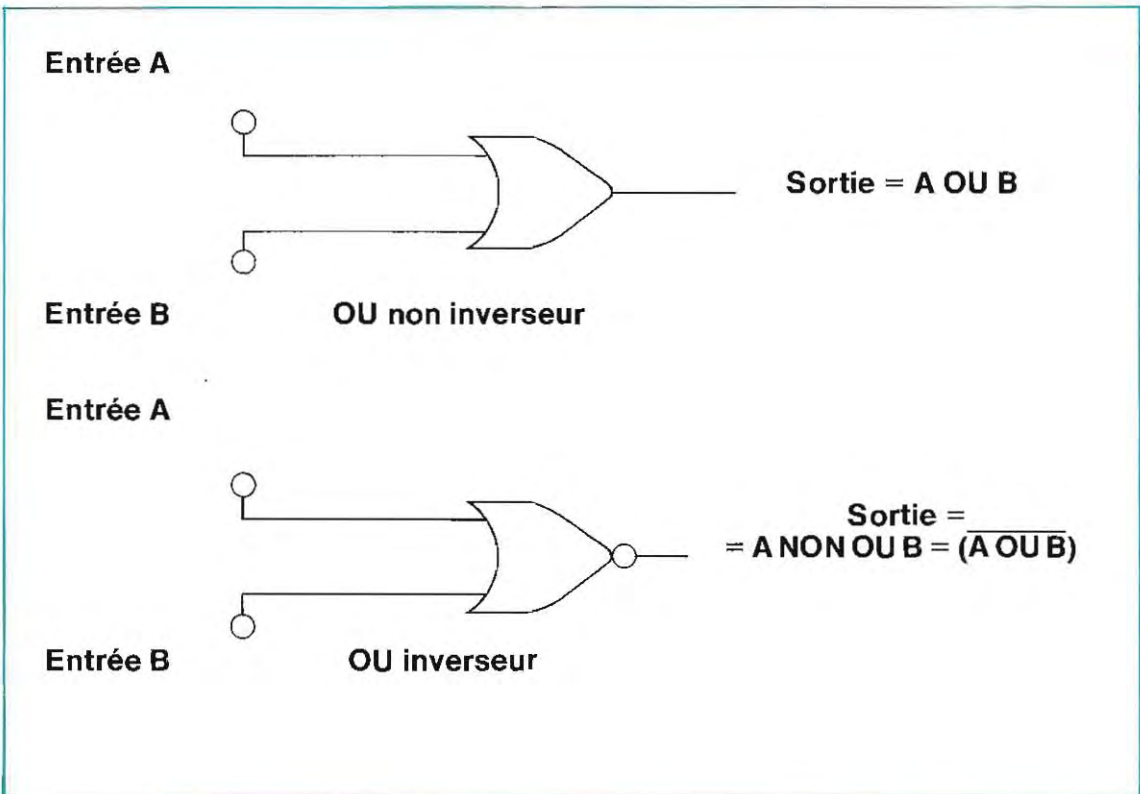
En utilisant la notation habituelle : 1 = **H** et 0 = **L**, le circuit qui réalise la fonction OU a la même table de vérité que l'opérateur.

Ce type de circuit existe également en non inverseur et inverseur. Leurs symboles respectifs figurent sur le schéma ci-dessous.

La table du OU inverseur s'obtient en inversant la table du OU non inverseur :

A	B	OU	$\overline{OU}^*$
<b>H</b>	<b>H</b>	<b>H</b>	<b>L</b>
<b>H</b>	<b>L</b>	<b>H</b>	<b>L</b>
<b>L</b>	<b>H</b>	<b>H</b>	<b>L</b>
<b>L</b>	<b>L</b>	<b>L</b>	<b>H</b>

\* Le trait horizontal au-dessus d'un symbole d'opérateur logique indique qu'il s'agit du même opérateur, mais inversé, c'est-à-dire négatif.  $\overline{OU}$  signifie OU négatif (inversé).



Le schéma de cette page figure un exemple de forme d'onde (de signal). Ici aussi, pour les deux entrées, nous avons supposé la présence d'un signal non répétitif (non périodique).

### Opérateur OUX

Pour ce circuit, on applique la table de vérité de l'opérateur qui lui donne son nom. Les symboles sont représentés dans le schéma en haut de la page 91.

Pour ce circuit, il est plus difficile de trouver des constructeurs qui fabriquent le type in-

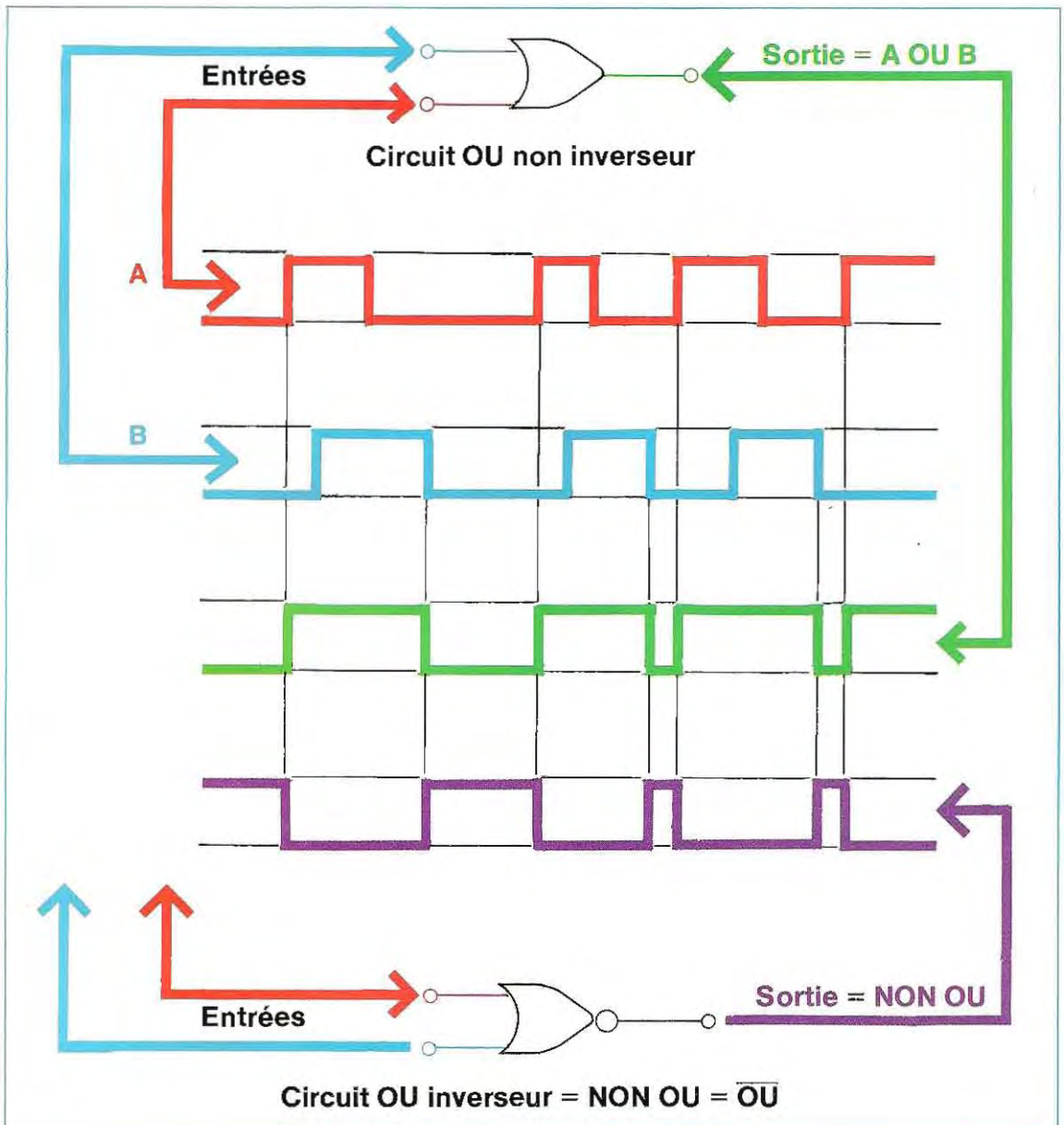
verseur ( $\overline{A \text{ OUX } B}$ ). On peut réaliser cette fonction en adjoignant un inverseur au circuit OU non inverseur.

Le schéma en bas de la page 91 figure les connexions et les formes de signaux.

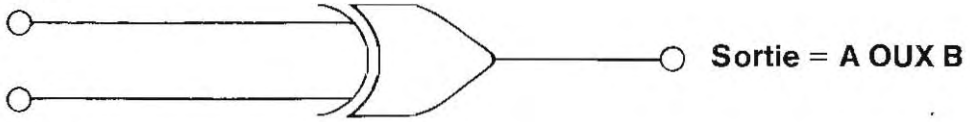
### Autres types d'inverseurs

Dans le circuit amplificateur (inverseur ou non), il y a une entrée et une seule.

Tous les autres circuits en ont deux (ou plus de deux). On peut donc les ramener au premier circuit (sauf le type OUX) en reliant symboliquement les deux entrées entre elles.



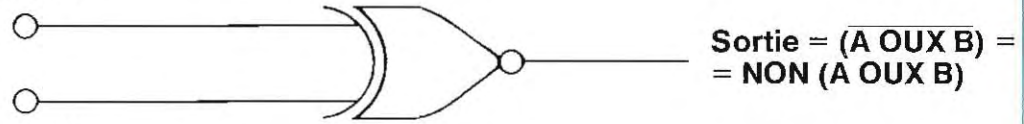
Entrée A



Entrée B

OU exclusif non inverseur

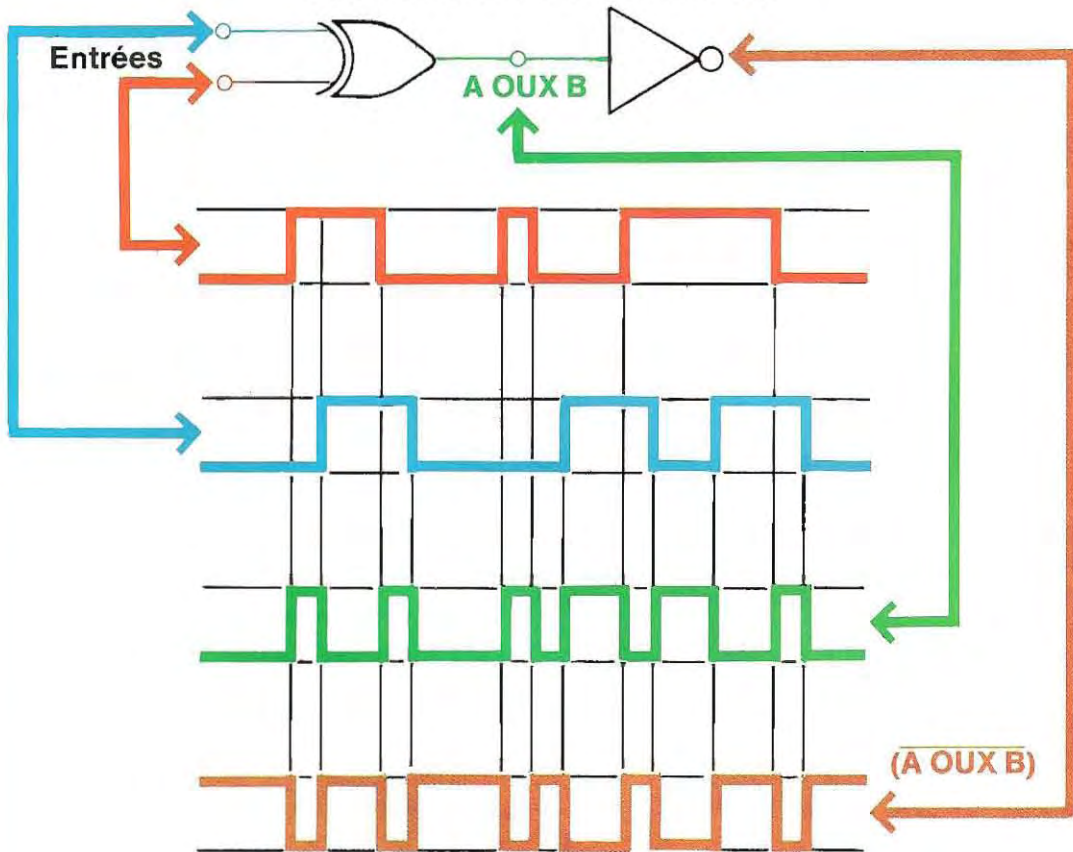
Entrée A



Entrée B

OU exclusif inverseur

CIRCUIT OUX AVEC INVERSEUR



## La banque de l'avenir

*Dans le monde entier, les banques s'efforcent de mettre au point les services qu'elles vont offrir à leur clientèle grâce à l'introduction de techniques nouvelles. Il est clair que les transactions bancaires vont en être radicalement transformées.*

*Il est encore un peu difficile de préciser ce que seront ces nouveaux services dans le détail, mais nous pouvons d'ores et déjà affirmer qu'ils rendront vite obsolètes les actuelles « calculettes ».*

*Les confédérations industrielles et commerciales s'inquiètent d'ailleurs de ces projets encore secrets. A juste titre : l'esprit inventif et l'ambition des jeunes générations de techniciens semblent sans limites. En revanche, ce qui est certain, c'est que les banques vont fonctionner dans un proche avenir selon des formules de self-service. Les clients pourront opérer leurs transactions pratiquement sans sortir de chez eux.*

*Cela ne signifie pas nécessairement la disparition des banques comme établissements*

*ouverts au public. Mais, avec l'informatisation des opérations de routine, certains changements interviendront ; le personnel bancaire sera plus disponible pour des services plus sophistiqués et plus personnalisés.*

\* \* \*

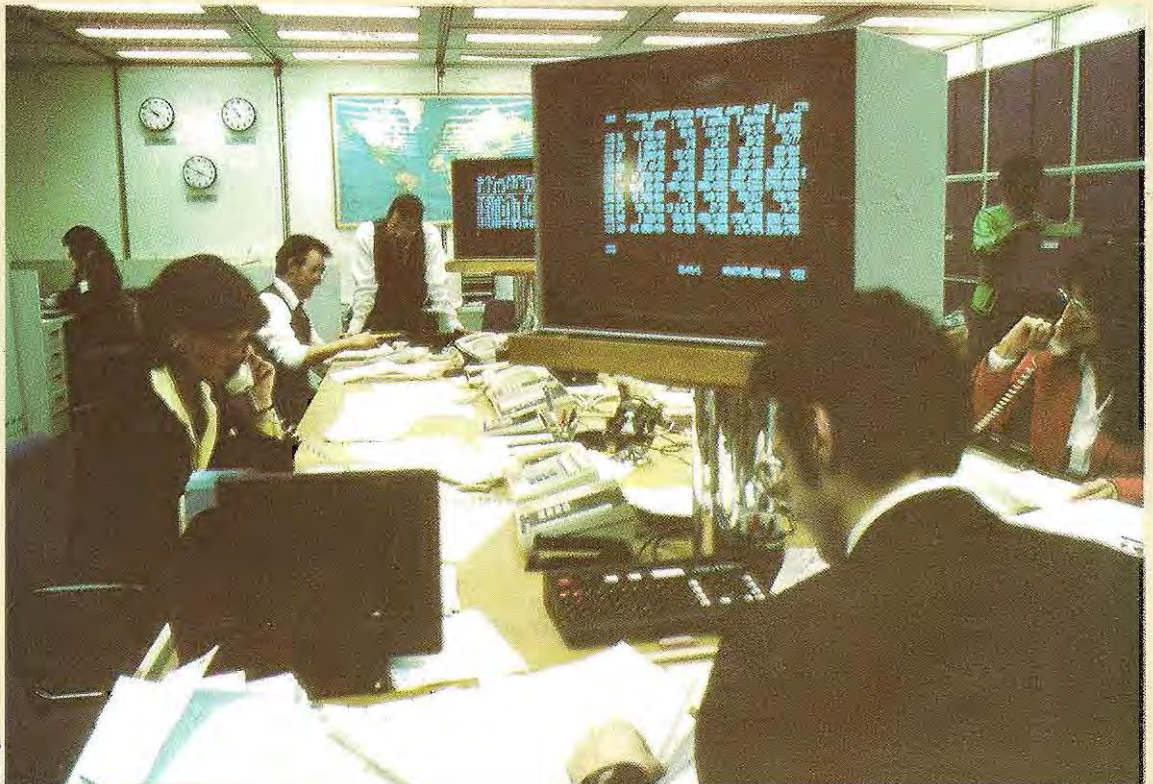
*Dans de nombreux pays, certaines banques ont décidé de tenter ce type d'expérience. La Barclays Bank est l'une des banques les plus à l'avant-garde pour les services « à domicile ». Nombre de ses clients disposent déjà de terminaux personnels chez eux ou à leur bureau.*

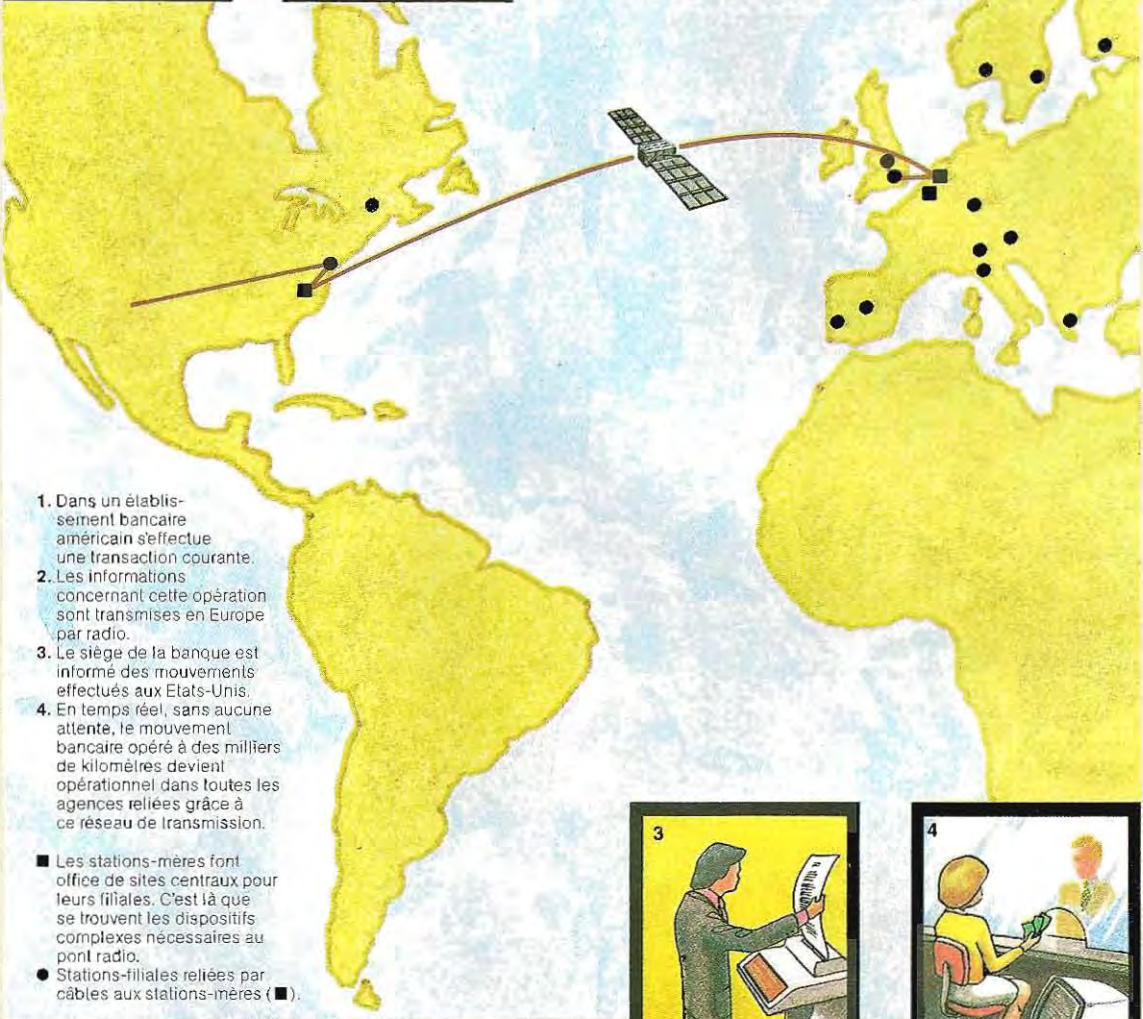
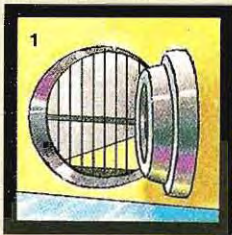
*Aux Etats-Unis, la City Bank et la Chemical Bank procèdent à des expériences analogues en collaboration avec deux cents de leurs propres clients.*

*Ces deux banques fournissent à leurs usagers des terminaux personnels qui permettent des transactions et répondent à certaines questions.*

*En Allemagne fédérale, un petit établissement, la Verbraucher Bank, a également tenté une expérience avec quelques clients. Ceux-*

### Terminaux de consultation de l'ordinateur de la Chase Manhattan Bank.





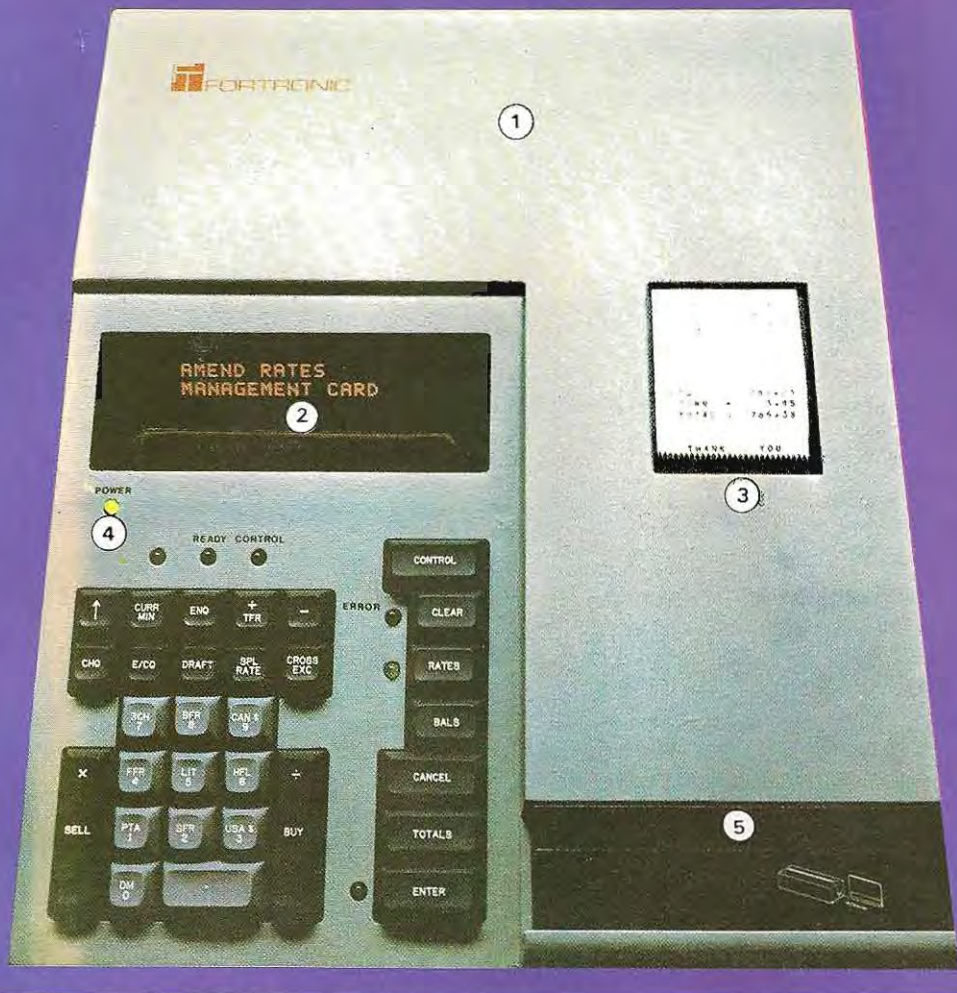
1. Dans un établissement bancaire américain s'effectue une transaction courante.
2. Les informations concernant cette opération sont transmises en Europe par radio.
3. Le siège de la banque est informé des mouvements effectués aux Etats-Unis.
4. En temps réel, sans aucune attente, le mouvement bancaire opéré à des milliers de kilomètres devient opérationnel dans toutes les agences reliées grâce à ce réseau de transmission.

- Les stations-mères font office de sites centraux pour leurs filiales. C'est là que se trouvent les dispositifs complexes nécessaires au pont radio.
- Stations-filiales reliées par câbles aux stations-mères (■).



ci ont reçu un petit terminal relié à un ordinateur central au moyen d'un système de type Vidéotex. Ils peuvent ainsi connaître le solde de leur compte, en obtenir un relevé, effectuer des virements sur d'autres banques, entre autres. Des banques plus grandes, comme la Deutsche Bank, se voient obligées de suivre cet exemple. En France, les banques se livrent à une expérience encore plus futuriste depuis le mois de janvier 1982, dans plusieurs villes. Tous les habitants possédant un compte courant y ont

reçu un module électronique et un terminal domestique. Chaque carte à mémoire, de la dimension d'une carte de crédit, contient un microprocesseur (ordinateur miniature), une mémoire ROM avec les programmes et les algorithmes de contrôle, et une mémoire capable de transcrire, mais non d'effacer, un grand nombre de transactions. Cette carte tient lieu de carnet de chèques ; on peut l'utiliser aux guichets des banques mais aussi dans les supermarchés, les grands magasins, les stations-service, les cabines télépho-



**Terminal bancaire (1) à affichage (2), imprimante (3), clavier (4) et lecteur de fiches magnétiques (5).**

niques, entre autres. Naturellement, de leur côté, les points de vente ont été équipés de terminaux spéciaux, dits POS (Point Of Sale terminal) capables d'effectuer des opérations avec les cartes de crédit normales aussi bien qu'avec les cartes à mémoire.

Dans ce cas des cartes à mémoire, l'« intelligence » nécessaire pour exécuter les contrôles n'est pas celle du terminal mais bien celle de la carte.

\* \* \*

En Belgique, la Caisse générale d'épargne et de retraite a mis en route un programme de réorganisation de ses agences dans cet esprit. Dans les nouvelles agences, les clients trouvent d'abord des consoles, sur lesquelles ils formulent d'abord leur demande. Ils sont alors dirigés vers un guichet s'il s'agit d'une opération de routine. S'ils ont à résoudre un problème plus spécifique ou plus personnel, ils sont orientés vers un bureau où ils vont

trouver la personne compétente. L'ordinateur évite les situations désagréables comme les attentes inutiles et les erreurs d'orientation du client.

Dans les banques self-service, plus d'employés derrière les guichets. Le client utilise un dispositif composé d'un écran où s'affichent les informations souhaitées, et d'un clavier pour communiquer les instructions à l'ordinateur de la banque. Clavier, écran, et éventuellement une petite imprimante, peuvent très bien trouver leur place au domicile de l'utilisateur (on emploie alors le téléviseur comme écran d'ordinateur), ou encore au bureau, à l'usine, dans les agences bancaires. Au Japon, on trouve même ces terminaux à l'extérieur des agences bancaires, à la disposition du public, un peu comme les cabines téléphoniques.

Comment s'opère la liaison entre le terminal domestique et l'ordinateur central de la banque ? Très simplement, à travers une ligne



téléphonique normale. Sur le clavier du terminal, le client appelle l'ordinateur, puis il compose son numéro de code (un mot de passe), puis il dialogue avec l'ordinateur, guidé par une série de programmes (menu) et de questions spécifiques. Dans certains cas, le client peut même se dispenser de composer son numéro d'accès : le terminal le lira lui-même directement sur la bande magnétique de la carte de crédit.

De nombreuses banques américaines fabriquent leur propres terminaux. C'est le cas de la Chemical Bank et de la City Bank. Pour donner un ordre de grandeur, le prix de chaque terminal individuel varie actuellement entre 300 et 1000 dollars. Une production de masse de ces terminaux en abaisserait le prix à environ 100 dollars.

\* \* \*

En Europe, il semble que les banques s'intéressent davantage à des systèmes de type Vidéotex (Minitel), avec branchement sur le réseau commuté, et utilisent donc un service déjà disponible aux usagers du téléphone. Les services nationaux seront rapidement reliés entre eux. Les clients auront ainsi directement accès à leur compte à partir de n'im-

porte quel terminal Vidéotex situé en Europe. Les banques sont en train d'élaborer un logiciel qui, au moyen d'instructions spécifiques et de contrôles appropriés, permet d'ouvrir l'accès à leur fichier à partir de terminaux de ce type.

Il est également possible d'adapter un terminal domestique au système Vidéotex. En effet, un terminal domestique est plus puissant qu'un téléviseur aménagé ; on pourrait même l'utiliser chez soi comme un vrai ordinateur personnel.

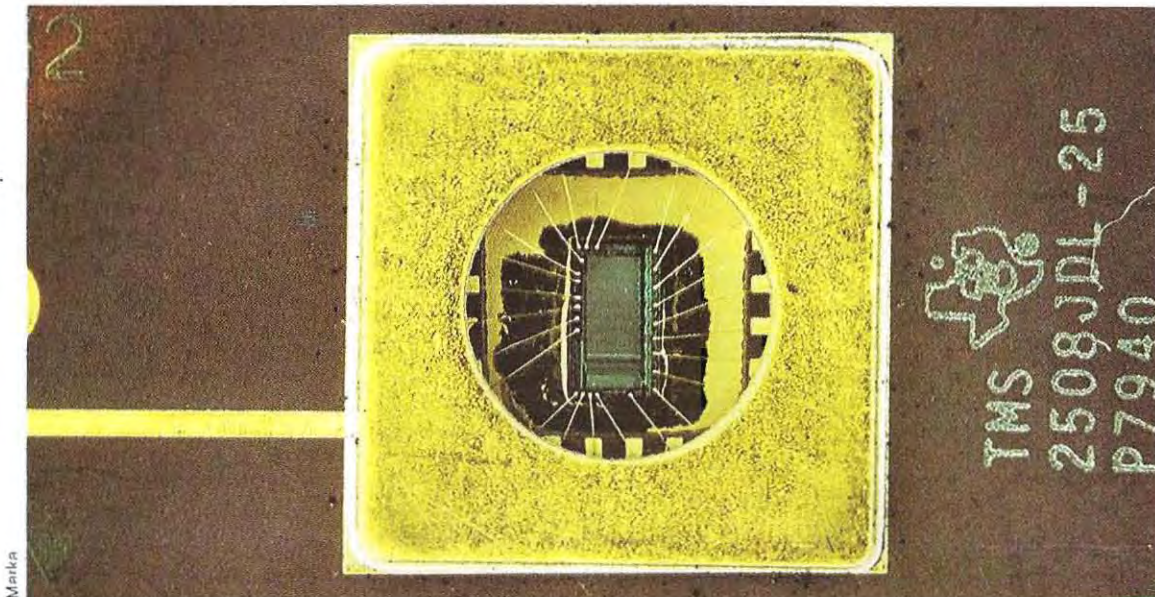
La Chemical Bank, au stade expérimental dont nous avons parlé, ne permet pas aux clients d'accéder directement à son fichier, mais seulement à une copie stockée en mémoire dans un ordinateur programmé à cet effet. La copie est mise à jour et enregistrée quotidiennement, en fin de journée.

La Verbraucher Bank, quant à elle, donne directement accès à son fichier, mais avec un contrôle très strict qui exige non seulement le numéro de compte mais aussi trois codes susceptibles de modifications, d'opération en opération.

(D'après un article de C. Merli et L. Gibin, INFORMATICA 70, n° 95, mars 1982.)

**Les ordinateurs modernes sont d'un emploi relativement facile ; ils sont utilisables même par des personnes peu familiarisées avec l'électronique.**





Marka

Dans un dispositif à deux entrées, chacune d'elles pouvant être dans l'état **H** ou **L**, les différentes combinaisons sont les suivantes :

<b>Première entrée</b>	<b>Seconde entrée</b>
H	L
H	H
L	H
L	L

Lorsque deux entrées sont liées entre elles, elles se trouveront dans le même état : ou toutes deux **H** ou toutes deux **L**.

Les états possibles sont donc :

A	B
H	H
L	L

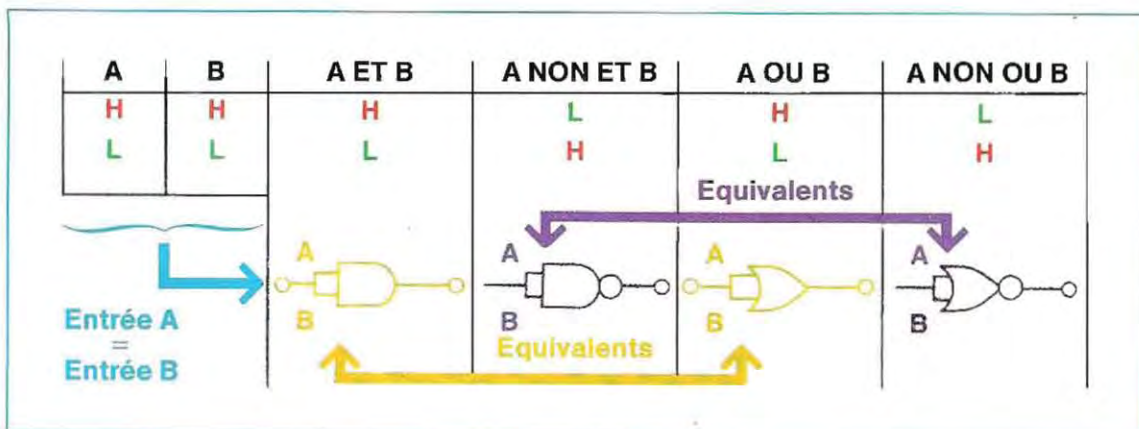
**Sur cette puce de silicium, véritable joyau de l'électronique, on distingue clairement les liaisons entre module et pattes (ou broches).**

Si l'on applique à ces états les opérateurs ET, NON ET, OU, NON OU, nous obtenons le schéma du bas de la page. On remarquera immédiatement qu'avec ce type de relation, on obtient :

ET = OU = amplificateur non inverseur  
NON ET = NON OU = amplificateur inverseur

Le circuit OUX ne peut pas être relié de cette manière puisque sa sortie serait dans tous les cas :

**L (H OUX H = L ; L OUX L = L).**



## Les circuits intégrés

Nous venons de parler des circuits capables d'effectuer certaines fonctions, et nous avons étudié leur comportement sous l'angle logique, mais sans envisager la façon de les fabriquer. Dans les débuts de l'électronique numérique, les circuits étaient constitués de composants discrets, c'est-à-dire réalisés séparément et qu'il fallait relier entre eux pour constituer un circuit complet. Par la suite, on a pu rassembler tous les constituants d'un circuit sur une seule plaquette et obtenir avec un seul composant des fonctions dont l'exécution exigeait jusque-là des dizaines d'éléments séparés. Ce nouveau type de composant a reçu le nom de **circuit intégré**; il est d'une utilisation extrêmement simple. Il suffit de brancher les entrées et les sorties et de prévoir une alimentation électrique.

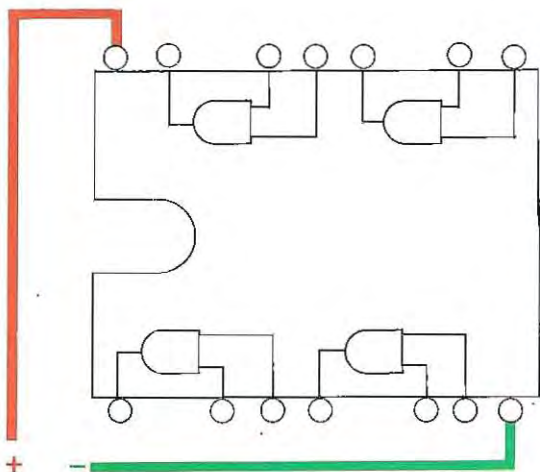
Les circuits intégrés les plus courants (que l'on appelle circuits TTL d'après leur structure) ne demandent que 5 volts et peuvent donc fonctionner avec une pile sèche.

Un même boîtier rassemble en général plusieurs circuits complets. L'un des plus répandus est celui qui accomplit la fonction ET, et où l'on trouve quatre de ces circuits, ou même davantage.

Ce boîtier comporte quatorze branches, trois par circuit (chaque circuit ET possède en fait deux entrées + une sortie) et deux broches pour l'alimentation générale.

Les constructeurs fournissent pour chaque modèle le schéma du branchement.

On trouvera, en bas de cette page à gauche, le schéma du circuit intégré ET.



En dépit de la dimension relativement réduite des circuits intégrés, leur utilisation pour la construction d'un ordinateur conduirait à réaliser des machines volumineuses.

La miniaturisation permet heureusement de rassembler dans un seul boîtier la totalité des circuits nécessaires à l'exécution des fonctions principales d'un ordinateur : c'est le **microprocesseur** ( $\mu P$ ); il constitue le cœur de l'ordinateur.

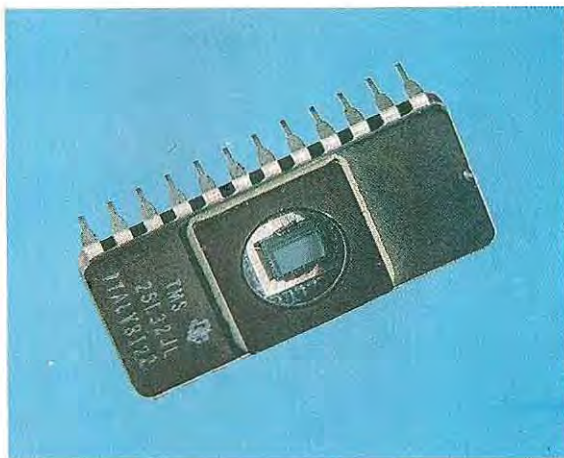
## Les circuits composites

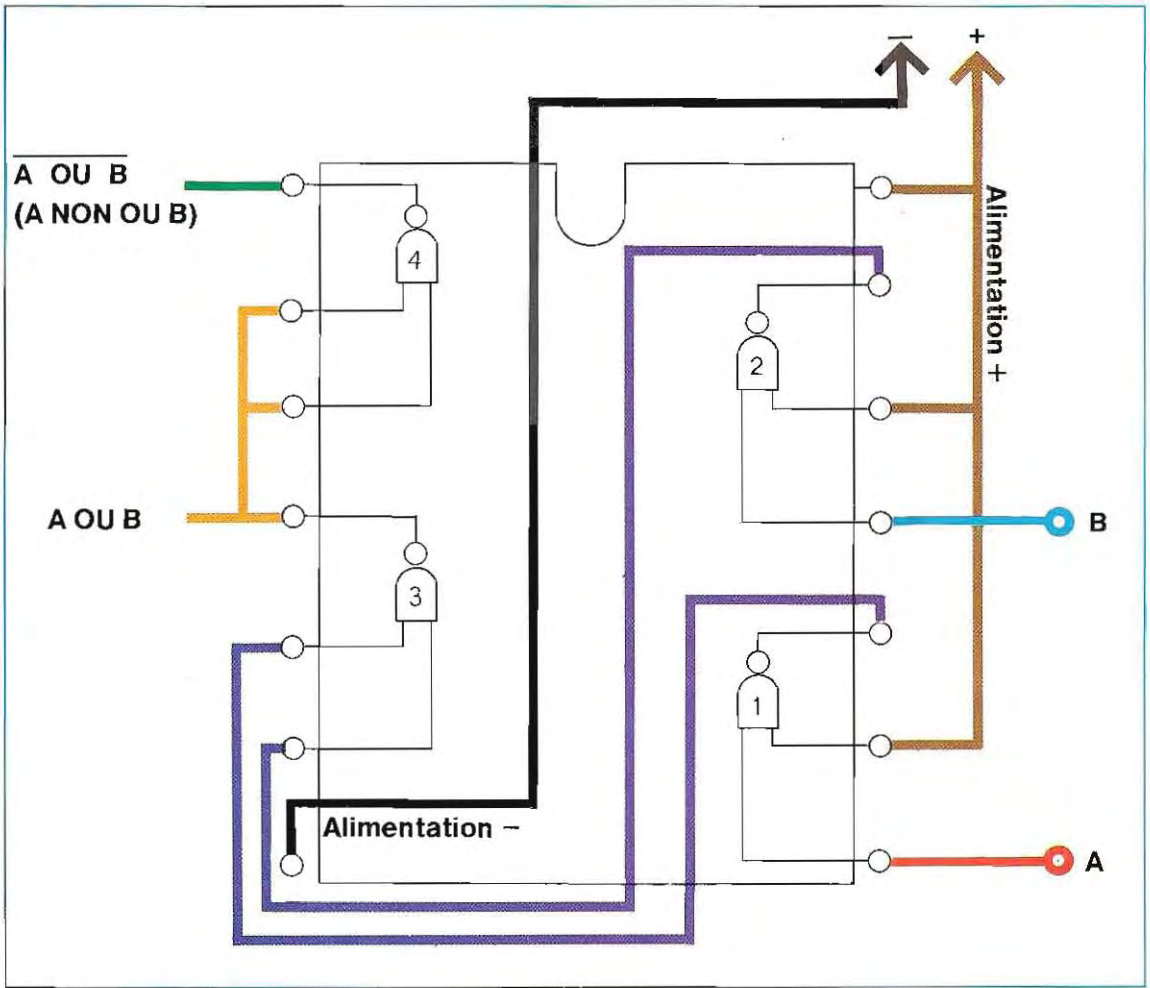
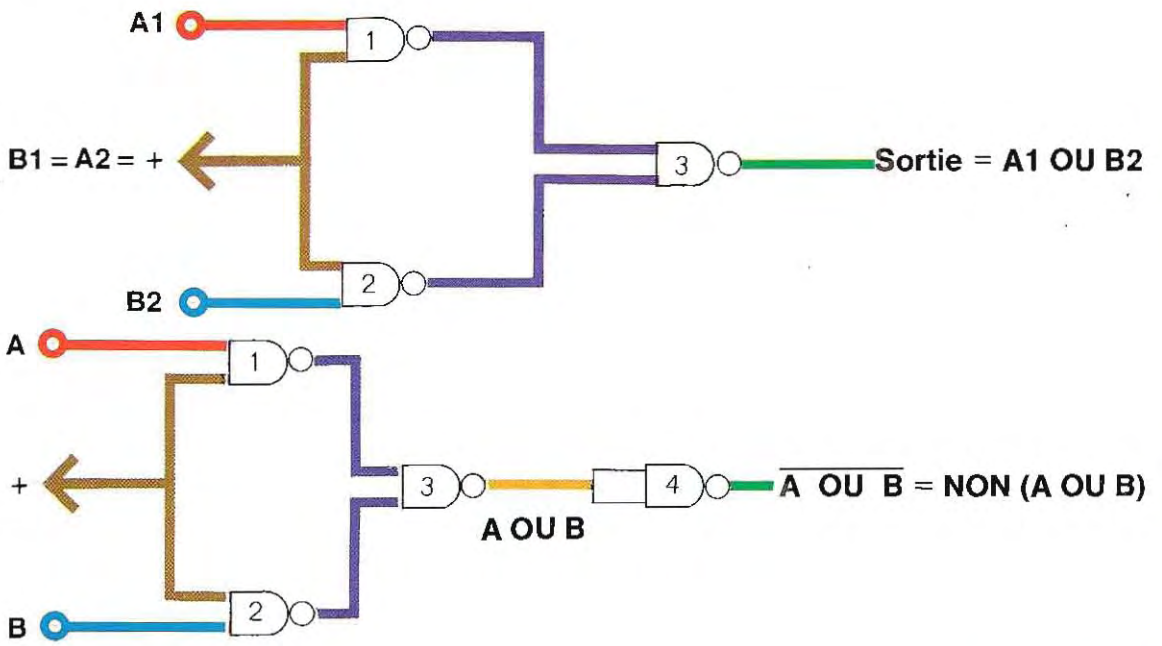
Les circuits que nous venons de voir peuvent être reliés entre eux pour obtenir une fonction logique; la connexion des différents circuits n'exige aucun soin particulier à condition toutefois qu'ils appartiennent à la même famille. La page 98 (en haut) montre un montage réalisé avec des circuits NON ET (NAND) pour accomplir la fonction OU.

On peut immédiatement vérifier son fonctionnement par la table de vérité et en observant que les circuits 1 et 2 ont l'une de leurs entrées (respectivement  $B_1$  et  $A_2$ ) au niveau H.

Rappelons que le niveau H peut être le pôle positif de l'alimentation, le niveau L étant le pôle négatif. Sur le schéma, les entrées  $B_1$  et  $A_2$  sont donc reliées au pôle plus (+). En connectant en sortie un autre NON ET on obtient la fonction NON OU (NOR); c'est ce qu'illustre le second schéma en haut de la page 98. Les branchements à effectuer pour obtenir les fonctions OU et NON OU figurent sur le graphique du bas de la page 98.

**Une Eprom de Texas Instrument. La fenêtre transparente protège la pastille de silicium et permet aux rayons ultraviolets d'effacer le contenu de la mémoire.**





## Les applications

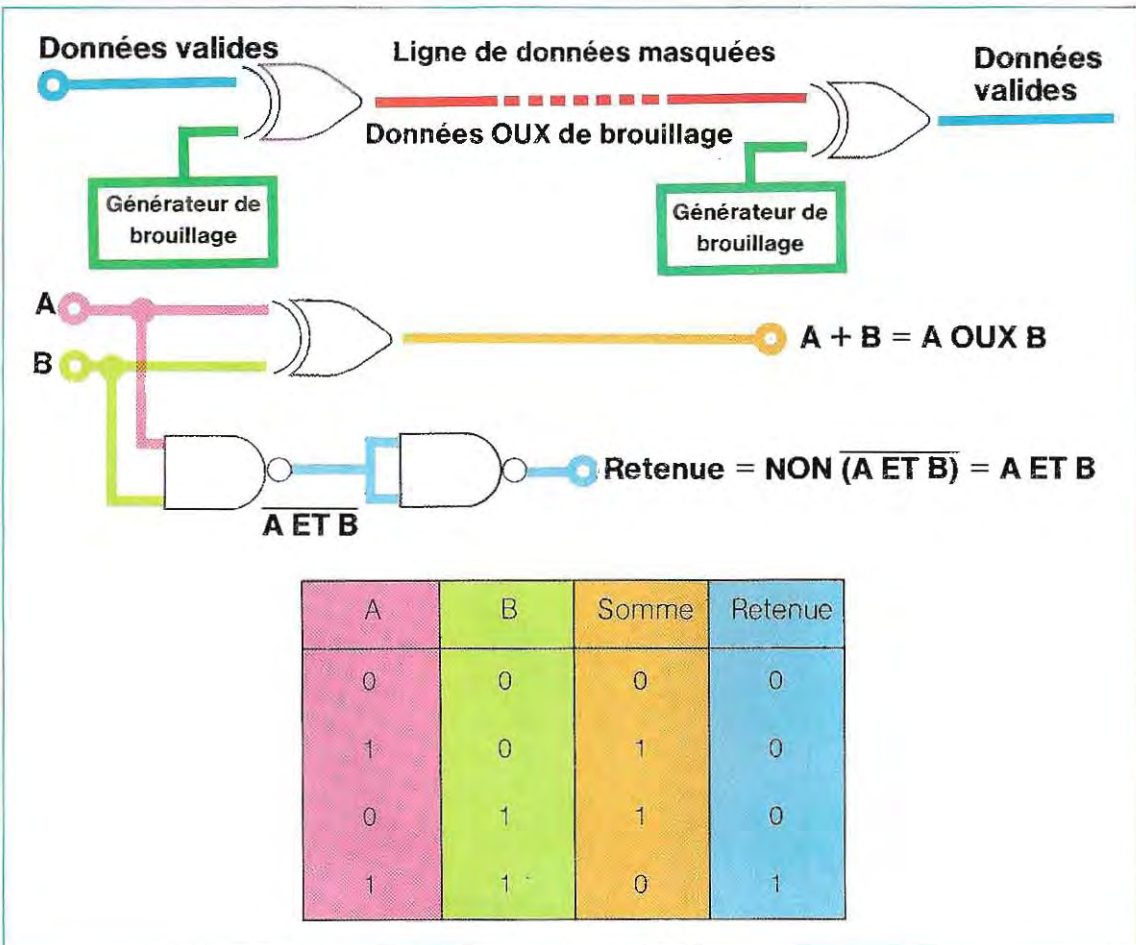
Nous avons expliqué que ces circuits ont une fonction principale de séparateur. Nous allons en étudier deux applications particulières.

La première concerne la sécurité du transport des données : les réseaux d'ordinateurs permettent en effet de transmettre à grande distance, et les données transmises risquent d'être interceptées et modifiées ; d'où la nécessité d'adopter des méthodes de protection.

Ces méthodes sont nombreuses et plus ou moins sûres. L'une des plus simples (quoique suffisamment sûre) consiste à envoyer sur la ligne et en le superposant aux données, un brouillage constitué d'une série d'impulsions aléatoires qui les rend indéchiffrables. A l'arrivée, il faut injecter un brouillage semblable en utilisant le même générateur de bruit pour obtenir un signal dépouillé, composé uniquement des données utiles. Le système qui effectue le mixage des signaux au départ et

leur séparation à l'arrivée est un simple circuit OUX (XOR), ce qui signifie l'un ou l'autre, mais pas les deux. On en trouvera le schéma sur cette page. Supposons qu'au départ les données soient de niveau H : si l'on injecte comme brouillage un bruit de niveau H, le résultat est un niveau L ( $H \text{ OUX } H = L$ ) ; à l'arrivée, il n'est possible de définir si ce niveau L est un véritable niveau L, ou bien un niveau H masqué, qu'en injectant le même brouillage qu'au départ (H) : on obtient ainsi un résultat (brouillage)  $\text{OUX } L$  (données d'arrivée) = H (données vraies).

La seconde application est le circuit qui permet de réaliser un additionneur binaire. Au bas du schéma de cette page, le lecteur trouvera la table de vérité correspondante. On notera qu'en utilisant les NON ET, il faut faire appel à deux circuits, le second ne fonctionnant que comme inverseur. Si l'on remplace le circuit 1 par un circuit ET, le second devient inutile, la sortie étant directement A ET B.



## Autres types de circuits intégrés

Les circuits intégrés utilisés dans la construction d'un système de micro-ordinateur ne se limitent pas aux types que nous avons présentés.

Les circuits accomplissant les fonctions d'opérateur logique (ET, OU, OUX, etc.) sont les plus simples et les plus utilisés; lorsqu'on augmente la spécialisation d'un circuit, on augmente aussi sa complexité, jusqu'au microprocesseur, qui renferme des milliers de composants sur un même circuit intégré.

Nous allons décrire quelques circuits plus complexes courants. Les circuits à microprocesseur seront abordés plus loin.

**Temporisateur** (timer en anglais) : dispositif fournissant des signaux périodiques sous forme d'ondes, ou oscillations; on l'emploie en général pour la création d'impulsions. Il en existe deux types fondamentaux: à réseau RC (circuit résistance et capacité, connecté à un oscillateur) et à quartz. Dans un oscillateur à circuit RC, la durée de l'impulsion est réglée par un circuit composé d'une résistance et d'un condensateur: ce sont les plus économiques et les moins précis. Le circuit à quartz utilise un cristal de quartz dont la vibration mécanique commande la durée de l'impulsion: ils sont plus coûteux, mais beaucoup plus précis. Ce dernier type est utilisé pour générer des signaux dans les systèmes à micro-ordinateur. La fréquence est alors de

quelques mégahertz (MHz), ou millions d'oscillations par seconde.

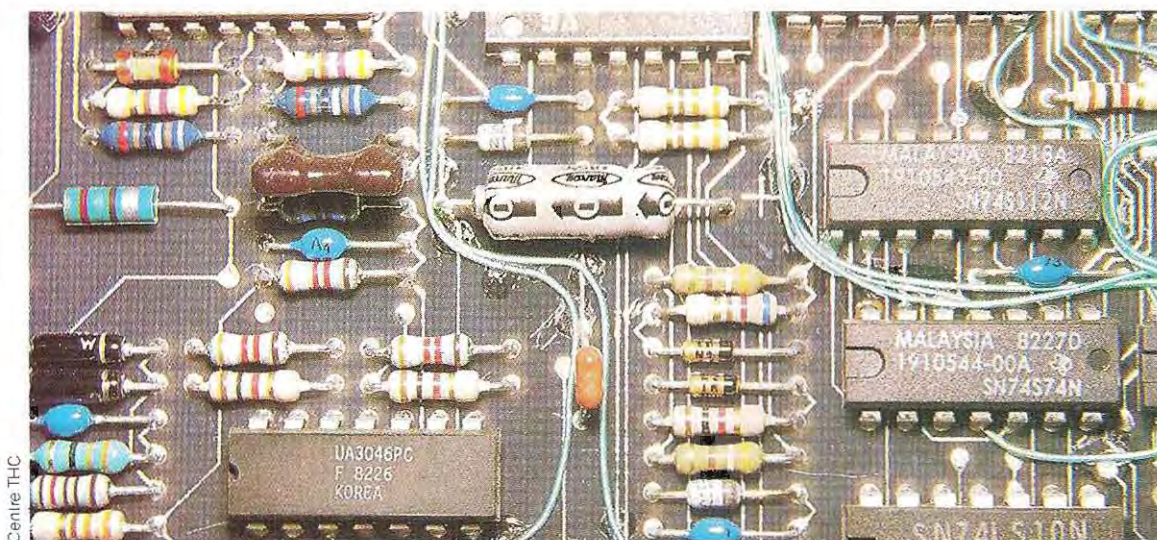
**Bascule** (trigger en anglais) : circuit à déclenchement, c'est-à-dire sensible au niveau du signal d'entrée. Quand ce niveau dépasse une valeur déterminée, le circuit « bascule » et la sortie change d'état, passant du niveau bas au niveau haut, ou vice versa. Ces circuits servent comme régénérateurs de signaux.

**Diviseur** (diviser en anglais) : circuit capable de fournir en sortie une quantité d'impulsions égale à celle d'entrée divisée par un nombre quelconque. Il en existe deux types, à division fixe ou programmable. Les diviseurs fixes fournissent en sortie le nombre d'impulsions d'entrées divisé par une quantité fixe, qui dépend du branchement du circuit. Un diviseur programmable permet au contraire un réglage du diviseur.

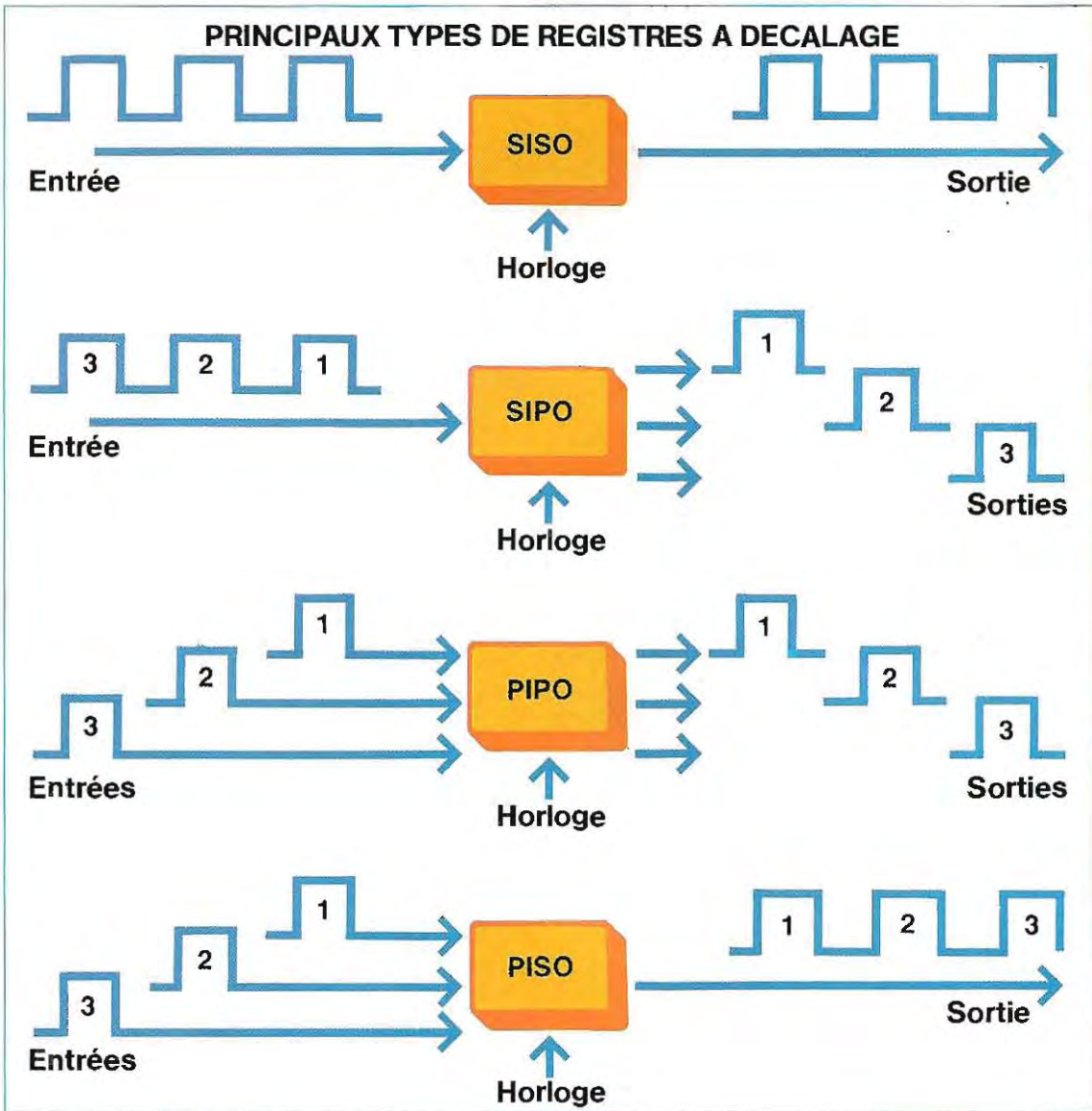
**Bascule bistable** (flip-flop en anglais) : circuit à deux états (ouvert-fermé, ON-OFF) dans lequel la sortie ne se modifie pas immédiatement à réception du signal d'entrée. Le circuit mémorise le changement d'état de l'entrée et le suit, à réception d'un autre signal (ou impulsion) dénommé signal d'horloge (clock).

Il s'agit d'un dispositif bistable, c'est-à-dire possédant deux états stables qui suivent l'impulsion d'entrée avec retard, à réception du signal d'horloge. Ce circuit sert pour la division par deux ou comme mémoire. Il est de type synchrone, la communication se faisant en coïncidence avec le signal d'horloge.

Détail de l'intérieur d'un ordinateur. Sur cette plaquette on voit plusieurs circuits intégrés.



## PRINCIPAUX TYPES DE REGISTRES A DECALAGE



**Registre à décalage** (shift-register) : dispositif constitué de bascules bistables correctement assemblées. La donnée (impulsion) se déplace d'une cellule à l'autre à l'arrivée de chaque signal d'horloge.

La donnée est chargée dans la première cellule ; au premier signal d'horloge, elle passe à la seconde cellule, et ainsi de suite. Il existe différents types de circuits de ce genre, selon leur mode de fonctionnement.

Les principaux sont :

Entrée série, sortie série (SISO, de l'anglais Serial Input Serial Output). Les données sont chargées en série (une à la fois) et sortent en série, en coïncidence avec les signaux d'horloge.

Entrée série, sortie parallèle (SIPO, de l'anglais Serial Input Parallel Output). Les impulsions d'entrée sont chargées, puis présentées sur un nombre équivalent de sorties diverses. Par exemple, quatre impulsions d'entrée sont mémorisées et présentées en même temps sur quatre sorties différentes.

Entrée parallèle, sortie parallèle (PIPO, Parallel Input Parallel Output). L'entrée et la sortie acceptent en même temps plusieurs impulsions.

Entrée parallèle, sortie série (PISO, Parallel Input Serial Output). L'entrée accepte des impulsions simultanées qui sont mémorisées et présentées en série sur une sortie unique (schéma de cette page).

## Codes secrets contre les pirates de logiciels

Les énormes progrès réalisés dans le domaine des techniques télématiques de transmission des données permettent de réduire considérablement le délai technique indispensable à l'échange des informations entre personnes ou bureaux que séparent des milliers de kilomètres. Des données confidentielles stockées dans les archives électroniques, concernant, par exemple, les comptes bancaires ou les secrets industriels, circulent quotidiennement sur les réseaux de télécommunications. Elles sont plus nombreuses chaque jour, et leur interception, ainsi que leur manipulation, seraient chose facile si on n'assurait pas leur protection.

Les cas les plus célèbres, ceux qui ont eu les honneurs de la presse mondiale, sont ceux dans lesquels sont intervenus les « pirates de logiciels ». Des personnes ingénieuses, et dépourvues de tout scrupule, ont trouvé le moyen d'accéder aux réseaux de transmission de données de plusieurs grosses banques internationales, s'informant ainsi sur d'importantes transactions financières.

Les précautions adoptées pour les procédures de transmission sont aussi variées que complexes, mais ne suffisent pourtant pas et n'ont pas empêché quelques spécialistes malhonnêtes d'organiser pour leur propre compte le transfert de sommes importantes. Des nouvelles de ce genre sont soigneusement étouffées par les établissements bancaires, qui ne tireraient aucun profit de cette publicité, mais elles ont plusieurs fois fait leur chemin jusqu'aux premières pages des journaux, et ont provoqué l'inquiétude du public. La piraterie au niveau du logiciel n'est pas toujours de type financier. Parmi les communications confidentielles qui circulent chaque jour sur les lignes téléphoniques du monde entier, on trouve aussi des informations sur les inventions et les procédés industriels couverts par des brevets, ou encore des messages échangés par des organes politiques internationaux. Il est plus que jamais indispensable de mettre au point de nouveaux systèmes de **cryptographie**.

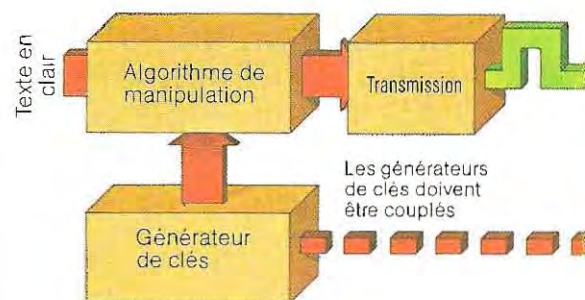
Le cryptage, ou codage, ou chiffrement, est le procédé qui consiste à traduire un texte original écrit en clair, lettre par lettre, en un mes-

sage codé contenant toutes les lettres du texte original, mais dans une autre position (**codage par transposition**), ou avec remplacement de certaines lettres par d'autres (**codage par substitution**). Dans la pratique, on peut combiner les deux méthodes en les appliquant une ou plusieurs fois dans le même texte. Un système ingénieux de codage par transposition où toutes les lettres sont mélangées plusieurs fois (**codage par transposition double ou triple**), appliqué à des textes relativement longs, peut constituer un système de cryptographie valable. Voici un exemple très simple de codage par transposition:

```
V I E N S I  
M M E D I A  
T E M E N T  
A V E C M A  
T E R I E L
```

Transcrit horizontalement, le texte se lit: VIENS IMMEDIATEMENT AVEC MATERIEL, mais le message pourrait être codé par lecture verticale: VMTAT IMEVE EEMER NDECI SINME IATAL. Un spécialiste se rendrait immédiatement compte qu'en rassemblant successivement les premières lettres de chaque groupe, puis les secondes et ainsi de suite, le texte prend une signification. Il faut donc adopter une méthode plus compliquée. Un autre codage simple par substitution consiste à déplacer d'un cran chacune des lettres de l'alphabet: A devient B, B devient C, etc. Le message « VIENS IMMEDIATEMENT » se lit alors: « WJFOT JNNFEJBUFNFU ». Une transformation de ce genre peut paraître

### SYSTEME CRYPTOGRAPHIQUE SIMPLIFIE





satisfaisante à première vue, mais chaque langue possédant une structure propre, la solution du problème n'est qu'un jeu.

Chaque lettre du texte original étant toujours représentée par la même lettre du code, ce dernier appartient à la catégorie mono-alphabétique.

Au lieu de déplacer d'un seul cran les lettres de la phrase VIENS IMMEDIATEMENT, on pourrait déplacer les lettres successives en augmentant chaque fois d'une unité le nombre de crans, ce qui donnerait pour VIENS le mot WKHRX. De cette manière, n'importe quelle lettre du texte original peut en théorie être représentée par n'importe quelle lettre du texte chiffré. Le décryptage du message exige alors une analyse mathématique beaucoup plus poussée.

Les systèmes de cryptage plus élaborés exigent l'application de deux procédures séparées. La première consiste à déplacer chaque lettre suivant un schéma prédéterminé, que l'on appelle **algorithme**, qui sera ensuite répété en sens inverse pour retrouver le texte original. La seconde procédure consiste à se munir d'une **clé** indiquant comment appliquer l'algorithme. De la longueur de cette clé dépend le degré de sécurité des codes de substitution. Une clé de longueur illimitée et comprenant des lettres et des nombres décimaux ou binaires synthétisés et disposés de manière aléatoire constituerait un code de chiffrement imbattable.

La plus grande garantie de sécurité que puisse offrir un code polyalphabétique (ou son équivalent sous forme numérique) réside dans le nombre des alphabets de substitution. En d'au-

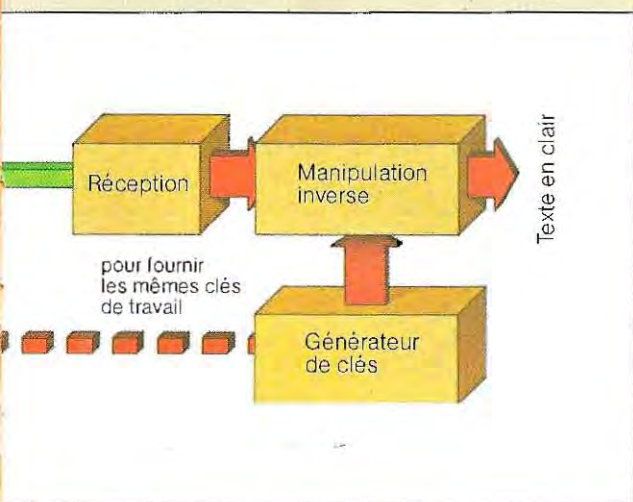
tres termes, la lettre A qui apparaît dans le texte dans des positions diverses peut être traduite selon les cas par n'importe quelle autre lettre de B à Z; c'est ce qu'il faudra déchiffrer correctement pour retrouver A. On peut aussi par mesure de précaution prévoir que les 25 alphabets de substitution (autant que les lettres de l'alphabet, moins la lettre codée) seront utilisés dans un ordre aléatoire, ce que l'on obtient facilement avec une clé de travail aléatoire de longueur illimitée.

Ce principe est celui de la chaîne à usage unique, visualisation d'une succession aléatoire de lettres ou de nombres décimaux ou binaires, dans laquelle chaque lettre clé indique de combien de crans chacune des lettres du texte doit être déplacée pour le déchiffrement.

Une cryptographie de ce type assure toute sécurité et défie les efforts du cryptographe pour découvrir le code; le système présente cependant des difficultés pour celui qui reçoit le message, et il est d'emploi compliqué et coûteux. Il exige aussi la production et la distribution de copies des clés d'une longueur suffisante pour couvrir le nombre total des lettres transmises sur une période de quelques semaines ou de quelques mois.

Dans la pratique, beaucoup de codes polyalphabétiques utilisent des clés de longueur limitée, fondées sur des dispositions comme les carrés de Vigenère ou les carrés inverses de Beaufort. Dans ce cas, la clé peut être une phrase brève, avec des clés additionnelles fournies, par exemple, par le message lui-même. Les techniques de résolution des codes polyalphabétiques ont d'ailleurs été renforcées par l'existence d'ordinateurs à grand débit. Cependant, on n'a pas toujours besoin de systèmes définitivement indéchiffrables: il suffit le plus souvent qu'ils imposent un retard à la découverte du texte original ou qu'ils impliquent un coût de déchiffrement démesuré et décourageant pour un éventuel utilisateur.

Au cours des dix dernières années, la cryptographie a remarquablement progressé grâce aux ordinateurs numériques eux-mêmes. En électronique, nous le savons, on dit qu'un système est numérique lorsque les signaux et les formes d'ondes qu'il élabore sont en nombre restreint, par opposition à un système analogique classique où les signaux traités peuvent adopter un nombre infini de formes et de niveaux. Le système binaire est



un système numérique qui ne prévoit que deux états différents, le signal étant représenté par l'une des deux conditions ON-OFF, ou signal présent-signal absent, ou encore 1-0. En un sens, toutes les transmissions chiffrées, par voie hertzienne ou par câble, exigent le recours à un code numérique. Le Morse et le langage des télécriteurs par exemple sont des codes numériques, même si pendant longtemps les messages étaient d'abord transcrits sous forme de lettres avant d'être transmis en alphabet Morse. On tend actuellement à codifier le texte en clair directement sous une forme numérique, en utilisant la modulation à impulsion codifiée. L'étape suivante consiste à transformer en chiffres le flux de chiffres binaires (bits) en les combinant avec une clé de travail numérique. Grâce aux appareils électroniques à grand débit, tout le processus peut être exécuté en temps réel quelle que soit la vitesse requise, qu'il s'agisse d'un texte écrit ou d'un discours

La méthode d'établissement de la séquence pseudo-aléatoire, nous l'avons vu, a un effet très important sur la sécurité du cryptage. En effet, si des systèmes numériques à base de «logique linéaire», offrent de nombreuses possibilités de codifications, celles-ci sont cependant aisément déchiffrables. Un petit ordinateur suffirait, à condition qu'une partie du message (par exemple l'adresse d'ouverture) soit connue de l'analyste.

C'est un risque qui a conduit des firmes comme IBM à confectionner des systèmes cryptographiques faisant appel à un algorithme non linéaire tout en utilisant un nombre relativement réduit de bits clés. Chacun des blocs du texte est soumis, avec sa clé, à une série complexe de transformations, telles que la transposition de l'ordre et les substitutions fondées sur de longues séquences-clés dérivées de quelques clés d'origine. Ces systèmes sont considérés comme imbattables.

#### Exemple 1

CLE	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
DEPLACEMENT	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
TEXTE EN CLAIR	V I E N S I M M E D I A T E M E N T A V E C M A T
CLE DE TRAVAIL	C F Z A L H T R P A S H R Y F N O O B E G R A M N
DEPLACEMENT	3 6 26 1 12 8 20 18 16 1 19 8 18 25 6 14 17 15 2 5 7 18 1 13 14
TEXTE CHIFFRE	Y O E O E Q G E U E B I L D S S E I C A L U N N H

#### Exemple :

La lettre H de la clé transforme le I en Q, ou le A en I.

La lettre R de la clé transforme le M en E, ou le T en L, ou le C en U.

parlé. L'utilisation d'une clé numérique constitue un chiffrement par substitution polyalphabétique. Si la clé est véritablement aléatoire et de longueur illimitée, le codage est très sûr.

Cela présente cependant des difficultés opératoires semblables à celles que l'on rencontre dans le déchiffrement analogique avec les clés à usage unique.

En pratique, on extrait une clé pseudo-aléatoire d'une autre clé relativement brève (en général inférieure à 100 bits), en l'agençant de telle manière que l'on rencontre de longues séries de 1 ou de 0 avant répétition de la séquence.

En 1977, l'organisme américain de normalisation, ANSI (American National Standard Institute), a donné son approbation à l'algorithme DES (Data Encryption Standard) fondé sur une clé de 56 bits. Chaque bloc de données subit dix-huit étapes de manipulation, pour lesquelles seize clés cryptographiques différentes sont dérivées d'une clé principale à 56 bits de manière à former une clé dont la longueur totale est de  $10^{17}$  bits. Les utilisateurs du système font appel à un algorithme publié, mais utilisent des générateurs de clés coupés pour lesquels la clé principale à 56 bits






(ou plutôt 64, avec la codification interne) est tenue secrète. Le générateur de clés peut être un petit module scellé, mais l'essentiel est de le garder en sécurité, ce qui entraîne des problèmes pour la gestion et la distribution des clés. On estime aujourd'hui que ce système présente une sécurité suffisante pour tous les usages commerciaux courants. Les systèmes numériques comme DES peuvent être appliqués directement aux ordinateurs pour protéger des archives confidentielles ou des transmissions par téléphone et par radio de données et de messages. Certains spécialistes pensent toutefois que la clé principale de 56 bits est trop courte pour que le système garantisse une sécurité totale. Bien que les systèmes cryptographiques DES soient progressivement adoptés par les organisations bancaires et commerciales, et même si personne n'a encore réussi à démontrer publiquement la possibilité de les violer, plusieurs groupes de spécialistes améri-

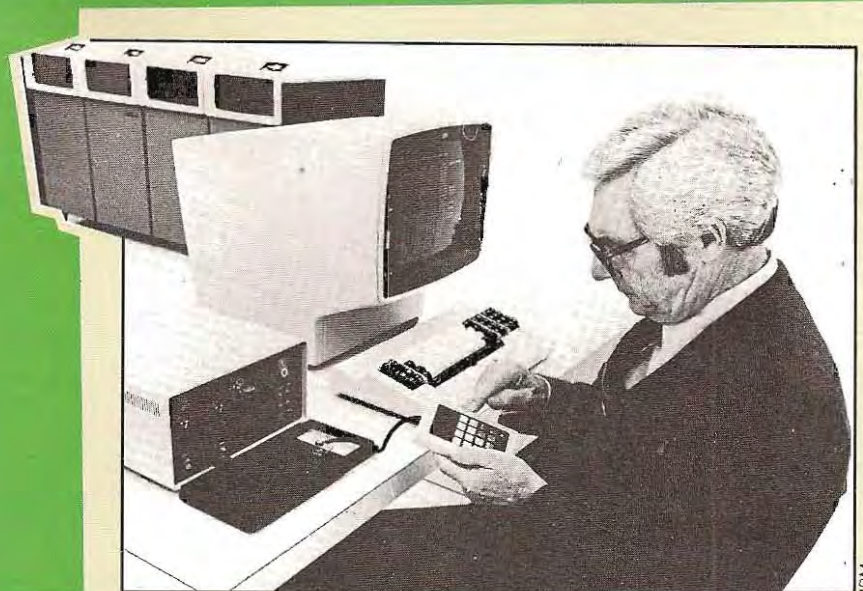
cains ont récemment proposé un système plus puissant. Il s'agit de la cryptographie à clé publique, dont l'intérêt est d'éliminer tous les problèmes de gestion et de distribution des clés. Dans un tel système, les fonctions de codage et de décodage sont distinctes. Ainsi, l'émetteur ne possède pas la clé de déchiffrement; mais il assure le codage du message pour un destinataire spécifique par l'emploi d'une liste connue de clés de codage. Ces systèmes cryptographiques ont aussi l'avantage de supprimer les difficultés liées à la distribution des clés, et à la protection des générateurs de clés ou des chaînes à usage unique.

Finalement, l'attention apportée par les spécialistes à l'établissement des systèmes de cryptographie permet d'espérer que l'on pourra profiter de la technologie télématique sans courir le risque de retrouver son compte en banque à découvert au profit d'un inconnu trop bien familiarisé avec l'ordinateur.

### Exemple 2

A	00001	N	01110
B	00010	O	01111
C	00011	P	10000
D	00100	Q	10001
E	00101	R	10010
F	00110	S	10011
G	00111	T	10100
H	01000	U	10101
I	01001	V	10110
J	01010	W	10111
K	01011	X	11000
L	01100	Y	11001
M	01101	Z	11010

TEXTE EN CLAIR	V	I	E	N	S
TEXTE TRADUIT EN SYSTEME BINAIRE	10110	01001	00101	01110	10011
CLE NUMERIQUE	01100	11001	00110	10101	01011
TRANSMISSION CODEE	11010	10000	00011	11011	11000
FORME D'ONDE					



## TEST 2



1 / Compléter la table de vérité suivante :

A	B	A OUX B	A ET (A OUX B)
1	0		
1	1		
0	0		
0	1		

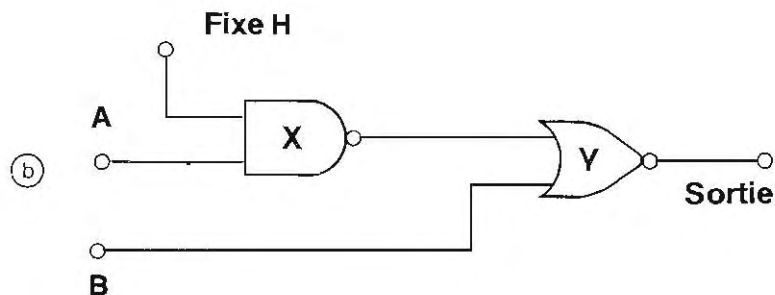
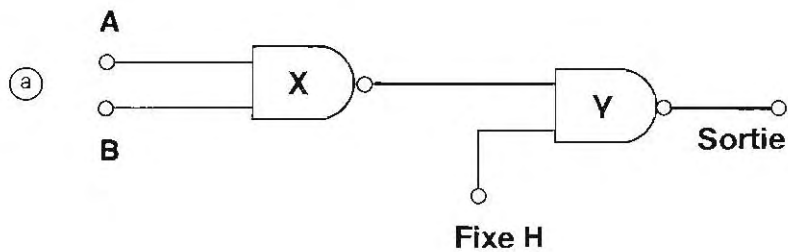
2 / Résoudre les opérations suivantes :

- a) 11011 ET 011
- b) 12 ET 7 (12 et 7 comme nombres décimaux)
- c) 12 ET 7 (12 et 7 comme nombres octaux)
- d) 1101 OU 0110
- e) 1011 OUX 1101

3 / Dessiner les symboles graphiques et la table de vérité des circuits :

- a) NON b) ET c) OU d) OUX

4 / Ecrire la table de vérité des circuits :



Les exercices 3 et 4 ne sont pas déterminants pour la poursuite de la lecture. On peut considérer le sujet comme assimilé si le lecteur sait répondre à l'exercice 1 et à 3 parties au moins de l'exercice 2. Les solutions du test se trouvent page 122.

# Les codes de transmission des données

Pour qu'un ordinateur fonctionne correctement, il faut lui permettre d'interpréter les données et les informations qui lui sont fournies. Si la transposition d'une donnée numérique en forme binaire est immédiate, les données non numériques doivent être traduites sous une forme

compréhensible par la machine. On utilise donc un code de transmission des données.

## Bit et byte

La représentation binaire d'un nombre est une série de symboles **0** et **1** : chacun de ces symboles prend le nom d'élément binaire (binary digit : **bit**). Par exemple, la notation binaire du nombre 35 est 00100011 (voir schéma ci-dessous).

La valeur binaire de chaque bit est déterminée par sa position : le bit placé en position 1 vaut **1**, le bit placé en position 3 vaut **4**, etc. Dans l'ordinateur, chaque bit est représenté par un signal électrique qui est de niveau haut (H) si le bit correspondant vaut **1** ou bas (L) si le bit vaut **0**.

Position	8	7	6	5	4	3	2	1
Puissance de 2	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
Valeur	128	64	32	16	8	4	2	1
35 décimal =	0	0	1	0	0	0	1	1

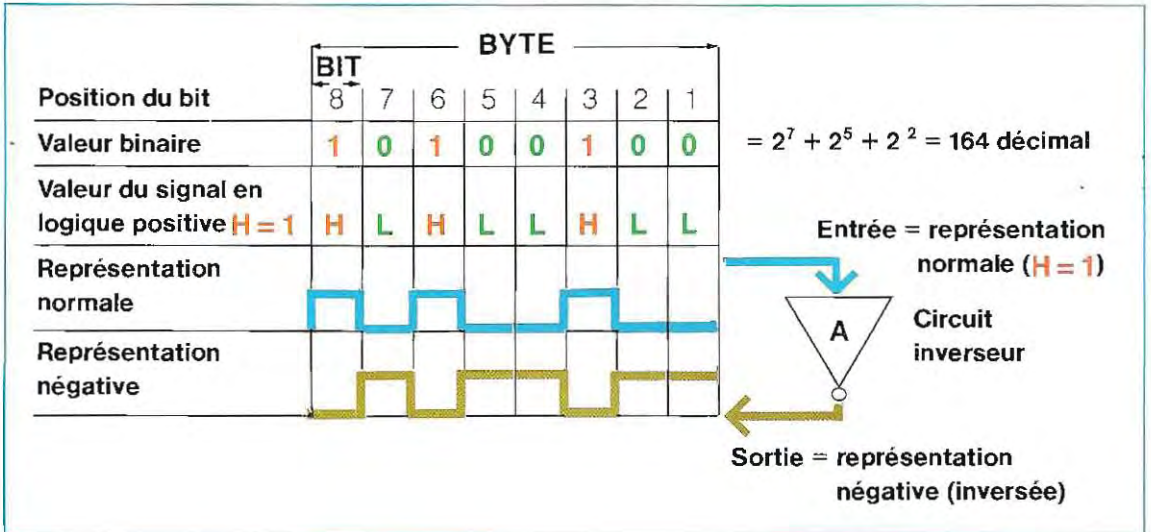
Le schéma ci-dessous représente une situation pour les signaux correspondant à 35 décimal.

Position = numéro du bit	8	7	6	5	4	3	2	1
Puissance de 2	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
Valeur binaire correspondant à 35 décimal	0	0	1	0	0	0	1	1
Niveau du signal	L	L	H	L	L	L	H	H
Représentation graphique								

Niveau haut = H  
Niveau bas = L

On suppose dans ce cas que la valeur binaire **1** correspond à un signal électrique de niveau haut ; rien n'empêche d'adopter la logique inverse (valeur binaire **1** = niveau bas), c'est d'ailleurs le mode de fonctionnement (logique négative) de nombreuses machines.

Voici la représentation du nombre 164 décimal, en logiques positive et négative :



Ces deux modes de fonctionnement ne posent aucun problème dans la pratique, et pour passer de l'un à l'autre il suffit de disposer d'un circuit inverseur (A sur la figure).

Comme l'illustre le schéma, pour représenter en binaire le nombre 164 décimal, il faut un « multiplet » de 8 éléments binaires, soit 8 bits, (la valeur binaire est 10100100) ; c'est-à-dire un **octet (byte** en anglais).

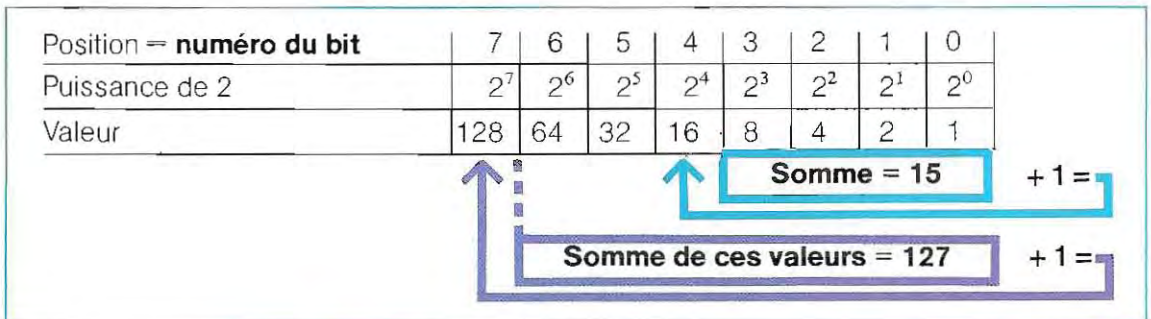
La valeur numérique maximale que l'on peut écrire avec un octet s'obtient en prenant les 8 bits qui le composent égaux à 1 :

Valeur maximale de 1 octet :  $11111111 = 2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 = 255$ .

Avec un bus de données à 8 bits (de 0 à 7), le nombre maximal que la machine peut utiliser est 255 : il ne suffit donc pas pour les applica-

tions pratiques. Pour éliminer cette entrave, il existe des circuits particuliers qui permettent de regrouper plusieurs octets afin de former de très grands nombres.

En regroupant 2 octets, on obtient par exemple 16 bits et donc une valeur numérique d'environ 32000. Les représentations les plus utilisées comportent 2, ou 4 octets (32 bits). On notera qu'ajouter 1 seul bit équivaut à multiplier par 2 la valeur numérique maximale : en fait, la valeur numérique d'un bit quelconque est égale à la somme des valeurs des bits qui le précèdent, plus 1. Par exemple, le bit 4 qui apparaît dans le schéma ci-dessous (somme des bits 0,1,2,3 = 1+2+4+8=15) vaut 16. Le bit 7 vaut 128 ; soit la somme des valeurs 64+32+16+8+4+2+1, plus 1.



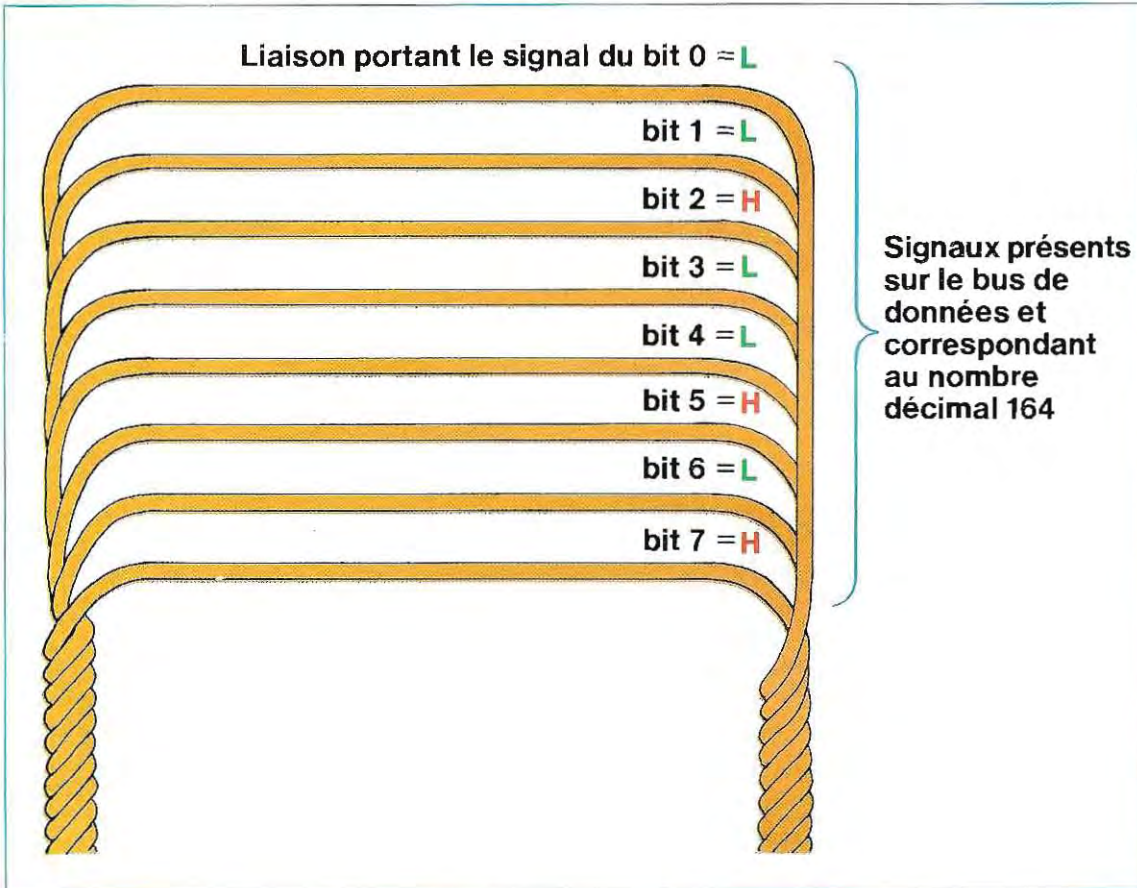
Pour représenter un nombre en langage machine, l'ordinateur dispose d'autant de liaisons électriques que de bits composant le nombre. La représentation d'un mot de 8 bits exige 8 liaisons, consacrées chacune à indi-

quer l'état du bit « transporté ». Ce groupe de liaisons prend le nom générique de **bus**. Les ordinateurs comportent deux bus pour assurer l'échange des données : l'un pour la circulation des données, l'autre pour les adresses de départ ou d'arrivée des données.

Dans un microprocesseur, le bus de données peut être de 8 bits (1 octet) et le bus d'adresse de 16 bits (2 octets). Le schéma en haut de la page

108 donne une numération des bits de 1 à 8 ; en utilisation courante, on préfère généralement la numération de 0 à 7 (voir le deuxième schéma de la page 108), car à partir de 0 le numéro successif de chaque bit en indique aussi la puissance de 2 (bit 0 :  $2^0$ , bit 1 =  $2^1$ , etc.).

Le schéma ci-dessous donne la situation des signaux pour un bus de données correspondant au nombre 164 décimal.



- En notation binaire, la plus petite quantité est un état, ou un niveau. Sa valeur est 0 ou 1, et elle prend le nom de bit, ou élément binaire.

- Un byte de 8 bits est un octet. La valeur maximale (décimale) que l'on peut écrire avec un octet est 255.

- Le canal électrique sur lequel circulent les données est le bus de données : dans la majorité des micro-ordinateurs, il comporte 8 ou 16 liaisons et peut donc transférer,

simultanément 8 ou 16 bits, c'est-à-dire 1 ou 2 octets.

- Pour prélever ou déposer une donnée, il faut fournir aux circuits électroniques l'adresse de cette donnée, c'est-à-dire sa position dans la mémoire.

- Les adresses des données circulent sur un connecteur électrique, le bus d'adresses. Ce bus est normalement de 16 bits, ou 2 octets pour une mémoire de capacité égale ou inférieure à 64 K octets.

## Le code ASCII

L'ordinateur, comme toutes les machines numériques, ne reconnaît que les symboles 0 et 1 (ou L et H). Pour pouvoir introduire des données (mots ou nombres) dans la machine, il faut donc avant tout les transformer en une série de 0 et de 1 ; de même, pour lire les données contenues dans l'ordinateur, il faut effectuer la transformation inverse, c'est-à-dire tirer d'une série de 0 et de 1 les lettres de l'alphabet ou les chiffres décimaux équivalents. La transformation en une série de 0 et de 1 s'appelle **codage** ; l'opération inverse est le **décodage**.

La forme de codage la plus simple est la

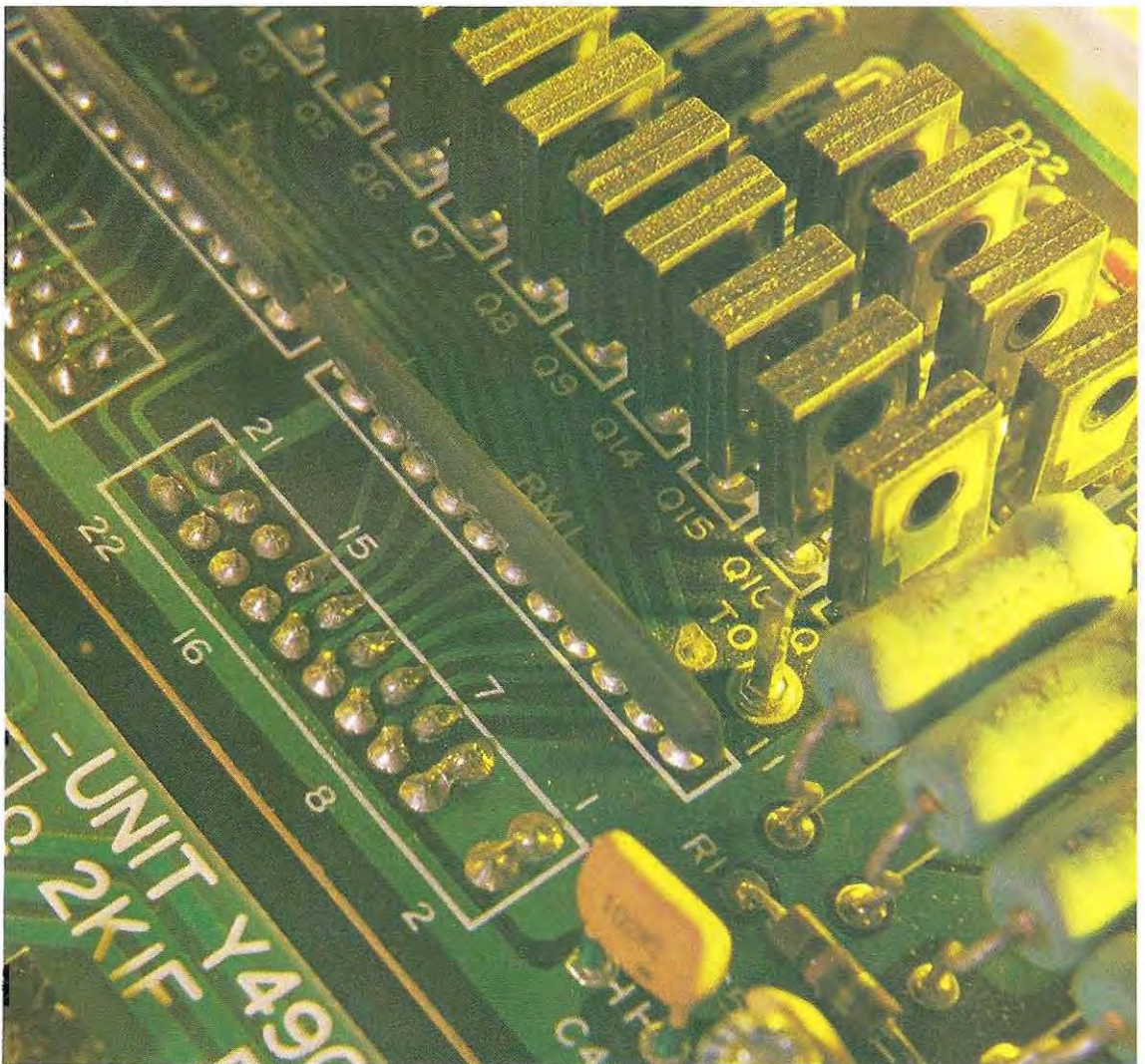
transformation d'un nombre décimal en son équivalent binaire, c'est-à-dire l'opération qui fournit la représentation d'un nombre sous la forme de symboles 0 et 1.

Pour les lettres de l'alphabet, il n'existe pas de codage mathématique ; la méthode consiste à établir une table de traduction, arbitraire dans sa conception, et qui fournit pour chaque lettre de l'alphabet (ou autre symbole) une représentation sous la forme d'une série de 0 et de 1.

Cette représentation s'appelle **code**. Chaque lettre de l'alphabet a son propre code, sous lequel l'ordinateur la reconnaît.

À l'inverse, quand l'ordinateur fournit des données, il faut disposer de circuits capables de

**Détail des circuits de commande d'une imprimante. En haut et à droite, on aperçoit les transistors de puissance qui pilotent la tête d'impression.**





traduire les informations binaires en symboles graphiques compréhensibles par l'utilisateur.

L'hypothèse d'un codage arbitraire n'est pas admissible car si les tables étaient différentes, les ordinateurs ne pourraient échanger des données que par l'intermédiaire d'opérations compliquées de conversion des codes. En outre, un code arbitraire n'offre aucune possibilité de vérifier si la donnée a été transmise correctement; elle ne permet donc pas de détecter d'éventuelles erreurs.

Telles sont précisément les raisons qui ont présidé à la création d'un code standard adopté par une majorité de constructeurs, le code **ASCII** (American Standard Computer Information Interchange).

Ce code utilise 7 bits pour traduire un symbole, d'où la possibilité d'utiliser 128 codes différents (de 0 à 1111111 = 127 décimal).

Le code ASCII est divisé en trois groupes, ayant chacun des caractéristiques et des fonctions bien déterminées :

- codes transparents
- symboles et nombres
- lettres

### Codes transparents

Ce terme recouvre tous les codes qui n'ont pas de correspondant alphabétique et qui sont utilisés pour transmettre des commandes.

Au cours de l'échange de données entre l'ordinateur et les périphériques, il est indispensable de transmettre, en dehors des données proprement dites, une série d'instructions assurant le fonctionnement de ces périphériques.

Par exemple, lorsqu'il pilote une imprimante, l'ordinateur doit envoyer les commandes de changement de lignes et d'avancement du papier; la suite des instructions d'impression est donc la suivante :

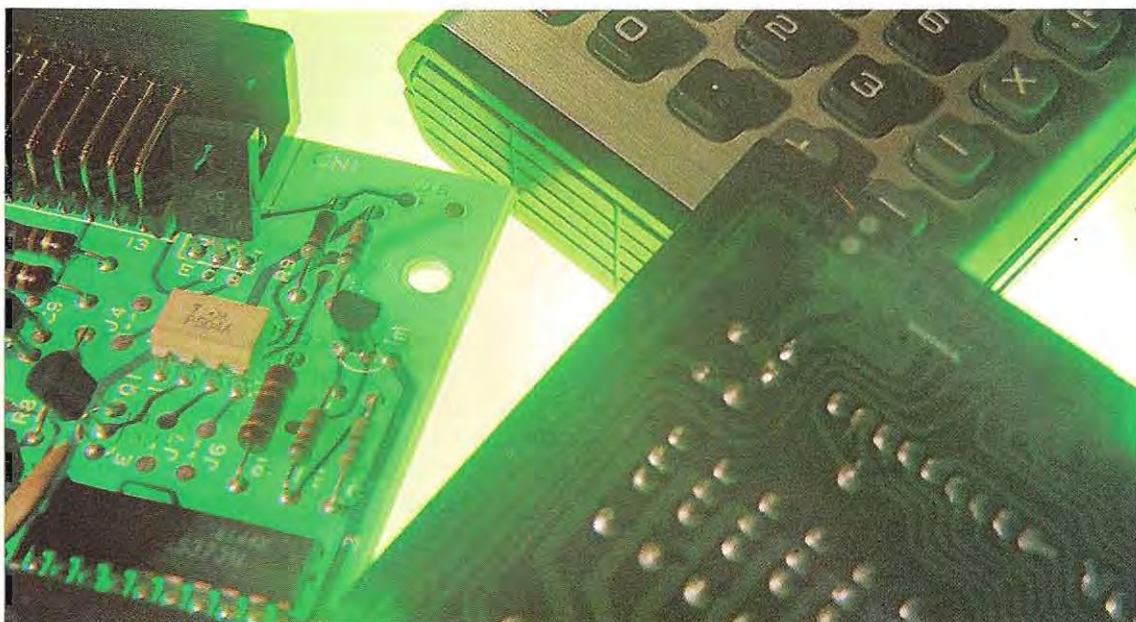
- 1 - Positionnement de la tête d'impression au début de la première feuille.
- 2 - Envoi et impression de la première ligne de caractères.
- 3 - Ordre donné à la tête d'impression de retourner à la marge gauche de la feuille.
- 4 - Ordre d'avancement du papier.
- 5 - Envoi et impression de la deuxième ligne de caractères.
- 6 - Reprise du cycle à partir du point 3, et répétition pour toutes les lignes à imprimer.

Les codes transmis aux points 1, 3 et 4 sont des codes transparents car ils ont pour fonction la commande de l'imprimante; ils sont envoyés aux circuits électroniques qui assurent la commande. Les codes transparents sont au nombre de 31 et ils occupent les premières positions, c'est-à-dire que leurs valeurs sont comprises entre 1 et 31 décimal (1 et 1F hexadécimal).

Nombre de ces codes transparents sont utili-

## PRINCIPAUX CODES TRANSPARENTS

	Sigle	DEC code décimal	HEX code hexadécimal
<b>1. Sonnerie (Bell)</b> Caractère d'appel dans le terminal	BEL	7	7
<b>2. Retour arrière (Backspace)</b> Caractère d'espacement arrière	BS	8	8
<b>3. Retour de chariot (Carriage Return)</b> Caractère de passage à la ligne	CR	13	D
<b>4. Echappement (Escape)</b> Caractère de changement de code	ESC	27	1B
<b>5. Présentation de formule (Form Feed)</b> Caractère d'alimentation en papier	FF	12	C
<b>6. Tabulation horizontale (Horiz. Tab.)</b>	HT	9	9
<b>7. Interligne (Line feed)</b> Caractère de changement de ligne	LF	10	A
<b>8. Tabulation verticale (Vert. Tab.)</b>	VT	11	B



Centre THC

### Une calculatrice démontée et ses circuits internes.

sés pour des applications particulières, comme l'échange de données entre ordinateurs, et non dans les programmes d'application.

On trouvera page 111 la liste des principaux codes utilisés dans les applications.

Pour plus de clarté, le tableau donne la traduction française des sigles qui proviennent de l'anglais.

**1 - Bell** (Décimal = 7, HEX = 7). Beaucoup de terminaux disposent d'un signal d'avertissement acoustique pour appeler l'attention de l'opérateur. C'est le code 7 qui active la sonnerie.

**2 - Backspace** (Décimal = 8, HEX = 8). C'est la commande de recul d'un caractère. On l'utilise pour corriger des erreurs : la commande se fait par le code 8, et l'on tape la valeur correcte à la place de l'erreur. Cela ne peut évidemment fonctionner qu'avec l'écran, puisque le nouveau caractère vient effacer le précédent.

**3 - Carriage Return** (Décimal 13, HEX = D). Commande de retour du chariot d'impression en début de ligne. Avec l'imprimante, ce code actionne véritablement le chariot, alors que sur l'écran, c'est simplement le curseur électronique qui se déplace.

**4 - Escape** (Décimal 27, HEX = 1B). Ce code n'a pas d'effet visible immédiat. Il sert à infor-

mer le périphérique (imprimante ou écran) que les caractères venant à la suite sont encore des codes de commande. On en trouve une application courante sur les écrans de visualisation graphique. Le terminal, s'il n'est pas informé que les nombres envoyés indiquent les positions du curseur, les interprète comme des caractères à inscrire, avec un tout autre résultat. C'est le code ESCAPE qui signale la nature particulière de ces nombres. L'instruction prend la forme suivante : code ESCAPE + nombre.

Si l'on entre le nombre sans le faire précéder du code ESCAPE, on obtient simplement l'écriture de symboles.

**5 - Form Feed** (Décimal 12, HEX = C). Il commande le passage d'une page à la suivante dans l'imprimante. Il est bon de l'utiliser avant et après chaque phase d'impression, pour que les listings soient séparés par du blanc.

**6 - Horizontal Tabulation** (Décimal 9, HEX = 9). Qu'il s'agisse d'une imprimante ou d'un écran, le nombre maximum de caractères pouvant être inscrit sur une ligne est généralement constant, chaque caractère ou espace occupant une position bien définie (toutefois la densité de l'impression peut être modifiée également par un code). La commande HT permet de sauter un certain nombre de ces positions par

une seule instruction. Ce code est utilisé par exemple lorsque l'on ne veut pas écrire trop près de la marge gauche : le code 9 déplace le chariot d'impression (ou le curseur) jusqu'à la première tabulation horizontale; deux codes 9 successifs permettent de sauter à la deuxième tabulation, etc.

**7 - Line Feed** (Décimal 10, HEX = A). Permet de sauter d'une ligne à la suivante.

**8 - Vertical Tabulation** (Décimal 11, HEX = B). Le fonctionnement est le même que pour le code HT, si ce n'est que le passage d'une tabulation à l'autre se fait dans le sens vertical. Les codes HT (tabulation horizontale) et VT (tabulation verticale) peuvent être programmés sur certains périphériques, c'est-à-dire que le nombre de caractères d'une HT et le nombre de lignes d'une VT peuvent être modifiés par l'entrée de codes ultérieurs (voir tableau).

### TABLEAU DES CODES ASCII TRANSPARENTS

Décimal	Hexadécimal	Sigle	Fonction dans les programmes d'application
0	0	NUL	N'actionne aucune commande
1	1	SOH	
2	2	STX	
3	3	ETX	
4	4	EOT	
5	5	ENQ	
6	6	ACK	
7	7	BEL	Active le signal acoustique
8	8	BS	Fait reculer le curseur d'un espace
9	9	HT	Tabulation horizontale
10	A	LF	Avance d'une ligne
11	B	VT	Tabulation verticale
12	C	FF	Change de page
13	D	CR	Renvoie le chariot
14	E	SO	
15	F	SI	
16	10	DLE	
17	11	DC1	
18	12	DC2	
19	13	DC3	
20	14	DC4	
21	15	NAK	
22	16	SYN	
23	17	ETB	
24	18	CAN	
25	19	EM	
26	1A	SUB	
27	1B	ESC	
28	1C	FS	
29	1D	GS	
30	1E	RS	
31	1F	US	

## Symboles et nombres

Ce groupe de codes va de 32 décimal (20HEX), à 64 décimal (40HEX) et comprend des symboles et des nombres. Le tableau (ci-dessous) donne également les nombres décimaux, considérant les nombres comme des symboles graphiques. On peut avoir deux formes de représentation :

- valeur numérique
- symbole ASCII

La représentation numérique est utilisée pour effectuer les calculs, la représentation ASCII pour les opérations d'impression. Par

exemple, le nombre décimal 5 peut être représenté selon les deux modes : numérique (binaire) = 1 0 1, ASCII : 35 HEX = 1 1 0 1 0 1. En utilisant le symbole 5 comme un nombre (pour les calculs), son code est 101, alors que dans les opérations d'impression (en ASCII) il est 110101.

## Lettres

Le dernier groupe de codes comprend les lettres majuscules et minuscules, et un nombre limité de symboles, comme on peut le voir dans les tableaux ci-dessous et à droite.

**TABLEAU ASCII  
DES SYMBOLES ET DES NOMBRES**

Décimal	Hexa-décimal	Symbole ou nombre représenté	
32	20	Espace	Symboles 1 <sup>re</sup> partie
33	21	!	
34	22	"	
35	23	#	
36	24	\$	
37	25	%	
38	26	&	
39	27	'	
40	28	(	
41	29	)	
42	2A	*	Nombres
43	2B	+	
44	2C	,	
45	2D	-	
46	2E	.	
47	2F	/	
48	30	0	
49	31	1	
50	32	2	
51	33	3	
52	34	4	
53	35	5	
54	36	6	
55	37	7	
56	38	8	
57	39	9	Symboles 2 <sup>e</sup> partie
58	3A	:	
59	3B	;	
60	3C	<	
61	3D	=	
62	3E	>	
63	3F	?	
64	40	@	

**TABLEAU ASCII POUR LES LETTRES**

Décimal	Hexadécimal	Symbole
65	41	A
66	42	B
67	43	C
68	44	D
69	45	E
70	46	F
71	47	G
72	48	H
73	49	I
74	4A	J
75	4B	K
76	4C	L
77	4D	M
78	4E	N
79	4F	O
80	50	P
81	51	Q
82	52	R
83	53	S
84	54	T
85	55	U
86	56	V
87	57	W
88	58	X
89	59	Y
90	5A	Z
91	5B	[
92	5C	\
93	5D	]
94	5E	^
95	5F	_
96	60	`
97	61	a
98	62	b
99	63	c

Décimal	Hexadécimal	Symbole
100	64	d
101	65	e
102	66	f
103	67	g
104	68	h
105	69	i
106	6A	j
107	6B	k
108	6C	l
109	6D	m
110	6E	n
111	6F	o
112	70	p
113	71	q
114	72	r
115	73	s
116	74	t
117	75	u
118	76	v
119	77	w
120	78	x
121	79	y
122	7A	z
123	7B	{
124	7C	
125	7D	}
126	7E	~
127	7F	DEL

### Applications

Voici la suite de codes qu'il faut envoyer à l'imprimante pour écrire : « Ceci est un exemple ».

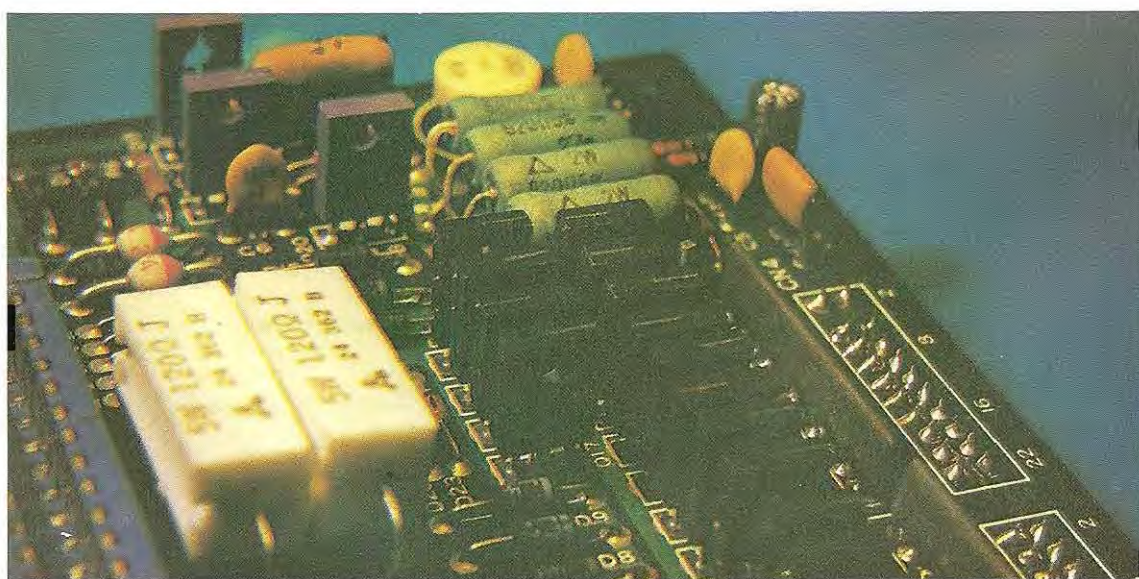
La procédure se compose des opérations :

- 1 - Envoi de la commande de mise en place du chariot en début de page (FF).
- 2 - Envoi des codes correspondant à la phrase à écrire.
- 3 - Envoi d'une nouvelle commande de changement de page (FF).

Le code correspondant aux étapes 1 et 3 est FF = 127 décimal = C (HEX). Pour le codage de la phrase, il faut procéder lettre par lettre, en se référant aux tables de conversion présentées page 114 et ci-contre.

Lettre	Code HEX
C	43
E	45
C	43
I	49
_____ espace = 20	
E	45
S	53
T	54
_____ espace = 20	
U	55
N	4E
_____ espace = 20	
E	45
X	58
E	45
M	4D
P	50
L	4C
E	45

Circuits de commande d'une imprimante. Au premier plan : deux résistances de puissance, plus grandes que des composants habituels.



FF  
12

C E C I E S T U N E X E M P L E  
43 45 43 49 20 45 53 54 20 55 4E 20 45 58 45 4D 50 4C 45

FF  
12

### Table ASCII en format compact

On trouvera ci-dessous la table des codes ASCII sous une forme plus pratique et plus compacte, c'est-à-dire organisée de manière à fournir les valeurs binaires de chaque symbole.

### TABLE DES CODES ASCII

		000	001	010	011	100	101	110	111
7 6 5 4 3 2 1	COLONNE LIGNE	0	1	2	3	4	5	6	7
0 0 0 0	0	NUL	DLE	SP	0	@	P	`	p
0 0 0 1	1	SOH	DC 1	!	1	A	Q	a	q
0 0 1 0	2	STX	DC 2	"	2	B	R	b	r
0 0 1 1	3	ETX	DC 3	#	3	C	S	c	s
0 1 0 0	4	EOT	DC 4	\$	4	D	T	d	t
0 1 0 1	5	ENO	NAK	%	5	E	U	e	u
0 1 1 0	6	ACK	SYN	&	6	F	V	f	v
0 1 1 1	7	BEL	ETB	'	7	G	W	g	w
1 0 0 0	8	BS	CAN	(	8	H	X	h	x
1 0 0 1	9	HT	EM	)	9	I	Y	i	y
1 0 1 0	10 (A)	LF	SUB	*	:	J	Z	j	z
1 0 1 1	11 (B)	VT	ESC	+	;	K	[	k	{
1 1 0 0	12 (C)	FF	FS	,	<	L	\	l	
1 1 0 1	13 (D)	CR	GS	-	=	M	]	m	}
1 1 1 0	14 (E)	SO	RS	.	>	N	^	n	~
1 1 1 1	15 (F)	SI	US	/	?	O	_	o	DEL

Par exemple, le symbole F occupe la ligne 6 et la colonne 4.

Le codage binaire de ce symbole s'obtient en réunissant les valeurs des bits de la ligne avec ceux de la colonne.

La ligne 6 a les valeurs

0 1 1 0 correspondant aux positions 4, 3, 2, 1

La colonne 4 a les valeurs

1 0 0

correspondant aux positions 7, 6, 5

Réunion

des deux nombres : F = 1 0 0 0 1 1 0 = 70 décimal = 46 hexadécimal



Ce résultat peut être obtenu très simplement en écrivant à la suite la colonne et la ligne.

**F** = colonne **4**, ligne **6** = **46** hexadécimal.

De même pour le symbole **m** qui se trouve dans la colonne 6 et la ligne 13 (= D) on a :

**m** = colonne **6**, ligne **13** (= D) = **6D** hexadécimal.

### Transmission des codes ASCII

La transmission des données entre l'ordinateur et les périphériques peut se faire de deux manières :

- par transmission en série
- par transmission en parallèle

La transmission en série consiste à envoyer un seul bit à la fois, l'un derrière l'autre. Au contraire, en mode parallèle, un certain nombre de bits sont transmis simultanément. La transmission en série n'exige qu'une seule liaison électrique puisque les signaux (bits) voyagent l'un derrière l'autre, alors que la transmission en parallèle nécessite autant de liaisons électriques que de signaux à envoyer simultanément. Les sché-

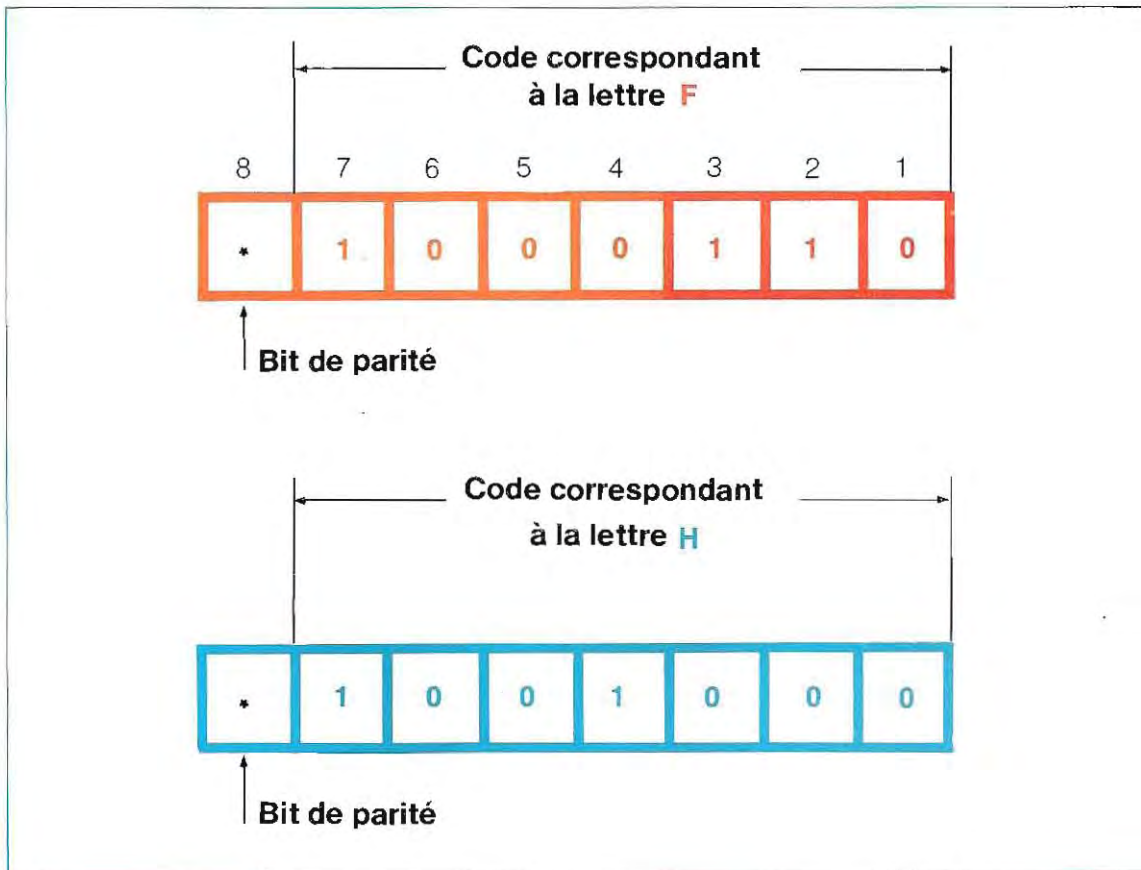
mas de la page 118 illustrent ces deux modes de transmission.

La transmission en parallèle permet une plus grande vitesse mais elle est plus coûteuse et presque impossible à réaliser sur de longues distances en raison du nombre très élevé des liaisons électriques; la transmission en série autorise au contraire des vitesses plus limitées mais elle est plus économique et donc plus largement utilisée.

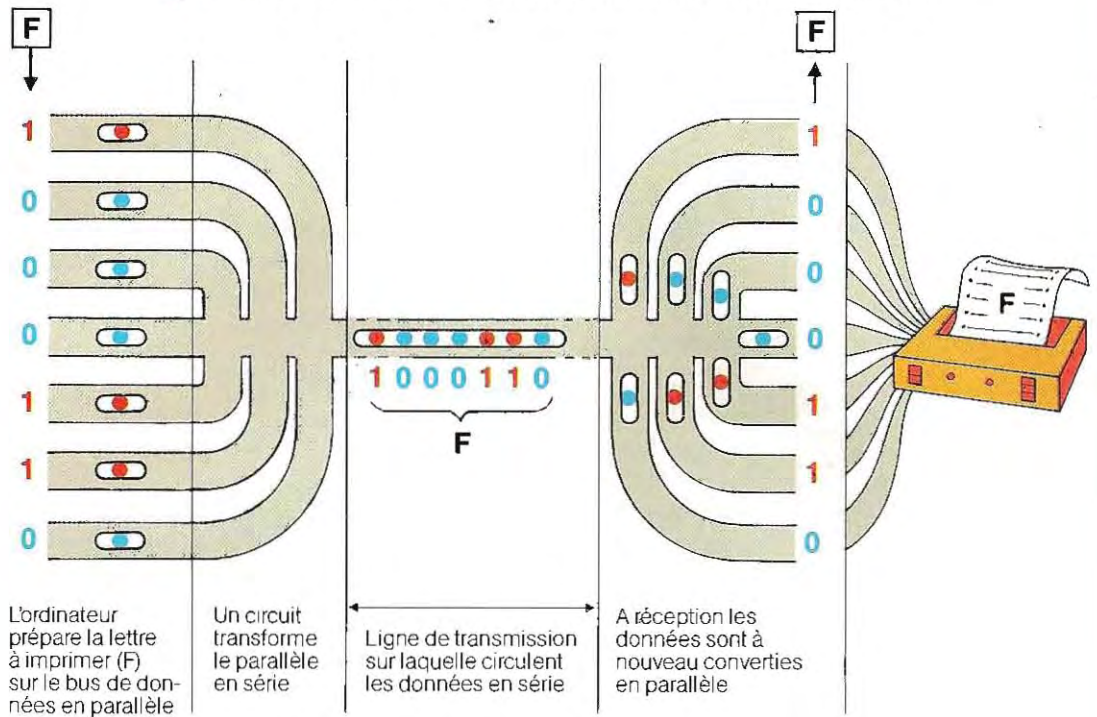
A partir de maintenant, nous travaillerons toujours dans l'optique de la transmission en série. Cette liaison prévoit pour le message envoyé deux formes différentes :

- mode asynchrone
- mode synchrone

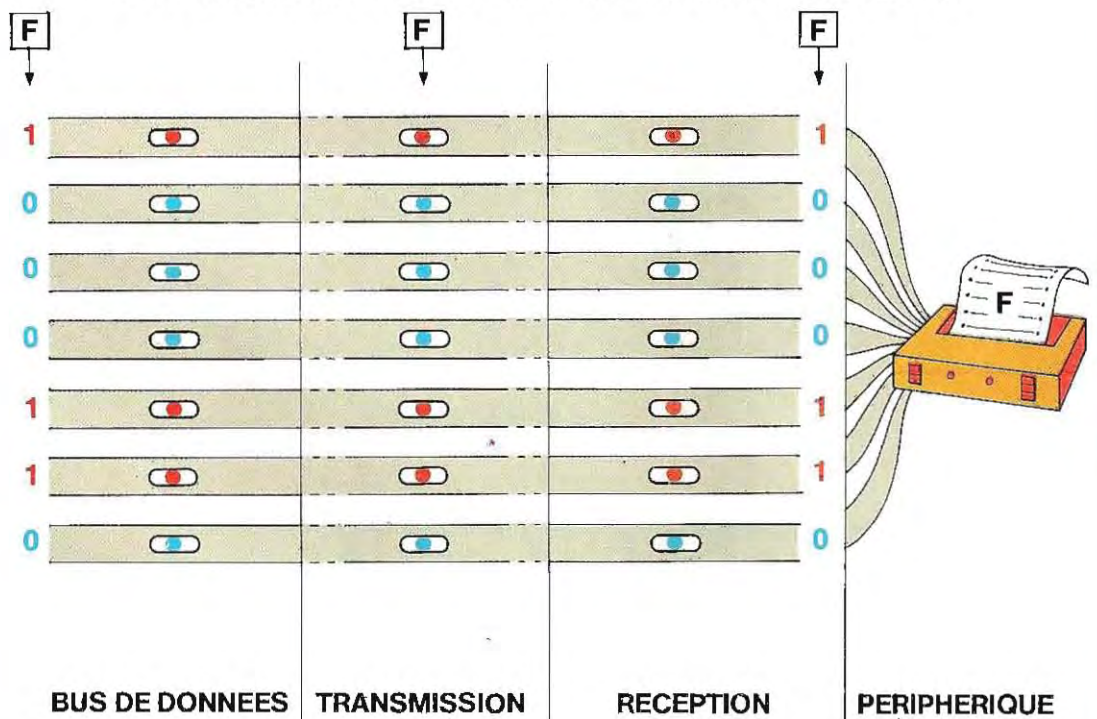
En mode asynchrone, l'espace (en termes de temps) entre un caractère à envoyer et le suivant est arbitraire, alors que la distance entre les bits de chaque caractère est fixe. Le schéma ci-dessous montre la transmission des caractères **F** et **H** en mode asynchrone.



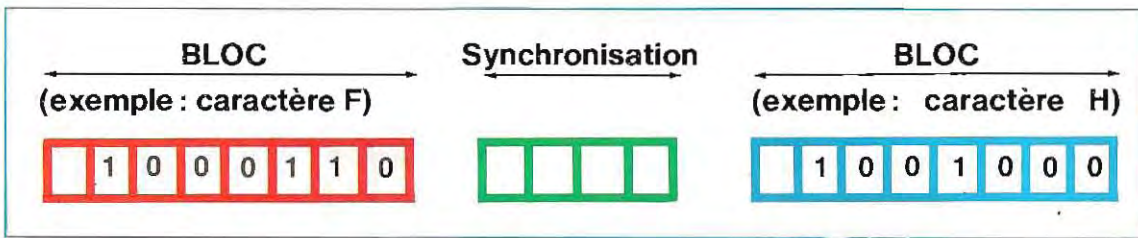
## SCHEMA LOGIQUE DE LA TRANSMISSION EN SERIE



## SCHEMA LOGIQUE DE LA TRANSMISSION EN PARALLELE





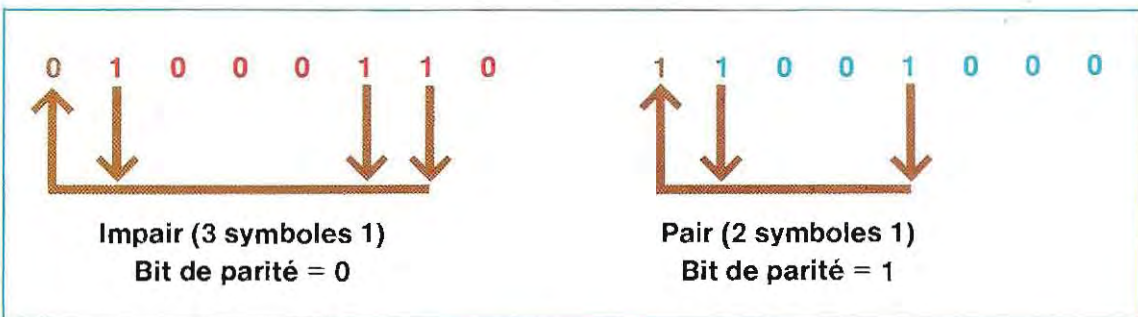


En mode synchrone, les données doivent être transmises par groupes que sépare un intervalle de temps déterminé et qui prennent le nom de **bloc**. Chaque bloc est séparé du suivant par un code particulier assurant la synchronisation du système.

Le schéma ci-dessus montre la structure en mode synchrone pour la transmission des deux mêmes lettres **F** et **H**.

Le schéma de la page 117 donne, en plus des 7 bits normaux du code ASCII, un bit signalé par le symbole\*. Ce bit de parité est un élément de

contrôle de la fiabilité du message reçu. Ce contrôle se fait en prenant le bit de parité (bit 8) égal à 1 si les symboles 1 sont présents en nombre pair dans les autres bits. Si la quantité de symboles 1 contenus dans le caractère à envoyer est un nombre impair, le bit de parité est égal à 0. Dans le schéma de la page 117, le caractère **F** (1000110) contient 3 symboles 1, son bit de parité est donc égal à 0; en revanche le caractère **H** (1001000) contient 2 symboles 1, son bit de parité prend la valeur 1. La forme du message devient donc :



Il arrive aussi que dans les applications on rencontre la logique inverse, c'est-à-dire :

bit de parité = 1 dans le cas impair,

bit de parité = 0 dans le cas pair.

En dehors du bit de parité, en mode de transmission asynchrone, il faut envoyer un signal de reconnaissance de début de caractère et un signal de fin de caractère.

Le signal de début de caractère prend le nom de bit de départ (START). Le signal de fin de caractère s'appelle bit d'arrêt (STOP). Un message se conclut normalement par deux bits d'arrêt.

Le format complet est donc le suivant :

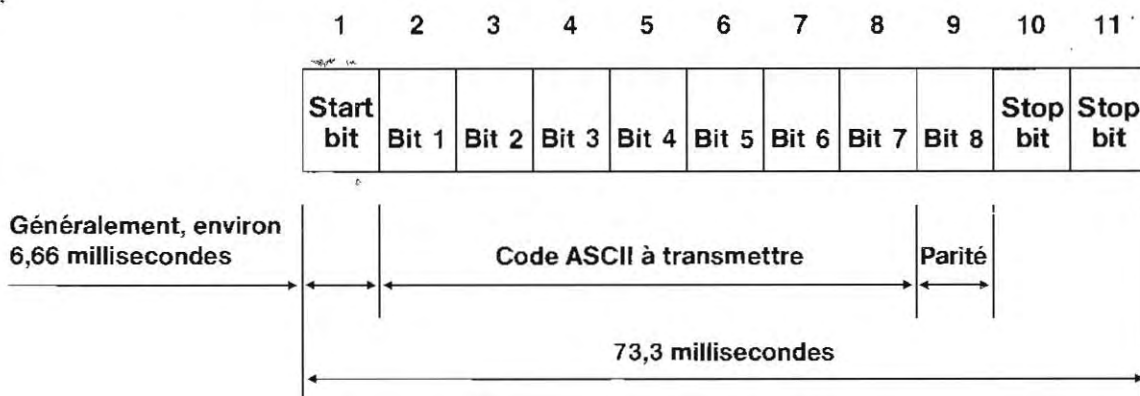
- bit de START (départ)
- 7 bits de la donnée
- bit de parité
- premier bit de STOP (arrêt)
- deuxième bit de STOP (arrêt).

La vitesse de transmission s'exprime en nombre de modulations par seconde, ou d'unités d'information par seconde. On parle plus volontiers

de bits par seconde (bps) : l'unité en est le **baud**.  
Le schéma ci-dessous représente un message dont la vitesse est 150 bauds. Dans ces trans-

missions, le bit de départ vaut 0 et les bits d'arrêt valent 1. L'état 1 prend le nom de MARQUE (MARK), l'état 0 celui de ESPACE (SPACE).

### STRUCTURE DES SIGNAUX EN TRANSMISSION A 150 BAUDS



- Les données (symboles, lettres et nombres) sont codées selon le code ASCII.

- Le code ASCII est utilisé dans les transmissions vers les périphériques (écrans, imprimantes) ou entre ordinateurs.

- Pour la gestion des périphériques, il existe un certain nombre de codes particuliers que l'on appelle **CODES TRANSPARENTS**.

- La transmission des données peut se faire selon deux modes : transmission **EN SERIE** et transmission **EN PARALLELE**.

- La transmission **EN SERIE** est la plus utilisée dans les échanges entre ordinateurs ; elle est **ASYNCHRONE** ou **SYNCHRONE**.

- En transmission **ASYNCHRONE**, en dehors des 7 bits constituant la donnée (le caractère), il faut transmettre :

Un premier bit de **START (DEPART)**, qui vaut normalement 0 ;

le bit de **PARITE**, après les 7 bits de la donnée ;

deux bits de **STOP (d'ARRET)**, qui valent normalement 1.

L'ordinateur à l'école : jeunes élèves aux prises avec des imprimantes en fonctionnement.



## Contrôle de réception

Dans la transmission d'un caractère en mode asynchrone, le bit de parité permet de contrôler la validité de la donnée. Il faut donc pouvoir vérifier à la réception la cohérence entre le nombre de symboles 1 présents dans la donnée et la valeur du bit de parité.

À l'arrivée du message, le circuit récepteur extrait le premier bit (Départ, Start) et les deux derniers (Arrêt, Stop ou Fin), et présente donc au périphérique (écran ou imprimante) un paquet de 8 bits.

Les 7 premiers bits constituent la donnée proprement dite, et le dernier le bit de parité.

Pour contrôler la validité de la donnée reçue :

- 1 / on isole le 8<sup>e</sup> bit (bit de parité) ;
- 2 / on compte le nombre de symboles 1 présents parmi les autres bits ;
- 3 / si le total « donnée » est pair et si le bit 8 vaut 1, ou si le total « donnée » est impair et si le bit 8 vaut 0, la donnée est valide.

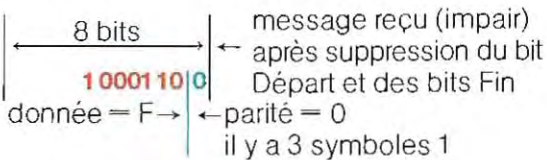
Pour notre exemple, nous avons choisi arbitrairement un message impair : l'inverse est parfaitement concevable.

Pour isoler le 8<sup>e</sup> bit des 7 autres (qui représentent la donnée), on peut adopter l'opérateur ET :

- pour isoler la parité :**
- message à l'arrivée ET 00000001 ;**
- pour isoler la donnée :**
- message à l'arrivée ET 11111110.**

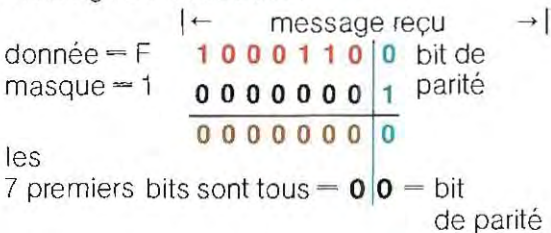
En fait, en utilisant l'opérateur ET entre un nombre binaire quelconque (message reçu) et la valeur 1, on obtient le dernier bit du message reçu, c'est-à-dire le bit de parité.

Par exemple, dans le cas de la lettre **F** = 10001100 :

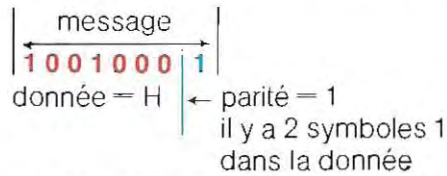


La vérification du bit de parité se fait avec l'opérateur ET.

Message ET 00000001 :

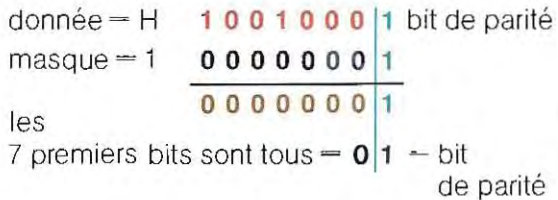


Mais dans le cas de la lettre **H** = 10010001 :



Dans ce cas également, pour extraire le bit de parité, il suffit d'appliquer l'opérateur ET et le masque 00000001.

Message ET 00000001 :



La valeur constante avec laquelle on effectue l'opération ET (1 dans les exemples ci-dessus) pour isoler une partie d'une donnée prend le nom de **masque**, car elle permet d'extraire la partie intéressante en la lisant comme s'il y avait un filtre sur la donnée.

Pour extraire du message reçu (8 bits) les 7 bits qui constituent la lettre transmise, il faut appliquer le masque = 11111110.

Message ET 11111110



Les exemples ci-dessus montrent comment on isole les données et les bits de parité à l'aide de programmes conçus à cet effet. Les circuits récepteurs modernes possèdent cette fonction au niveau du matériel, c'est-à-dire sans apport d'aucun programme.

## Codes de sécurité

Les listings de la page 78 donnent les résultats obtenus en appliquant les différents opérateurs logiques entre le nombre fixe 11 (masque) et le nombre J variable de 65 à 90. Ces valeurs sont les codes ASCII correspondant aux lettres en majuscule (A = 65 ; Z = 90).

La colonne OUX fournit un excellent instrument de codification car elle associe à chaque lettre une autre valeur différente et qu'on ne peut déduire sans connaître la valeur du masque. Sur la première ligne du listing 3 page 78, on lit :

$$11 \text{ OUX } 65 = 74$$

La lettre A (65) devient J (74).

En changeant de masque (listing 4 page 78)

$$32 \text{ OUX } 65 = 97$$

La lettre A devient a (97).

Cette méthode permet une certaine sécurité dans la transmission des données. Il existe cependant des méthodes pour forcer les systèmes de sécurité à base de codification ; elles consistent à exploiter la vitesse de calcul considérable de la machine pour effectuer un grand nombre de tentatives, jusqu'à découvrir par approximations successives la clé de code utilisée pour le chiffrement.

## Solutions du test 2

1 / Le test propose le calcul du résultat à l'aide de l'opérateur  $\overline{A \text{ OUX } B}$ , c'est-à-dire NON (A OUX B).

Pour faciliter la recherche, il faut d'abord calculer A OUX B, puis sa négation.

A	B	A OUX B	$\overline{A \text{ OUX } B}$	A ET ( $\overline{A \text{ OUX } B}$ )
1	0	1	0	0
1	1	0	1	1
0	0	0	1	0
0	1	1	0	0

On notera que A ET ( $\overline{A \text{ OUX } B}$ ) est identique à A ET B.

2 / a)  $11011 \text{ ET } 011 = 00011$

b) Avant de résoudre l'opération logique il faut convertir les deux nombres décimaux 12 et 7 en binaire :

$$12 \text{ décimal} = 1100 \qquad 7 \text{ décimal} = 0111$$

$$12 \text{ ET } 7 = 1100 \text{ ET } 0111 = 0100$$

c'est-à-dire, en reconvertissant tout en décimal ( $0100 = 4$ )

$$12 \text{ ET } 7 = 4 \text{ (en décimal)}$$

c) Les nombres 12 et 7 octaux possèdent les représentations binaires suivantes :

$$12 \text{ octal} = 001010 \qquad 7 \text{ octal} = 000111$$

donc :

$$12 \text{ ET } 7 = 0001010 \text{ ET } 000111 = 000010$$

d'où :

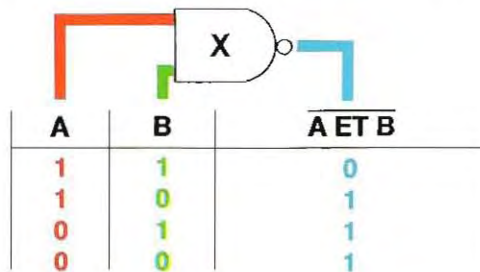
$$12 \text{ ET } 7 = 2 \text{ (en octal)}$$

d)  $1101 \text{ OU } 0110 = 1111$

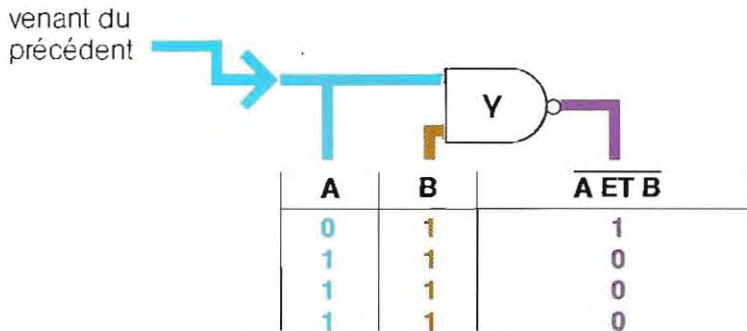
e)  $1011 \text{ OUX } 1101 = 0110$

3 / Pour trouver la solution, se reporter au texte relatif aux opérateurs logiques : NON, ET, OU, OUX, pages 80 à 83 et pages 88 à 91.

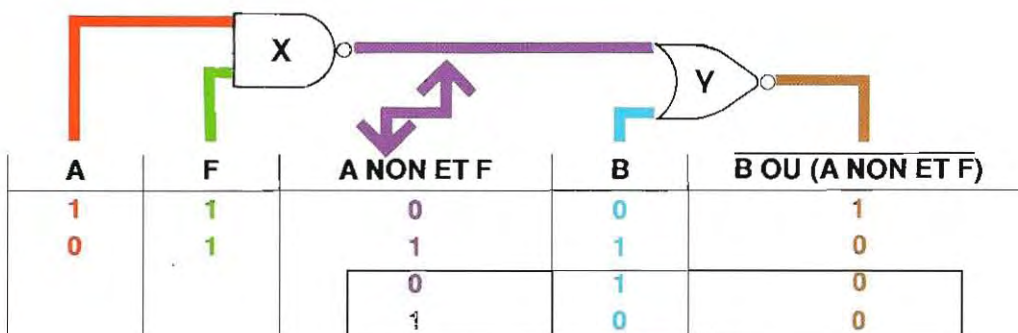
4 / a) Dans les circuits composites, il faut toujours partir du premier, résoudre sa table de vérité puis l'appliquer au suivant. Cette méthode donne pour le circuit X:



La colonne  $\overline{A \text{ ET } B}$  (0111) devient l'une des deux entrées du circuit suivant (Y). La seconde entrée de ce circuit est fixe, et vaut 1. De ce fait:



b)

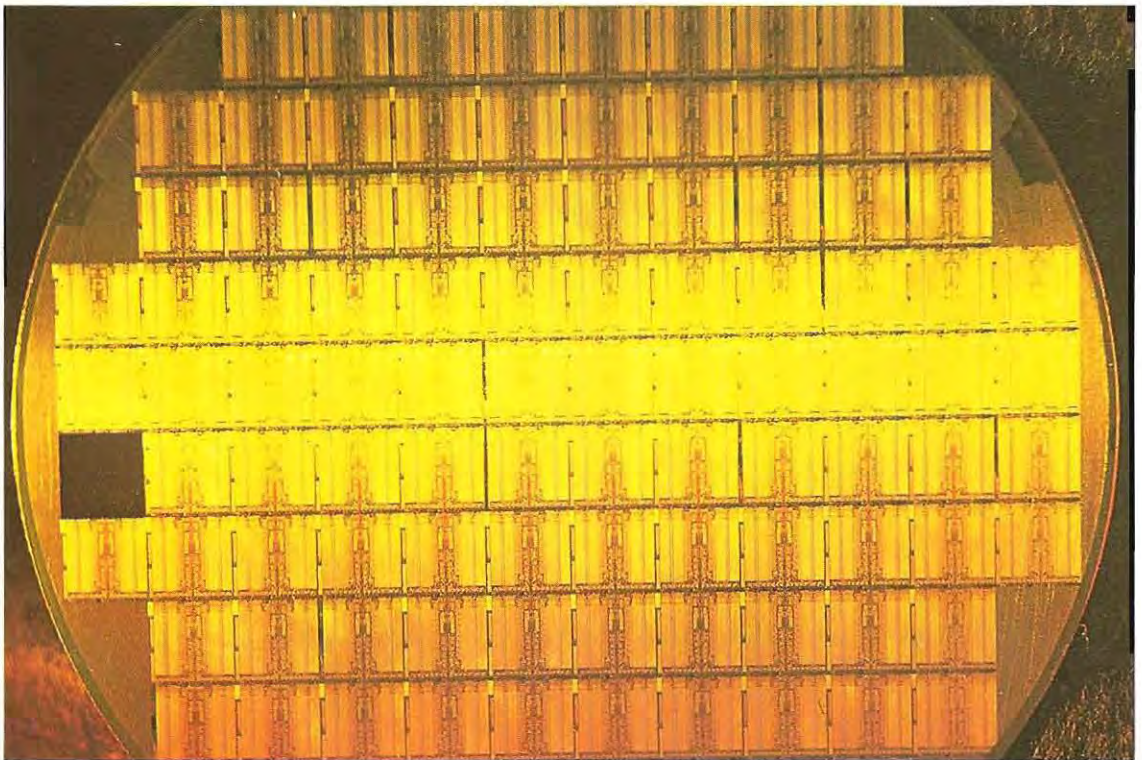


On a répété dans la seconde partie du tableau les états 0, 1 de A NON ET F pour obtenir toutes les combinaisons possibles avec la colonne B.

## TEST 3



- 1 / Ecrire les chiffres 7, 9, 0, 3 en binaire, en nombres et en code ASCII.
- 2 / Enumérer les principales différences entre la transmission en série et la transmission en parallèle.
- 3 / Construire le message complet qui doit être envoyé pour transmettre le caractère K en mode asynchrone.
- 4 / Traduire en code décimal les opérations suivantes réalisées sur imprimante :
  - a) présentation de formule (saut de page)
  - b) écriture des caractères X et Y
  - c) interligne
  - d) écriture des mêmes caractères en minuscules
  - e) saut de page
- 5 / En étudiant la table des codes ASCII, dire quelle est l'opération qu'il faut accomplir pour transformer une lettre majuscule en minuscule.
- 6 / Ecrire le masque qui extrait d'une donnée quelconque les bits 3, 7, 1 (les bits étant numérotés à partir de 0). Contrôler l'exactitude de la réponse en appliquant le masque sur la représentation binaire des nombres décimaux 12, 21, 6, 15.  
*Les solutions du test se trouvent pages 150 et 151.*



Archives Curcio/IBM

**Le dispositif numérique à jonction Josephson permet des vitesses de commutation très élevées.**

# Microsystèmes

Pour bien utiliser un système à microprocesseur (ordinateur personnel ou micro-ordinateur professionnel) il faut connaître son architecture interne.

On peut distinguer trois niveaux d'utilisation du micro-ordinateur.

- **Utilisation de programmes figés.**
- **Utilisation de certains programmes évolutifs et adaptables.**
- **Développement de programmes spécifiques (« à la demande »).**

Le degré de connaissance est naturellement fonction du type d'application. Dans tous les cas, un examen, même sommaire, de la structure du système est indispensable.

La structure physique de l'ordinateur est du **matériel** (hardware), tandis que ce qui concerne la programmation prend le nom de **logiciel** (software).

En micro-informatique, c'est la technologie du matériel qui est à l'origine des progrès les plus décisifs.

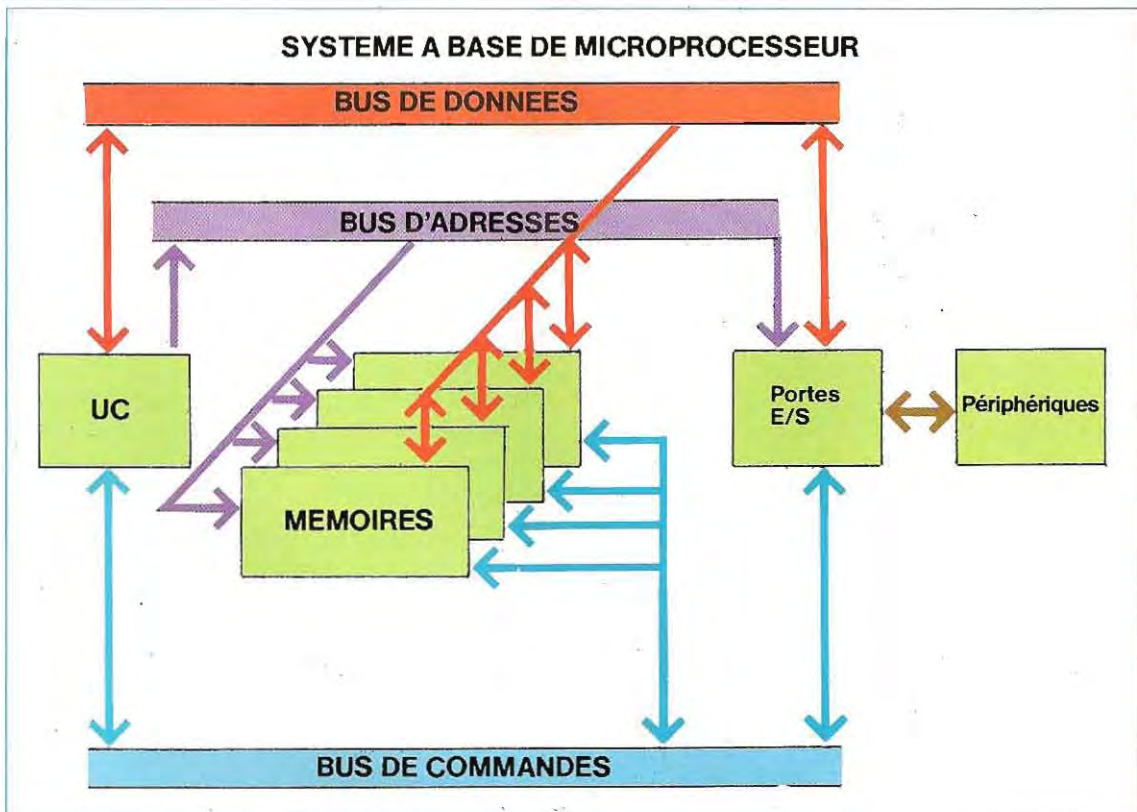
Cela se traduit par la production de systèmes à la fois plus performants et moins coûteux. C'est même à tel point qu'aujourd'hui certains programmes coûtent plus cher que la machine elle-même.

Les possibilités considérables de tels équipements permettent en outre de vulgariser l'emploi des micro-ordinateurs dans un nombre croissant d'applications nouvelles. Cette prolifération ne peut que se poursuivre.

## Les principaux organes

Le schéma ci-dessous montre l'organisation interne d'un système à base de microprocesseur. Les éléments principaux en sont :

- **UC = unité centrale (ou CPU = Central Processing Unit)**
- **MEMOIRES**
- **PORTES ENTREE/SORTIE (E/S)**
- **PERIPHERIQUES**
- **BUS DE DONNEES**
- **BUS D'ADRESSES**
- **BUS DE COMMANDES**



## UNITE CENTRALE (UC)

Elle exécute deux fonctions principales : commande de tous les autres composants et exécution des calculs.

## MEMOIRES

Dispositif renfermant sous une forme binaire les instructions qui constituent le programme et les données qui seront utilisées dans l'exécution d'un programme.

## PORTES E/S

Circuits permettant la communication du système avec le monde extérieur. Le sigle E/S (entrée/sortie) correspond à l'anglais I/O (Input/Output).

## PERIPHERIQUES

Tous les dispositifs extérieurs à l'ordinateur et lui servant d'E/S.

Les principaux périphériques sont :

le **visuel** (ou **moniteur**) : écran d'affichage permettant à l'opérateur de dialoguer avec la machine ;

le **clavier** : très comparable au clavier d'une machine à écrire, c'est l'élément qui permet d'envoyer des caractères et des ordres à la machine (saisie) ;

l'**imprimante** : organe de sortie sur papier ;

l'**unité à disque, disquette, ou bande magnétique** : ils constituent la mémoire de masse du système et permettent de stocker d'une manière permanente une grande quantité de données. Ces dispositifs ne sont pas de véritables périphériques mais des extensions de la mémoire centrale de la machine.

**des dispositifs spéciaux** : dans des applications spécifiques (contrôle de procédés dans l'industrie), ils sont indispensables pour piloter des machines ou en surveiller le fonctionnement.

## BUS DE DONNEES

Ensemble de lignes physiques assurant le transfert des données. Les systèmes utilisent un bus de données constitué de circuits parallèles pouvant transporter 8 ou 16 bits (1 ou 2 octets).

Dans l'usage courant, l'octet est souvent appelé « mot » (ou caractère), mais la réciproque n'est pas vraie : dans certains systèmes, le mot comprend 16 bits, c'est-à-dire deux octets, et parfois plus.

## BUS D'ADRESSES

Liaison sur laquelle circulent les signaux constituant l'adresse c'est-à-dire la localisation en mémoire des données.

## BUS DE COMMANDE

Troisième liaison, inséparable d'un système, par laquelle l'UC envoie ses instructions aux autres composants et en reçoit les signaux en réponse.

Le schéma en haut de la page ci-contre représente de façon détaillée les composants d'un système.

## Structure de l'unité centrale

L'unité centrale (UC) est le composant le plus complexe. C'est elle qui génère tous les signaux nécessaires au fonctionnement des autres circuits et qui effectue le traitement des données. Le schéma en bas de la page ci-contre montre que sa structure est comparable à celle du système tout entier. Ses composants principaux sont les suivants.

Le **bus interne** : ses fonctions sont analogues à celles des bus externes (voir schéma p. 127 en haut), et consistent à assurer le transfert de l'information.

Les **accumulateurs** et les **registres** : dispositifs de mémorisation des données avant et après le traitement.

Le **décodeur d'instructions** : circuit traduisant les instructions binaires d'un programme en une série d'impulsions électriques.

Les **circuits de commandes** : pilotés par le circuit précédent, ils servent à générer divers types de signaux pour assurer le fonctionnement des autres composants, à l'intérieur comme à l'extérieur de l'unité centrale (mémoires, portes, etc.).

L'**unité arithmétique et logique** ou **UAL** (en anglais ALU, Arithmetic and Logic Unit). C'est l'organe chargé d'exécuter les calculs et les opérations logiques.

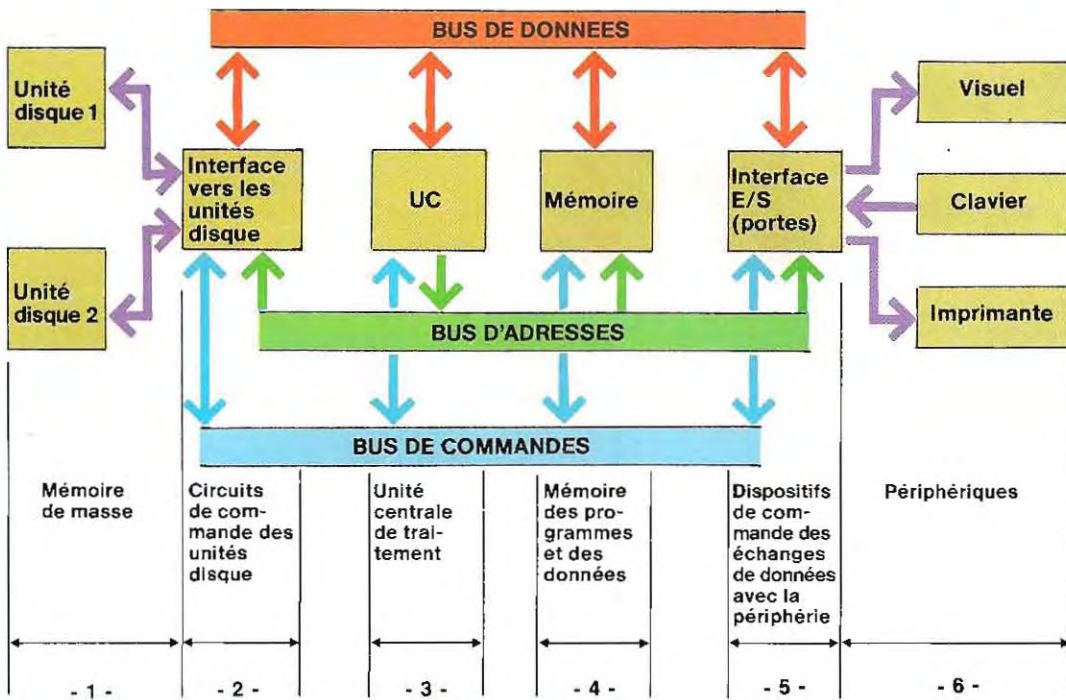
## Fonctionnement de l'unité centrale

Un programme se compose d'une série d'instructions, divisées au moins en deux parties :

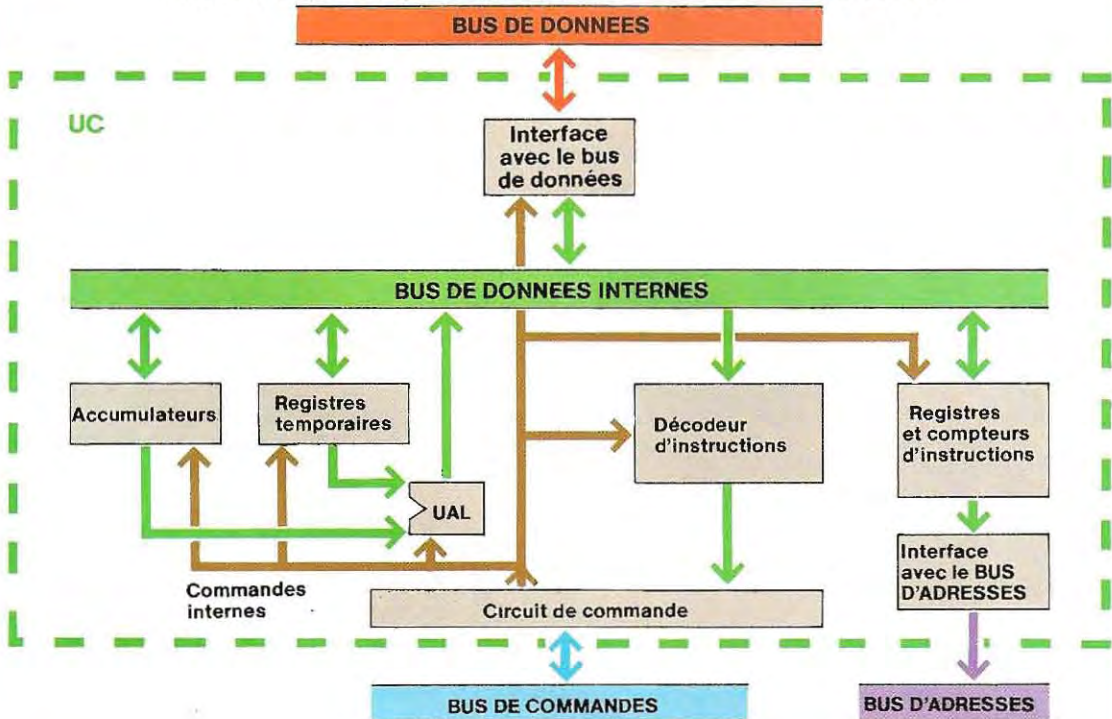
- **le code opération (opcode) :**  
**indique la nature de l'opération à effectuer**
- **la donnée (ou opérande) :**  
**quantité sur laquelle l'opération doit être effectuée.**



## SYSTEME COMPLET A MICROPROCESSEUR



## STRUCTURE CARACTERISTIQUE D'UNE UNITE CENTRALE



Les codes et les données (ou leurs adresses) sont inscrits dans les mémoires du système. L'ensemble de ces mémoires peut être considéré comme un livre composé de pages portant chacune un certain nombre de caractères. Considérons par exemple une mémoire divisée en pages de 1024 octets (voir schémas p. 125 et 128) ; pour connaître le contenu de la mémoire 3 de la page 1, il faut donner l'adresse 1026 (la première page porte le numéro zéro et contient les mémoires de 0 à 1023 ; la page 1 commence à 1024).

Chaque zone de mémoire ( composée d'une ou plusieurs pages) est réservée à un usage précis. Le tableau donnant la topographie des différentes zones et leur emploi s'appelle **image mémoire** ; il varie selon la machine et les programmes qu'elle accepte.

Le « déroulement » d'une instruction de programme se fait selon une séquence d'opérations appelée **cycle** :

■ **Extraction de l'instruction**  
(en anglais **FETCH**)

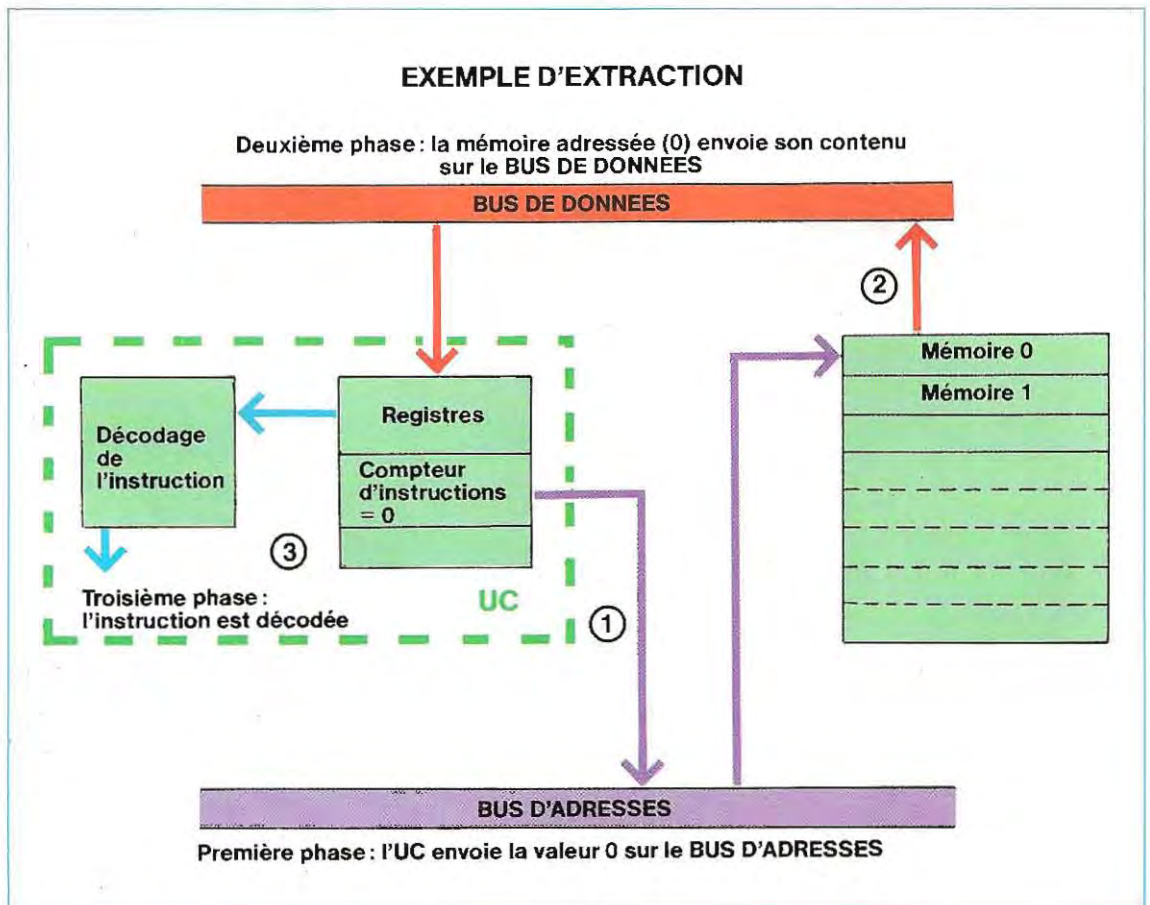
■ **Décodage et exécution**  
(en anglais **EXECUTE**)

C'est la forme la plus simple ; beaucoup d'instructions, plus complexes, exigent plusieurs opérations d'exécution.

Au branchement de la machine, ou lorsqu'on donne l'ordre de commencer un travail, l'unité centrale extrait la première instruction, l'exécute, passe à la seconde, l'exécute, et ainsi de suite pour toutes les instructions.

Il existe à l'intérieur de l'unité centrale une mémoire que l'on appelle un **compteur d'instructions**, ou **compteur ordinal** (en anglais : Program Counter, PC). C'est elle qui stocke l'adresse des instructions successives à exécuter. Si toutes les instructions étaient séquentielles (c'est-à-dire si l'exécution du programme ne comportait pas de saut d'un point à un autre), il suffirait, pour connaître l'adresse des différentes instructions, d'ajouter 1 au contenu du compteur ordinal.

Le schéma ci-dessous illustre un exemple d'extraction (Fetch) d'instruction.



- ① L'UC envoie le contenu du compteur d'instructions (ici = 0) sur le bus d'adresses.
- ② La mémoire correspondante (adresse 0) envoie son contenu (instruction à exécuter par l'unité centrale) sur le bus de données.
- ③ Le contenu du bus de données (dans ce cas, une instruction à exécuter) est chargé dans une mémoire interne de l'unité centrale (registre); puis elle passe dans l'unité de décodage où elle est transformée en signaux électriques, par l'intermédiaire de **microprogrammes** qui sont inscrits par le constructeur dans l'unité centrale.

A la fin de ce cycle, on ajoute 1 au contenu du compteur d'instructions; dans l'instruction suivante, la mémoire portera l'adresse 1 et ainsi de suite. Il arrive fréquemment que le déroulement d'un programme n'entraîne pas l'exécution obligatoire de toutes les instructions de manière séquentielle. Un choix logique peut conduire à sauter une partie du programme. Pour stocker en mémoire le contenu du compteur d'instructions et pouvoir le

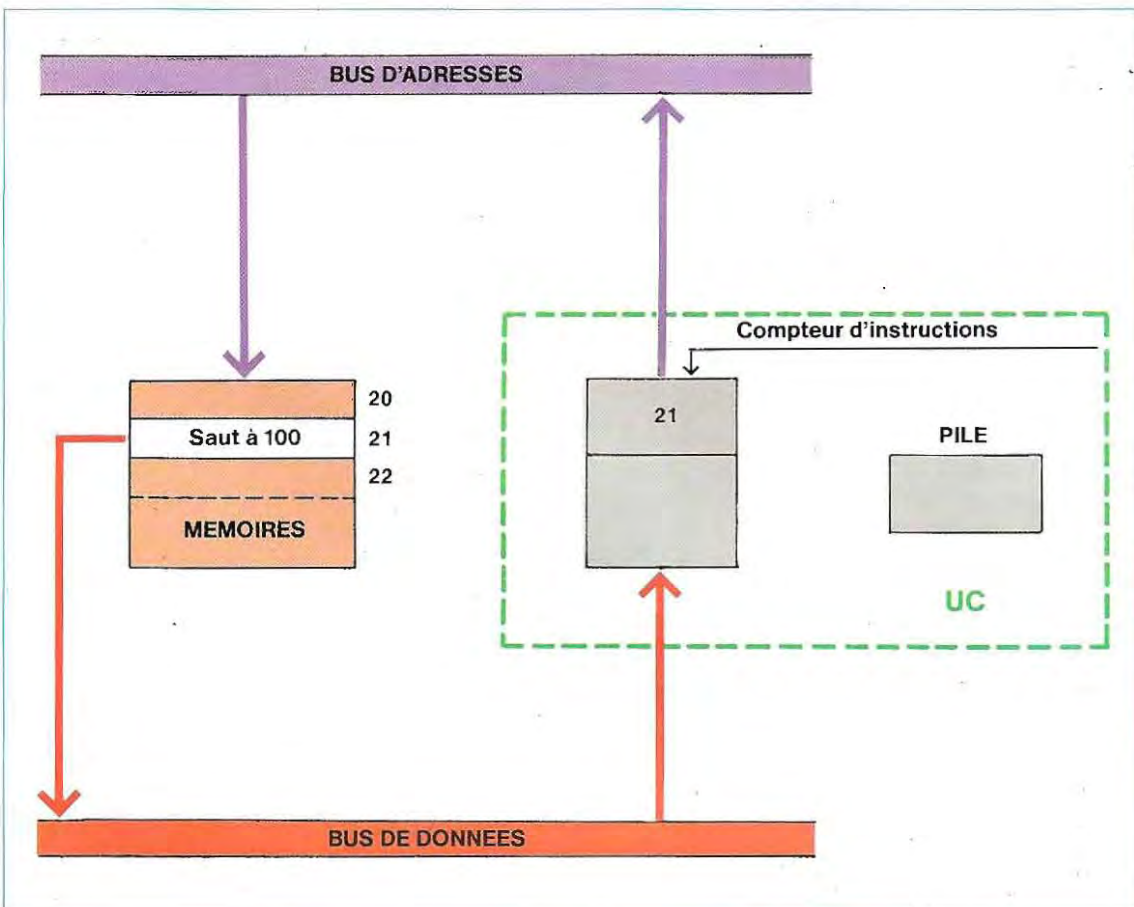
remettre à zéro, on prévoit à la fin du saut une zone de mémoire dénommée **pile** (stack) qui est une file d'attente.

A l'arrivée d'une instruction de rupture de séquence, l'unité centrale place le contenu du compteur d'instructions dans la pile; le compteur contient alors la valeur nouvelle (adresse), dont on extrait et exécute l'instruction. L'unité centrale reprend ensuite la valeur primitive du compteur d'instructions, ajoute l'**incrément**, et procède de manière séquentielle jusqu'à la prochaine instruction de saut.

### Phases d'une instruction de saut

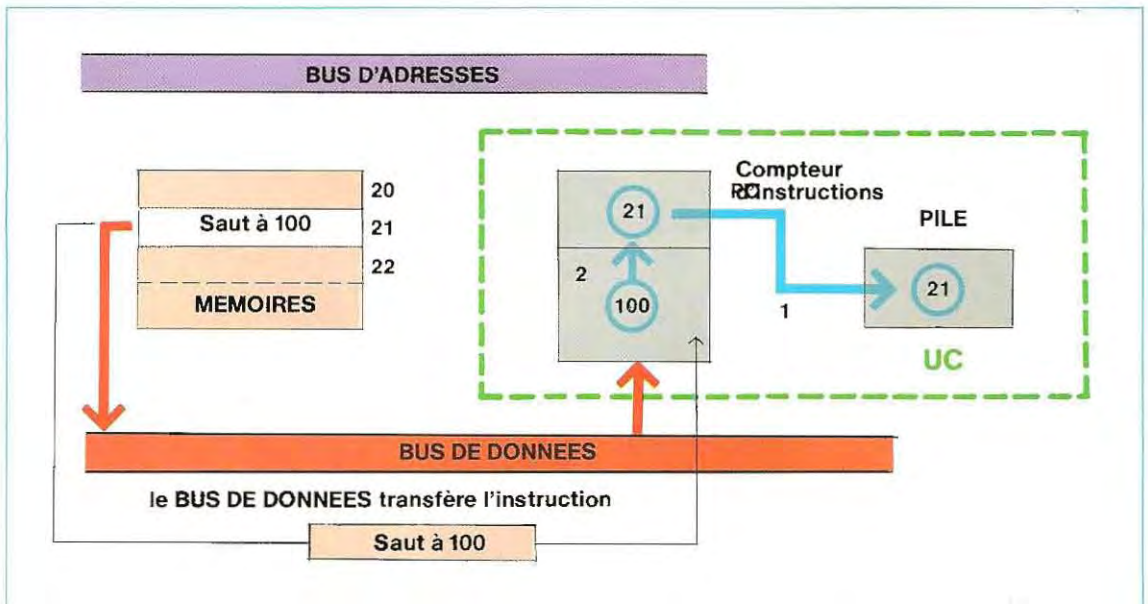
#### Première étape

Dans l'exécution du programme, le compteur d'instructions est parvenu (par exemple) à la valeur 21. Cette valeur est envoyée sur le bus d'adresses. La mémoire n° 21 répond en émettant sur le bus de données son contenu, par exemple le code correspondant à l'instruction « saut à 100 ». La file d'attente (pile) n'est pas encore concernée.

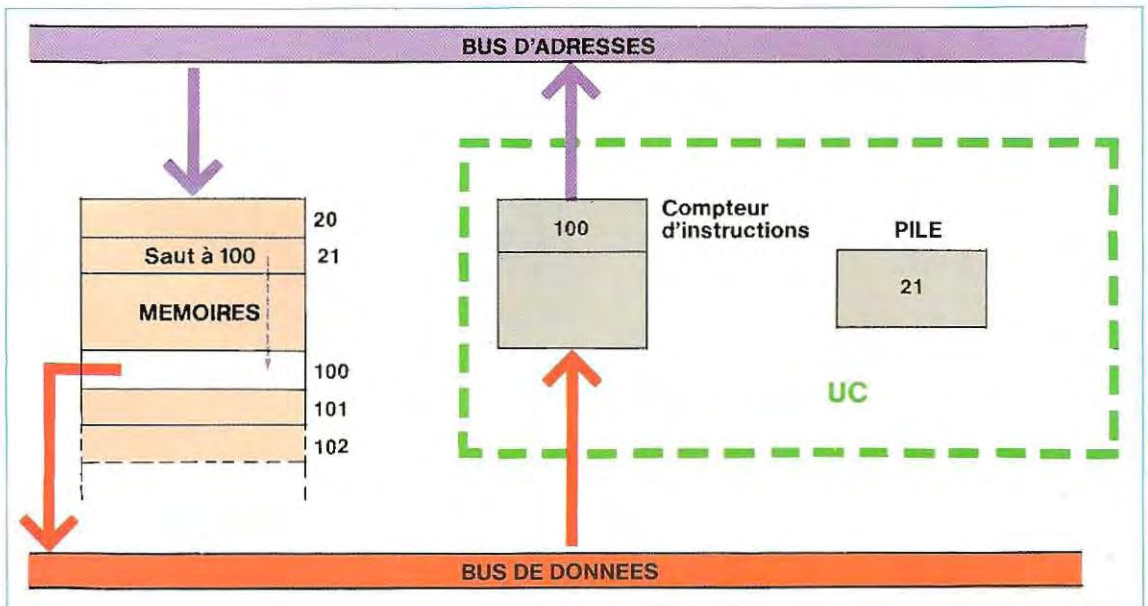


### Seconde étape

L'unité centrale reconnaît l'instruction de saut; elle archive le contenu du compteur d'instructions dans la pile 1 et transfère la nouvelle adresse (100) dans le compteur d'instructions 2. Cette valeur 100 représente l'instruction ultérieure à exécuter.



Le fonctionnement reprend alors son allure normale; l'unité centrale envoie sur le bus d'adresses le contenu du compteur d'instructions (100); la mémoire correspondante répond en envoyant son contenu sur le bus de données.



Cette nouvelle instruction est exécutée, et l'incrément 1 ajouté au compteur d'instructions ; l'unité centrale exécute à présent l'instruction portant l'adresse de mémoire 101, et ainsi de suite jusqu'à ce qu'elle rencontre une instruction particulière qui commande le retour (Return).

Quand l'unité centrale rencontre cette instruction, elle se contente de prélever le contenu de la pile et de le déposer avec incrément dans le compteur d'instructions.

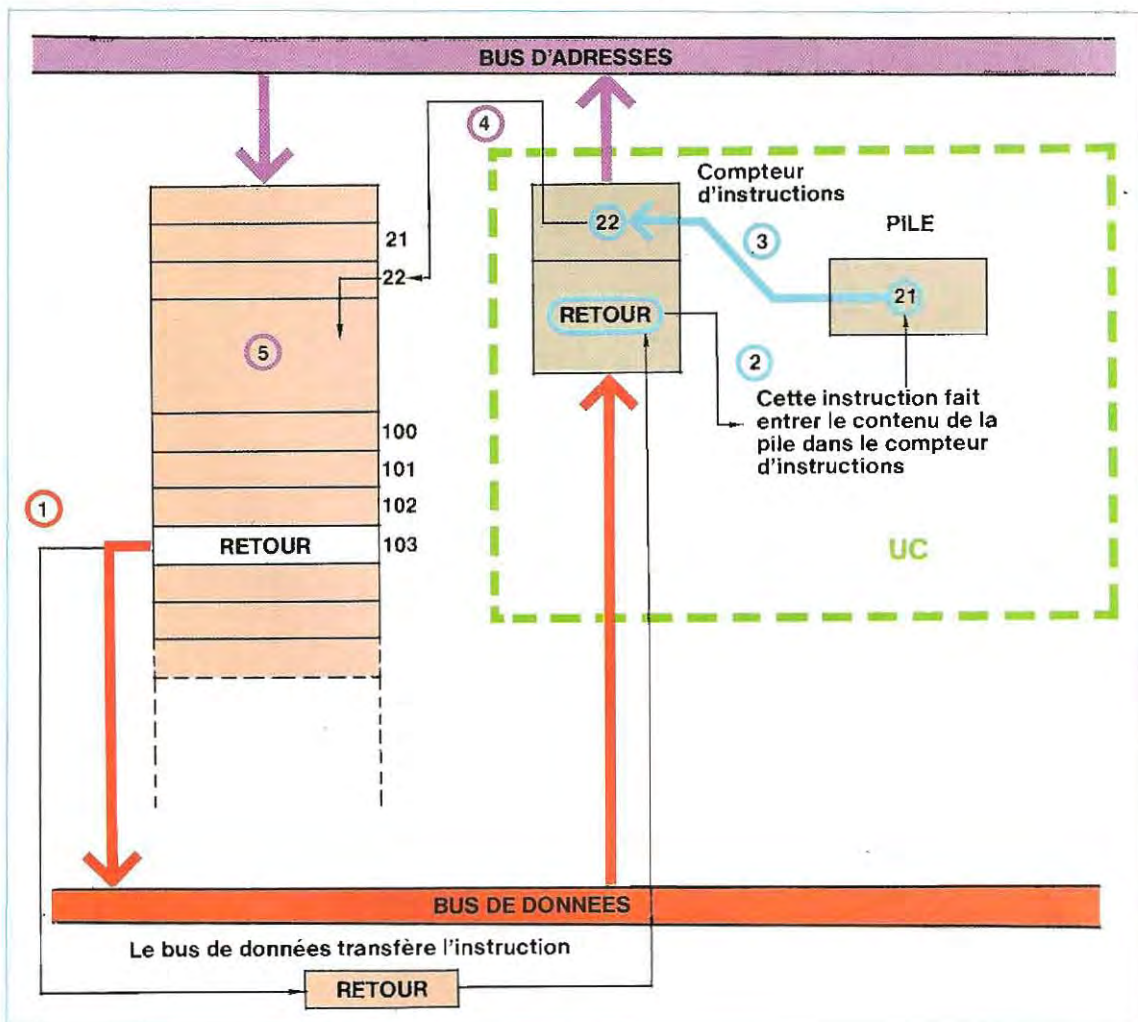
Le programme repart alors sur l'instruction qui suit le saut. Après le saut à 100 (voir schéma sur cette page), l'unité centrale exécute de manière séquentielle les instructions contenues en 101, 102 et 103. La mémoire 103 contient l'instruction Retour. Dès qu'elle identifie ce code, l'UC prélève le contenu de la file d'attente (21 dans cet exemple), l'incrémente et l'inscrit en compteur d'instructions (voir

schéma ci-dessous : ①, ②, ③).

Dès ce moment, le programme reprend son cheminement initial (instructions 22 et suivantes : ④, ⑤).

Cette technique de sauvegarde de l'adresse de retour peut être utilisée pour des sauts successifs. Par exemple, si la mémoire 101 contenait à son tour une instruction de saut, le fonctionnement se ferait suivant le même mécanisme, mais avec deux sauvegardes de l'adresse de retour, la première pour le premier saut (21) et la seconde pour le second saut (101). Le premier retour en arrière se ferait à 102 (101 + 1) et le second à 22 (21 + 1).

Dans ce cas, la pile occupe deux cases de mémoire. On appelle **pointeur de pile** l'adresse de l'emplacement de mémoire contenant le dernier article rangé dans une pile. Dans les systèmes les plus généralement utilisés, la pile peut être réalisée sur la mémoire



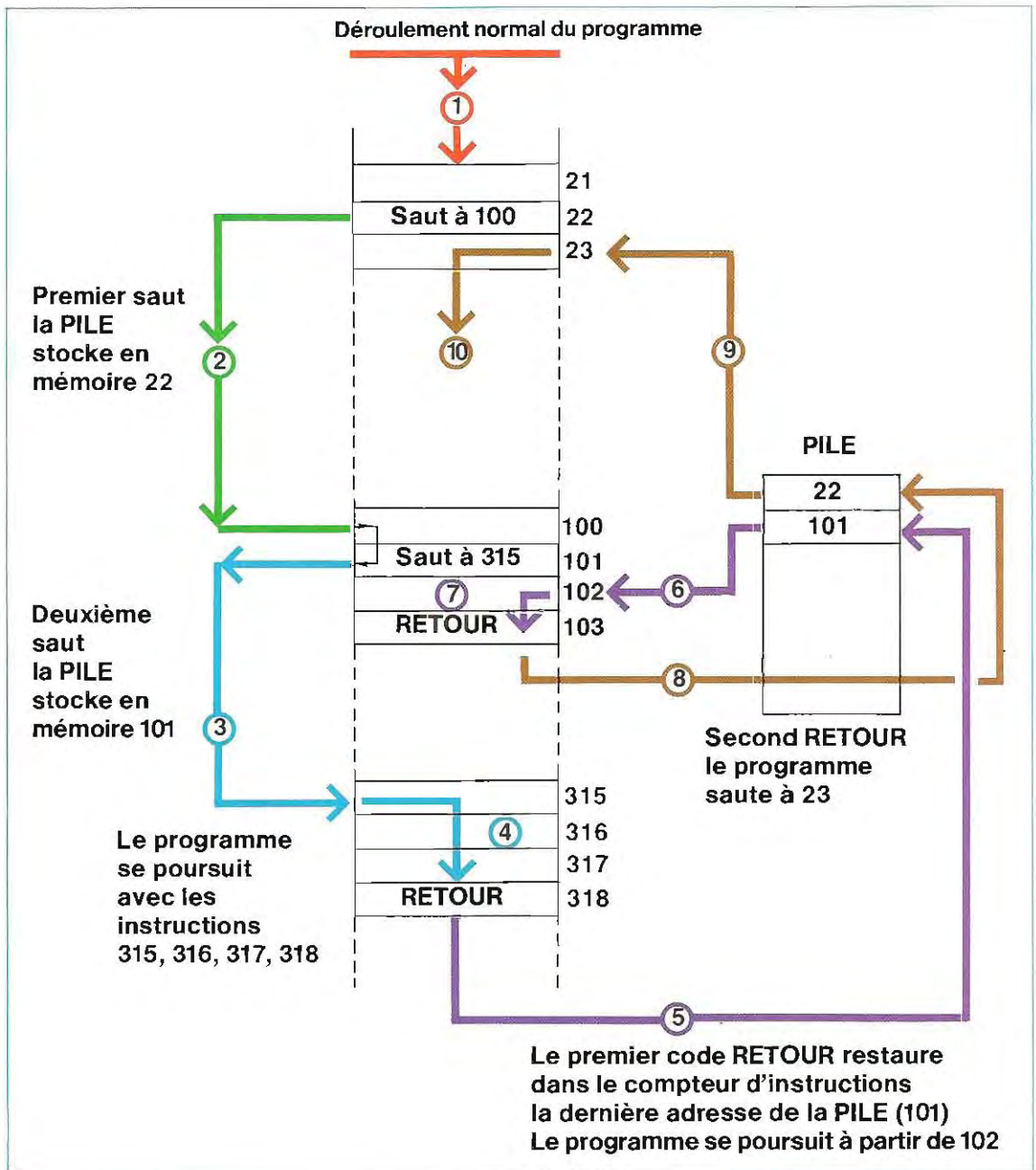
toute entière et ne connaît donc pas d'autres limites que celles de la mémoire elle-même. Le schéma de cette page illustre la logique de fonctionnement avec deux sauts successifs. La séquence des opérations est numérotée de 1 à 10.

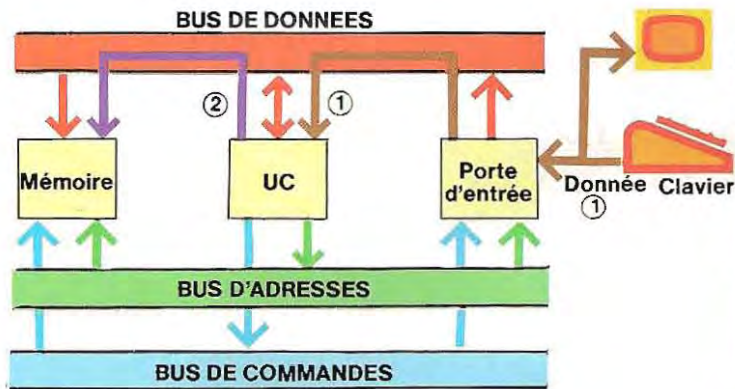
### Gestion des interruptions

Le déroulement normal d'un programme peut être momentanément arrêté par l'apparition d'un signal d'interruption (Interrupt).

Ce signal arrête momentanément les opérations en cours pour permettre le déroulement d'une autre fonction, et à la fin de celle-ci, le retour au point où le programme a été interrompu.

Dans tous les systèmes, les périphériques ont des niveaux de priorité différents : normalement, l'élément prioritaire est le clavier, moyen d'entrée des commandes ; les autres (écran, imprimante) peuvent fonctionner si le clavier n'exerce pas sa priorité.

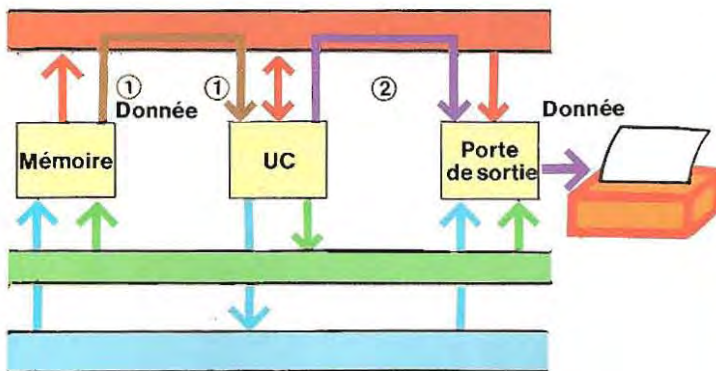




### LECTURE D'UNE DONNEE AU CLAVIER

1 - La donnée passe la porte d'entrée, et se place dans l'UC.

2 - De l'UC, elle vient se localiser en mémoire.



### ECRITURE D'UNE DONNEE PAR L'IMPRIMANTE

1 - La donnée est prélevée en mémoire et déposée dans l'UC.

2 - De l'UC elle est envoyée à la porte de sortie et de là à l'imprimante

Cette logique de fonctionnement ne peut être gérée que par les signaux d'interruption.

Les interruptions de programme sont nécessaires pour laisser place aux demandes de service provenant de l'extérieur et non prévues dans le programme d'application. Ainsi en est-il de l'opération qui consiste à interrompre une impression (par exemple si les données ou la mise en colonnes sont erronées). Le programme d'impression lui consacre toutes les ressources du système ; l'unité centrale prélève les données en mémoire, leur attribue un code et les envoie à l'imprimante. Dans ces conditions, le clavier, organe de commande, est totalement inactif, le programme d'application ne prévoyant pas d'entrée de données. L'introduction de nouvelles instructions ne peut se faire qu'à la fin du programme qui se déroule, une fois l'unité centrale libérée de la fonction d'impression.

En réalité, il suffit d'entrer au clavier un code d'interruption : l'UC arrête le déroulement du

programme d'application et identifie le périphérique « appelant » avant de lui donner la main. L'opérateur peut alors communiquer avec le système et entrer l'ordre d'interruption de l'imprimante.

### Lecture des données

Les périphériques d'un système se classent en deux catégories principales :

- **périphériques d'entrée** : ce sont les éléments qui permettent d'introduire les données dans la machine. Pour les ordinateurs individuels et les micro-ordinateurs, cette catégorie ne comprend généralement qu'un clavier.
- **périphériques de sortie** : ce sont les éléments permettant à la machine de communiquer avec l'opérateur. Il s'agit essentiellement de l'écran de visualisation et de l'imprimante.

Les opérations d'E/S (entrée/sortie) sont illustrées par le schéma de cette page.

## Des mémoires qui n'oublent rien

Même dans les ordinateurs les plus puissants, l'unité centrale de traitement ne peut effectuer qu'une opération à la fois. Après avoir exécuté un calcul, elle stocke le résultat en mémoire, puis passe au suivant, qu'elle stocke ailleurs, etc. L'ordinateur traite ainsi des milliers d'opérations dans une fraction de seconde. Quand il a besoin d'un résultat, il faut qu'il puisse y avoir accès immédiatement. La structure de la mémoire doit donc autoriser l'introduction et l'extraction à volonté de toutes les données, sans avoir à parcourir tout le circuit comme on le ferait avec les pages d'un carnet.

Ce type de mémoire à accès sélectif, ou direct, s'appelle mémoire vive, ou RAM (de l'anglais Random Access Memory). Elle permet d'adresser et de classer toute information reçue ou émise.

Nous avons vu que l'ordinateur travaille sur les données après les avoir transformées en binaire. Une fois codée, la donnée se présente sous la forme d'une chaîne de chiffres binaires adressés, un par un, en mémoire (adressage en série).

On peut aussi présenter à la mémoire tous les chiffres binaires en même temps, sur des conducteurs parallèles, correspondant chacun à l'une des puissances de la base 2 (adressage en parallèle).

La mémoire est constituée de rangées et de colonnes de transistors, alignés ou disposés sur une unique pastille, ou puce (chip), de silicium. Cette pastille doit également être pourvue des circuits nécessaires pour commander, sélectionner, envoyer, rechercher et identifier les informations qui entrent dans la mémoire ou en sortent.

Il existe deux types de mémoires vives à semi-conducteurs : les mémoires dynamiques et les mémoires statiques.

Une mémoire dynamique ne conserve son contenu qu'un certain temps, il faut donc la « rafraîchir » régulièrement à l'aide d'un signal d'horloge, ce qui exige des circuits complémentaires. Chaque cellule de mémoire (ou unité de stockage sur la micropastille) est cependant plus petite que dans le cas d'une

mémoire statique où une cellule exige au moins deux transistors pour stocker un bit. Si la mémoire dynamique tend à perdre son contenu, c'est que les chiffres 0 ou 1 y sont emmagasinés sous forme d'une charge électrique (égale à 0, ou différente de 0). Le transistor se comporte donc comme un simple condensateur et perd graduellement sa charge sous l'effet du phénomène de dispersion.

Dans la mémoire statique, au contraire, le signal est enregistré par un dispositif à bascule (flip-flop, ou plus exactement bascule bistable). Dans la pratique, c'est un interrupteur électronique ne pouvant être que ouvert (ON) ou fermé (OFF), ce qui donne la possibilité de représenter les deux chiffres binaires 0 et 1 sans avoir à emmagasiner de charges électriques.

Le transistor le plus couramment utilisé pour les mémoires est de type MOSFET (sigle anglais correspondant à transistor en métal oxyde semi-conducteur à effet de champ).

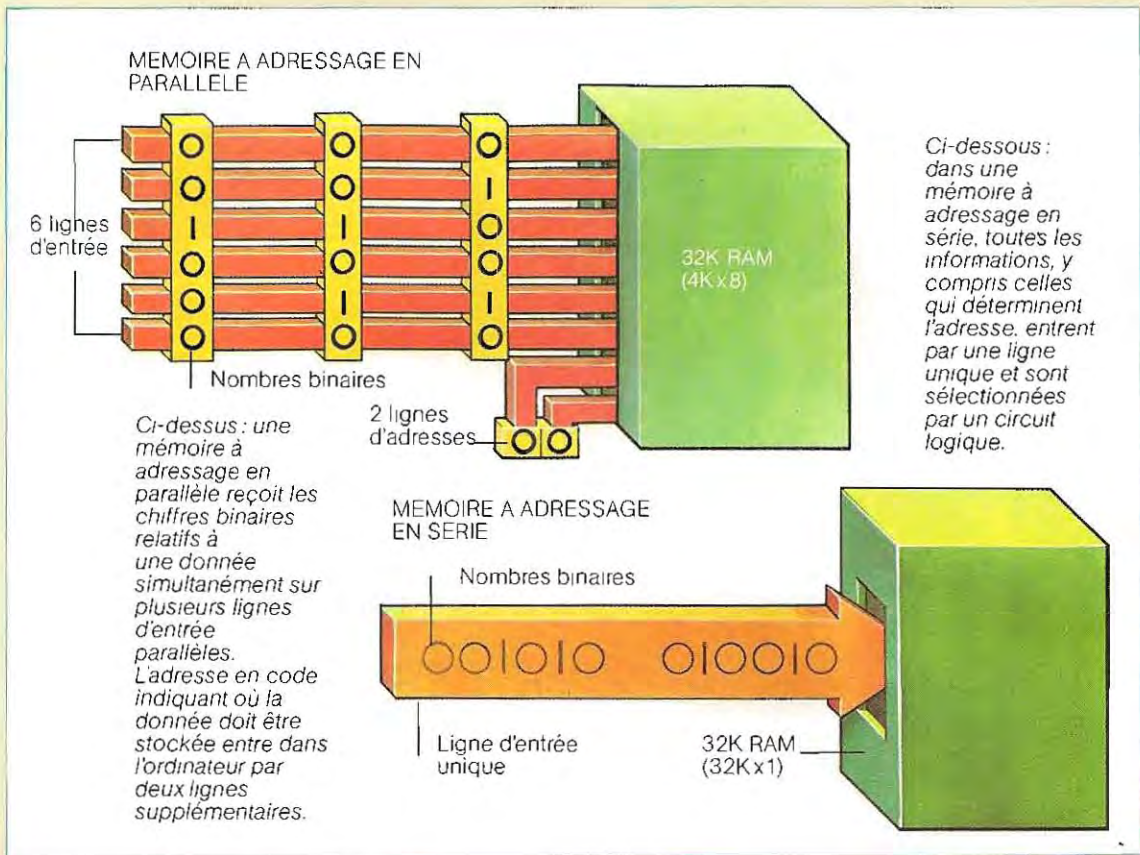
Des milliers de transistors de ce type peuvent être imprimés par photogravure sur un centimètre carré de silicium semi-conducteur. Chaque transistor est composé d'un microscopique cristal de matériau semi-conducteur, traversé par un courant électrique circulant entre deux électrodes, que l'on appelle émetteur et collecteur. Dans la masse du semi-conducteur, une troisième électrode, assurant la régulation du courant, s'appelle porte en raison de sa fonction.

Le transistor à effet de champ faisant partie d'une mémoire est employé comme un interrupteur qu'on peut fermer en envoyant un signal à sa porte.

En position de fermeture du circuit (ouverture de la porte), le courant circule de l'émetteur au collecteur en traversant le dispositif et va charger un condensateur relié au transistor. Ce condensateur prend ainsi un état de charge correspondant au chiffre 1 du code binaire, ou un état de non-charge (chiffre 0). Cette technique permet de stocker les informations en binaire.

L'ensemble du MOSFET et du condensateur constitue la cellule de mémoire. La porte de chaque cellule est connectée à une ligne de sélection que l'on active par une instruction appropriée lorsqu'on veut qu'un chiffre soit chargé dans le condensateur de la cellule.





Dans une mémoire à adressage en série, chaque chiffre binaire s'accompagne d'une adresse à sélection de colonne à deux bits qui définit dans quelle colonne l'information doit entrer. Un circuit logique assure l'envoi des chiffres à leur adresse exacte.

Dans une mémoire à adressage en parallèle, un mot se présente sur plusieurs lignes d'entrée. Les signaux circulant sur les deux lignes d'adresses stockent le mot dans la zone voulue. Cette méthode permet de stocker un grand nombre d'informations sur une seule pastille de petite dimension.

Le recours aux résistances proprement dites est restreint dans les mémoires RAM statiques à MOSFET. On les remplace en effet par un transistor dont la porte est connectée au collecteur et qui constitue une résistance non linéaire de surface réduite.

Pourtant, la société Mostek a réhabilité l'usage des résistances, en choisissant un silicium polymorphe à implantation ionique qui, pense-t-on, n'occuperait guère plus d'espace que le système actuel mais également

avec une moindre consommation d'énergie.

Les mémoires vives ne sont pas toutes construites sur la technologie MOS (métal oxyde semi-conducteur). On utilise aussi des mémoires dites bipolaires, que l'on tend toutefois à réaliser à l'aide de couples de bascules, tout comme les mémoires statiques MOS, d'où une augmentation de la surface nécessaire pour chaque cellule. Les mémoires bipolaires peuvent être beaucoup plus rapides que les mémoires dynamiques, même si celles-ci restent plus économiques.

Il existe une variante de la mémoire vive statique MOS : la mémoire MOS complémentaire, ou CMOS, dans laquelle les résistances de charge du couple croisé de transistors constituant un flip-flop sont remplacées par des transistors polarisés en direction opposée à celle des véritables transistors de commutation. La résistance présentée par les deux transistors de charge est infinie en théorie, le courant ne passe donc pas et il n'y a pas dissipation d'énergie. Il en résulte que les

mémoires vives statiques CMOS consomment encore moins d'énergie, ce qui explique leur utilisation dans le domaine spatial, malgré un fonctionnement plus lent. Si, en théorie, aucun courant ne doit passer à travers les transistors des mémoires CMOS, on constate dans la pratique un flux de courant au cours de la brève période de commutation de la bascule passant d'un état à l'autre. Cet inconvénient, dû aux effets des capacités parasites, a été récemment éliminé par la société RCA, qui a mis au point une méthode de réalisation de mémoire CMOS sur substrat de saphir. Cette technique multiplie par quatre la vitesse de fonctionnement. Mais, le saphir étant beaucoup plus coûteux que le silicium, les recherches se poursuivent, sans que l'avenir de ce procédé soit assuré.

Toutes les mémoires vives à semi-conducteurs sont volatiles. On a étudié de nombreuses méthodes pour remédier à ce défaut. En cas de coupure de l'alimentation électrique de la mémoire, on peut effectuer le transfert immédiat des informations les plus importantes dans une mémoire magnétique à disque, ou à ruban.

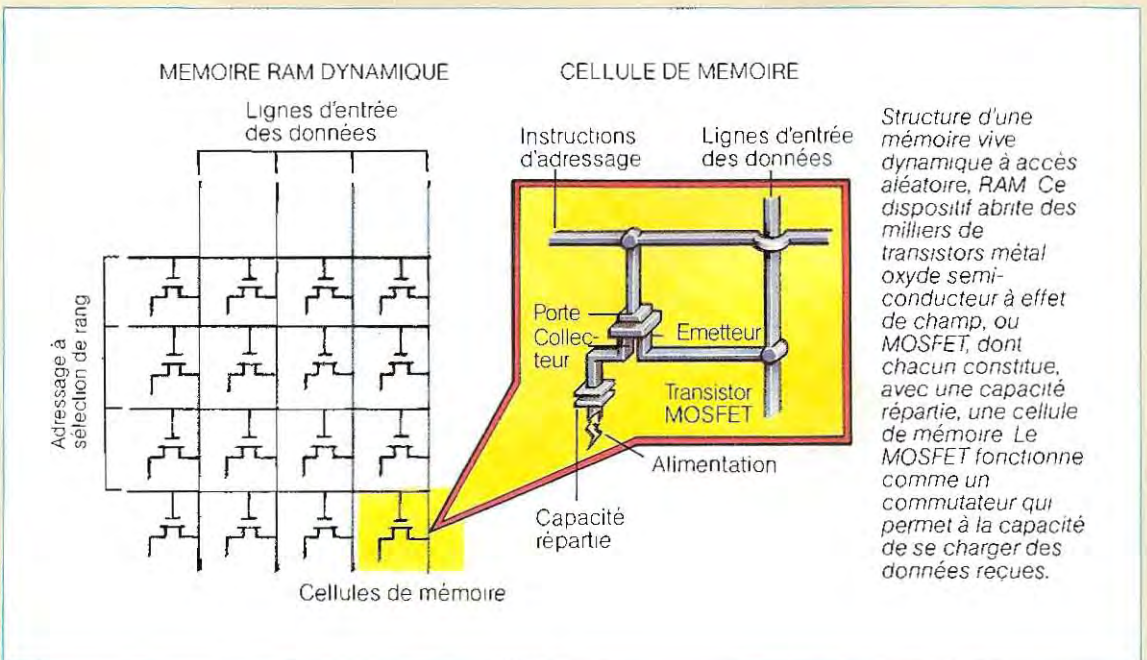
On emploie aussi des mémoires à semi-conducteur non volatiles, pour stocker des programmes à durée indéfinie, par exemple celui d'un jeu vidéo, ceux qui commandent

les opérations internes d'un ordinateur, ou qui indiquent à la machine les phases d'une procédure mathématique.

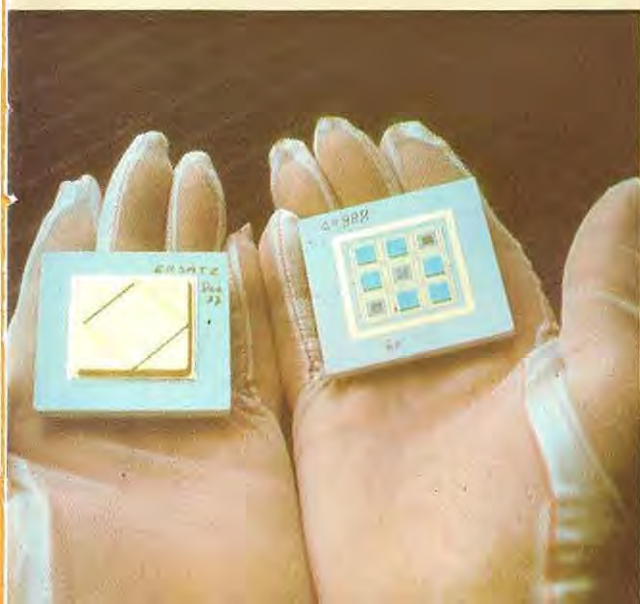
Ces mémoires ne peuvent fournir qu'un ensemble figé d'informations, lues et utilisées continuellement par l'ordinateur. Ce sont les mémoires mortes ROM (Read Only Memory). Elles sont formées d'une simple matrice à diodes et leur programmation est effectuée une fois pour toutes en faisant fondre certaines diodes par application d'un courant de haute tension. La présence ou l'absence de connexion entre les lignes peut servir à indiquer un état logique 1, ou 0. La simplicité d'une mémoire morte explique son coût très faible et la haute densité d'informations que l'on peut y stocker.

Une fois programmées, les mémoires mortes ne peuvent plus être modifiées. Cependant, il existe aujourd'hui d'autres familles de mémoires mortes, des composants que l'on peut effacer en exposant la matrice de diodes à un faisceau ultraviolet à travers une petite fenêtre pratiquée au sommet du dispositif, puis reprogrammer. Ces mémoires ont été dénommées EPROM (Electrically Programmable Read Only Memory), par opposition aux mémoires mortes programmables (PROM).

Les mémoires EPROM ont à présent atteint



Ricky Blakely



Archives Curcio/IBM

**Deux phases du montage des pastilles (chips) de mémoire sur support céramique.**

**A droite : les plaquettes de support.**

**A gauche : les plaquettes isolées et recouvertes.**

**A ce stade, le module est prêt à être mis en place.**

une capacité de 64 Kbits (kilobits), les travaux de réalisation des versions à 128 et 256 Kbits sont déjà très avancés.

Dans une autre variante de la mémoire morte, dénommée EAROM (Electrically Alterable Read Only Memory), il n'est pas nécessaire d'effacer toute la mémoire pour en reprogrammer une partie puisque la simple application de tensions plus élevées que la normale provoque la reprogrammation des cellules individuelles.

Il existe aussi d'autres systèmes prometteurs pour stocker des données numériques, mais ils ne sont pas encore apparus en grande quantité sur le marché. Les plus intéressants sont les mémoires à bulles magnétiques et ceux qui utilisent les dispositifs à couplage de charge (CCD, Charge Coupled Device).

Dans le cas des bulles magnétiques, les informations sont stockées en utilisant la propriété de certains cristaux qui, sous l'action d'un champ magnétique, emmagasinent ce que l'on pourrait considérer comme des bulles dotées d'une polarité magnétique. Le sens

de cette polarité détermine si c'est un 1 ou un 0 qui se trouve en mémoire. La signification 1 ou 0 peut aussi être donnée par la présence ou l'absence de bulle. Ces dispositifs permettent de stocker un très grand nombre de données (20 000 bits sur 5 mm<sup>2</sup>).

Contrairement aux disques magnétiques, le lecteur, support des mémoires à bulles, étant fixe, ne peut déchiffrer une information que lorsque celle-ci défile devant lui sous l'effet du mouvement des bulles à l'intérieur du support. C'est le déplacement du champ magnétique qui provoque le mouvement des bulles. Tout cela rallonge le temps d'accès, mais, comme dans tous les dispositifs magnétiques, les bulles ont l'avantage de ne pas être volatiles; en outre, la densité de stockage autorisée par ce procédé est à peu près dix fois supérieure à celle d'une mémoire à semi-conducteurs.

De telles mémoires sont désormais largement utilisées par des entreprises comme IBM, Texas Instrument et des constructeurs japonais, entre autres.

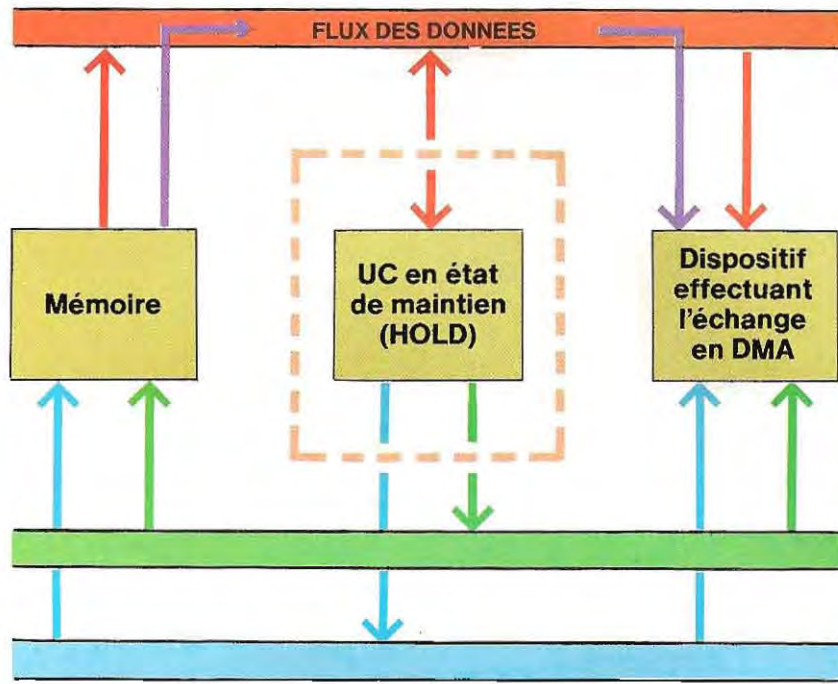
Les dispositifs à couplage de charge appliquent une technique fondée sur les semi-conducteurs qui les rend comparables à d'immenses fichiers tournants. Les informations sont stockées en une longue suite et l'on ne peut y accéder, comme dans le cas de mémoires à bulles, que lorsqu'elles parviennent devant des têtes de lecture spéciales. Physiquement, une mémoire CCD est un transistor à effet de champ de grande longueur, doté d'un certain nombre de sorties (ou électrodes) disséminées entre les deux extrémités, exactement comme un tuyau d'arrosage pour pelouse percé de nombreux trous.

On peut évidemment faire varier la quantité de charge qui s'écoule le long du transistor avec plus de précision que pour un tuyau d'arrosage, et l'on présente à chaque électrode un tableau complet de la série de signaux qui parviennent à l'entrée du dispositif.

Pour que les informations progressent le long du dispositif à couplage de charge, il faut fournir régulièrement un signal d'horloge, et le temps d'accès à chaque bit est plus long qu'avec d'autres dispositifs comparables.

On pense que ce système devrait être peu coûteux, mais il n'est pas sûr qu'il convienne à un emploi généralisé.

## FLUX DES DONNEES DANS L'ACCES DIRECT A LA MEMOIRE



En lecture, par le clavier par exemple, la donnée est transférée d'abord à la porte d'entrée puis à l'unité centrale et enfin à la mémoire. Au contraire, en sortie, la donnée est prélevée en mémoire, transférée à l'unité centrale, puis envoyée à la porte de sortie.

Cette gestion des opérations d'E/S, qui implique un passage par l'unité centrale, ne peut se faire à très grande vitesse et s'applique donc au transfert des données vers les périphériques lents (le clavier et la plupart des imprimantes). Pour obtenir des échanges de données plus rapides on utilise plutôt la technique de l'accès direct à la mémoire (DMA, Direct Memory Access) qui consiste en un échange direct entre la mémoire et le dispositif qui reçoit ou transmet les données. L'unité centrale n'a pas à intervenir dans cet échange, son activité est donc suspendue jusqu'à la fin de celui-ci.

Cet état d'activité suspendue s'appelle maintien (HOLD).

Le schéma de cette page illustre un échange

en accès direct à la mémoire.

Ce type d'échange est généralement adopté pour des unités à disque, dispositifs ayant à traiter de grandes quantités de données.

Une disquette peut contenir 80 000 à 1 200 000 caractères; les disques rigides pour micro-ordinateurs peuvent dépasser 10 millions de caractères; une économie de temps minime sur chaque caractère se traduit donc au total par un gain appréciable.

En outre, dans les programmes d'application, les données sont continuellement extraites de l'unité à disque, soumises à des calculs puis à nouveau stockées sur ces unités: cet accès continu au disque rend indispensable l'utilisation de la procédure d'accès direct.

Le choix de la procédure est automatique, les programmes se chargeant de la sélection et de l'exécution sans que l'utilisateur ou le programmeur aient à s'en préoccuper.

Cette catégorie de programmes fait partie du système opératoire fourni avec la machine.

## Les mémoires

Les mémoires sont les dispositifs dans lesquels on stocke les programmes et les données utilisés ou produits. Il s'agit donc d'une mise en mémoire provisoire, puisqu'en écrivant le nouveau programme on détruit le précédent; de même, lorsqu'on coupe l'alimentation électrique de la machine, le contenu de la mémoire disparaît (sauf avec certains types particuliers). Le stockage permanent se fait par transfert sur disques (ou sur ruban magnétique).

La zone de mémoire est divisée en quatre blocs fondamentaux:

- 1 / zone des signaux d'interruption
- 2 / zone de système
- 3 / zone des programmes de service
- 4 / zone utilisateur

La zone 1 stocke les instructions qui permettent de gérer les signaux d'interruption de programme. Les zones 2 et 3 sont consacrées au système d'exploitation, ensemble de programmes effectuant les fonctions fondamentales d'impression, lecture au clavier, gestion des disques, etc. La zone 4 est disponible pour les programmes d'application (et les données qui s'y rattachent).

Les ordinateurs individuels et micro-ordinateurs disposent d'un minimum de 16 000 à 64 000 caractères de mémoire, dont en général 28 000 ou 58 000 sont réservés à l'utilisateur, selon la complexité du système d'exploitation. Dans les micro-ordinateurs les plus perfectionnés et les mini-ordinateurs, on dé-

passé 260 000 caractères jusqu'à atteindre 2 000 000 (2 Mo) ou 4 000 000 (4 Mo) pour les machines plus importantes.

Le schéma de cette page énumère les divers types de mémoires.

**Mémoires à lecture et écriture** : ce sont les mémoires vives ou RAM (Random Access Memory). Elles permettent d'effectuer les opérations de lecture et d'écriture et se divisent en :

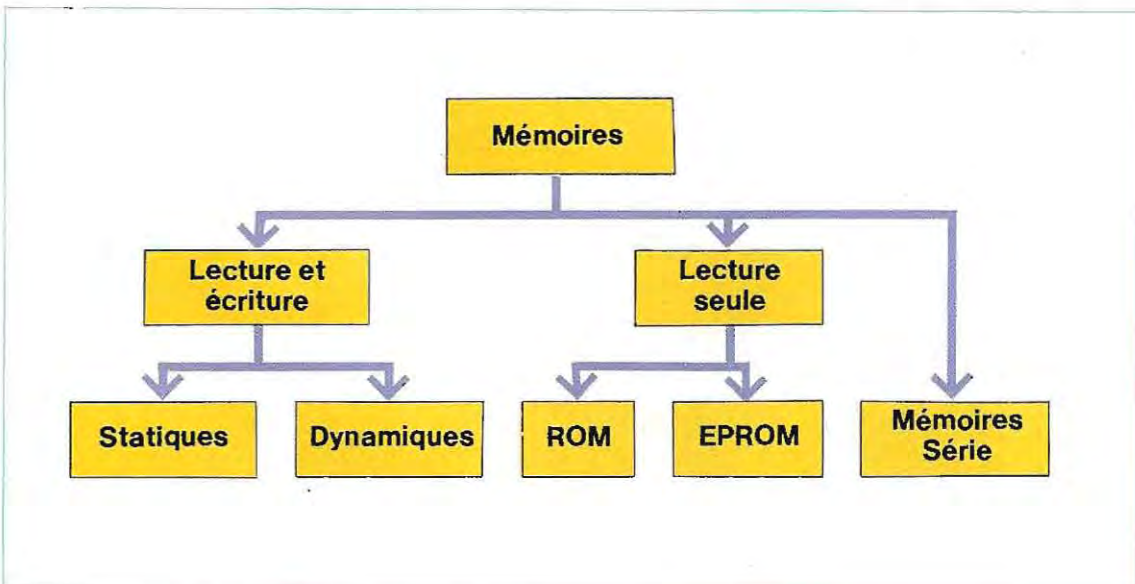
- mémoires dynamiques
- mémoires statiques

Le type dynamique exige un entretien, un rafraîchissement (Refreshing en angl.) régulier des données (quelques centaines de fois par seconde) pour en éviter l'effacement. Le type statique ne nécessite aucune opération d'entretien de la mémoire, une fois la donnée écrite : celle-ci peut être extraite même au bout d'un certain temps.

**Mémoires à lecture seule** : ce sont des mémoires sur lesquelles on ne peut intervenir. Leur emploi est réservé aux programmes qui n'ont pas à être modifiés. Elles se divisent en deux types :

- ROM (mémoire morte)
- EPROM (mémoire morte reprogrammable)

Le type ROM (Read Only Memory) comporte toutes les données nécessaires dès sa construction ; il n'existe donc aucun moyen de le modifier. Le type EPROM (Electrically Programmable ROM) peut être effacé par exposition aux rayons ultraviolets puis reprogrammé.





### Appareil portatif de programmation de mémoire EPROM.

**Mémoire série :** ce sont des mémoires d'emploi très particulier qui donnent des possibilités très grandes mais exigent des circuits complexes. De ce fait, elles sont peu utilisées.

### Les périphériques

**Ecran de visualisation (moniteur) :** il ressemble beaucoup à un poste de télévision normal, au point que certains micro-ordinateurs économiques utilisent simplement un appareil de télévision classique (avec un modulateur). Toutefois, l'image sera moins bonne.

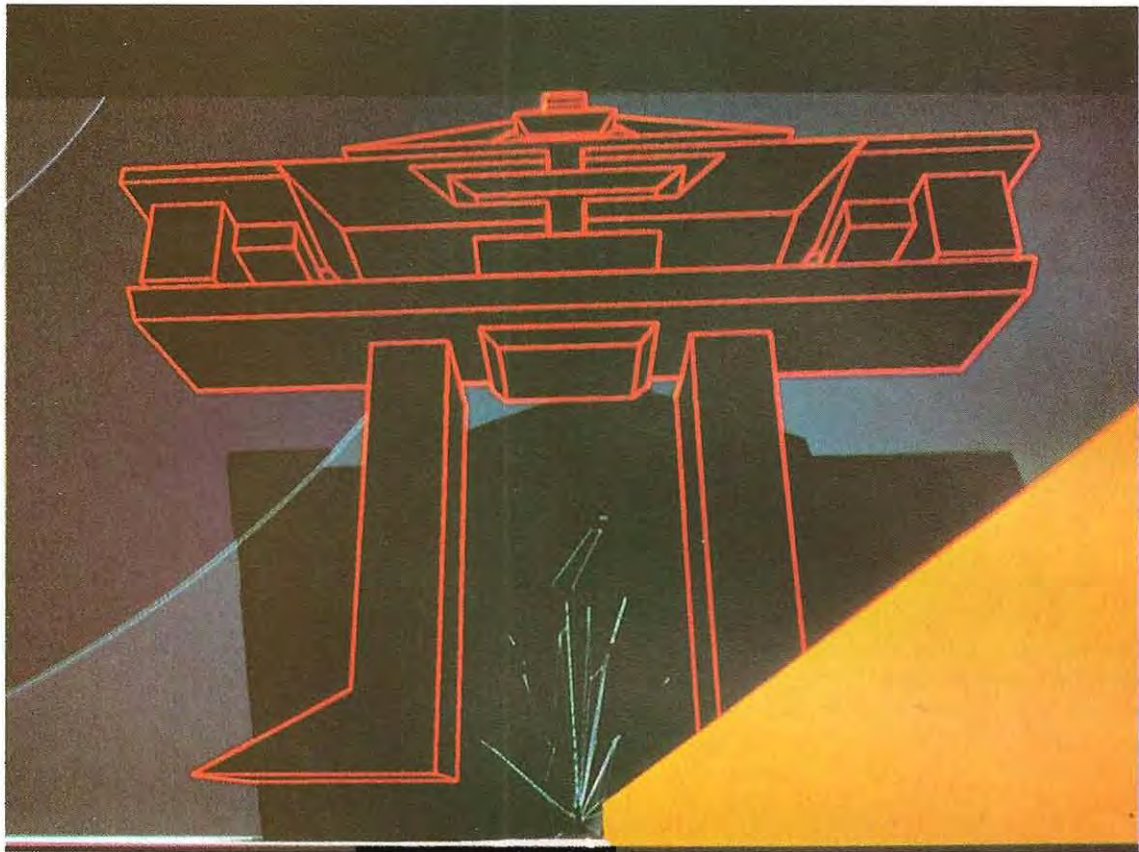
Les moniteurs peuvent être monochromes (gris, vert, ou ambré), ou polychromes, et se différencient par leur taille. Dans le cas d'un moniteur monochrome, la qualité de l'affichage est déterminée par sa bande passante et par son prix. Il permet d'afficher avec plus ou moins de stabilité et de netteté un nombre variable de caractères alphanumériques (lettres et chiffres) et graphiques (dessins). La finesse d'une image graphique dépend du

nombre de points élémentaires (picture element, ou pixel) qui la constituent. Le nombre de ces points varie selon le micro-ordinateur utilisé et la qualité de la restitution est fonction du moniteur. 256 x 192 (pixels) constitue une bonne résolution, mais il y a mieux... et moins bien !

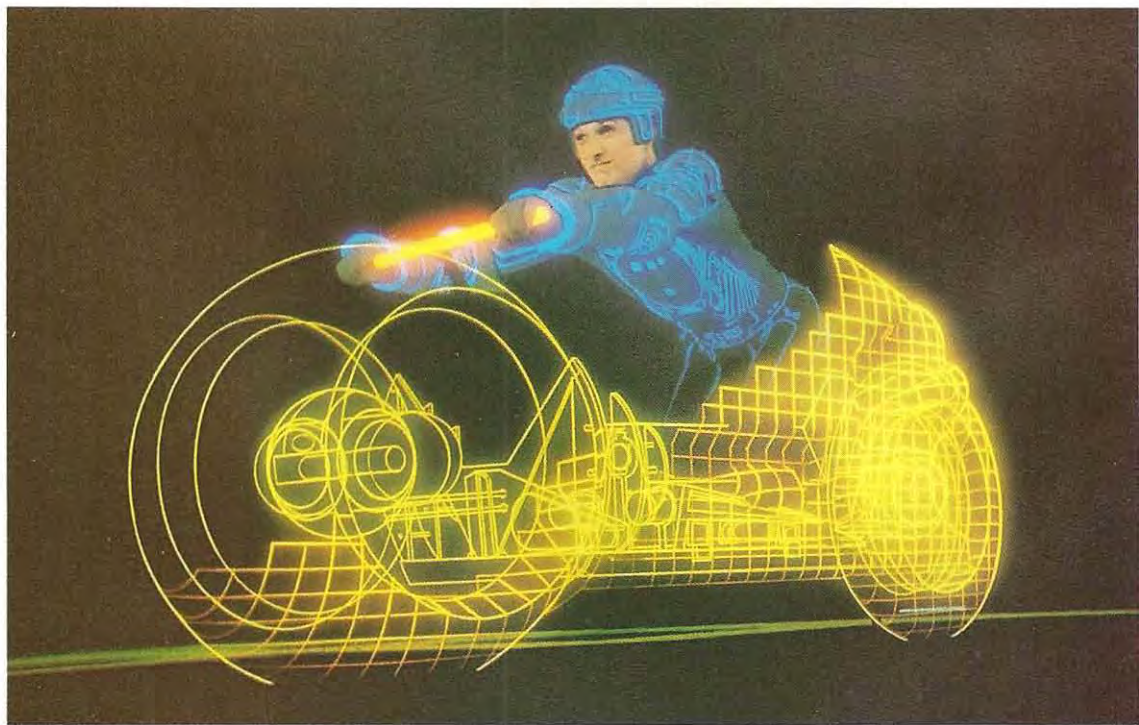
Les écrans couleurs peuvent être classés en deux catégories :

- téléviseur couleurs récent et disposant d'une entrée vidéo dite Péritel. C'est le cas le plus courant et le moins cher.
- vrai moniteur couleurs (ne pouvant pas recevoir les chaînes de télévision) ayant une bande passante beaucoup plus élevée qu'un téléviseur standard et permettant un affichage plus fin. Cependant, son coût le met hors de portée de l'amateur moyen, le réservant à des applications professionnelles.

Le nombre de caractères par ligne et le nombre de lignes par écran varient avec le type de micro-ordinateur utilisé.



1982 © Walt Disney Productions



1982 © Walt Disney Productions

Deux images du film « Tron » illustrant l'emploi du graphisme informatisé dans le cinéma d'animation.

**Le clavier.** Il est généralement divisé en deux sections : la partie alphanumérique qui contient (comme le clavier d'une machine à écrire) les caractères alphanumériques et les chiffres, et la partie numérique ne contenant que des chiffres. Sur certaines machines, chaque touche a deux significations : d'abord le caractère alphanumérique normal et ensuite une fonction déjà programmée (et que l'on obtient grâce à un code particulier). L'utilisation de ce type de clavier permet de gagner beaucoup de temps car il évite, dans l'écriture d'un programme, de rédiger en clair certaines instructions. Il suffit d'appuyer sur une touche pour en écrire une.

**Les imprimantes.** Dans les systèmes à microprocesseur, on utilise surtout les imprimantes :

- à aiguilles
- à marguerite
- par ligne

L'imprimante à aiguilles, ou matricielle : son nom lui est donné par le procédé de formation des caractères. La tête d'impression est composée d'un certain nombre d'aiguilles (sélectionnées par un électro-aimant) qui avancent en fonction de la lettre ou du symbole à imprimer. Le caractère n'est pas tracé d'un trait continu mais formé d'un ensemble de points très rapprochés.

On trouvera dans le tableau ci-dessous quelques-uns des caractères que l'on peut obtenir avec une imprimante à aiguilles.

Les avantages de ces machines sont leur coût modeste, leur vitesse moyenne, leur possibilité d'impression de caractères différents.

L'imprimante à marguerite, ou caractère par caractère : dans ce modèle chaque caractère tout préparé n'a plus qu'à être imprimé sur le papier. Les roues d'impression (marguerites) sont identiques à celles des machines à écrire. Ces imprimantes sont plus coûteuses que les précédentes, beaucoup moins rapides, et ne permettent de changer de caractère qu'en changeant de marguerite. En revanche, la qualité de l'impression est très soignée (qualité courrier).

L'imprimante par ligne : beaucoup plus rapide, elle imprime plus de 600 lignes par minute. Le coût en est très élevé et son emploi ne se justifie que dans les systèmes à mini-ordinateur de plus grande taille (listings). Citons encore les imprimantes à jet d'encre, les imprimantes thermiques, les imprimantes à laser connectables à des micro-ordinateurs.

**Unités à disques.** Les ordinateurs individuels et les micro-ordinateurs en utilisent deux types, à disque dur ou souple.

### CARACTERES D'IMPRIMANTES A AIGUILLES

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n

O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

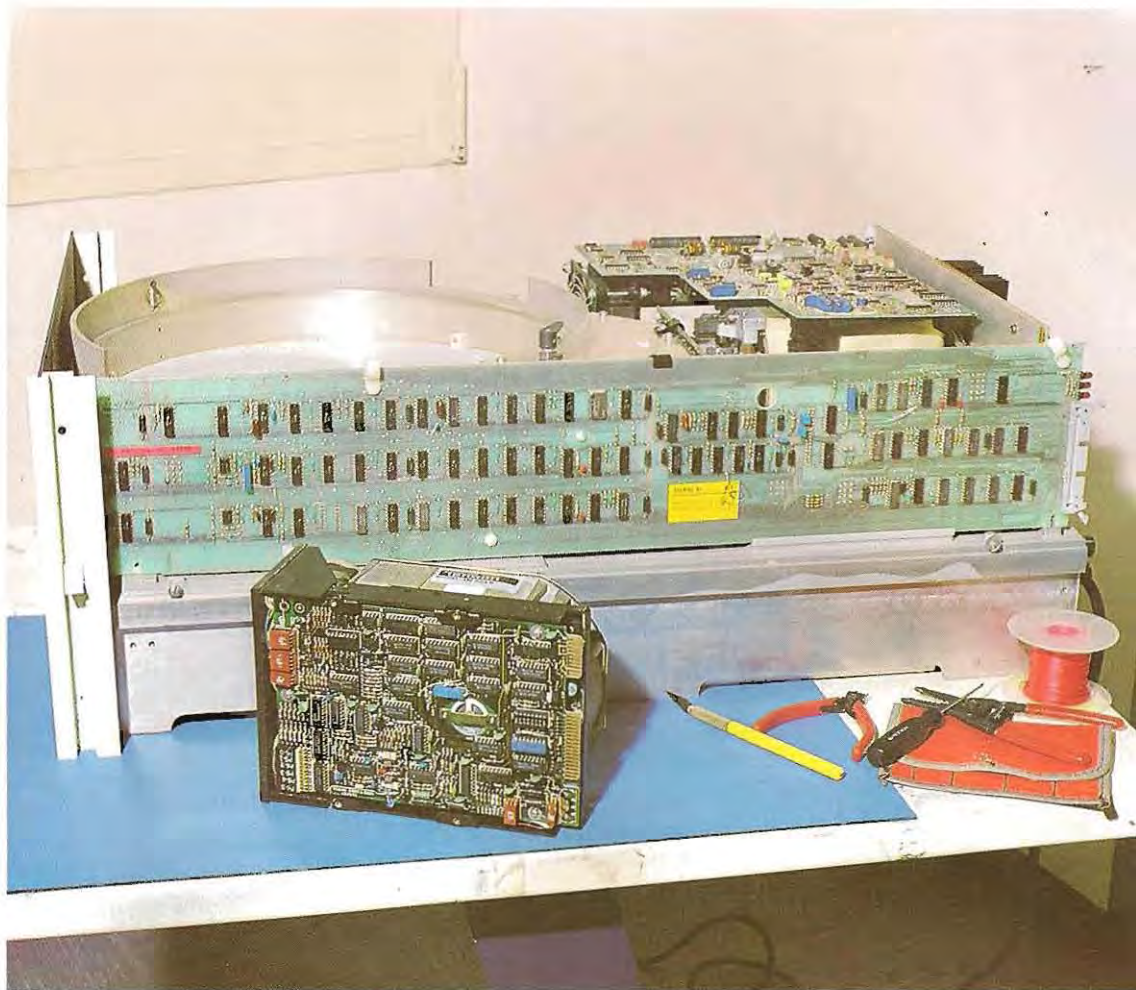
e b c d e f g h i j k l m n

O P Q R S T U V W X Y Z

1 2 3 4 5 6 7 8 9 0

1 2 3 4 5 6 7 8 9 0





Archives Curcio/Buffetti Data

**Comparaison de deux types d'unité à disque. En dépit de la grande différence de taille, la capacité de stockage des deux dispositifs est la même.**

Le disque dur, qui est généralement fixe, a une capacité minimum de 5 ou 10 millions de caractères.

Le disque souple, ou disquette, est amovible, et sa capacité de mémoire varie entre 80 000 et 1 200 000 caractères (selon ses dimensions et son type).

Les disquettes peuvent être :

- à simple ou double face
- à simple ou double densité.

La première caractéristique indique si l'on peut utiliser un côté du disque ou les deux ; la seconde donne une idée de la densité de stockage des données.

C'est évidemment la disquette simple face-simple densité (SF-SD) qui contient le moins de données ; la capacité est nettement plus

grande sur les disquettes double face-double densité (DF-DD).

Les systèmes de plus grandes dimensions (mini-ordinateurs) emploient aussi des disques durs fixes ou amovibles.

Le type fixe a une grande capacité de mémoire ; il est utilisé dans les applications d'une certaine complexité. Le second type (dur amovible) a une capacité de l'ordre de 10 millions d'octets (caractères) et un coût plus limité.

La photo de cette page compare deux unités à disque, ayant toutes deux une capacité de 10 Mootets (10 Méga-octets, 10 Mo, ou 10 millions de caractères) ; la plus grande est à disque dur amovible, l'autre est une unité à disquettes multiples. La photo montre clairement la différence d'encombrement.

## Fonctionnement des mémoires mortes

Les mémoires mortes, ou à lecture seule (ROM), nous le savons, servent à stocker des algorithmes permanents pour commander l'exécution de certaines opérations. Ce logiciel permanent, ou microprogramme, est utilisé pour accomplir toute une gamme de fonctions répétitives dans de nombreuses machines. L'exemple ci-dessous décrit l'emploi d'une mémoire morte pour la commande d'un écran à sept segments, destiné à la présentation d'un nombre sur le visuel d'une calculatrice de poche.

Nous avons déjà appris que les signaux provenant du microprocesseur de la mémoire sont codés en binaire.

Un affichage numérique à sept segments est un dispositif numérique à base 10, et doit avoir comme tel la possibilité de recevoir en entrée dix signaux différents.

Le problème de la conversion est résolu par le

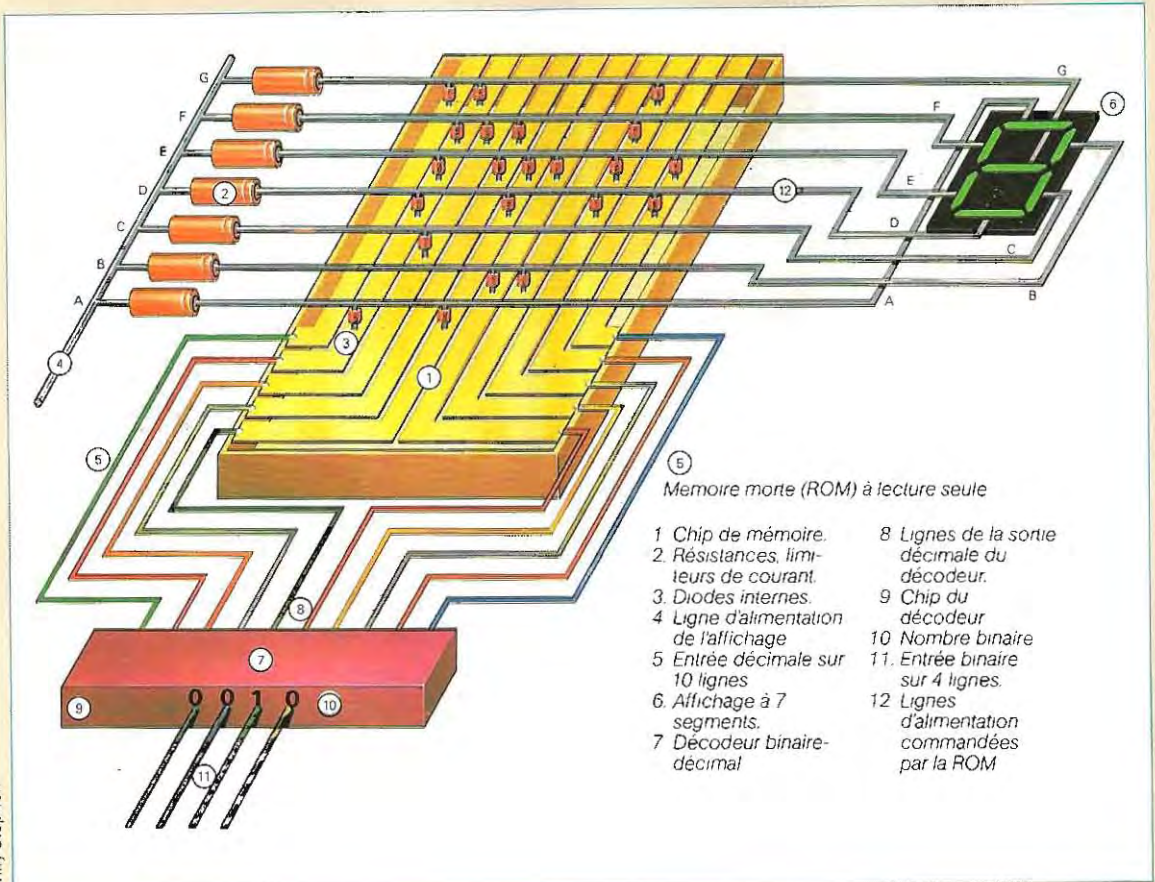
recours à un décodeur binaire-décimal, composé d'un ensemble de portes logiques.

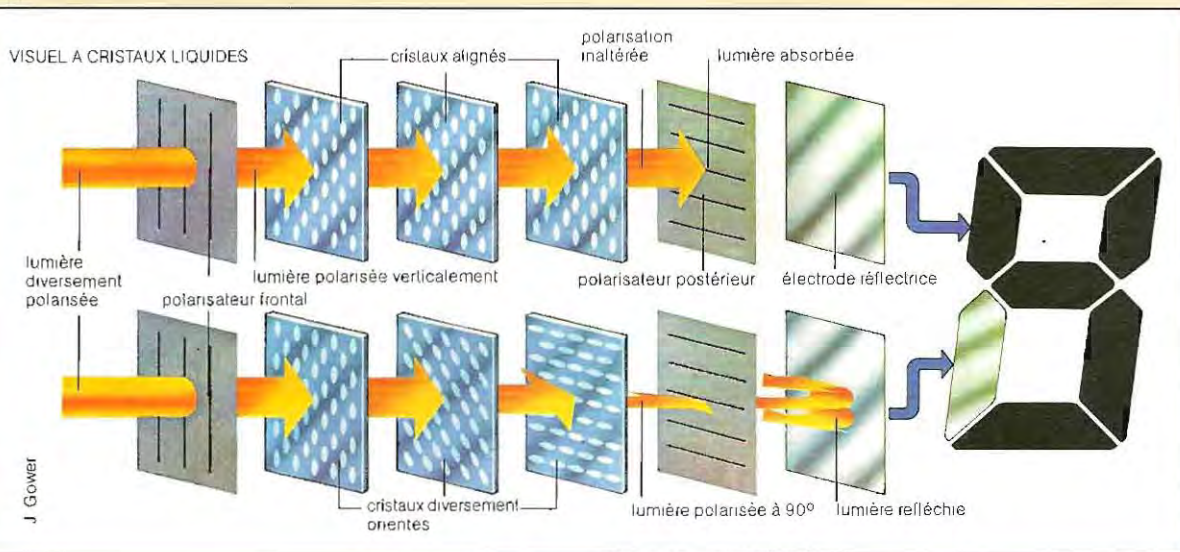
Les quatre lignes d'entrée étudiées dans cet exemple peuvent fournir au décodeur les chiffres binaires qui représentent les nombres de 0 à 9; pour convertir ces chiffres il faut, en revanche, dix lignes de sortie.

En l'absence de signaux binaires en entrée, les dix lignes de sortie sont parcourues en permanence par une tension continue, celle du fonctionnement de l'affichage (en général 5 volts).

Quand, par exemple, le décodeur reçoit le nombre binaire 0000 (0 dans le système décimal), il fournit en sortie un 0, ce qui s'obtient en produisant une absence de tension sur la première ligne de la sortie décimale (celle du 0) et un 1 (présence de tension) sur toutes les autres lignes.

Si le décodeur reçoit le nombre binaire 0001 (1 dans le système décimal) il produit un 0 (absence de tension) sur la deuxième ligne (qui représente le chiffre décimal 1) alors que toutes les autres lignes transportent un 1 (présence de tension).





On peut répéter le même procédé pour tous les autres nombres décimaux.

On utilise cette tension nulle comme indicateur, comme on le fait avec les logiques inversées parce que l'absence de tension dans une ligne signifie qu'elle est reliée à la masse, et que, par conséquent, elle possède la propriété d'annuler la tension sur toute ligne qui entre en contact avec elle. Elle peut donc couper l'alimentation de l'un quelconque des segments du visuel, et l'empêcher de s'allumer. L'exclusion sélective des segments permet d'obtenir sur un visuel à sept segments les chiffres compris entre 0 et 9.

C'est la mémoire ROM qui élimine du visuel les segments appropriés, pour chacune des opérations du décodeur. Par exemple, si on a un 0 sur la première ligne de sortie du décodeur (chiffre décimal 0), il faut déconnecter le segment horizontal au centre du visuel, ce qui fera apparaître le 0.

Une mémoire ROM très simple, capable d'accomplir certaines fonctions nécessaires, se construit par étapes successives. Il faut d'abord composer une table qui indique quels sont les segments du visuel qu'il convient d'allumer, pour chacune des combinaisons réalisées avec les quatre bits susceptibles de se présenter en entrée.

Ensuite, on met en place les connexions sur une matrice à diodes, comme on le voit sur la figure. Si on insère une diode entre la première ligne de sortie du décodeur et la ligne d'alimentation G, chaque fois qu'il se produit une absence de tension sur la première ligne,

l'alimentation du segment central du visuel est interrompue. Les diodes qui relient la ligne d'alimentation de A à F à la ligne 0 du décodeur deviennent inutilisables. Elles sont délibérément fondus, éliminées de la matrice, au cours de l'élaboration du programme de la ROM, grâce à un courant intense sous une haute tension. On peut donc, par cette méthode simple de la matrice à diodes, programmer une mémoire ROM, qui remplira une fonction donnée.

Un visuel numérique est donc commandé par une unité de sortie (interface) qui reçoit de l'ordinateur central la donnée à faire apparaître. Elle la décode et fournit en sortie les impulsions nécessaires pour activer le visuel. L'unité d'affichage à diodes électroluminescentes (LED) peut faire apparaître les chiffres de 0 à 9 en déclenchant les combinaisons appropriées des sept segments qui la composent. Si l'on utilise des cristaux liquides, il faut également sept segments pour former le chiffre 8; en activant les segments appropriés, on peut former les chiffres de 0 à 9.

Cette visualisation se fonde sur les propriétés de cristaux nématiques (du grec *nēmatos* : fil) en suspension dans un liquide emprisonné dans une plaque transparente placée à l'avant, et une plaque réfléchissante placée à l'arrière. Si l'on applique une différence de potentiel entre les segments de la plaque antérieure et ceux de la plaque postérieure, le liquide devient opaque et, grâce à une altération des indices de réflexion, produit un segment visible.

## Les instructions dans l'UC

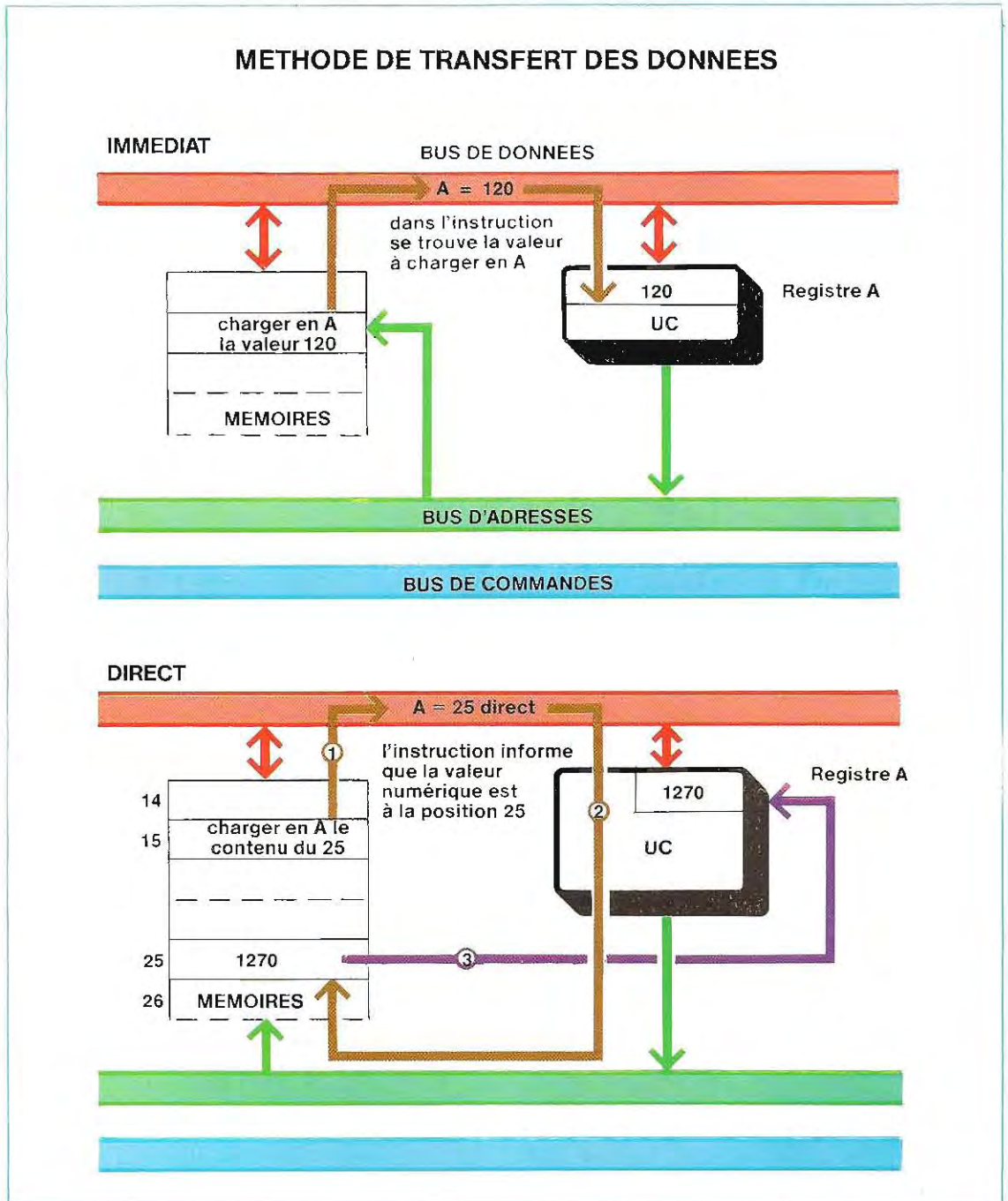
Le fonctionnement d'un système à microprocesseur consiste à exécuter la série d'instructions constituant le programme. Dans l'unité centrale, une instruction est interprétée et convertie en signaux électriques, par l'intermédiaire de microcodes.

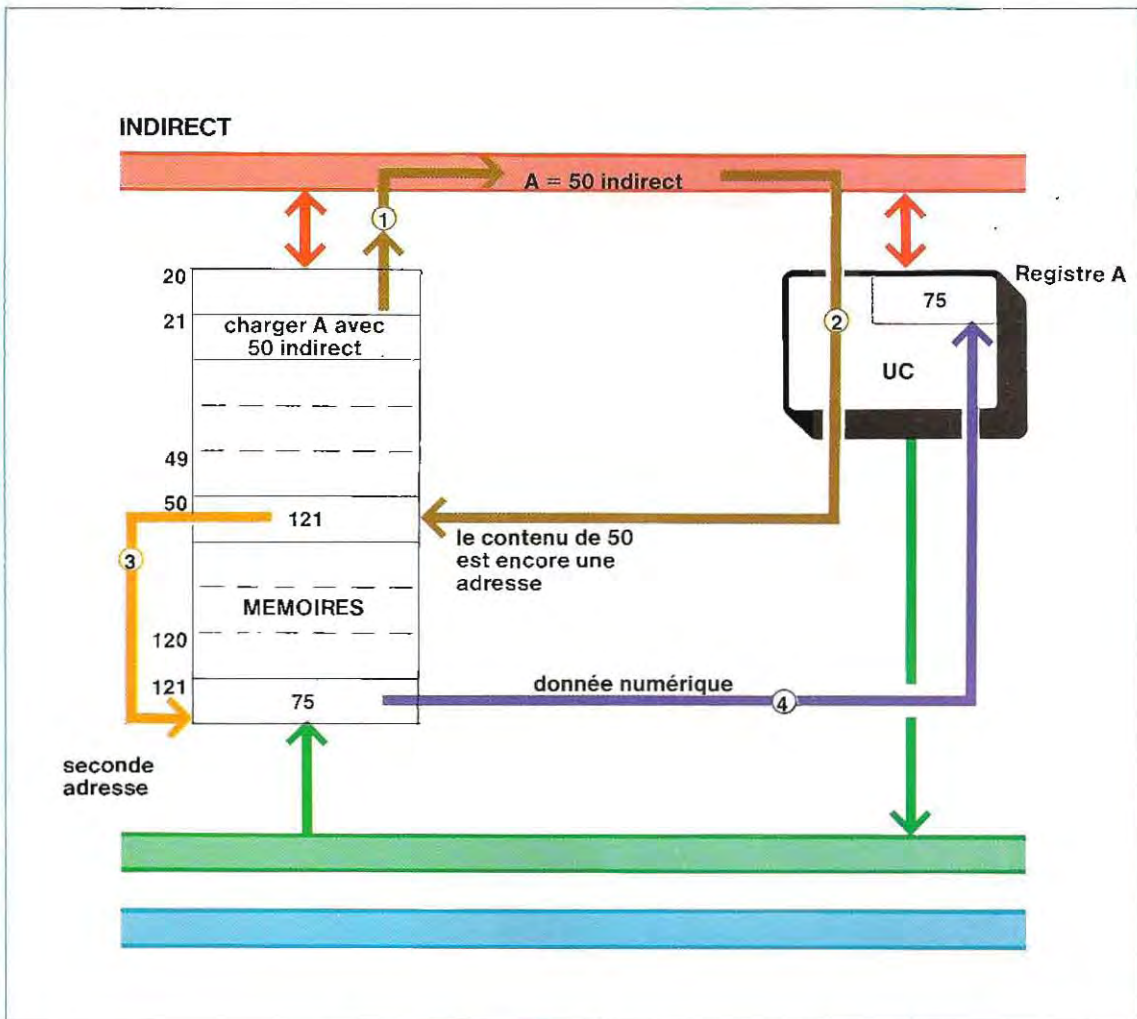
Une unité centrale interprète les cinq types

d'instructions suivants :

- 1 / transfert de données
- 2 / opérations arithmétiques et logiques
- 3 / instructions de saut
- 4 / fonctions d'E/S (Entrée/Sortie)
- 5 / instructions spéciales

Selon le type de l'unité centrale, chacun des





cinq groupes se compose d'une quantité plus ou moins grande d'instructions plus ou moins complexes.

### Instructions de transfert des données

Ces instructions de mouvement des données sont celles qui permettent d'extraire une donnée et de l'adresser en mémoire.

Dans les microprocesseurs ( $\mu P$ ), la transmission se fait toujours entre la mémoire du système et une mémoire particulière, contenue dans l'unité centrale, et appelée registre (voir schémas). Ces instructions sont :

**immédiates** : dans le registre désigné, on introduit une valeur numérique donnée. Par exemple, la valeur 120 dans le registre A ;

**directes** : on charge le registre du contenu d'une mémoire spécifique. Le registre A reçoit, par exemple, la valeur numérique contenue

dans la mémoire 25. A l'inverse de ce qui se passe avec les instructions immédiates, le registre n'est pas chargé d'une donnée spécifique (immédiate) mais de tout le contenu de la mémoire 25 ;

**indirectes** : c'est une extension du type précédent. Le registre est chargé de deux adresses successives : la première produit la seconde, la seconde produit la donnée. Par exemple : chargement du registre A de 50 indirect. L'instruction se déroule de la façon suivante :

- ① - ② extraction du contenu de la mémoire 50 ;
- ③ ce contenu, par exemple 121, est interprété comme une nouvelle adresse ;
- ④ extraction de la donnée contenue dans la mémoire 121.

Dans ces exemples, nous n'avons considéré que des transferts de la mémoire au registre, mais l'inverse est possible.

## Opérations arithmétiques et logiques

Ces opérations sont réunies dans un même groupe. Pour mener à bien une opération, il faut en général fournir les adresses où rechercher les données et l'adresse où déposer le résultat. La séquence logique complète des instructions à fournir comprend trois étapes :

- ① introduire dans le registre A la première donnée (par exemple le contenu de la mémoire 26) ;
- ② exécuter l'opération arithmétique (par ex. la somme) entre le contenu de A (première donnée) et celui de la mémoire concernée (par ex. 41) ; le résultat est en A ;
- ③ transférer le contenu de A (résultat) dans la mémoire désignée (par ex. 53).

## Instructions de saut

Ce groupe d'instructions permet de sauter d'un point à l'autre du programme au moment de son exécution par l'UC. On distingue :

- les instructions de saut proprement dites (**Jump** en anglais),
- les instructions de saut avec retour (**Return** en anglais).

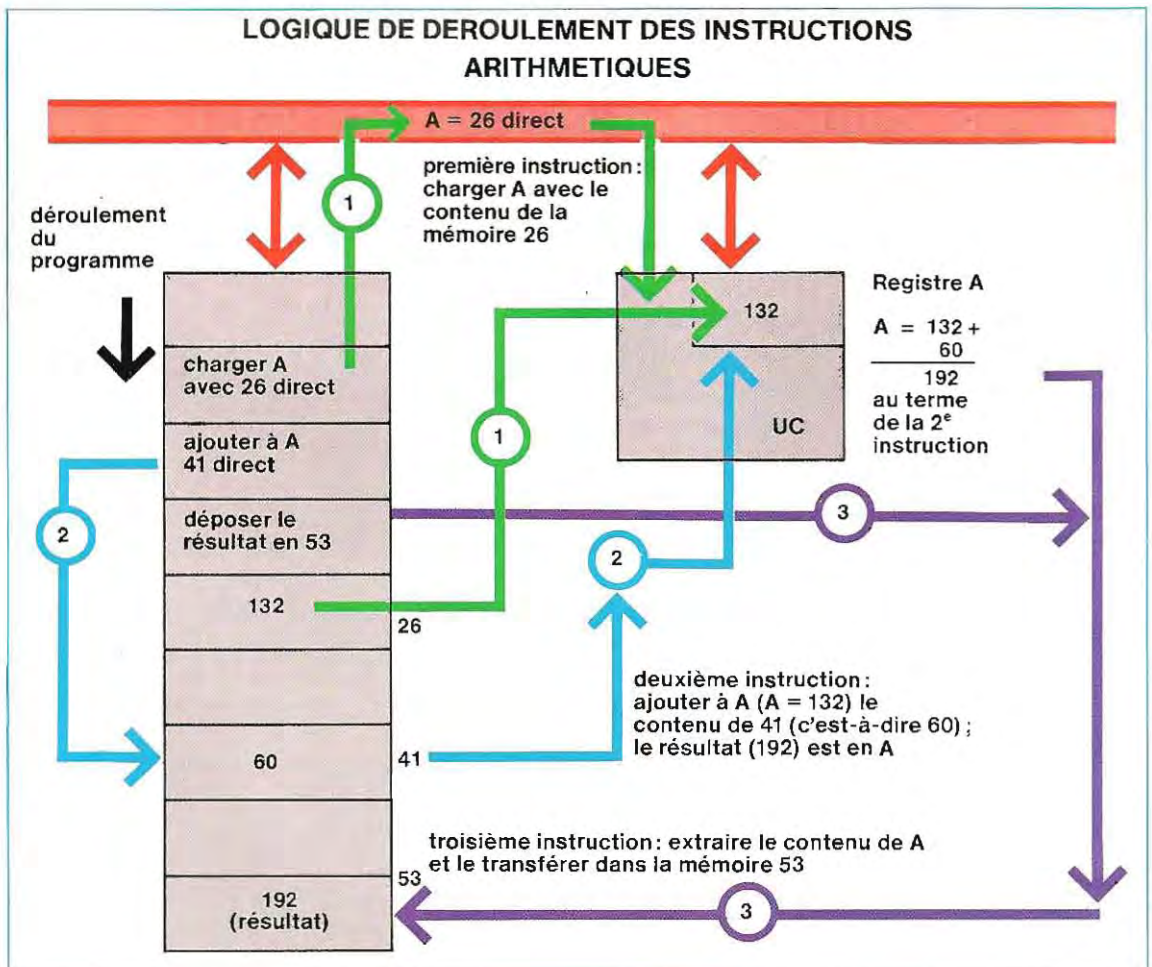
Dans le premier cas (schéma en haut de la p. 149), l'opération se poursuit du point d'arrivée du saut jusqu'à la fin du programme. Dans le second, lorsqu'on arrive à l'instruction Retour, l'opération reprend à l'instruction qui suit celle de saut (schéma en bas de la p. 149).

## Fonctions E/S (Entrée/Sortie)

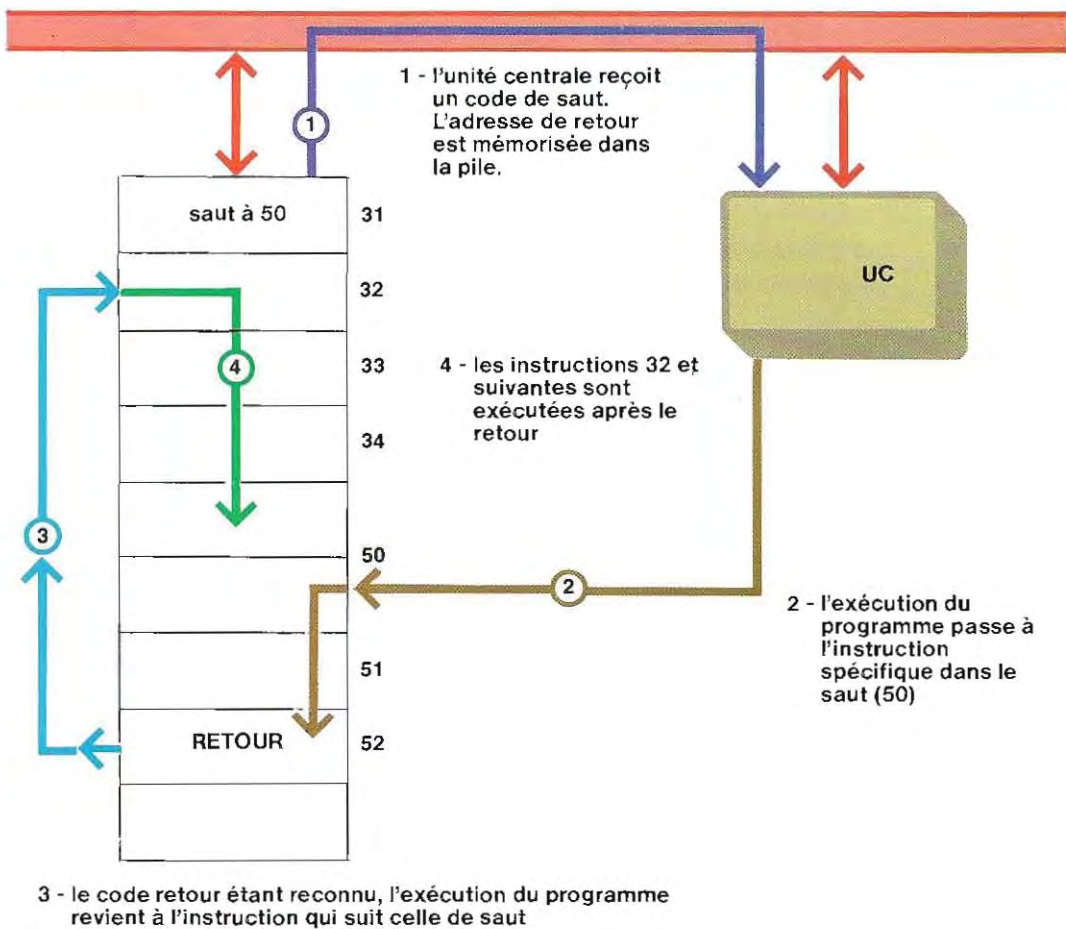
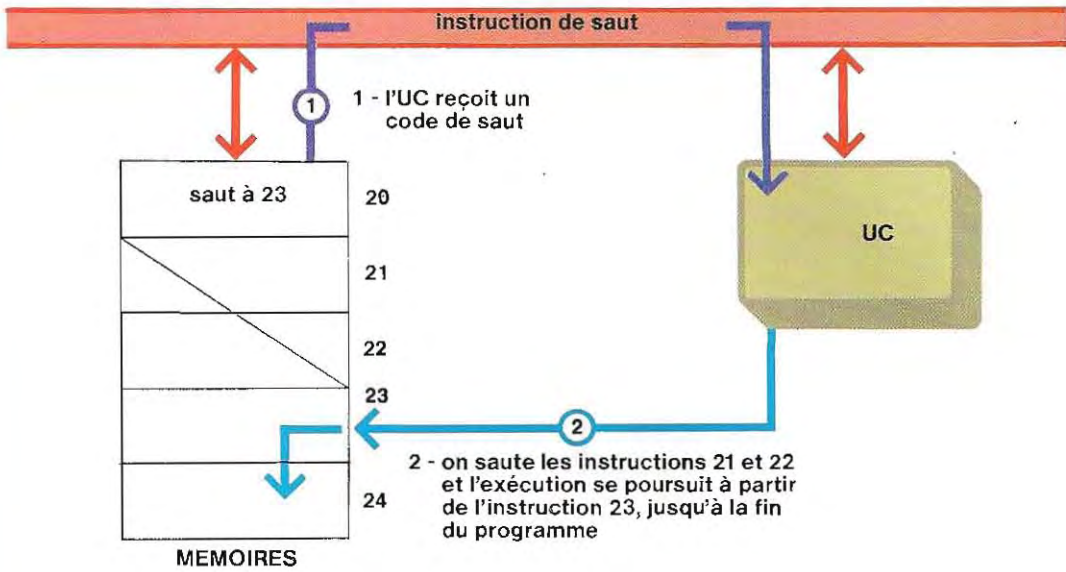
C'est dans ce groupe qu'on trouve toutes les instructions qui permettent de lire ou d'écrire une donnée au moyen d'un périphérique.

## Instructions spéciales

C'est un groupe d'instructions particulières, souvent propres à des unités centrales déterminées. Les principales concernent la fonction d'interruption (anglais Interrupt).



## LOGIQUE DE DEROULEMENT DES INSTRUCTIONS DE SAUT



# Solutions du test 3

1 / Conversion en binaire sous forme chiffrée :

Décimal	Binaire	
7	0111	$(2^2 + 2^1 + 2^0 = 4 + 2 + 1)$
9	1001	$(2^3 + 2^0 = 8 + 1)$
0	0000	0
3	0011	$(2^1 + 2^0 = 2 + 1)$

Conversion en binaire sous forme de caractères (symboles ASCII) :

Décimal	ASCII	
7	37 hexadécimal	= 55 décimal
9	39 hexadécimal	= 57 décimal
0	30 hexadécimal	= 48 décimal
3	33 hexadécimal	= 51 décimal

La première forme (numérique) est utilisée dans les calculs, la seconde dans les opérations d'E/S et de mémorisation.

Le programmeur doit préciser, au moyen d'instructions appropriées, quelle forme il a adoptée pour éviter toute erreur.

2 / La transmission en série consiste à envoyer les données, une à la suite de l'autre (bit par bit). Dans la transmission parallèle, on lance un bloc entier d'informations à la fois (par exemple un caractère complet). Ce dernier mode est certes plus rapide, mais exige des lignes à conducteurs multiples.

3 / La codification du caractère K en ASCII est 4B (hexadécimal) = 75 (décimal) = 1001011 binaire. Les symboles 1 sont au nombre de quatre, le bit de parité est donc égal à 1. Pour compléter le message, il faut encore un bit de départ (Start = 0) et 2 bits d'arrêt (ou Fin, Stop = 1). La forme complète est donc :



- 4 / a) Saut de page = FF = 12 décimal  
 b) Caractères X, Y = 88, 89 décimaux  
 c) Saut de ligne = LF = 10 décimal  
 d) x, y (minuscules) = 120, 121 décimaux  
 e) Saut de page = FF = 12 décimal

Le message complet (en décimal) est le suivant :

12 FF	88	89	10 LF	120	121	12 FF
commande	X	Y	commande	x	y	commande
caractères		caractères		caractères		commande



Si on ne précisait pas qu'il s'agit de caractères ASCII, le message serait erroné puisque ces données, interprétées comme nombres et non comme symboles ASCII, n'ont aucune signification pour l'imprimante.

5 / Les lettres majuscules ont pour codes des nombres allant de 65 à 90 (décimaux), les minuscules, des nombres allant de 97 à 122. Il suffit d'ajouter 32 (décimal).

**65 +** (valeur décimale de A)  
**32** transformation  
**97** (valeur décimale de a)

De la même façon :

$$z (=122) = Z (90) + 32$$

6 / Pour extraire (au moyen de ET) un bit quelconque, il faut avoir recours à un masque contenant le symbole 1 dans la position à extraire.

Le masque requis est :

Bit	7	6	5	4	3	2	1	0
Masque	1	0	0	0	1	0	1	0

En décimal, ce masque vaut :  $2^7 + 2^3 + 2^1 = 128 + 8 + 2 = 138$ .

Pour la seconde partie, il faut tout d'abord convertir les nombres 12, 21, 6, 15 en binaire (voir listing 1 p. 63).

<b>12 =</b>	0	0	0	0	1	1	0	0
<b>21 =</b>	0	0	0	1	0	1	0	1
<b>6 =</b>	0	0	0	0	0	1	1	0
<b>15 =</b>	0	0	0	0	1	1	1	1

Si l'on applique l'opérateur ET entre les nombres et le masque, on obtient pour le nombre 12 :

Bit	7	6	5	4	3	2	1	0
Nombre 12 =	0	0	0	0	1	1	0	0
Masque =	1	0	0	0	1	0	1	0
12 ET masque	0	0	0	0	1	0	0	0

De façon analogue :

**21 ET masque = 00000000**  
**6 ET masque = 00000010**  
**15 ET masque = 00001010**

## Test 4



- 1 / Quelle fonction le compteur d'instructions remplit-il ?
- Il contient le nombre des instructions exécutées.
  - Il contient l'adresse de la mémoire d'où sera extraite la prochaine instruction à exécuter.
  - C'est une mémoire auxiliaire qui sert à effectuer une somme.
- 2 / Dans les tableaux de mémoires ci-dessous, quelle est l'instruction qui sera exécutée (dans les deux cas) après l'instruction 106 ?

	18
sauter à 103	19
	20
	102
	103
	104
	105
RETOUR	106
	107

a

	18
sauter à 103	19
	20
	102
	103
	104
	105
	106
	107

b

- 3 / Dans quel élément de l'Unité Centrale les calculs arithmétiques sont-ils exécutés ?
- Dans la pile.
  - Dans l'UAL.
  - Dans le compteur d'instructions.
- 4 / Qu'est-ce qu'un système d'exploitation ?
- Un ensemble de programmes remplissant des fonctions de base telles que la gestion des périphériques.
  - Les programmes qui servent à effectuer des calculs.
  - Les éléments spéciaux qui commandent l'échange des données dans le DMA.
- 5 / Qu'est-ce que le DMA ?
- Pour l'utilisateur, l'accès direct à la mémoire.
  - La modification des contenus de la mémoire, opérée par l'Unité Centrale.
  - Un échange de données sans passer par l'Unité Centrale.
- 6 / Dans la figure suivante, on voit un tableau de mémoire. Le programme commence à la position 31. Dites s'il y a une erreur, et, si c'est le cas, laquelle :



Ajouter 12 au contenu de A	31
Sauter à 55	32
RETOUR	33
	55
	56
RETOUR	57
	58
	59

7 / Laquelle de ces définitions de « Interruption » (Interrupt) est exacte ?

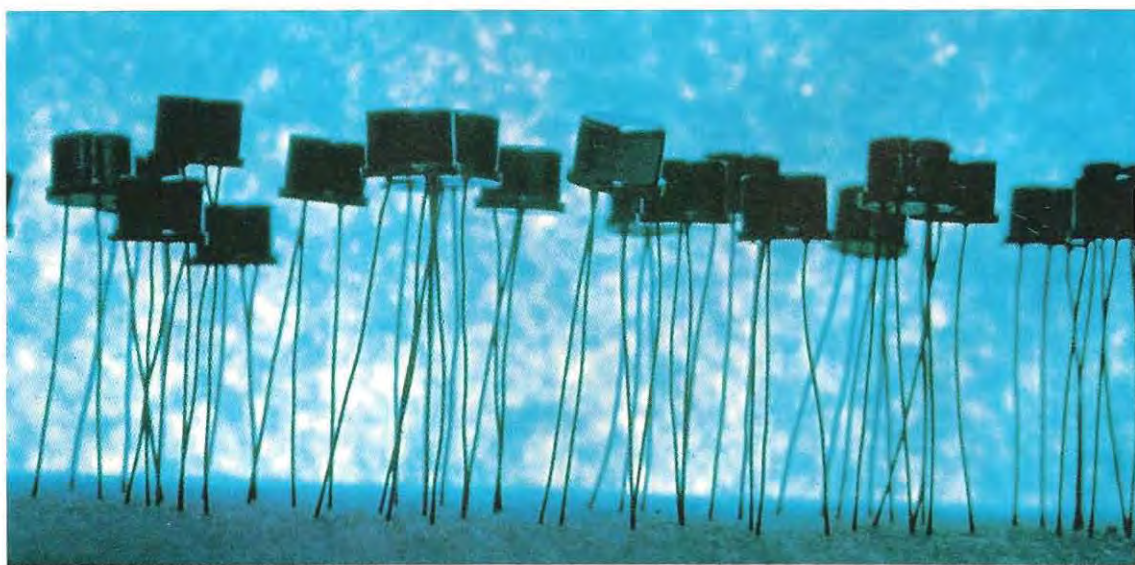
- a) Interruption du fonctionnement de l'Unité Centrale.
- b) Interruption du programme à la suite d'une demande extérieure.
- c) Interruption des échanges avec la mémoire.

8 / Parmi les affirmations suivantes, lesquelles sont exactes ?

- a) Une mémoire EPROM ne peut pas contenir de programme.
- b) L'unité de disque contient environ 64 000 caractères.
- c) Une mémoire ROM ne peut pas être écrite à partir du programme.
- d) A l'arrivée d'une Interruption, l'unité centrale se place en position Attente (Hold).
- e) L'afficheur et le clavier ne peuvent pas déclencher une Interruption.
- f) L'unité disque, normalement, échange des données dans le DMA.

*Les solutions du test se trouvent à la page 183.*

**Dans les systèmes numériques, les transistors sont le plus souvent remplacés par des circuits intégrés.**



Marka

# La Programmation

Ecrire un programme consiste à traduire toutes les opérations à effectuer dans un langage compréhensible pour la machine.

Lorsque nous manipulons une calculatrice courante, nous donnons nos instructions en appuyant sur les touches. Les circuits internes ainsi activés répondent aux ordres transmis par les touches. Les signaux-résultats, émis par ces circuits, sont traduits et viennent s'afficher sur l'écran de la calculatrice.

Dans les machines programmables (ordinateurs), on ne commande pas les opérations une par une. Les fonctions sont chargées en mémoire, et sont exécutées à la mise en route du programme sans autre intervention de l'opérateur. Une machine programmable est ainsi capable d'effectuer des tâches beaucoup plus complexes qu'un simple calcul arithmétique. Sa principale caractéristique est sa capacité de traiter une grande quantité de données. L'ordinateur doit être capable d'acquiescer (de lire) ces données, de les mémoriser, de les traiter, d'effectuer sur ces données les recherches souhaitées par l'utilisateur. Pour déclencher ce traitement informatique, il faut un programme, ou série d'instructions, indiquant à la machine quelles fonctions exécuter, et dans quel ordre. Lorsqu'on recherche, par exemple, le nom de Martin dans un répertoire, on doit prévoir les trois fonctions suivantes :

- 1 / lire un nom figurant dans la liste ;
- 2 / si ce nom est Martin, inscrire O.K., afficher Martin et les données associées, puis terminer le programme ;
- 3 / si le nom n'est pas Martin, revenir à l'étape 1, et lire le nom suivant.

Mais la machine n'est pas en mesure de comprendre des phrases formulées en langage naturel. Pour mettre en mémoire de telles instructions, il faut d'abord les traduire, leur donner une forme susceptible d'être interprétée et convertie en signaux électriques.

La forme d'expression la plus accessible à la machine est le langage binaire. De nombreuses solutions existent pour mettre en corrélation la démarche de l'esprit humain et le

fonctionnement de l'ordinateur. Ce sont les outils de la programmation. Ils sont écrits en langages artificiels mais évolués.

## Les langages de programmation

Le choix de tel ou tel langage est déterminé par le type de problème à résoudre : gestion d'une entreprise, gestion de production, calculs scientifiques, etc. Il peut arriver, cependant, que le choix du langage de programmation n'ait qu'une incidence marginale : il s'agit de gagner du temps, d'économiser sur le nombre d'instructions à donner à la machine. Des circonstances extérieures, ou des préférences personnelles, guideront alors la décision du programmeur.

Parfois, au contraire, un problème donné ne peut se résoudre qu'en employant un langage donné (c'est le cas du Fortran pour les applications mathématiques). Tous les langages de programmation sont symboliques. Ils expriment, dans une formulation compréhensible par l'homme et par la machine, les tâches que la machine doit accomplir. La traduction à l'usage de la machine est assurée par des programmes particuliers intermédiaires : les interpréteurs et les compilateurs.

Outre les langages symboliques, il y a des langages « absolus », où les instructions prennent directement la forme binaire. En général, on n'emploie pas ces langages machine pour rédiger des programmes d'application (ce serait trop fastidieux), mais uniquement pour certaines applications.

## Langages compilés et interprétés

Ces langages (dits de haut niveau) s'adressent au programmeur. Toutes les instructions y sont codées et décrivent l'action à accomplir : par exemple Print (écrire) lorsqu'on veut faire apparaître un mot, ou une formule, sur l'écran, etc. L'ordinateur ne peut pas exécuter directement ces ordres, puisque l'unité centrale ne reconnaît qu'un nombre limité d'instructions fondamentales (voir « Les instructions dans l'UC », p. 146). Il est donc nécessaire de traduire la commande écrite en langage de haut niveau en une série d'instructions élémentaires. Cette conversion se fait au moyen de deux catégories de programmes : les compilateurs et les interpréteurs.

Les compilateurs travaillent sur le programme d'application global, et en fournissent la traduction complète. Les interpréteurs procèdent instruction par instruction, ce qui permet de détecter immédiatement une erreur éventuelle.

Les trois langages de haut niveau les plus utilisés dans les micro-ordinateurs sont :

■ le Basic ■ le Fortran ■ le Cobol

D'autres langages, également employés, seront présentés dans le chapitre consacré aux systèmes d'exploitation.

Le Fortran est conçu pour un usage scientifique, le Cobol pour des utilisations dans le domaine de la gestion et de l'administration. Le Basic est le langage le plus souple : il se prête – bien que d'une manière moins puissante – aux deux emplois.

Le Fortran et le Cobol sont toujours compilés ; le Basic existe sous les deux formes, Basic compilé et Basic interprété.

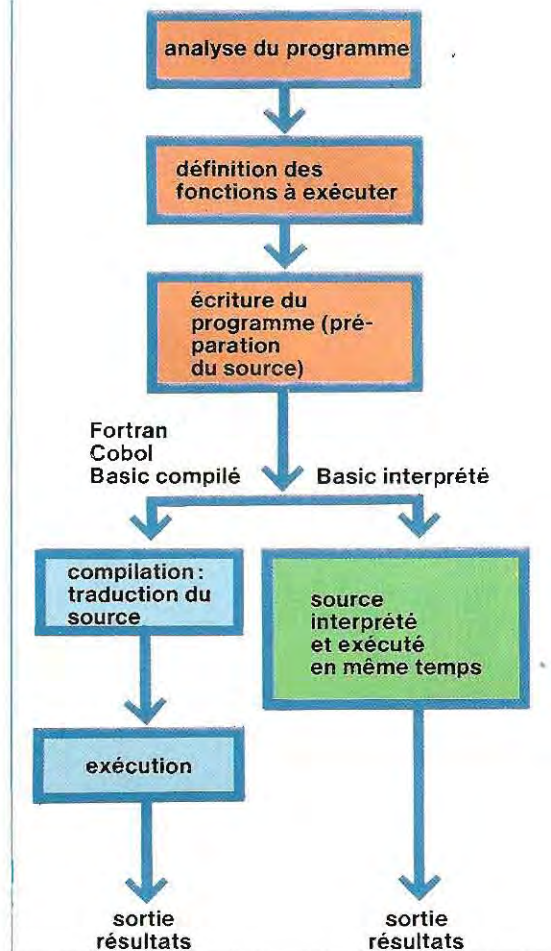
Un programme d'application, quel que soit le langage dans lequel il est écrit, prend le nom de **programme source** (on dit aussi le source).

Le schéma ci-contre illustre les différentes phases du processus de préparation et d'exécution d'un programme d'application, avec un langage compilé ou interprété.

Dans les langages interprétés (Basic interprété), chaque instruction du programme source est traduite (interprétée) au moment même de son exécution. La traduction des instructions se fait en même temps que s'effectue le programme d'application. L'exécution est donc plus longue. Mais l'avantage est que, dès l'introduction du programme source, on obtient un diagnostic. Le programme source est introduit, de façon codée, sur le clavier, et c'est le système d'exploitation qui le charge en mémoire. Au cours de cette phase, si des erreurs se révèlent, l'interpréteur les signale, ce qui permet de les corriger immédiatement. Des erreurs banales telles que les fautes de frappe sont immédiatement repérées et corrigées. En revanche, dans les langages compilés, les erreurs n'apparaissent qu'au moment de l'exécution du programme et il faut attendre la fin du processus de compilation.

Résumons les caractéristiques de ces deux types de langage.

## SCHEMA LOGIQUE DU PROCESSUS D'ECRIURE ET D'EXECUTION D'UN PROGRAMME



**Langage interprété :** possibilité de correction immédiate. Exécution plus lente.

**Langage compilé :** chaque correction implique une nouvelle compilation, donc une attente. En revanche, l'exécution est rapide. L'idéal est de disposer d'un langage qui puisse être aussi bien interprété que compilé. L'interpréteur est donc très utile lors de la rédaction du programme d'application.

La phase d'essai avec compilation intervient seulement après vérification que le programme d'application est correct.

Grâce à sa souplesse, le Basic se prête bien à cette méthode : on rédige en interprété et on teste en compilé : c'est pourquoi c'est le langage le plus utilisé dans les micro-ordinateurs. A l'origine, le Basic avait été conçu pour les

débutants et il reste le plus adapté pour une initiation à la programmation.

## Compilation

La compilation d'un programme source se fait en deux étapes successives.

- La première étape aboutit à la production d'une forme intermédiaire qui prend le nom de **translatable**.
- La seconde étape consiste à relier entre elles les différentes parties du translatable. Cette opération s'appelle le **lien** (en anglais link). Au terme de cette opération, on obtient le **programme objet**, qui sera chargé en mémoire avant d'être exécuté.

Habituellement, dans les systèmes comportant un compilateur, il faut prévoir des unités

disque, chargées de mémoriser le translatable. Dans les langages interprétés, les unités disque ne sont pas indispensables, puisque l'interpréteur peut résider en mémoire de façon permanente.

## Basic

Ce langage est adapté à la fois aux usages scientifiques et de gestion puisque, outre les caractéristiques du Fortran, il a aussi la capacité de traiter des textes et différentes instructions destinées à l'impression de rapports économiques. Le traitement des données sur disque (fichiers) est de moindre puissance que celui du Cobol, mais il convient tout à fait à la structure des micro-ordinateurs.

Le Basic est plus récent que les autres langages, et les formes sous lesquelles il a été adopté ne sont pas encore unifiées. Mais les différences (mineures) qui existent d'un micro-ordinateur à l'autre sont destinées à disparaître.

## Fortran

C'est un langage à l'usage des scientifiques. Voici quelles en sont les caractéristiques principales :

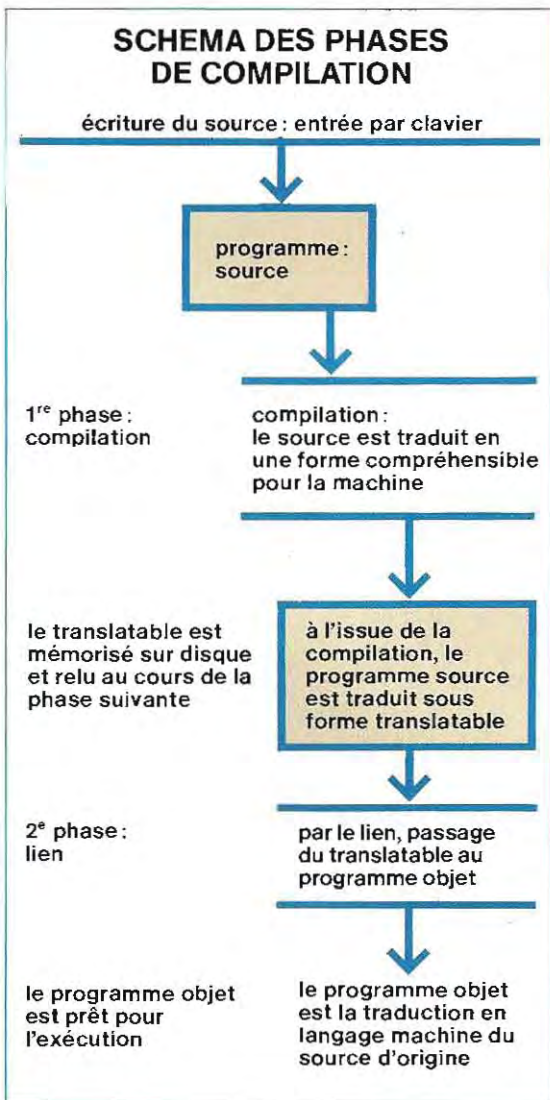
- nombre restreint d'instructions fondamentales ;
- peu de texte à traiter ;
- des données normalement sous forme de nombres exponentiels ;
- relativement peu de données à traiter, mais une exploitation très poussée de ces données ;
- nécessité d'instruments de calcul particuliers (algorithmes mathématiques).

L'emploi de ce langage en gestion n'est pas exclu, mais il se révèle lourd et moins maniable, surtout par manque d'instructions permettant d'élaborer des caractères alphanumériques.

## Cobol

C'est un langage destiné exclusivement à des applications de gestion. Ses caractéristiques principales sont les suivantes :

- pas d'instructions de calcul (sauf les opérations principales) ;
- possibilité de bonne gestion des données sur disque (archives) ;
- instructions pour l'édition de rapports de caractère économique.



# Analyse et organigrammes

La rédaction d'un programme n'est que la phase ultime, l'aboutissement d'un travail complexe d'analyse d'un problème donné, qu'il faut ensuite traduire sous forme de structures synthétiques compréhensibles pour la machine.

**L'analyse** du problème consiste à décomposer les opérations à réaliser en une série de fonctions élémentaires, auxquelles correspondent des symboles graphiques. A partir de cette représentation graphique, **l'organigramme**, ou **ordinogramme** (en anglais flowchart), on passe à l'écriture proprement dite des instructions. Nous allons tenter d'illustrer les principales méthodes d'analyse et les techniques de préparation des organigrammes.

## Etablissement et écriture d'un programme

Pour chaque application spécifique, il faut fournir à l'ordinateur le programme adéquat. Lorsque l'application est complexe, on peut recourir à des sous-programmes, dont l'ensemble constitue une **procédure**.

Par exemple, dans la gestion d'un commerce, on distinguera trois étapes fondamentales :

- **entrée des marchandises**
- **sortie des marchandises**
- **inventaire des stocks.**

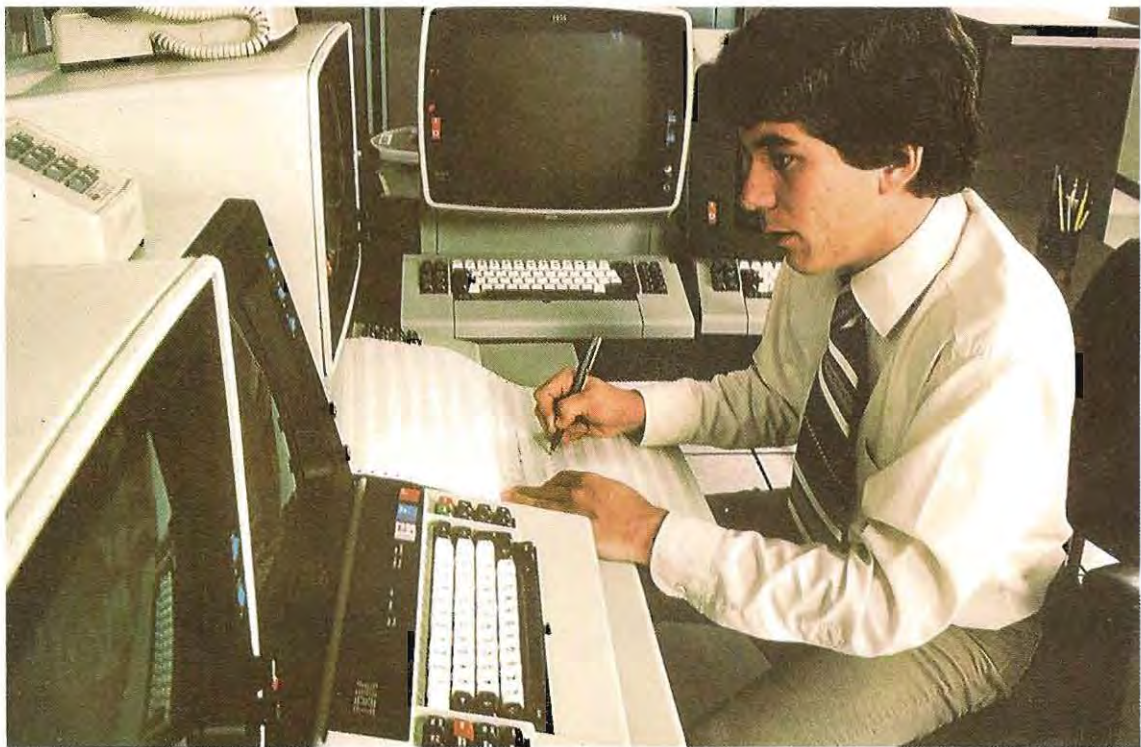
A chacune de ces parties correspondra un programme, l'ensemble constituant la procédure de gestion du magasin.

Avant de commencer l'écriture d'un programme, il faut connaître les différents aspects du problème et définir une méthode. Cette phase du travail est la plus délicate, car c'est d'elle que dépend toute la qualité du programme.

L'établissement d'un programme peut se diviser en trois phases :

- **analyse du problème**
- **synthèse**
- **mise au point des organigrammes.**

**Dans les compagnies d'assurances, la capacité de mémorisation et la rapidité de recherche de l'ordinateur permettent un traitement rapide des dossiers.**



Marka

## Analyse

Un programme, quel qu'il soit, ne peut traiter que des données préalablement fournies à la machine. Dans l'exemple précédent (inventaire des stocks), le programme ne donnera de résultats corrects que si l'on a introduit tous les mouvements (entrée et sortie) des marchandises. Ces mouvements doivent en outre porter sur des articles connus de la machine. La première phase de l'analyse consiste à définir les résultats recherchés et à vérifier que l'on dispose bien de toutes les données nécessaires.

Supposons par exemple que nous voulions obtenir l'inventaire des stocks sous forme d'une liste contenant les informations suivantes :

- 1 / code de l'article
- 2 / description
- 3 / quantité en stock
- 4 / prix unitaire
- 5 / valeur globale du stock.

En analysant les résultats recherchés, on définit les différents programmes qui constitueront la procédure (schéma à droite). Dans notre exemple, les premières données à écrire sont :

- 1 / code de l'article
- 2 / description.

Il faut d'abord fournir ces données à la machine. Il faut donc, en premier lieu, un programme pour créer les rubriques correspondant aux articles du magasin.

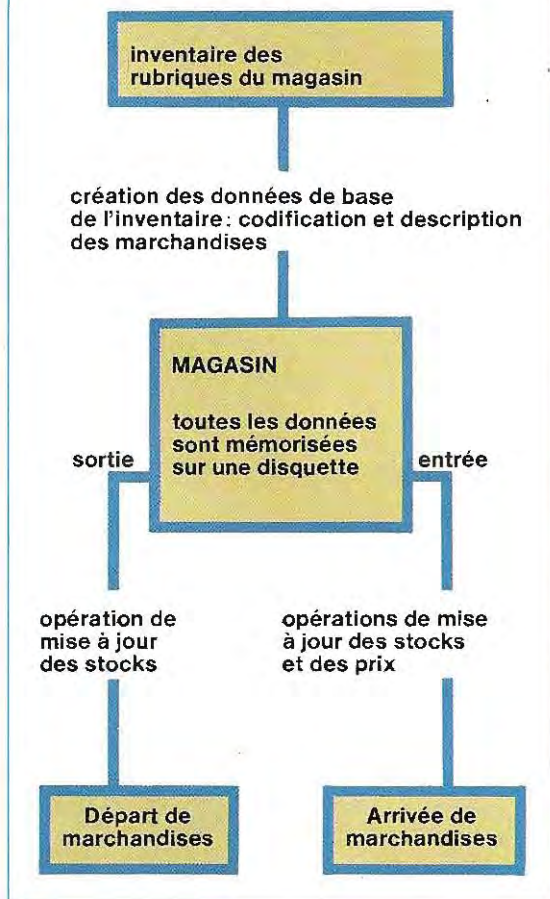
La demande suivante (quantité en stock) est celle du chiffre représentant la quantité existant en magasin. Ce chiffre variera en fonction de deux opérations :

- départ d'un article
- arrivée d'un article.

Le programme doit, à chaque sortie et à chaque entrée de marchandise, effectuer la soustraction et l'addition correspondantes, c'est-à-dire mettre à jour l'état des stocks.

Le quatrième point (prix unitaire) est une donnée à fournir au moment de l'entrée de la marchandise. Ainsi, pour les points 3 et 4, il faut

## SCHEMA LOGIQUE DE LA PROCEDURE DE GESTION D'UN MAGASIN



deux programmes :

- sortie de marchandises
- entrée de marchandises.

Enfin, le point 5 requiert un programme qui, utilisant les résultats des précédentes opérations, effectue un calcul final. On peut résumer les résultats de l'analyse en un tableau des principales opérations à exécuter :

- 1 / Constitution des rubriques (codification et description des articles)
- 2 / Mouvements d'entrée (état des stocks) et nouveaux prix
- 3 / Mouvements de sortie (état des stocks)
- 4 / Impression des listings des stocks (calcul de la valeur totale).



Tableau des programmes et des données nécessaires :

### PROGRAMMES

Création des rubriques  
Entrées  
Sorties  
Impression des listings

### DONNEES

Codification des articles et description  
Volume du stock et prix unitaire  
Volume du stock  
Valeur totale

Les fonctions principales de chacun des programmes peuvent se résumer ainsi :

#### Création des rubriques

L'utilisateur introduit, pour chaque article existant dans le magasin, un code permettant d'identifier l'objet, et sa description.

#### Entrées

Il s'agit d'ajouter au stock chaque nouvelle arrivée de marchandise, et de mettre à jour les prix. Les opérations à effectuer sont :

quantité actuelle = quantité précédente + quantité arrivée ;

prix = prix de l'article arrivé.

#### Sorties

Il suffit ici de mettre à jour l'état du stock après livraison au client, soit : quantité actuelle = quantité précédente - quantité prélevée.

#### Impression du listing

Avant d'imprimer le listing d'état des stocks, le programme doit calculer, pour chaque article, la valeur totale de ce qui est en magasin, soit :

valeur totale = stock x prix unitaire,

soit : Rubrique 5 = Rubrique 3 x Rubrique 4.

Les opérations sont illustrées par les schémas des pages 160 et 161.

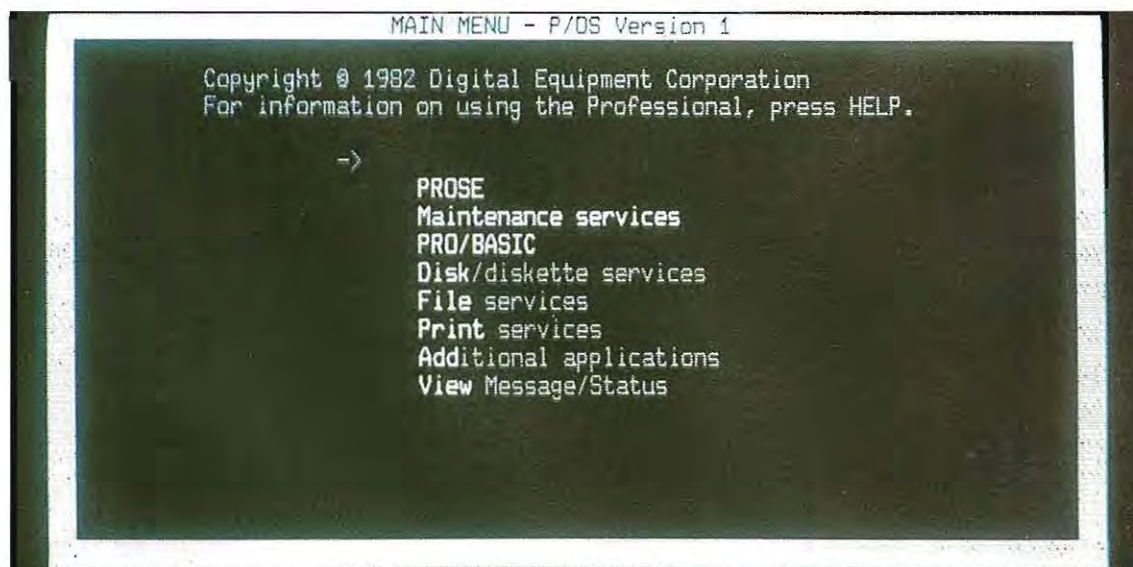
Les données relatives à chaque rubrique doivent être mémorisées par l'ordinateur. Pour cela on utilisera une mémoire de masse, par exemple une disquette. Et on choisira le type de machine en fonction de la capacité de mémoire nécessaire, de la vitesse requise, du nombre d'utilisateurs appelés à travailler simultanément sur la machine. Le choix définitif se fera après une analyse précise de l'utilisation qu'on veut en faire et se portera sur :

**Ordinateur domestique.** Pour un seul utilisateur à la fois. Peu de données à mémoriser. Pas d'exigence de rapidité. Coût minimal. Peu, ou pas de possibilité d'extension.

**Micro-ordinateur.** Un seul utilisateur, éventuellement plusieurs. Données assez importantes. Possibilité d'obtenir une bonne rapidité de traitement. Coût moyen. Assez bonnes possibilités d'extension.

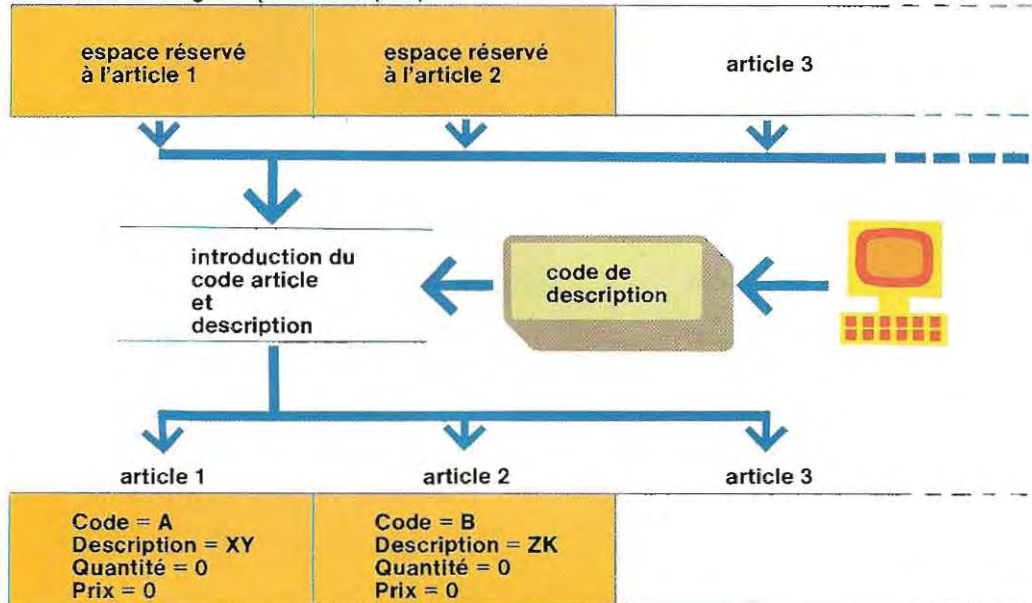
**Mini-ordinateur.** Possibilité d'utilisateurs multiples, simultanément et éventuellement avec des programmes différents. Quantité de données selon les besoins. Rapidité moyenne ou élevée. Coût élevé. Possibilités d'extension très grandes.

Informations visualisées sur l'écran d'un ordinateur.



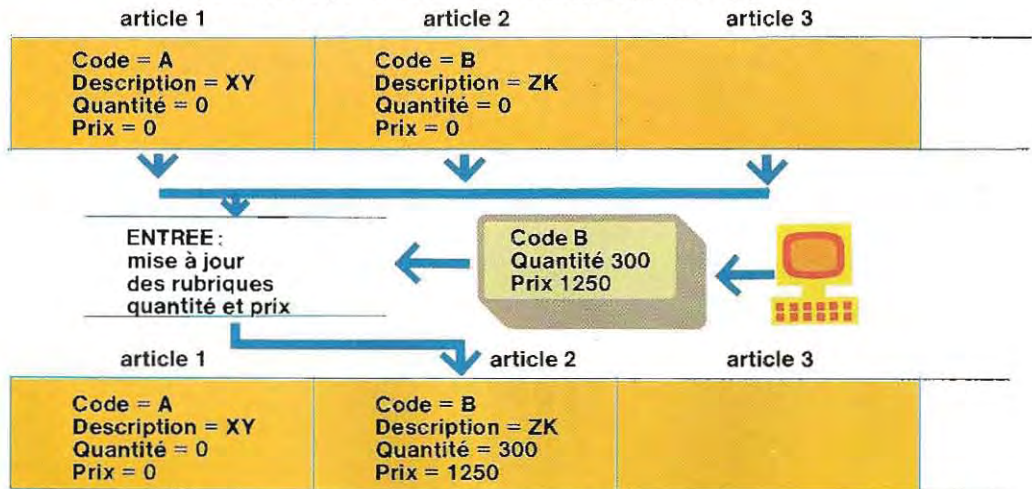
## CREATION DES RUBRIQUES DU MAGASIN

inventaire du magasin (vide au départ)

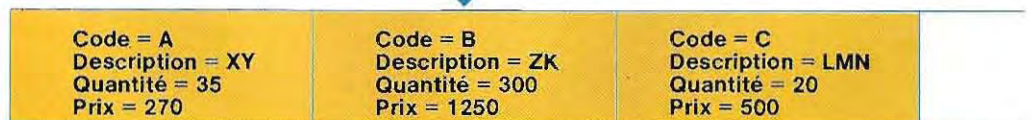


On a introduit dans l'inventaire du magasin les codes et les descriptions de chaque article.  
Quantités et prix sont égaux à 0 puisque le chargement n'est pas encore effectué.

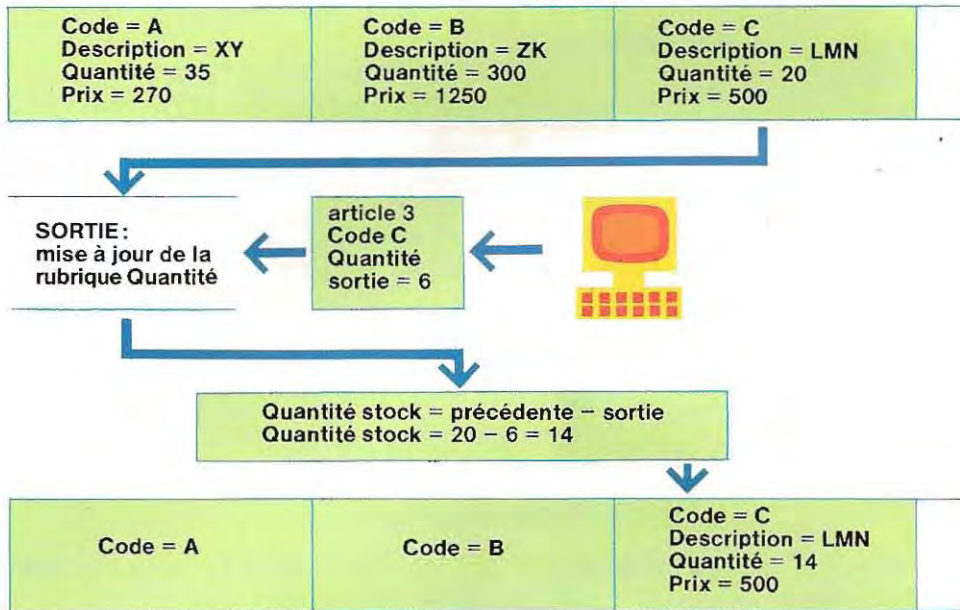
## OPERATIONS ENTREE DE MARCHANDISES



L'opération entrée (par exemple de l'article 2) met à jour la quantité (300) et le prix (1250) ; une fois tous les articles arrivés, l'inventaire du magasin est complet.



## OPERATIONS SORTIE DE MARCHANDISES



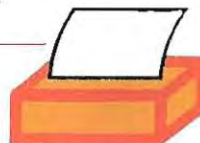
L'opération de sortie (prélèvement sur stock) soustrait la quantité prélevée de la quantité stock précédente et met à jour les données dans l'inventaire.

## EDITION LISTINGS DU STOCK

Code = A Description = XY Quantité = 35 Prix = 270	Code = B Description = ZK Quantité = 300 Prix = 1250	Code = C Description = LMN Quantité = 14 Prix = 500
---	---	--

Lecture de l'inventaire total.  
Pour chaque article,  
calcul de la valeur totale.  
Valeur totale = Prix × Quantité

Les valeurs totales de  
tous les articles sont  
additionnées: on obtient  
le total général.



Code	Description	Quantité	Prix	Valeur totale
A	XY	35	270	9450
B	ZK	300	1250	375000
C	LMN	14	500	7000
<b>Total général</b>				<b>391450</b>

Le choix de base se fera en fonction des besoins prévisibles.

**Les objectifs de la phase d'analyse sont :**

- 1 / contrôler l'existence de toutes les données d'entrée,
- 2 / définir les éventuels programmes annexes,
- 3 / choisir le type de machine.

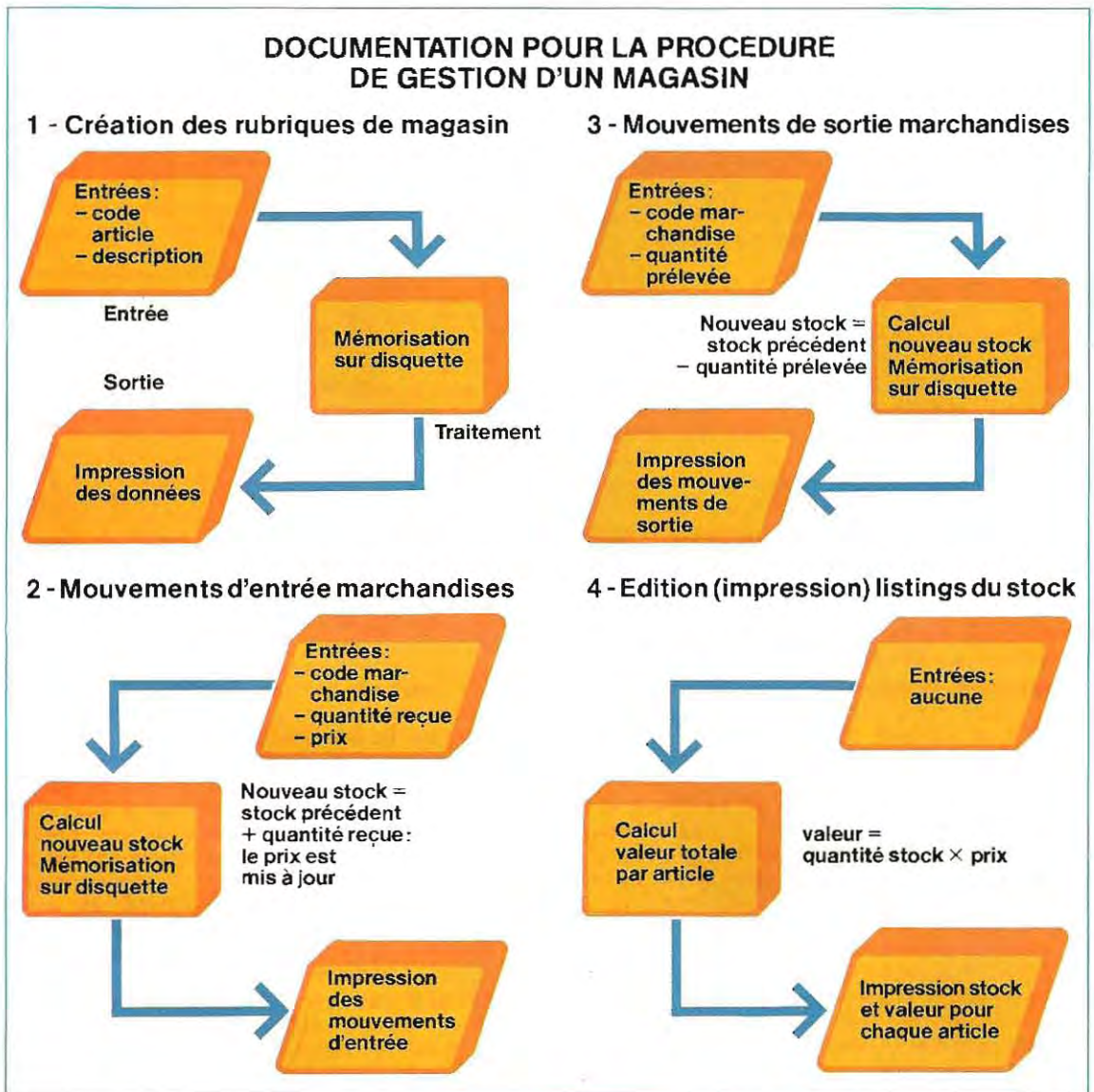
### Synthèse

La synthèse consiste à préparer un ou plusieurs schémas qui montrent de façon synthétique les fonctions d'un programme, ou des programmes d'une procédure. Le plus simple est d'utiliser la technique HIPO (Hierarchy plus Input-Process Output).

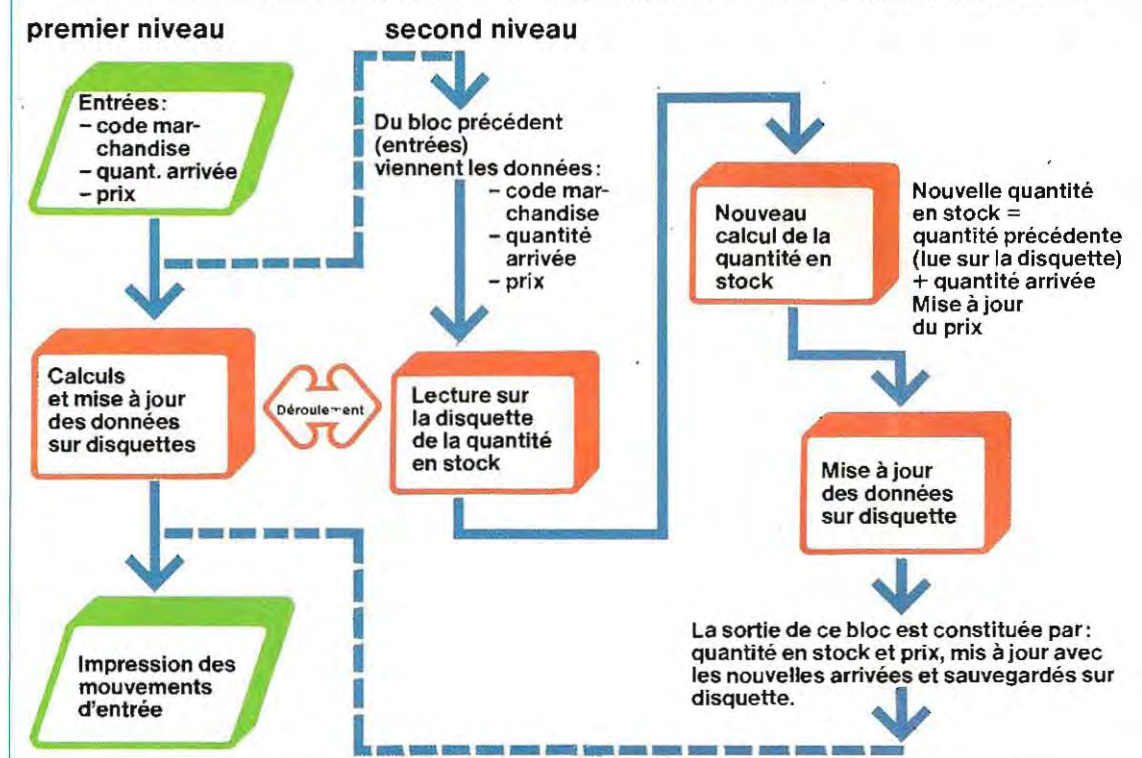
Cette technique consiste à dresser pour chaque programme (ou partie de programme) un schéma en trois blocs.

**Premier bloc : entrée.** Contient la description de toutes les entrées nécessaires à l'établissement du programme.

**Deuxième bloc : traitement.** Contient la des-



## EXEMPLE DE DEROULEMENT D'UN SCHEMA D'ENTREE DE MARCHANDISES



cription de tous les calculs et, en général, de toutes les opérations à effectuer sur les données.

**Troisième bloc : sortie.** Contient la description de toutes les sorties prévues.

Dans les cas les plus simples, un seul organigramme HIPO peut suffire. Pour des programmes plus complexes, il en faudra plusieurs, plus ou moins détaillés. La technique HIPO est non seulement un auxiliaire précieux dans le déroulement du travail, mais fournit également une bonne documentation sur l'application (dossier de programmation).

Le schéma de la page 162 présente des organigrammes HIPO appliqués à une procédure de gestion de magasin.

Ensuite on pourra développer les blocs les plus complexes. Par exemple, dans la partie Mouvements d'entrée, on peut développer le bloc traitement de la façon suivante :

- a/ lecture sur disque, pour chaque code marchandise, du stock existant ;
- b/ addition au stock ancien de la nouvelle quantité arrivée ;
- c/ remplacement de l'ancien prix unitaire par le nouveau ;

d / remplacement sur le disque des données anciennes par les nouvelles.

Chacun des traitements énumérés a trait à un article déterminé ; la donnée d'entrée est le code de l'article.

Au terme du traitement, on a sur disque les données mises à jour : c'est la sortie, le résultat du traitement.

Le schéma ci-dessus représente l'état initial et le développement (deuxième état). Avec ce deuxième niveau d'information, nous n'avons pas encore suffisamment de détails pour procéder à l'écriture d'un programme. Il faut passer à un niveau supérieur de description et préparer un schéma qui illustre, à l'aide de symboles, toutes les opérations à accomplir. Dans le schéma en haut de cette page, il y a lieu de prévoir une vérification des données (voir par exemple si on n'a pas essayé d'introduire un code marchandise qui n'existe pas). A la sortie sur imprimante, il faut prévoir quelques lignes pour en-tête, etc.

Le schéma des opérations à effectuer, à ce niveau de détail, prend le nom d'organigramme. C'est l'ultime étape préparatoire avant l'écriture du programme proprement dite.

## Des machines qui savent lire

Si l'on considère le rôle joué par les ordinateurs dans notre vie quotidienne, on comprend que pour d'innombrables applications on a besoin de machines capables de lire automatiquement.

Chaque année, plusieurs milliards de chèques bancaires sont rédigés. Il faut les classer de telle manière que chaque chèque puisse être échangé avec l'agence émettrice ou avec une chambre compensatrice.

Seules les machines peuvent effectuer cet énorme travail. Elles lisent en effet les caractères imprimés à l'encre magnétique au bord inférieur du chèque et opèrent une rapide sélection automatique.

Dans certains pays, le volume du courrier est encore assez modeste et le tri peut se faire manuellement ou avec des équipements semi-automatiques : c'est un employé qui lit l'adresse et compose sur un clavier le code postal ; une machine classe ensuite la lettre automatiquement.

Au Japon et aux Etats-Unis, le volume du courrier dépasse les capacités en main-d'œuvre disponible. On emploie des machines lectrices appelées des lecteurs. Ces machines rejettent environ 8% des lettres parce qu'elles ne réussissent pas à lire l'adresse sur l'enveloppe.

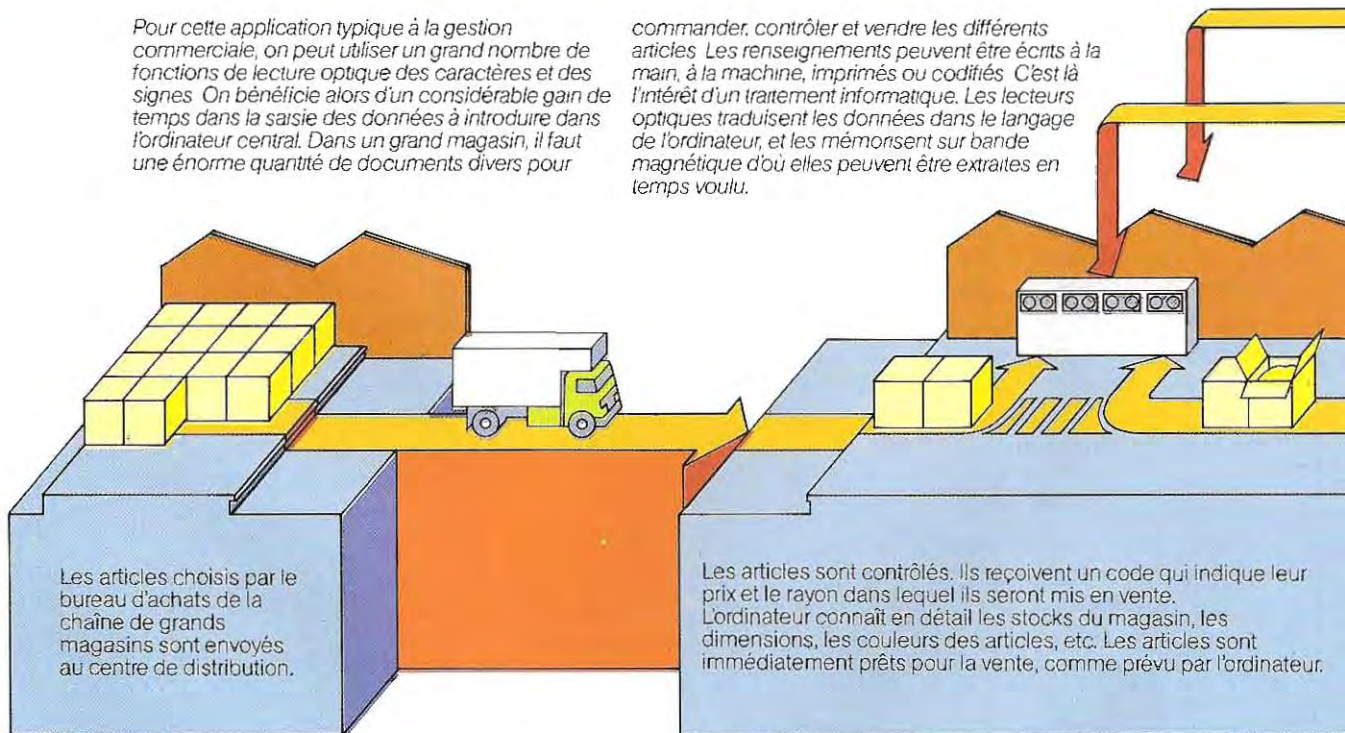
Ces matériels ne se bornent pas à classer des documents, elles peuvent aussi introduire des données dans les ordinateurs. Par exemple, si l'on veut tenir à jour la liste des noms et des adresses d'un million de personnes ou d'établissements, il faut enregistrer environ 25 000 modifications par semaine. On peut fournir les adresses à un ordinateur, par l'intermédiaire de la console. Mais on peut aussi écrire les adresses sur une simple feuille de papier, à la machine à écrire, et communiquer cette feuille à l'ordinateur.

Dans plusieurs pays d'Europe, les Postes utilisent déjà des machines capables de lire les caractères imprimés. Le rythme moyen de ces machines est de 1 000 caractères par seconde.

La Bank of America en possède une capable de lire 40 000 caractères par

Pour cette application typique à la gestion commerciale, on peut utiliser un grand nombre de fonctions de lecture optique des caractères et des signes. On bénéficie alors d'un considérable gain de temps dans la saisie des données à introduire dans l'ordinateur central. Dans un grand magasin, il faut une énorme quantité de documents divers pour

commander, contrôler et vendre les différents articles. Les renseignements peuvent être écrits à la main, à la machine, imprimés ou codifiés. C'est là l'intérêt d'un traitement informatique. Les lecteurs optiques traduisent les données dans le langage de l'ordinateur, et les mémorisent sur bande magnétique d'où elles peuvent être extraites en temps voulu.



seconde. Mais ce sont là des équipements compliqués et très coûteux.

On utilise les lecteurs électroniques également pour introduire des textes dans des machines de traitement de texte. Un dactylographe compose le texte sur une machine appropriée, l'auteur le revoit et le corrige s'il y a lieu. Ensuite, une machine lectrice explore le texte et le transmet à la machine de traitement de texte, qui permet à l'opérateur d'apporter très rapidement les modifications nécessaires.

Malheureusement, ces lecteurs peu coûteux utilisés avec les processeurs de texte sont peu fiables lorsqu'il est nécessaire de lire un texte qui n'est pas écrit en caractères spéciaux, comme par exemple une correspondance portant des corrections. Il faut une écriture spéciale et des intervalles nets entre les caractères.

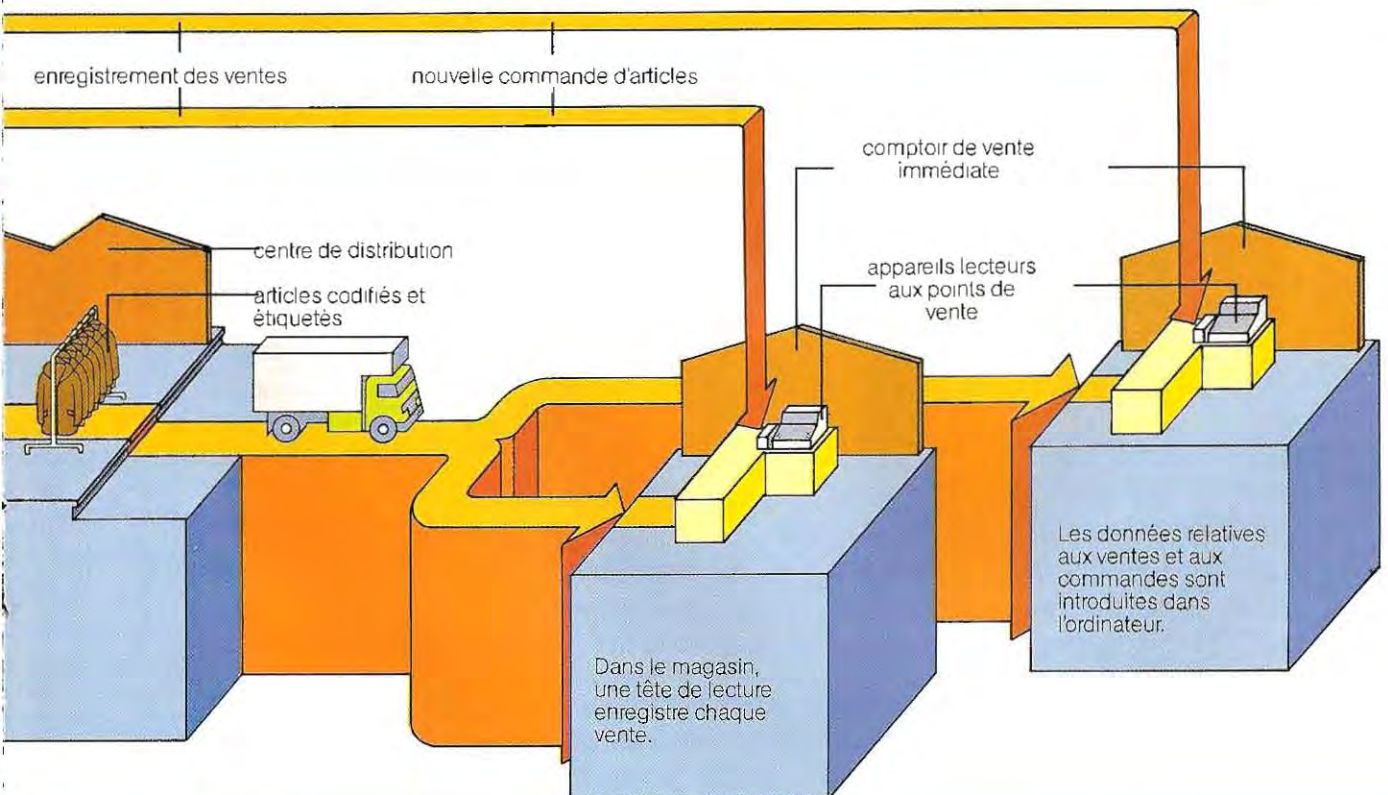
Les compagnies européennes du gaz et de l'électricité utilisent des lecteurs de factures qui font immédiatement apparaître les arriérés de paiement.

Pour acquérir une machine ultra-rapide, capable de lire lettres, chiffres et ponctuation, il faut investir des sommes importantes. Cependant, le marché propose des machines à des prix très différents et on peut, par exemple, faire des économies considérables si l'on demande à une machine de lire, non pas des caractères, mais des séquences codifiées de barres parallèles plus ou moins épaisses et espacées.

Ces codes à barres, déchiffrables même par des appareils portatifs, sont d'un emploi désormais courant dans les bibliothèques et les magasins. Les appareils de lecture optique pour ce type de code sont beaucoup plus simples et économiques que ceux auxquels on demande de déchiffrer des caractères d'écriture normale.

Par rapport aux hommes, les machines tolèrent moins les différences de forme et de qualité des caractères, mais elles sont infiniment plus rapides.

Le prix d'un lecteur dépend de la gamme de caractères qu'il aura à lire, et de sa rapidité. Les progrès de la micro-électronique



ont permis de réduire davantage le coût de la partie électronique de l'appareil que celui de sa partie mécanique (chargée par exemple de la transmission des cartes). Beaucoup de machines pourtant récentes, mais peu perfectionnées sur le plan mécanique, ne peuvent lire qu'une dizaine de caractères par seconde. Une solution particulièrement économique consiste à faire passer une tête lectrice sur la ligne de texte à lire.

Si l'encre, le papier et le type de caractères correspondent exactement aux normes établies par le fabricant, une bonne machine est capable de lire chiffres, lettres (majuscules et minuscules) et ponctuation avec une très faible marge d'erreur. Une erreur typique est la confusion entre deux caractères (par exemple 5 et S), mais le taux normal d'erreur est de moins de 1 pour 100 000. Autre inconvénient possible: la machine signale (environ 1 fois sur 10 000) qu'elle n'a pas pu reconnaître un certain signe.

Par ailleurs, si une machine est destinée à lire seulement des chiffres, il n'est pas nécessaire d'avoir une aussi bonne qualité d'écriture. Avant de reconnaître la forme d'un caractère, un lecteur optique explore le texte au moyen d'une série de cellules photo-électriques, ou d'un dispositif laser ou, plus rarement, d'une télécaméra. Après analyse, le caractère est représenté par la machine comme une matrice de bits.

Dans le diagramme (reproduit ci-dessous) qui représente le nombre 222, le chiffre 1 cor-

000000000	000000000	000000000
0011111000	0011111000	0011111000
0111111100	0111111100	0111111100
0110001110	0110001110	0110001110
0000000110	0000000110	0000000110
0000000110	0000000110	0000000110
0000001100	0000001100	0000001100
0000011000	0000011000	0000011000
0000110000	0000110000	0000110000
0011100000	0011100000	0011100000
0111111110	0111111110	0111111110
0111111110	0111111110	0111111110
0000000000	0000000000	0000000000
0000000000	0000000000	0000000000

respond au noir et le 0 au blanc. L'homme comprend immédiatement que le code lu correspond au chiffre 2, mais, pour arriver à la même conclusion, la machine a besoin de faire des calculs.

Une méthode consiste à lui faire comparer cette matrice de 130 bits à d'autres matrices de 130 bits, précédemment mémorisées par la machine et pourvues chacune de son propre code. Lorsqu'on lui présente les deux modèles, l'appareil compte le nombre de bits qui diffèrent. La reconnaissance consiste à trouver le code du modèle qui possède le moins de bits discordants. Ce schéma est bien sûr simplifié ici; en réalité, il est beaucoup plus complexe. Et cependant, il n'est pas encore assez efficace. Il peut en effet facilement fournir un code inexact à cause de caractères déformés, mal imprimés, empâtés etc., ou d'une panne dans le mécanisme d'exploitation électronique.

Pour remédier à ces inconvénients, on a mis au point une autre technique de reconnaissance, fondée sur l'emploi de relations logiques, qui permet de reconstituer un bit défectueux.

Si l'on considère par exemple la matrice de bits suivante, dans laquelle certains bits-clés ont été représentés par des lettres de l'alphabet,

0000000000	0000000000	0000000000
000aabb000	000aabb000	000aabb000
00aeef fb00	00aeef fb00	00aeef fb00
0ae0000fb0	0ae0000fb0	0ae0000fb0
0ae0ii0fb0	0ae0ii0fb0	0ae0ii0fb0
0ae0ii0fb0	0ae0ii0fb0	0ae0ii0fb0
0ae0000fb0	0ae0000fb0	0ae0000fb0
0cg0000hd0	0cg0000hd0	0cg0000hd0
0cg0jj0hd0	0cg0jj0hd0	0cg0jj0hd0
0cg0jj0hd0	0cg0jj0hd0	0cg0jj0hd0
0cg0000hd0	0cg0000hd0	0cg0000hd0
00cgghhd00	00cgghhd00	00cgghhd00
000cc0d000	000cc0d000	000cc0d000
0000000000	0000000000	0000000000

le caractère prédéfini sera reconnu comme 0 si:

au moins cinq **a** sont **1**

ou

au moins quatre **e** sont **1**;





BFI-Letton

**Un micro-ordinateur est placé dans les caisses enregistreuses. Ces machines contiennent un système miniature avec unité centrale, mémoire et interface E/S.**

et  
 au moins cinq **b** sont **1**  
 ou  
 au moins quatre **f** sont **1** ;  
 et  
 au moins cinq **c** sont **1**  
 ou  
 au moins quatre **g** sont **1** ;  
 et  
 au moins cinq **d** sont **1**  
 ou  
 au moins quatre **h** sont **1** ;  
 et  
 au moins trois **i** sont **0** ;  
 et  
 au moins trois **j** sont **0**.

Certaines des machines les plus coûteuses sont capables de reconnaître les majuscules écrites à la main en lettres d'imprimerie, à condition que l'espace entre les lettres soit respecté.

Avant de reconnaître un caractère écrit à la main, la machine (lecteur) doit déterminer sur

le document une zone qui ne contient que ce caractère. Il faut donc qu'elle puisse l'isoler des caractères proches. D'où la nécessité de l'espacement.

La prolifération des ordinateurs à base de microprocesseur a provoqué une explosion de demandes de machines destinées à la conversion de données numériques en représentation codifiée à l'usage de l'ordinateur lui-même.

C'est ce marché qui pousse à la construction de machines lectrices.

Mais on constate que, vers la fin des années 1970, le nombre de brevets accordés pour ces appareils a fortement diminué.

Nombre de firmes américaines et japonaises ont parfois perdu beaucoup d'argent à construire des machines qui se sont finalement révélées invendables parce que trop chères.

Il semblerait pourtant qu'ampleur du marché et prix tendent progressivement à s'équilibrer, grâce aux progrès continuels de l'électronique.

## Etablissement des organigrammes

L'élaboration d'un organigramme est la représentation d'un programme par une succession de symboles graphiques. L'ensemble des diagrammes qui en résulte est l'organigramme du programme. Chaque fonction accomplie par un ordinateur a en effet son symbole graphique propre. De ce fait, lorsque l'organigramme a été correctement établi, l'écriture du programme se réduit à un travail de traduction des symboles en instructions, telles qu'elles existent dans le langage de programmation choisi.

Les symboles graphiques utilisables peuvent être répartis en trois groupes fondamentaux. Ces groupes sont ceux des

- symboles liés aux opérations de l'ordinateur (calculs, décisions, etc.);
- symboles des opérations d'entrée et de sortie des données;

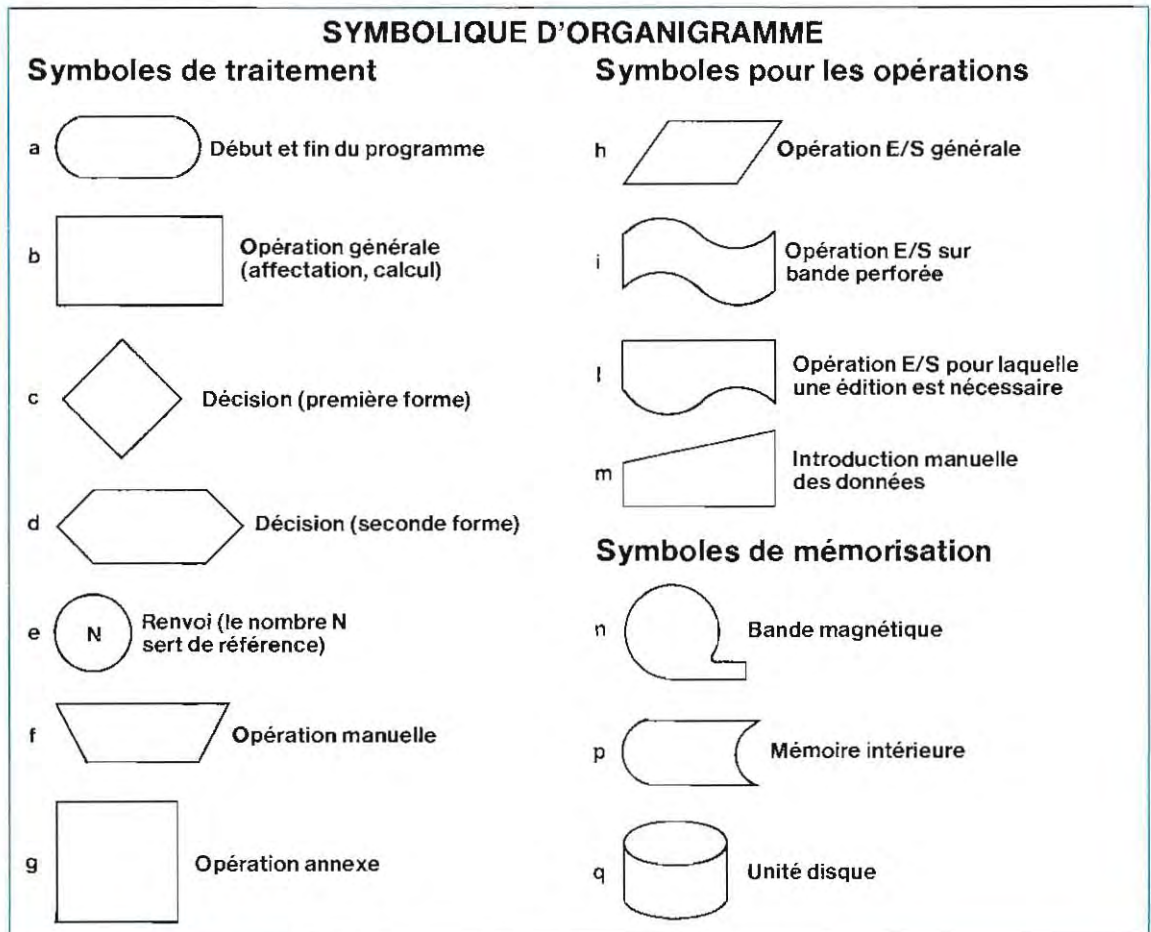
- symboles liés à la mémorisation.

Dans le schéma de cette page, on trouvera les symboles les plus couramment utilisés. Nous les décrirons plus en détail par la suite. Nous décrirons plus sommairement ceux qui sont peu utilisés et qui ne nous serviront pas dans nos exemples d'applications. Mais il faut savoir que dans les applications de programmation sur micro-ordinateurs, la gamme des symboles utilisés est beaucoup plus restreinte.

### Symboles d'usage courant

Ce sont les symboles de traitement puisque les opérations de traitement constituent l'essentiel d'un programme.

Le symbole **a** indique le début et la fin d'un programme. Tout programme, aussi complexe soit-il, revêt la forme générale qui apparaît dans le schéma de la page 169 (en haut). Les flèches indiquent le sens suivi par le traitement.





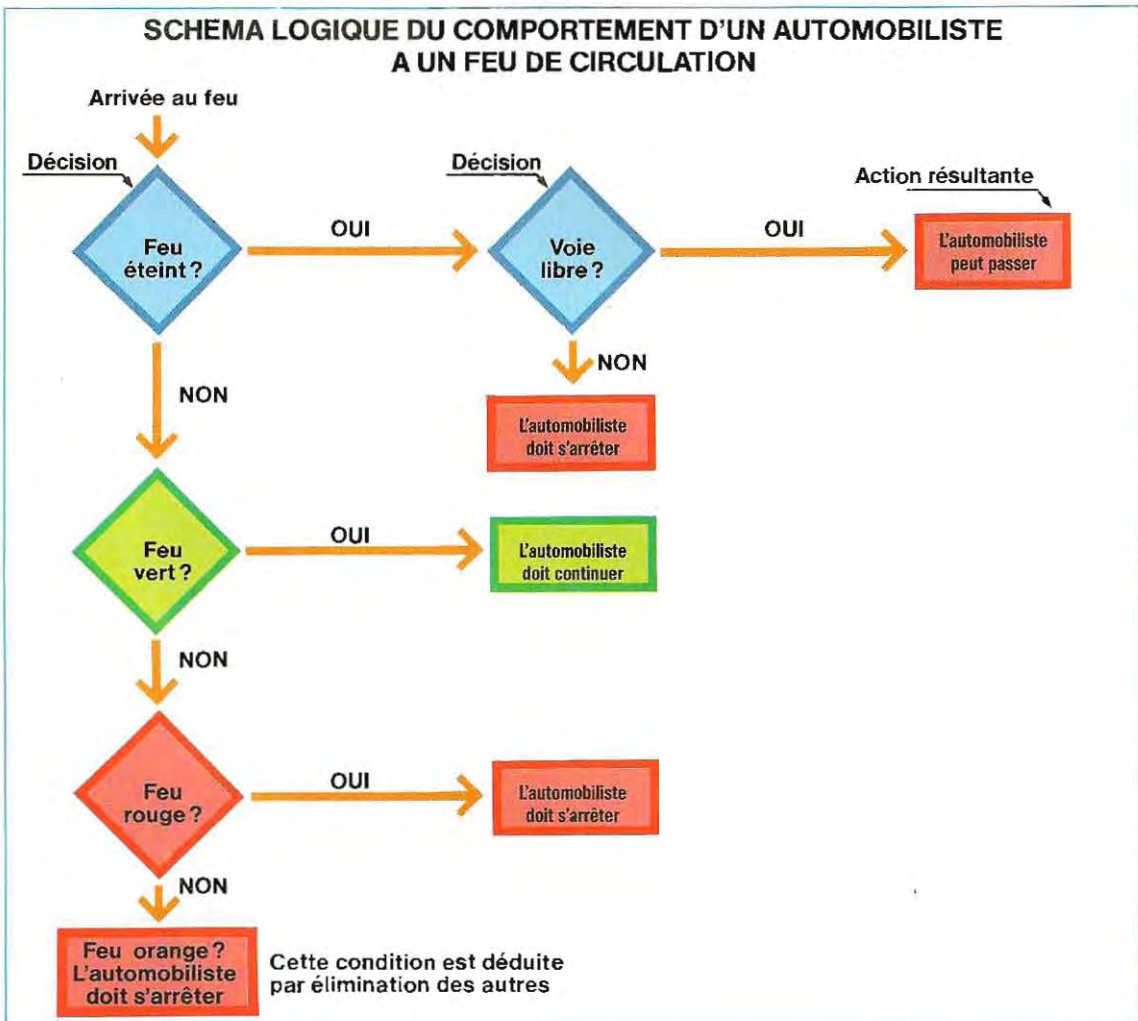
$$\text{Coût total} = \text{Quantité} \times \text{Coût unitaire}$$

Les symboles **c** et **d** sont équivalents et expriment une décision (appelée aussi test). On choisit l'un ou l'autre de ces symboles selon la longueur du commentaire que l'on veut y introduire. La décision se décompose en deux phases.

Le symbole **b**, accompagné du commentaire nécessaire, indique un traitement général, par exemple un calcul. Dans la procédure de gestion d'un magasin que nous venons d'étudier, le coût total est obtenu en faisant le produit de la quantité (nombre d'articles) par le coût unitaire : il faudra indiquer cette opération par le symbole **b** de la façon suivante :

- Première phase : comparaison entre deux (ou plusieurs) valeurs ;
- Seconde phase : choix de la marche à suivre en fonction du résultat de la comparaison.

Prenons l'exemple d'un automobiliste qui arrive à un feu de signalisation. On peut sché-



matiser ses différents comportements possibles par les symboles **b** et **c** (voir schéma p. 169).

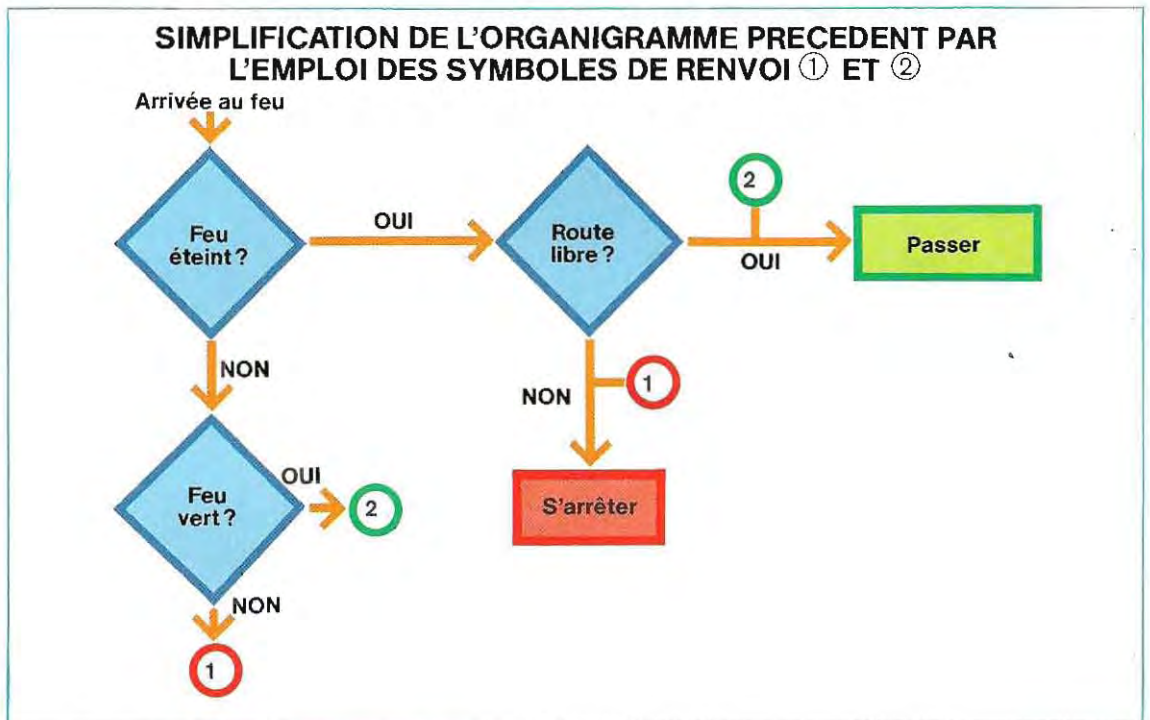
On utilise le symbole **e** (renvoi) du schéma de la page 168 pour indiquer le point de reprise du programme après exécution du bloc d'instructions précédent. On évite ainsi de tracer la ligne de liaison, ce qui facilite la lecture des organigrammes et permet de les dessiner sur plusieurs pages lorsqu'ils sont longs ou compliqués. Le schéma de cette page illustre le déroulement de ce processus de renvoi.

On peut alors éviter la question "feu rouge ?" dans la mesure où, quelle que soit la réponse

## Symboles graphiques moins utilisés

Le symbole **f** représente une intervention manuelle, telle que le remplacement d'une disquette ou l'alimentation du bac de papier destiné à l'imprimante. Comme il s'agit d'une opération manuelle, le programme doit prévoir une pause qui laisse à l'opérateur le temps de l'effectuer. Le plus souvent, on définit cette pause en subordonnant la poursuite de l'exécution du programme à un accord explicite de l'opérateur.

Lorsqu'arrive le moment de l'opération manuelle, le programme exprime une demande



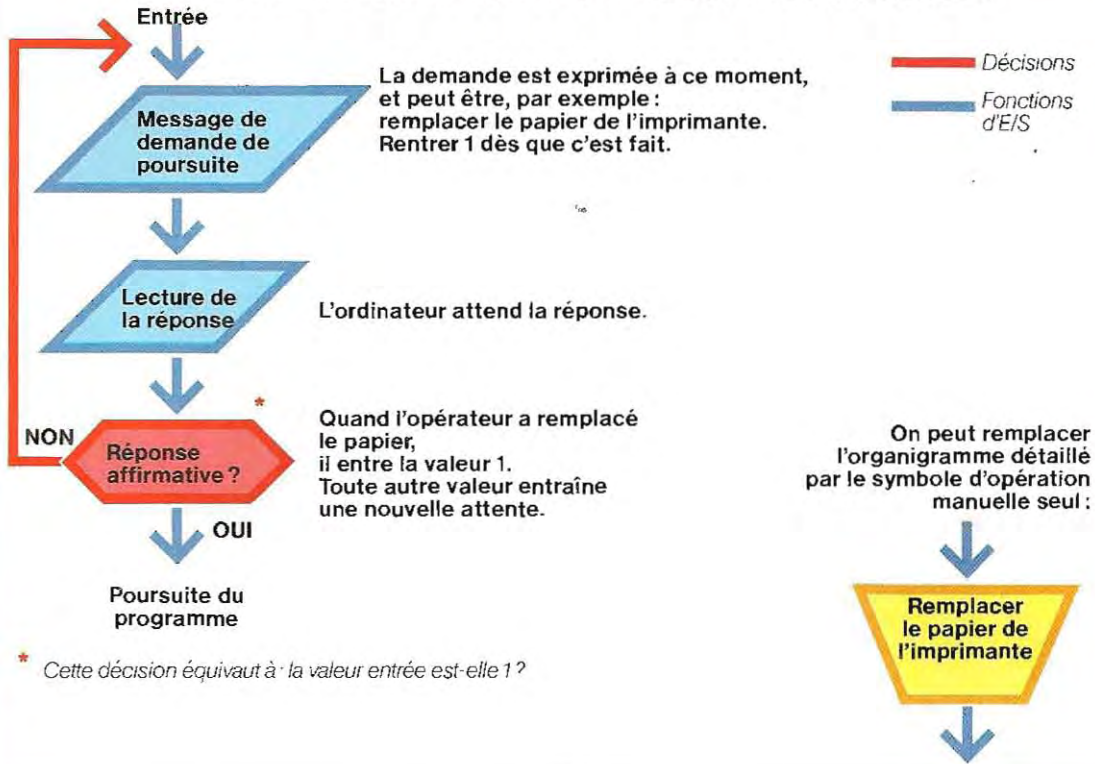
(oui = feu rouge, non = feu orange), l'action qui en découle est toujours l'attente.

Un programme peut s'accomplir si la machine a reçu toutes les données qu'elle doit traiter. L'introduction de ces données (par ex. à l'aide du clavier) est une opération d'entrée (input), et leur restitution par la machine une fois traitées (sur une imprimante, par exemple), une opération de sortie (Output).

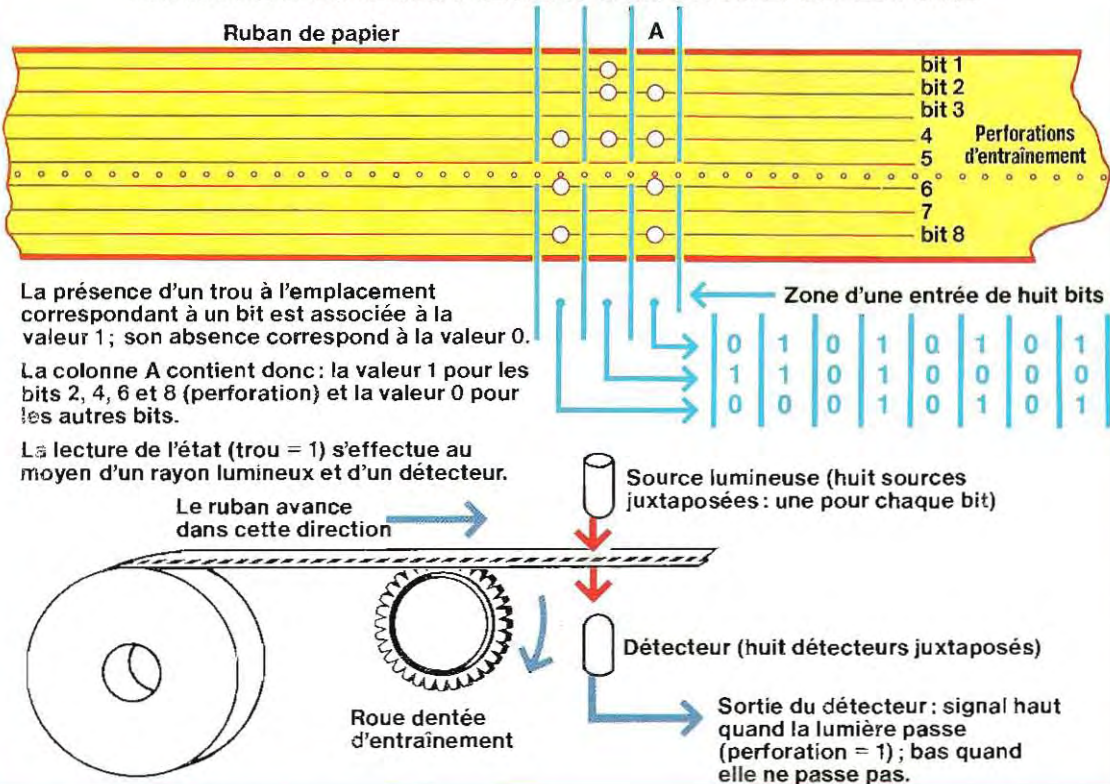
Le symbole graphique le plus couramment utilisé pour représenter une opération générale d'entrée-sortie (E/S) est le symbole **h** (voir p. 168). Le contenu du symbole indiquera s'il s'agit d'une entrée ou d'une sortie.

d'accord et attend la réponse; l'opérateur a alors le temps d'effectuer l'opération. Dès qu'il est prêt, il entre l'ordre de poursuivre. Le schéma en haut de la page 171 représente l'organigramme simplifié de cette procédure. Le symbole **g** indique toute opération auxiliaire, sans lien étroit avec le programme, tandis que le symbole **i** désigne l'émission ou la lecture d'une bande perforée. Ce support d'information permet l'enregistrement des données par des perforations (l'existence d'un trou correspond au niveau 1, voir schéma p. 171). Ce ruban de papier a été supplanté par les supports magnétiques.

## METHODE D'ATTENTE POUR LES OPERATIONS MANUELLES



## EXEMPLE D'ENREGISTREMENT SUR UN RUBAN PERFORE





**Utilisation de l'ordinateur pour l'établissement de plans : la machine fournit des données d'une extrême précision et des idées nouvelles en un temps record.**

Le symbole **l** représente une opération d'entrée-sortie qui nécessite l'édition d'un document. On l'utilise surtout pour attirer l'attention sur les opérations particulières d'impression comme l'impression des bulletins de paie. Le symbole **m** indique une introduction manuelle des données. Il convient d'utiliser la même logique que pour le symbole **f** (opération manuelle). Le symbole **n** désigne une bande magnétique. Ce support a désormais presque totalement remplacé le ruban perforé et sert à l'enregistrement des données auxquelles on ne recourt pas très souvent, le disque ou la disquette étant préférables en cas d'accès fréquent aux données.

Quant au symbole **p**, il désigne la mémoire et les opérations qui s'y rattachent, tandis que le symbole **q** représente les disques, c'est-à-dire les mémoires de masse les plus employées.

### Exemple d'application

Nous voilà donc en mesure d'établir l'organi-

gramme d'un problème donné. Nous compléterons par la suite l'illustration de cette méthode, mais nous pouvons, d'ores et déjà, étudier un exemple complet d'établissement d'un organigramme.

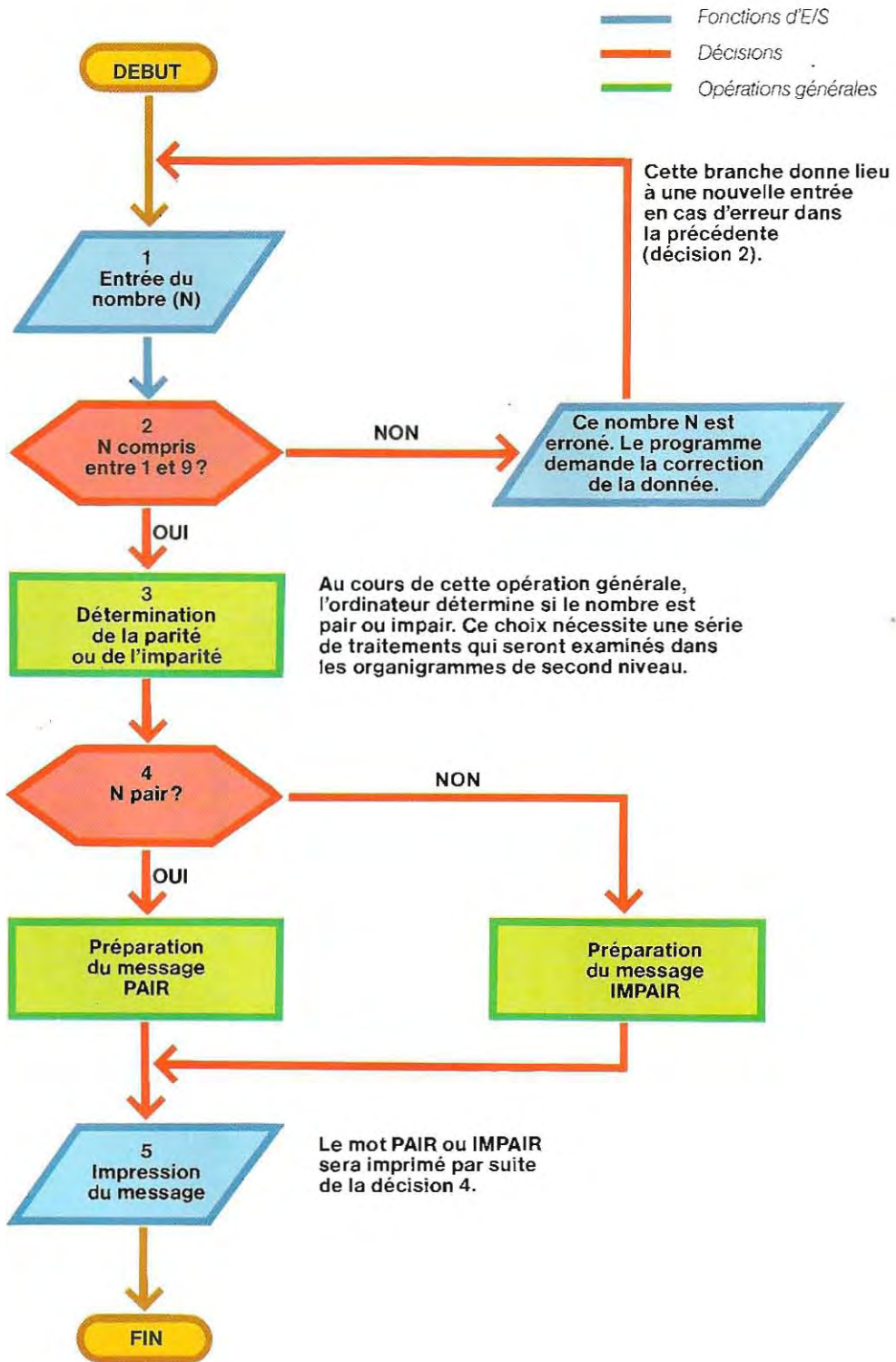
Prenons le problème suivant : lire un nombre  $N$  compris entre 1 et 9, et écrire s'il est pair ou impair.

Il faut commencer par décomposer le problème en un certain nombre d'actions plus simples. Ici on aura :

- 1 / lecture du nombre  $N$  (opération d'E/S) ;
- 2 / vérification :  $N$  est-il bien compris entre 1 et 9 (décision) ? ;
- 3 / détermination de la parité ou de l'imparité de  $N$  (action générale) ;
- 4 / choix de la réponse à fournir (décision) ;
- 5 / impression de la réponse (opération d'E/S).

L'ensemble des symboles correspondants

# EXEMPLE D'APPLICATION D'UN ORGANIGRAMME DE PREMIER NIVEAU



constitue l'organigramme de premier niveau illustré par le schéma de la page 173.

L'application de la méthode HIPO (Hierarchy + Input-Process-Output; ou en français HETS, Hiérarchie Entrée Traitement Sortie) au schéma du programme donne :

- entrée : N ;
- traitement : vérification de la validité de N (compris entre 1 et 9) et détermination de sa parité ou de son imparité ;
- sortie : message d'erreur si N n'est pas compris entre 1 et 9 ; sinon, message PAIR ou IMPAIR.

### Les boucles

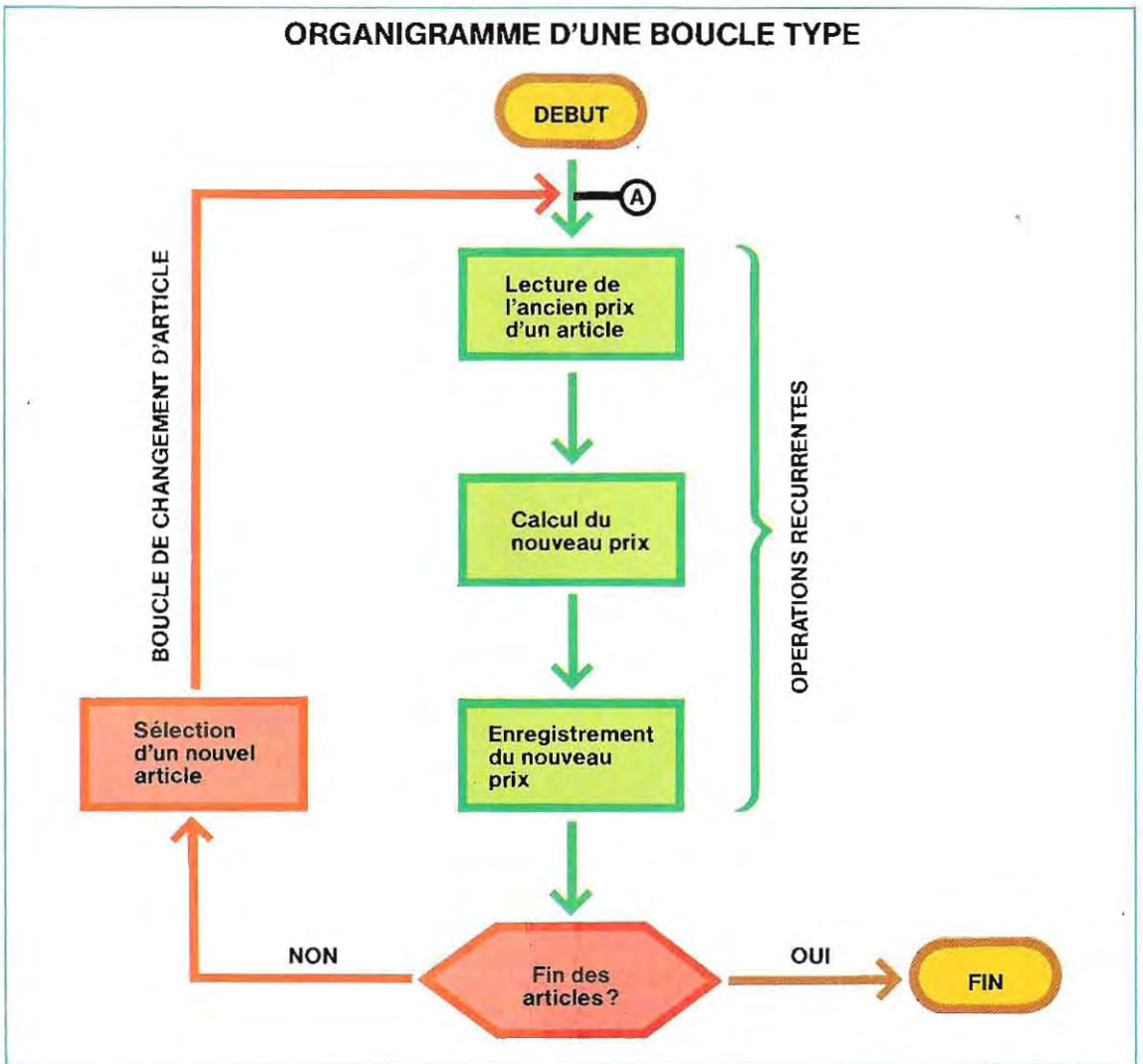
Il est tout particulièrement utile d'employer un système à microprocesseur pour effectuer des calculs récurrents portant sur différentes

données. On résout ce problème par l'emploi d'une **boucle** (loop en anglais), c'est-à-dire d'un programme ou d'une partie de programme capable, à la fin d'un traitement, de revenir à son propre début, de prélever une nouvelle donnée et d'y effectuer les mêmes opérations. Ce cycle (boucle) doit être répété jusqu'à épuisement des données.

Par exemple, à la suite d'une augmentation de 12 %, on veut connaître les nouveaux prix des articles d'un magasin. Pour chaque article, il faut effectuer le calcul récurrent suivant :

$$\text{nouveau prix} = \text{ancien prix} + \left( \frac{\text{ancien prix}}{100} \times 12 \right)$$

Le schéma ci-dessous, illustrant l'organigramme de premier niveau de ce calcul, fi-

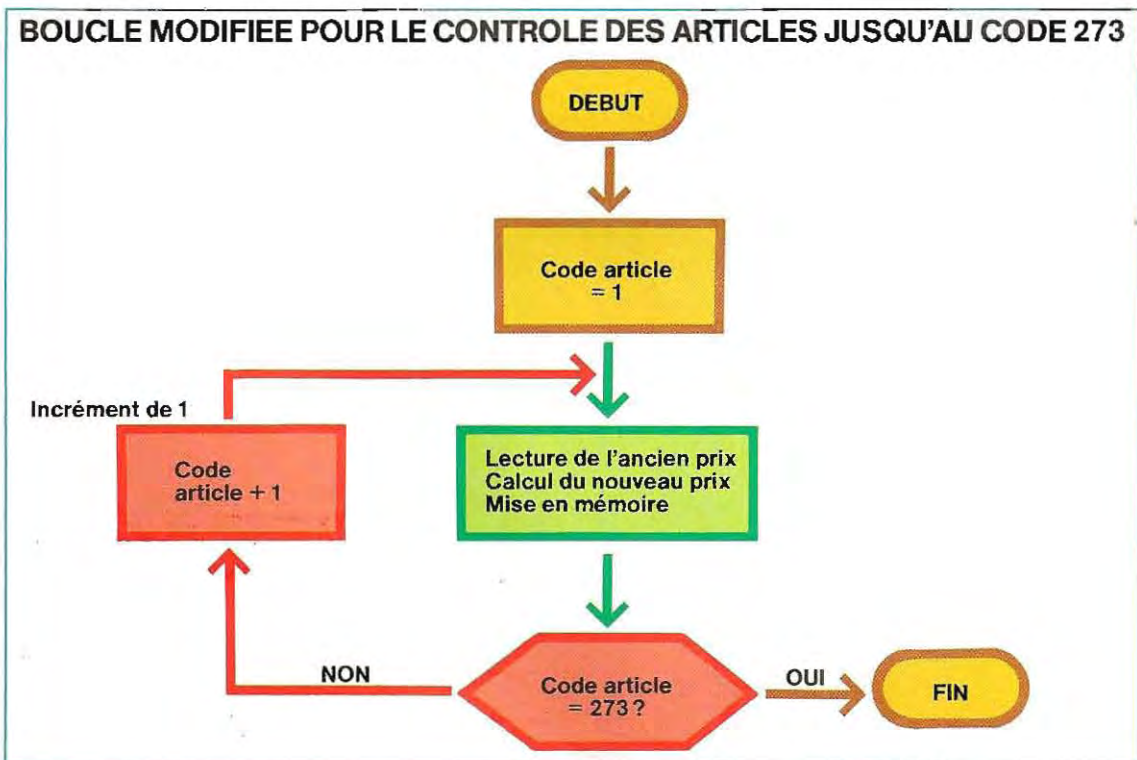




gure en rouge les parties constituant la boucle. Pour chaque article, le branchement, c'est-à-dire le retour au début du traitement, se fait au point A.

### Boucle à limites explicites

Si un magasin contient 273 articles différents codifiés selon une numérotation allant de 1 à 273, il faudra répéter la boucle 273 fois (une fois par article) en commençant par l'article numéro 1. Pour savoir si tous les articles ont été traités, il suffira de demander si l'on en est au numéro 273. Si ce n'est pas le cas, on pourra sélectionner un nouvel article en ajoutant simplement 1 au code de l'article précédent. L'organigramme représenté ci-dessous tient compte de ces modifications. Dans ce programme, le code article est un nombre



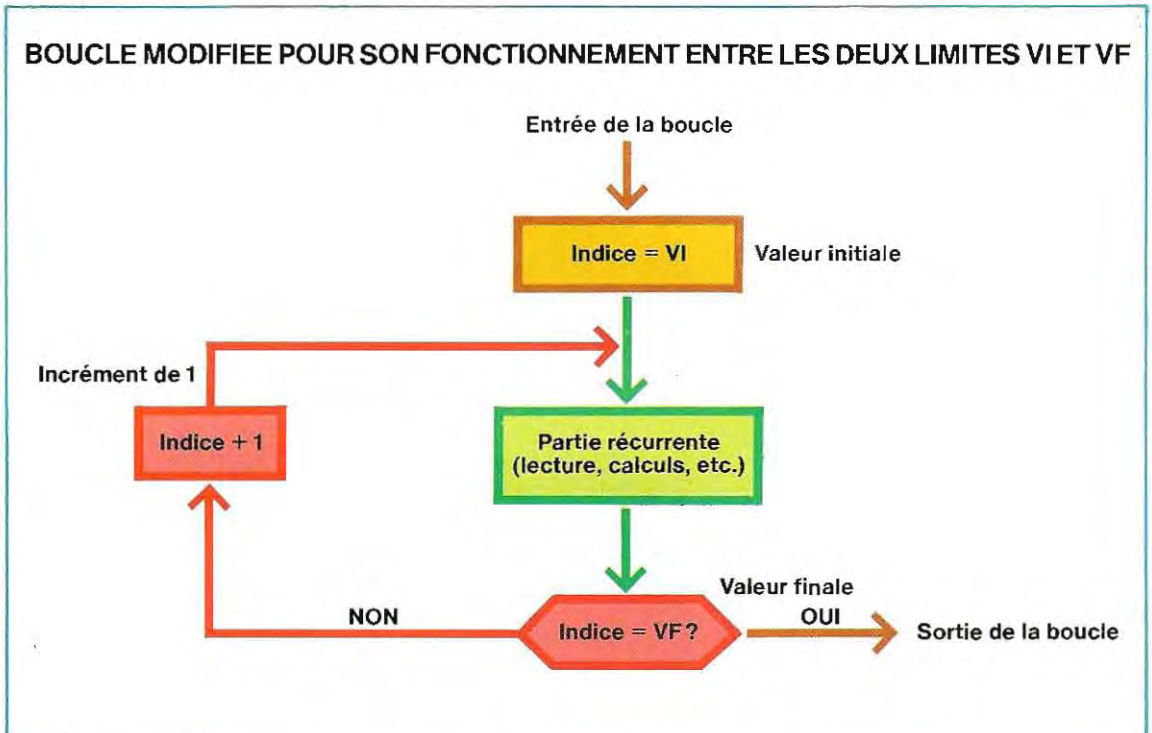
compris entre 1 et 273, et qui désigne l'article en cours de traitement. On l'appelle l'**indice de boucle**. La plupart des boucles sont indicées : elles se répètent de la valeur initiale à la valeur finale de leur indice (ici, 1 à 273).

### Boucle à limites paramétrées

Sur le schéma de cette page, les limites de la boucle, c'est-à-dire les valeurs initiale et fi-

nale de l'indice (1 et 273) sont exprimées sous une forme numérique explicite. On peut également les exprimer de façon indirecte en leur affectant un symbole.

Si l'on appelle **VI** (valeur initiale) et **VF** (valeur finale) ces deux limites, on peut représenter l'organigramme d'une boucle type par le schéma ci-dessous. La désignation symbolique des limites d'une boucle permet de les paramétrer. On dit qu'une partie de pro-



gramme est paramétrée lorsqu'au lieu de se répéter entre des valeurs numériques explicites, elle accepte en entrée des noms symboliques, les **paramètres**. Il faut alors, avant l'exécution des instructions paramétrées, déclarer la valeur numérique des paramètres utilisés. Cette méthode permet d'appliquer les mêmes instructions à des valeurs numériques différentes. Dans notre exemple, si l'on veut effectuer le calcul pour tous les articles, on donnera aux paramètres les valeurs :

- Valeur initiale = VI = 1
- Valeur finale = VF = 273.

Si, au contraire, on ne voulait effectuer le calcul que pour les articles allant des numéros

20 à 56, il faudrait insérer, avant la boucle, deux instructions d'affectation :

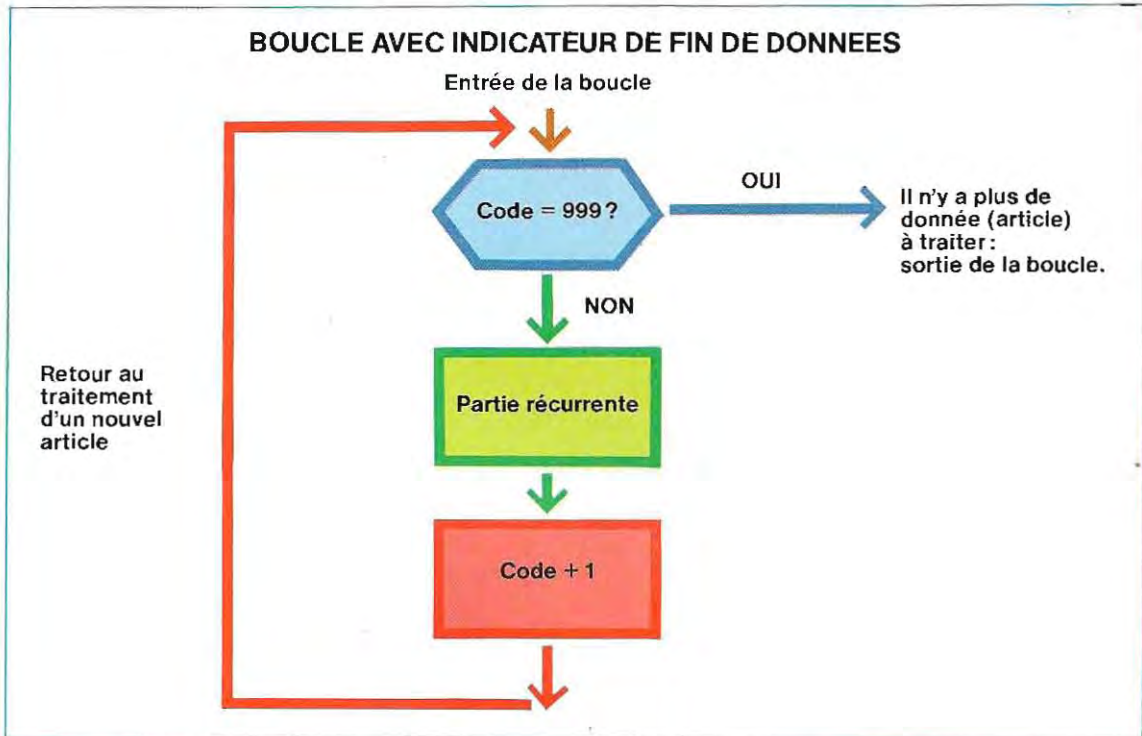
- VI = 20 (commencement au code 20)
- VF = 56 (fin au code 56).

### Boucle avec indicateur de fin de données

Les boucles que nous avons vues jusqu'ici, c'est-à-dire à limites explicites ou à limites paramétrées, ne s'achèvent qu'après le traitement de toutes les données. Elles présentent toutefois un inconvénient : on doit connaître à l'avance le nombre des données (273 dans l'exemple).

Or, on ne connaît pas ce nombre dans toutes les applications. Il faudra alors adopter un

type de boucle qui permette de déterminer la valeur finale. Pour cela, on écrit à la fin des données une valeur conventionnelle facile à reconnaître ; lorsque la boucle rencontre cette valeur, elle s'arrête. On écrit la valeur symbolique de **fin de données** au terme de la phase d'introduction des données. On peut aussi appeler cette valeur une **fin de fichier** (EOF, End of File) ou un **label de fin**. Reprenons l'exemple du magasin. On peut si-



gnaler la fin des données en inscrivant un code article inexistant : la rencontre de ce code signifiera que les données sont terminées et que la boucle peut être abandonnée. Si ce code fictif est 999, on obtient l'organigramme ci-dessus. On prend alors la décision de sortir ou non de la boucle avant la partie récurrente afin de ne pas effectuer de calculs sur des valeurs erronées. Le code de fin de fichier s'écrit en phase d'introduction des données et sa valeur ne doit pas être celle d'une donnée. Dans l'exemple du magasin, on doit, avant de fixer cette valeur, prévoir la possibilité de création de nouveaux articles. On modifie alors l'organigramme (schéma p. 178).

### Boucle à rupture de code

Ce type de boucle permet d'effectuer des

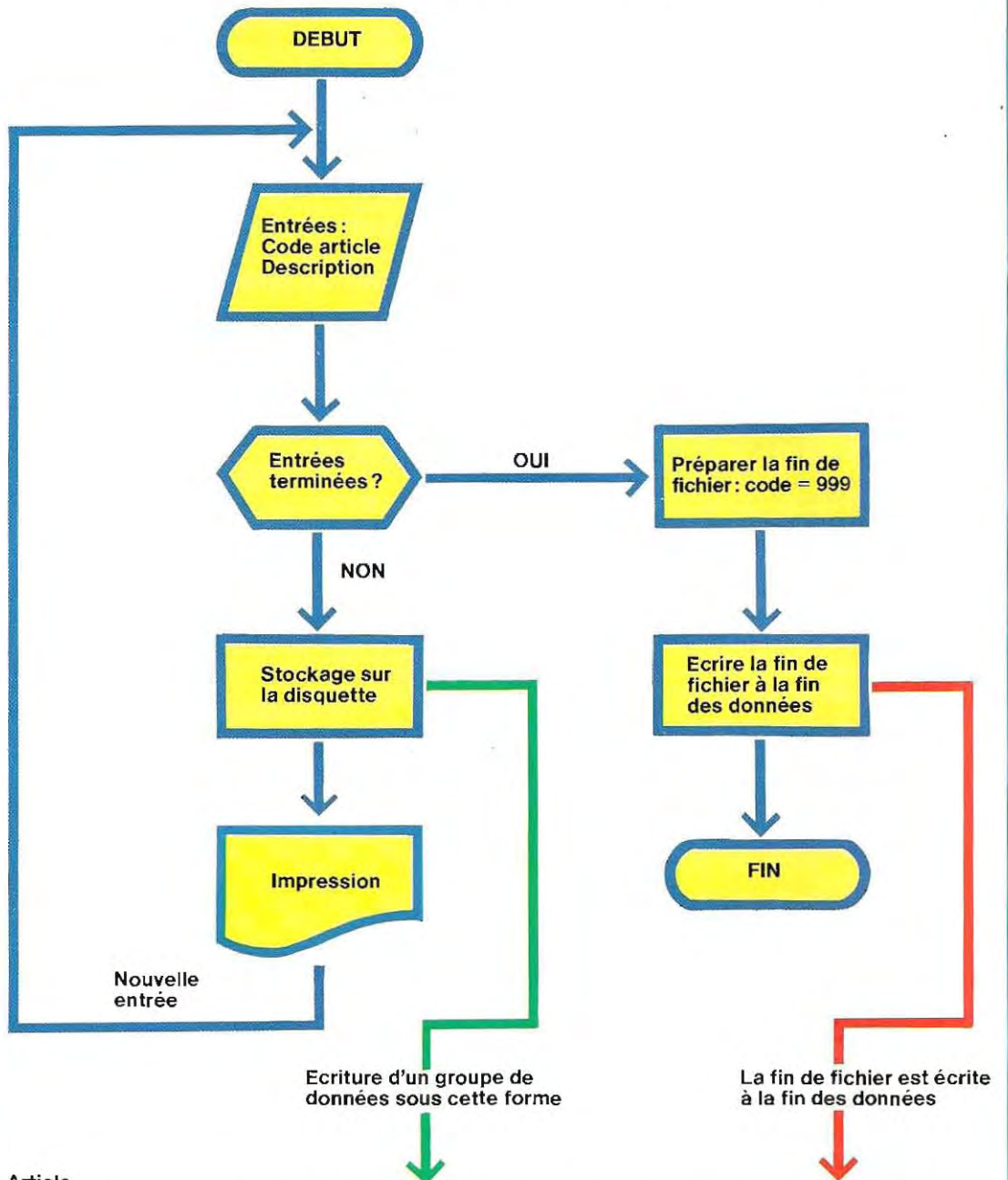
opérations itératives tant qu'un paramètre donné conserve sa valeur initiale. Quand ce paramètre change de valeur, on dit qu'il y a **rupture de code** et la boucle est interrompue.

On utilise couramment cette définition dans les langages Cobol et RPG\*, alors que le Basic ne dispose pas d'un nom particulier pour désigner les boucles de ce type.

La technique de la rupture de code est d'une grande utilité dans les traitements sélectifs (c'est-à-dire sur des données déterminées, celles dont le code a la même valeur). Le code désigne n'importe quel élément de la

\* Très proche du Cobol, le RPG (Report Program Generator), ou GAP (Générateur Automatique de Programmes), en français, est cependant moins complexe et n'est utilisé que sur certains ordinateurs (IBM).

## CREATION DES RUBRIQUES DU MAGASIN



Ecriture d'un groupe de données sous cette forme

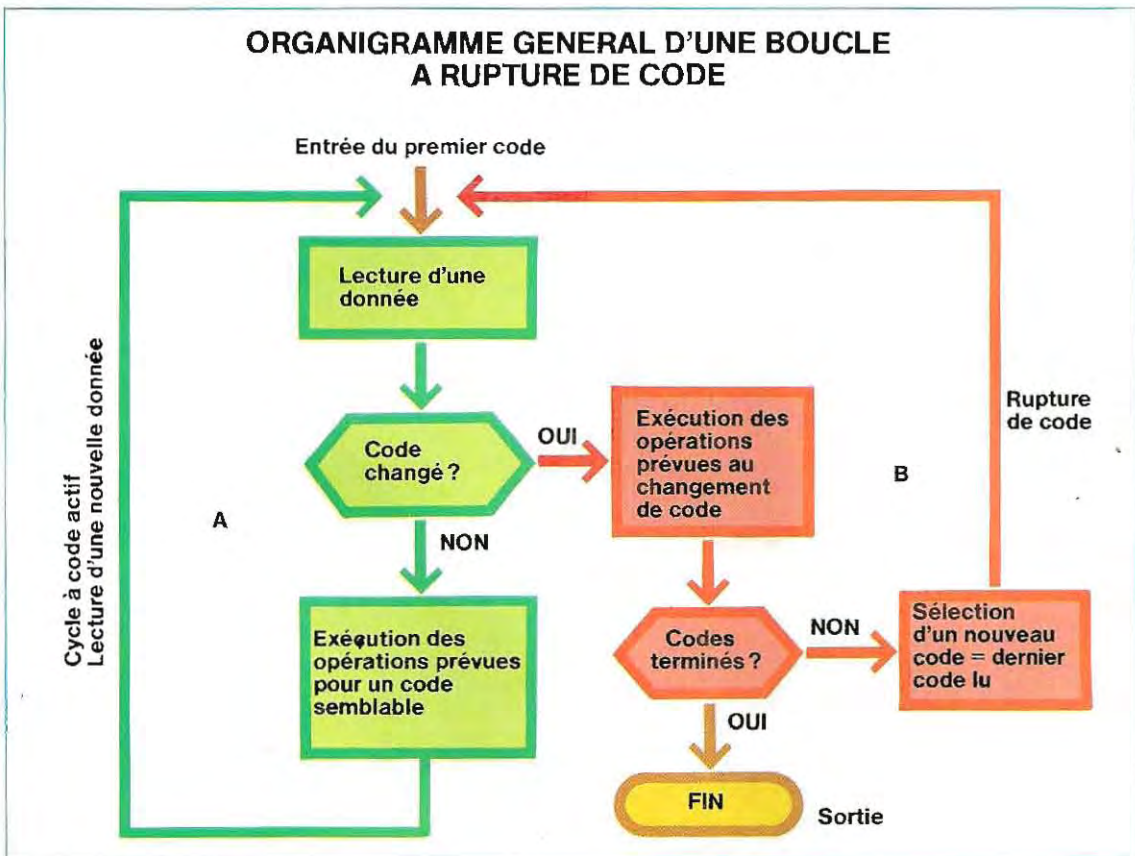
La fin de fichier est écrite à la fin des données

Article précédent	Zone contenant les données d'un article				Indicateur de fin de fichier
C O D E	Description	Quantité = 0	Coût = 0	9 9 9	

donnée. Ainsi, dans la gestion de notre magasin, nous pouvons prendre comme code la quantité en stock. Les instructions définies dans la boucle ne seront alors exécutées que pour les articles de stock égal. On peut voir sur cette page l'organigramme simplifié d'une boucle à rupture de code.

Les deux lignes de travail correspondent, la première aux opérations à accomplir sur les données de même code (bloc fonctionnel A), et la seconde aux opérations à effectuer en

### ORGANIGRAMME GENERAL D'UNE BOUCLE A RUPTURE DE CODE



cas de rupture de code (bloc fonctionnel B). Lors de la rupture de code, on doit vérifier, après l'exécution du programme, si tous les codes ont été traités. Si c'est le cas, le programme s'arrête. Sinon, le nouveau code est sélectionné et on reprend la boucle.

L'analyse des ventes effectuées par un groupe de représentants est un bon exemple de l'utilisation de cette méthode. Il s'agit de déterminer, au terme d'une période donnée, le chiffre des ventes réalisées par chacun d'entre eux. On prend comme code le numéro du représentant. L'opération (traitement) à effectuer sur

## Une révolution dans l'imprimerie

On a salué la photocomposition comme la plus grande révolution dans l'imprimerie depuis l'invention des caractères mobiles. Les procédés électroniques de reproduction permettent désormais, grâce aux transmissions par satellite, de publier simultanément journaux et revues dans le monde entier. Ainsi, on n'a plus besoin de déléguer les travaux d'impression à de grands ateliers qui doivent en sous-traiter le transport. Aujourd'hui, ce sont des petits établissements automatisés qui effectuent les différentes opérations de traitement des textes et leur transmission par ordinateur. Depuis un certain temps déjà, plusieurs hebdomadaires internationaux envoient, sur film, des pages entières prêtes à imprimer aux quatre coins du monde. L'un des plus grands quotidiens des Etats-Unis procède à la transmission par satellite d'exemplaires de son édition principale. Mais il ne s'agit là que de deux exemples d'une révolution qui a touché tous les secteurs de l'imprimerie, des journaux aux billets de banque.

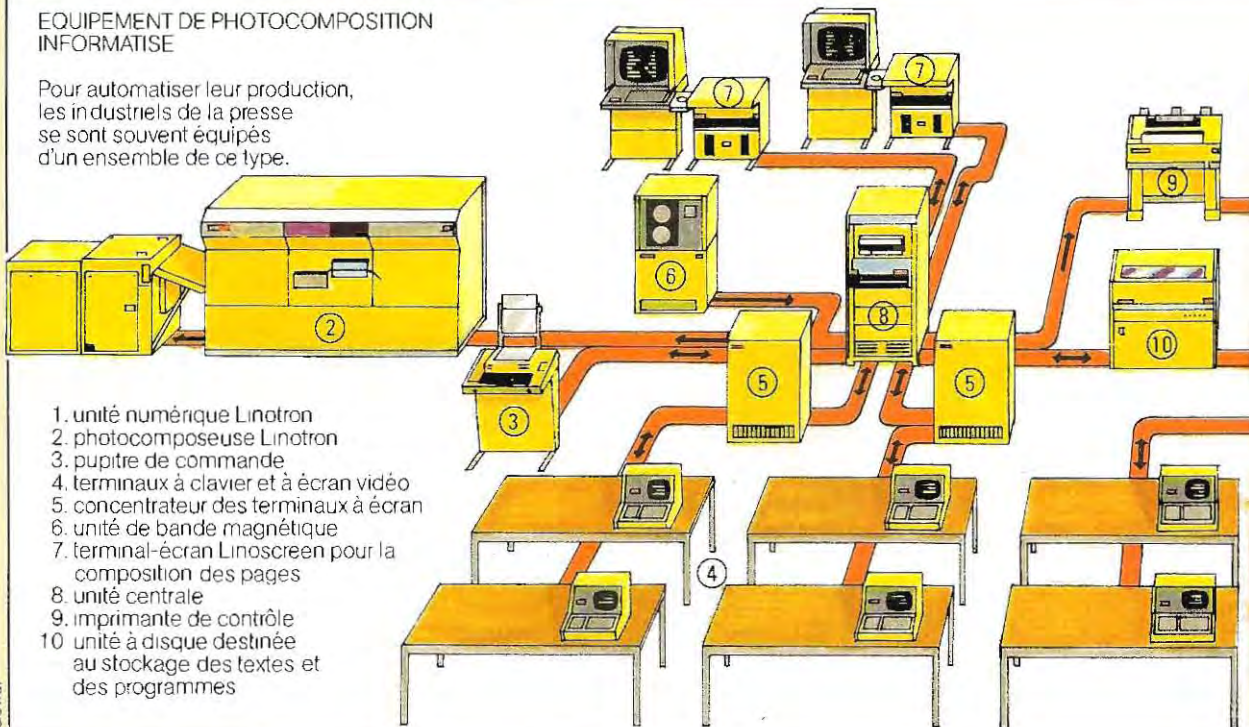
Les journaux et les revues d'actualité, dont l'utilisation est immédiate, doivent être publiés rapidement. Ils ont donc été parmi les premiers à bénéficier de la technologie électronique

moderne, recourant aux techniques les plus efficaces de manière à conserver leur compétitivité. C'est le cas, en particulier pour les grandes revues financières qui doivent parvenir rapidement à leurs lecteurs, aussi éloignés soient-ils. C'est pourquoi elles ont recours à la télétransmission pour s'assurer une diffusion à peu près simultanée vers tous les points du globe, tandis qu'elles se contentent des transports aériens pour la distribution régionale.

La publicité spécialisée, les informations boursières et les guides des spectacles, entre autres, sont autant de sujets qui se prêtent à une diffusion électronique par les systèmes de Télétex (vidéographie diffusée). Les services rendus par les ordinateurs (et tout particulièrement la lecture des informations sur leurs écrans) constituent dans certains domaines (les agences de compagnies aériennes par exemple) une inestimable amélioration de la diffusion des renseignements. Malheureusement, le Télétex n'a pas encore remporté un succès commercial suffisant pour qu'on puisse généraliser la diffusion télématique des journaux. Selon certains experts, cela n'est pas dû seulement à la résistance naturelle du public aux changements, mais aussi à la difficulté réelle qu'il y a

### EQUIPEMENT DE PHOTOCOMPOSITION INFORMATISE

Pour automatiser leur production, les industriels de la presse se sont souvent équipés d'un ensemble de ce type.



1. unité numérique Linotron
2. photocomposeuse Linotron
3. pupitre de commande
4. terminaux à clavier et à écran vidéo
5. concentrateur des terminaux à écran
6. unité de bande magnétique
7. terminal-écran Linoscreen pour la composition des pages
8. unité centrale
9. imprimante de contrôle
10. unité à disque destinée au stockage des textes et des programmes

à s'adapter aux mutations qui se succèdent à un rythme extrêmement rapide.

Malgré sa mécanisation, l'imprimerie a fonctionné selon des principes à peu près inchangés pendant plus de quatre siècles. La composition se faisait lettre par lettre, avec des caractères en alliage de plomb, ou en bois pour les affiches de grandes dimensions. Le métal offrait cependant l'avantage de pouvoir être fondu et réutilisé.

On employait alors des planches à imprimer planes, et c'est le quotidien londonien «Times» qui bouleversa la technique de transfert du texte sur le papier en introduisant la première rotative, machine à imprimer munie de plaques courbes enroulées à la surface d'un cylindre.

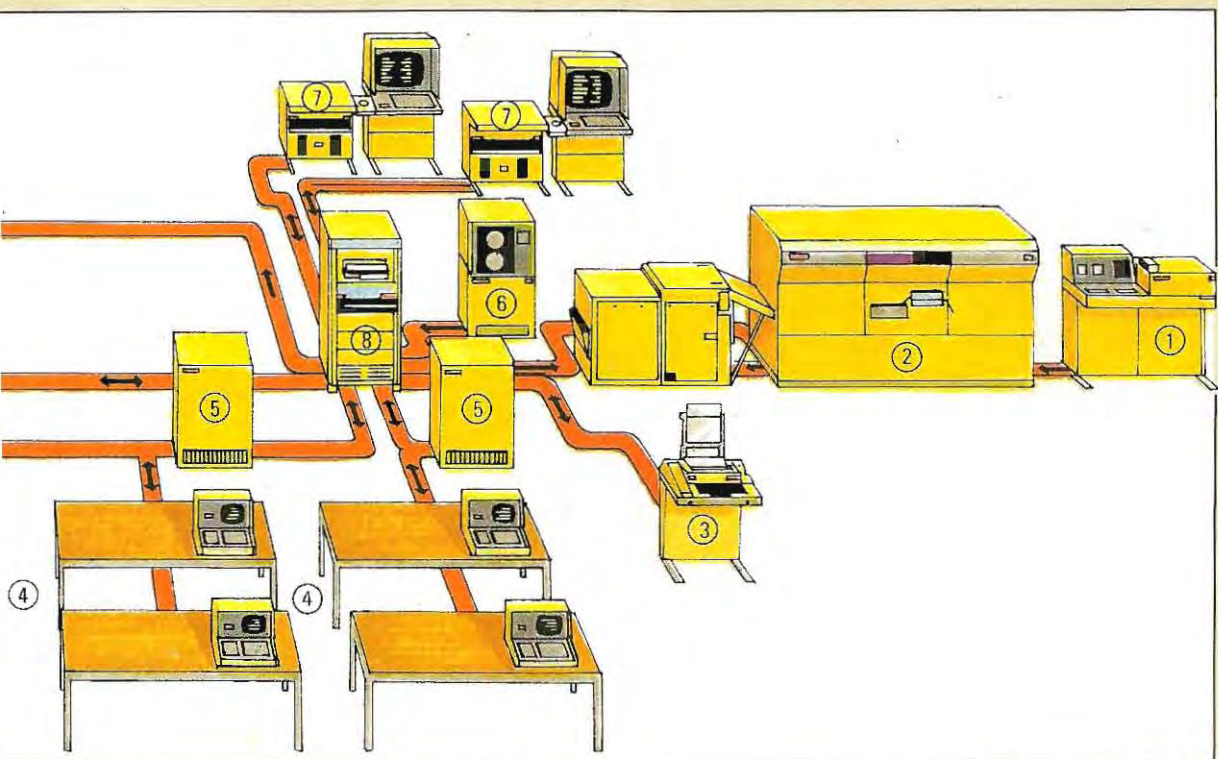
Aujourd'hui, tous les systèmes mécaniques de composition, de reproduction et d'impression sont peu à peu supplantés par les procédés électroniques. La composition se fait désormais essentiellement par photographie. L'opérateur dispose d'un clavier et d'un écran, et tape le texte qui, à sa demande, est automatiquement traité par l'ordinateur et enregistré sur une bande ou un disque magnétique. L'ordinateur effectue les espacements entre les lettres et les mots ainsi que la

mise en place du texte dans la page, comme le veut l'opérateur. Les signaux enregistrés magnétiquement sont ensuite communiqués à une photocomposeuse.

Les premières photocomposeuses ont été de simples adaptations des machines munies d'un clavier et qui servaient à composer les caractères de plomb. Aujourd'hui, les photocomposeuses de la « quatrième génération » fonctionnent à des vitesses si extraordinaires qu'une seule suffit à traiter les textes provenant de centaines de clavistes. La reproduction des images bénéficie également des améliorations apportées au matériel électronique.

On estime qu'aujourd'hui, près de 50% des reproductions en couleurs sont faites par analyse électronique. Des analyseurs décomposent les couleurs de l'original en deux, trois ou quatre « sélections » d'une seule couleur reproduites séparément sur des films qui sont ensuite superposés et imprimés de façon à restituer une image en couleurs semblable à l'original.

La reproduction électronique des images et des textes en un seul processus, ou « pre-press », doit son apparition à l'industrie de la presse quotidienne.





Sci-Tex Europe S.A.

**Grâce au système Sci-Tex, la mise en page apparaît directement sur l'écran.**

Souvent annoncée, la réunion de ces deux éléments essentiels dans un produit imprimé n'a abouti que récemment. Le texte et les images sont transformés en signaux électroniques, puis introduits sous cette forme dans un ordinateur. Les informations sont alors manipulées par l'intermédiaire d'un clavier et mémorisées. Le clavier peut facilement apporter modifications et corrections car il suit sur l'écran les corrections au fur et à mesure qu'il les tape sur le clavier. Puis il donne à l'ordinateur les instructions nécessaires pour qu'il enregistre la nouvelle version.

Au moment de l'impression, les signaux conservés dans la mémoire de masse de l'ordinateur sont transférés d'abord sur un film, puis sur des plaques sensibles et, enfin, sur du papier. Les procédés purement électroniques coûtent cher. C'est pourquoi on a récemment mis au point une méthode simplifiée de reproduction électronique en couleurs qui convient surtout aux petites entreprises.

Les systèmes électroniques n'intéressent pas que les éditeurs de journaux et de revues, mais aussi tous ceux qui publient des catalogues de vente par correspondance et ont également besoin d'une méthode très rapide pour imprimer des mises à jour de dernière minute.

Si les ordinateurs travaillent à très grande vitesse, ils exigent néanmoins un temps assez important de programmation et de codification des informations. Dès que le travail de mise en

page est compliqué, de nombreuses personnes, surtout si elles sont adroites et minutieuses, préfèrent travailler directement sur le film. En effet, facile à manipuler, le film offre en outre une image directe de ce que l'on fait.

L'héliogravure, procédé d'impression basé sur un cylindre gravé à l'acide, offre la meilleure qualité possible sur du papier léger et bon marché. On l'emploie dans les revues et les catalogues de grande diffusion. Depuis l'adoption des techniques électroniques les plus récentes, la phase de gravure des cylindres fait partie intégrante de la conversion directe d'une image en couleurs en une image à imprimer. Or, la lenteur du processus de gravure et son coût élevé constituent les deux principaux inconvénients de ce procédé.

Pour effectuer la gravure au laser, on recouvre les cylindres d'un revêtement de plastique sur lequel un rayon laser commandé par un analyseur de couleurs ou par d'autres unités grave l'image à imprimer. Cette méthode permet d'espérer une diminution des coûts. Cependant, et notamment dans les établissements les plus petits, la plus grande partie de la production sur cylindre se fait encore par gravure chimique à l'acide sous contrôle électronique.

Dans l'imprimerie américaine Harris à mouvement automatique, l'impression se fait véritablement en continu. Tandis que deux unités impriment des parties d'un livre en une ou deux couleurs, on prépare deux autres unités d'impression pour la partie suivante. Un microprocesseur augmente la vitesse des unités de préparation jusqu'au maximum tandis qu'il arrête automatiquement les autres unités d'impression dès que le nombre d'exemplaires prévu est imprimé.

On peut aujourd'hui panacher les techniques d'impression. C'est ce que prouve le succès du procédé Néosan selon lequel on peut utiliser des typons (films offset) respectant une angulation de trame compatible avec la préparation des cylindres pour l'impression en héliogravure.

Les mêmes films de sélection des couleurs peuvent ainsi servir à l'impression par l'un ou l'autre procédé.

C'est cette grande souplesse de fonctionnement qui peut donner à la presse toutes les chances de faire face aux exigences modernes de communication rapide des informations.



## Solutions du test 4

1 / b

2 / a : l'instruction 106 contient le mot RETOUR qui déclenche le mécanisme de retour à l'instruction venant après l'instruction de saut. Après l'instruction 106, le programme reprend donc à l'instruction 20.

b : dans ce cas, l'instruction RETOUR est absente ; le programme continue donc en série à partir de l'instruction 106 : il exécute les instructions 107 et suivantes.

3 / b : UAL est l'abréviation d'unité arithmétique et logique, qui est, comme son nom l'indique, destinée à effectuer des calculs.

4 / a

5 / c

6 / L'erreur réside dans la présence du mot RETOUR dans l'instruction 33. Ce mot ne doit être utilisé que pour le retour au point de débranchement après un saut. Or, ce programme ne comporte aucun saut qui conduise à la ligne 33. Inversement, le mot RETOUR est justifié dans l'instruction 57 puisqu'il y a un saut (instruction 32).

7 / b : l'Interruption permet de suspendre le traitement en cours afin d'utiliser les ressources du système pour un service demandé par l'extérieur.

8 / a : faux. Les EPROM sont destinées à stocker des programmes. On ne peut les effacer ou y écrire de nouveau qu'au moyen d'appareils spéciaux et elles ne peuvent donc pas servir à mettre des données en mémoire.

b : faux. L'unité de disque peut contenir de 250 000 à plusieurs millions de caractères (jusqu'à des centaines de millions pour les plus grandes). Le chiffre 64 000 est de l'ordre de grandeur de la mémoire d'un micro-ordinateur.

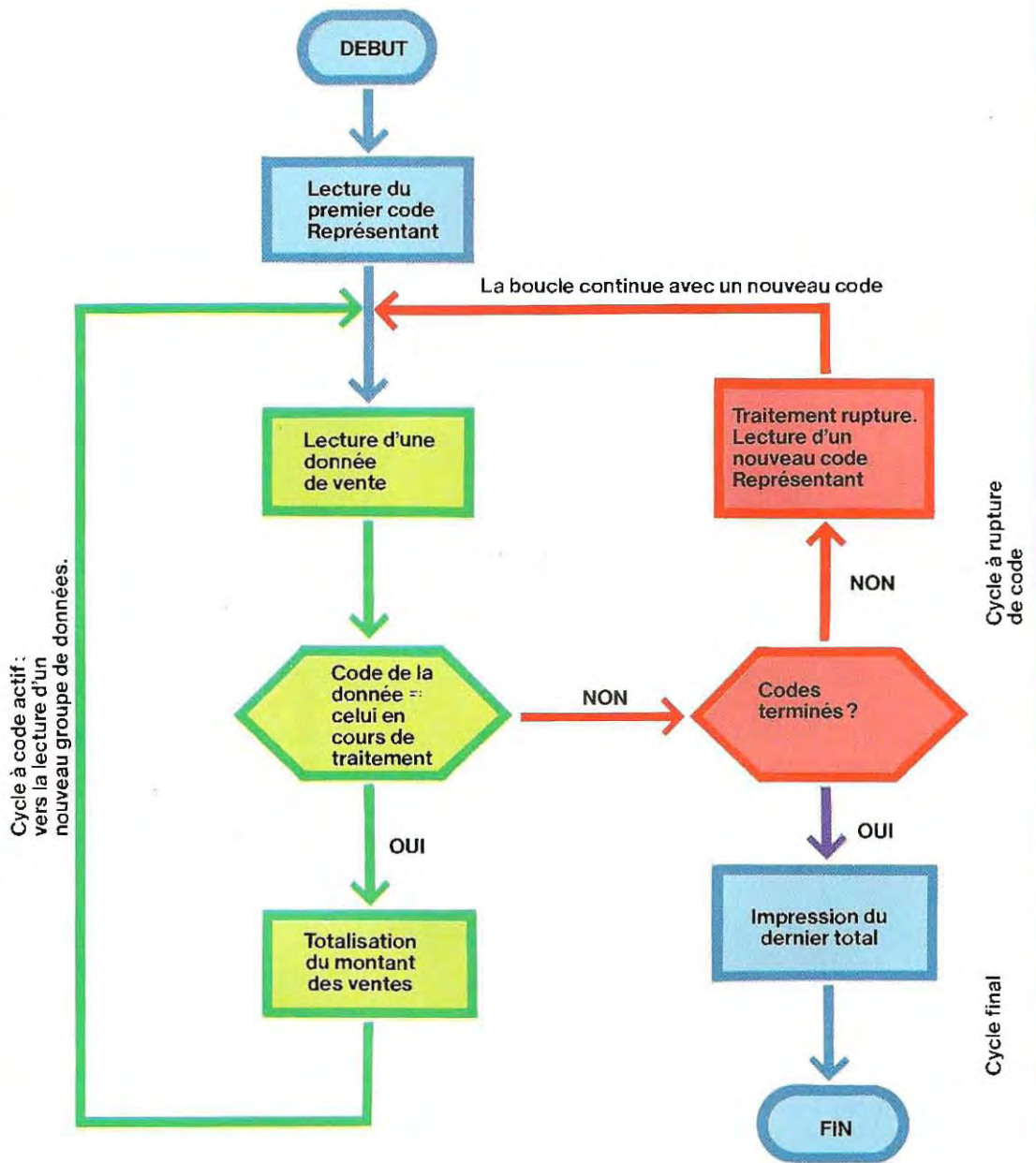
c : vrai. Il faut un équipement spécial pour programmer une ROM.

d : faux. L'UC (unité centrale de traitement) met en mémoire ce qui est en cours d'exécution, et se consacre à la nouvelle activité. Ensuite, elle reprend le déroulement normal du programme.

e : cette affirmation est partiellement vraie. Normalement, l'écran ne génère pas d'interruption, alors que cela fait partie du fonctionnement normal du clavier.

f : vrai. Le DMA (accès direct mémoire) est le seul moyen d'accélérer la vitesse de transfert, étant donné le nombre élevé de données échangées avec ce périphérique.

# ORGANIGRAMME D'UNE BOUCLE A RUPTURE DE CODE : EXEMPLE D'APPLICATION



Format des ventes

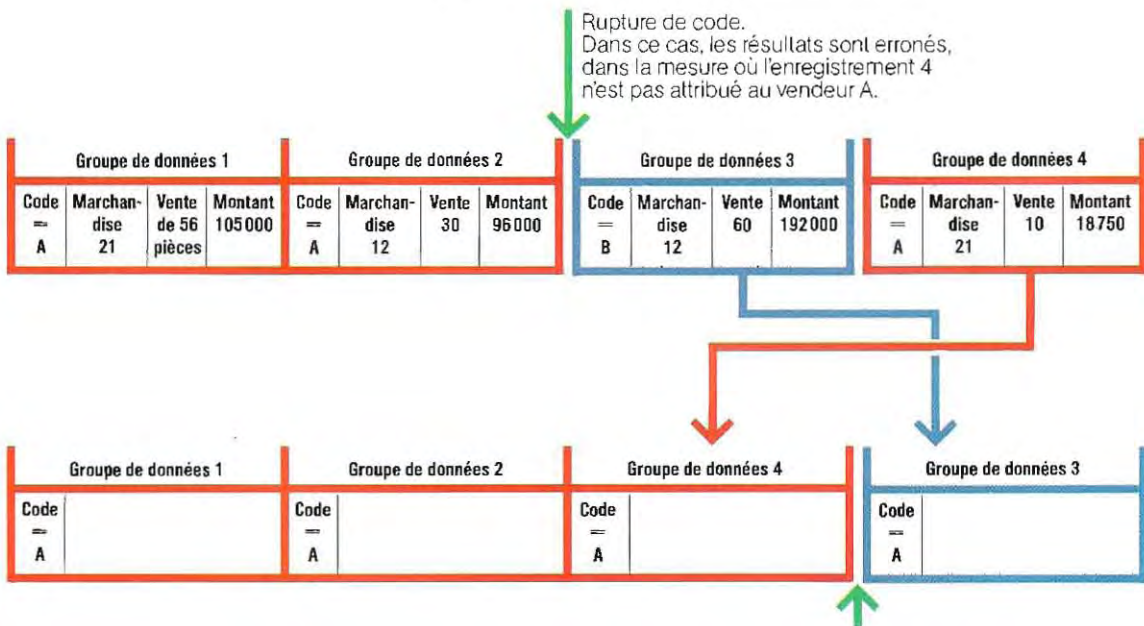
Code Représentant	Code Marchandise	Quantité vendue	Montant	Code Représentant	Code Marchandise
← Groupe des données concernant une vente →					

ce code actif consiste à fournir la somme des ventes réalisées. L'opération à effectuer à la rupture de code est l'impression du total.

On voit page 184 l'organigramme de ce problème, ainsi qu'un exemple de représentation des données concernant les ventes.

L'exécution d'une rupture de code nécessite un classement des données sur le code par ordre croissant ou décroissant sous peine de fausser les résultats. Supposons qu'il n'y ait que deux représentants (codés A et B), et que les informations sur les ventes aient été introduites dans le désordre comme il apparaît sur le schéma ci-dessous. La rupture de code (de A à B) n'indiquerait alors pas la fin des données sur le premier vendeur et le total des ventes serait donc faux puisqu'il ne tiendrait pas compte du contenu de l'enregistrement 4. En effet, à la deuxième entrée dans la boucle, après la rupture de code A, seules les données (informations) concernant B seront prises en compte et l'enregistrement 4 ne sera pas traité. Il convient donc, avant d'exécuter la boucle, de hiérarchiser les données, par exemple en ordre croissant. Les enregistrements 3 et 4 seront alors inversés et, à la rupture de code, on pourra totaliser toutes les ventes d'un même vendeur (A).

### RUPTURE DE CODE



Rupture de code.  
Dans ce cas, les résultats sont justes :  
à la rupture de code, on est sûr d'avoir  
traité toutes les données concernant le vendeur A



Marka

Opérateurs au travail sur des terminaux de gros ordinateurs.

**En résumé, les principaux types de boucle sont les suivants :**

- **Boucle à limites explicites :** les valeurs initiale et finale sont indiquées dans la boucle sous forme numérique.
- **Boucle paramétrée :** les valeurs initiale et finale sont désignées par des noms symboliques auxquels il faut attribuer une valeur numérique avant exécution. Cette méthode permet d'utiliser la boucle à plusieurs endroits du programme.
- **Boucle à indicateur de fin de données :** la valeur initiale est définie normalement, tandis que la valeur finale est déduite de la reconnaissance d'un code particulier.
- **Boucle à rupture de code :** elle traite les enregistrements du premier au dernier et comporte deux branches : l'une pour effectuer les opérations sur un même code, l'autre les traitements de fin de code (à la rupture). Il est donc indispensable que les données soient classées.

### Les boucles à pas différent de 1

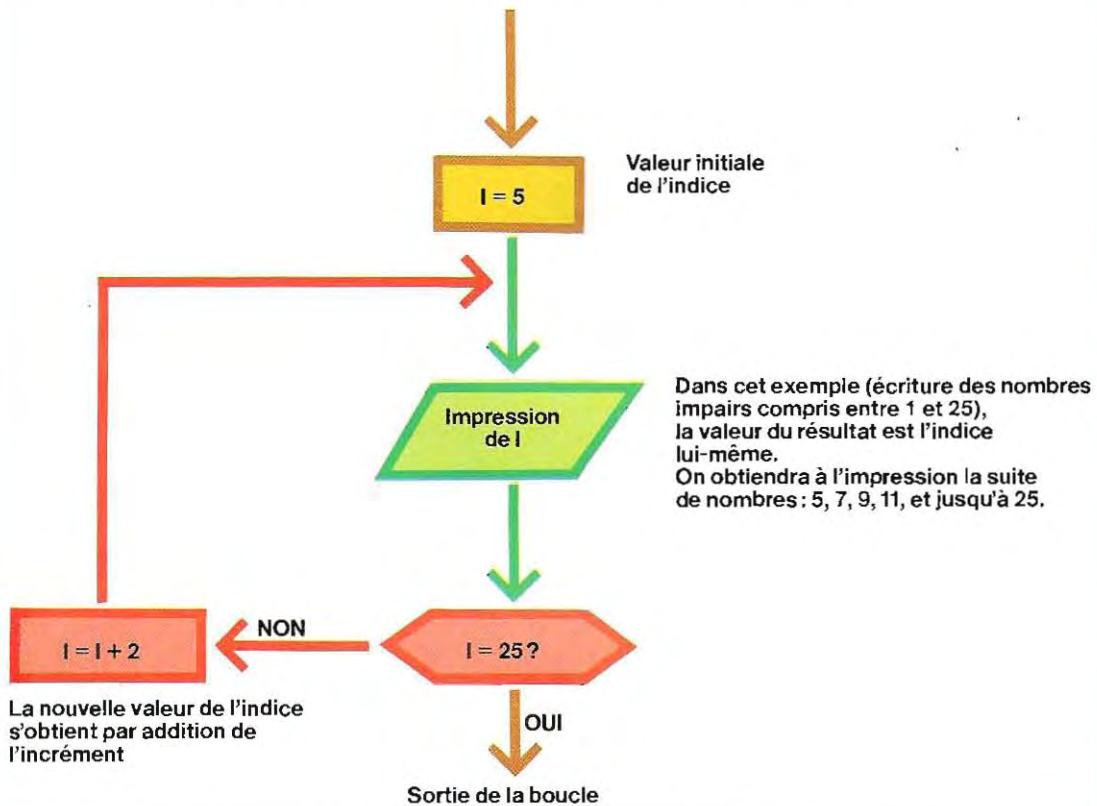
L'indice d'une boucle, c'est-à-dire la variable qui est incrémentée après chaque passage par la partie récurrente (voir schéma en haut de la page 187), n'a pas forcément un incrément de 1, ce qui permet d'opérer une sélection parmi les données.

Supposons qu'on veuille écrire tous les nombres impairs compris entre 5 et 25 : on choisira 5 comme valeur initiale, 25 comme valeur finale et 2 comme **pas** (step en anglais), c'est-à-dire comme incrément. Au premier passage, l'indice aura la valeur 5, celle du premier nombre à imprimer. Au deuxième passage, l'indice sera  $5 + 2 = 7$ , et ainsi de suite jusqu'à 25 (voir l'organigramme de la page 187, en haut).

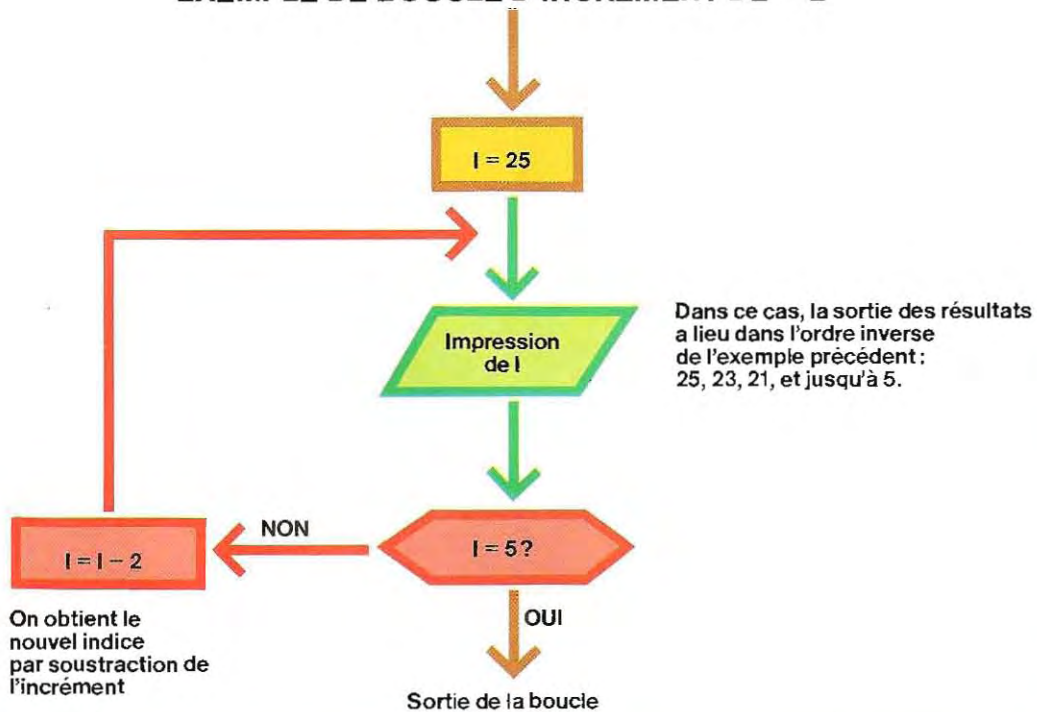
L'incrément peut aussi être négatif : la valeur initiale est alors supérieure à la valeur finale. Ainsi, dans notre exemple, en commençant à 25 pour terminer à 5, avec un pas de  $-2$ , la liste aurait été composée des mêmes nombres mais en sens inverse (25, 23, 21..., 5). Le second schéma de la page 187 figure l'organigramme de cette boucle.

Imaginons, comme application, que nous avons douze noms à imprimer quatre par quatre, sur trois lignes. On voit, en haut de la page 188, le format d'impression désiré. Les données à imprimer (noms de 1 à 12) sont enregistrées séquentiellement dans une zone-mémoire

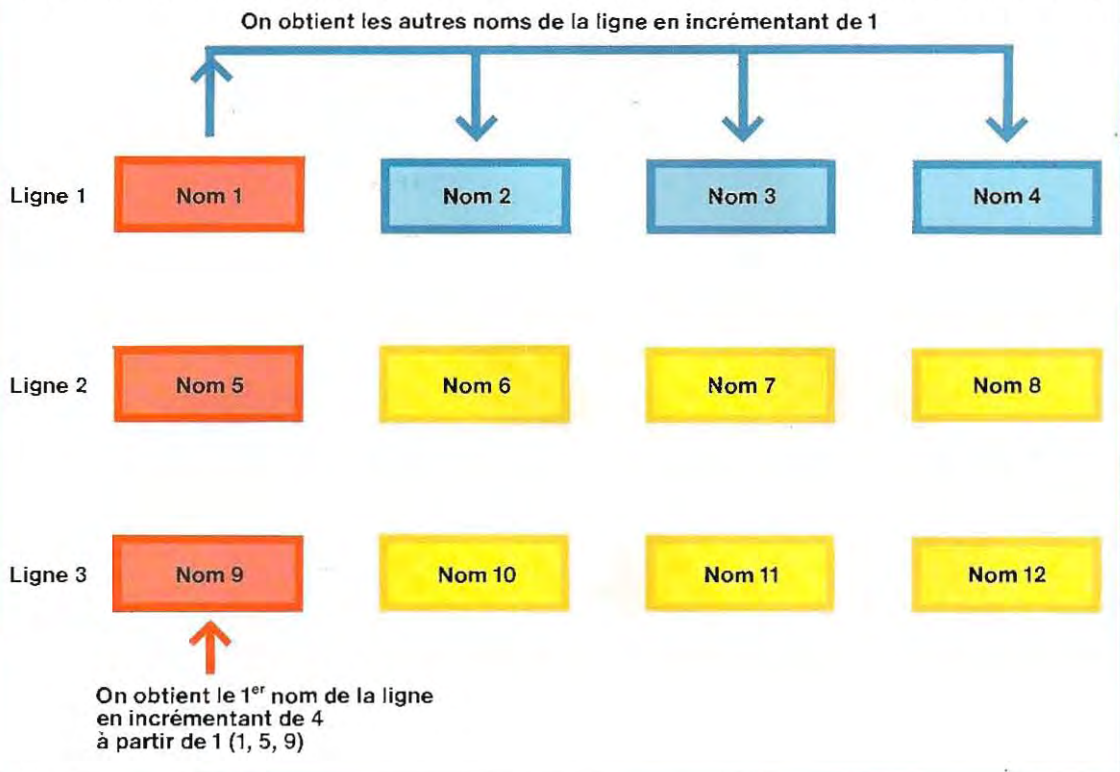
## EXEMPLE DE BOUCLE D'INCREMENT DE 2



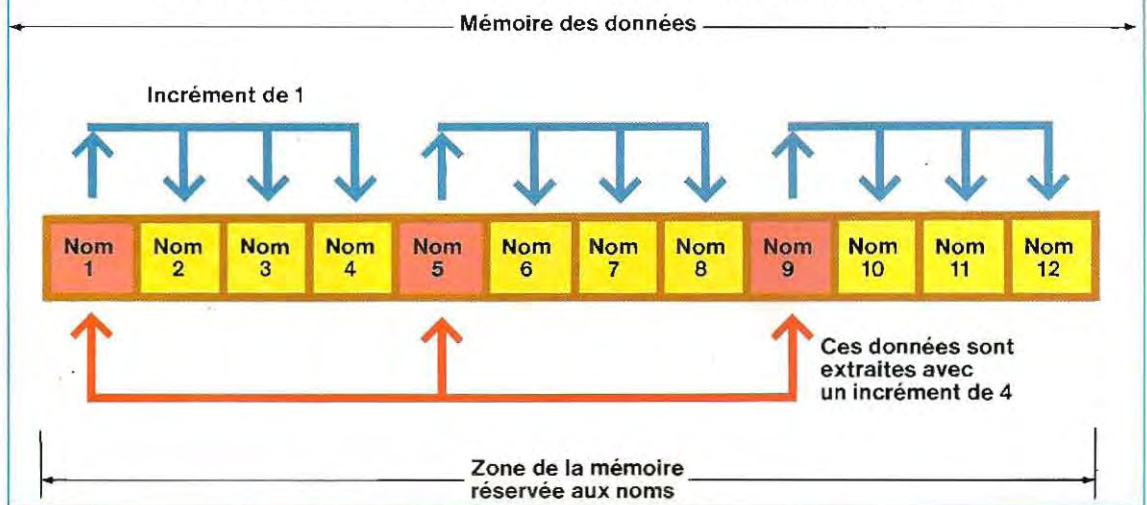
## EXEMPLE DE BOUCLE D'INCREMENT DE - 2



## FORMAT D'IMPRESSION DANS LEQUEL ON VEUT OBTENIR LES DONNEES



## POSITIONS OCCUPEES PAR LES DONNEES EN MEMOIRE



(voir schéma ci-dessus).

Pour obtenir le format désiré, il faut procéder comme suit :

- a** / sélection de la donnée de début de ligne : respectivement nom 1, nom 5, nom 9 pour les lignes 1, 2 et 3 ;

**b** / impression de la donnée sélectionnée (noms 1, 5 et 9), et des trois données suivantes sur la même ligne ;

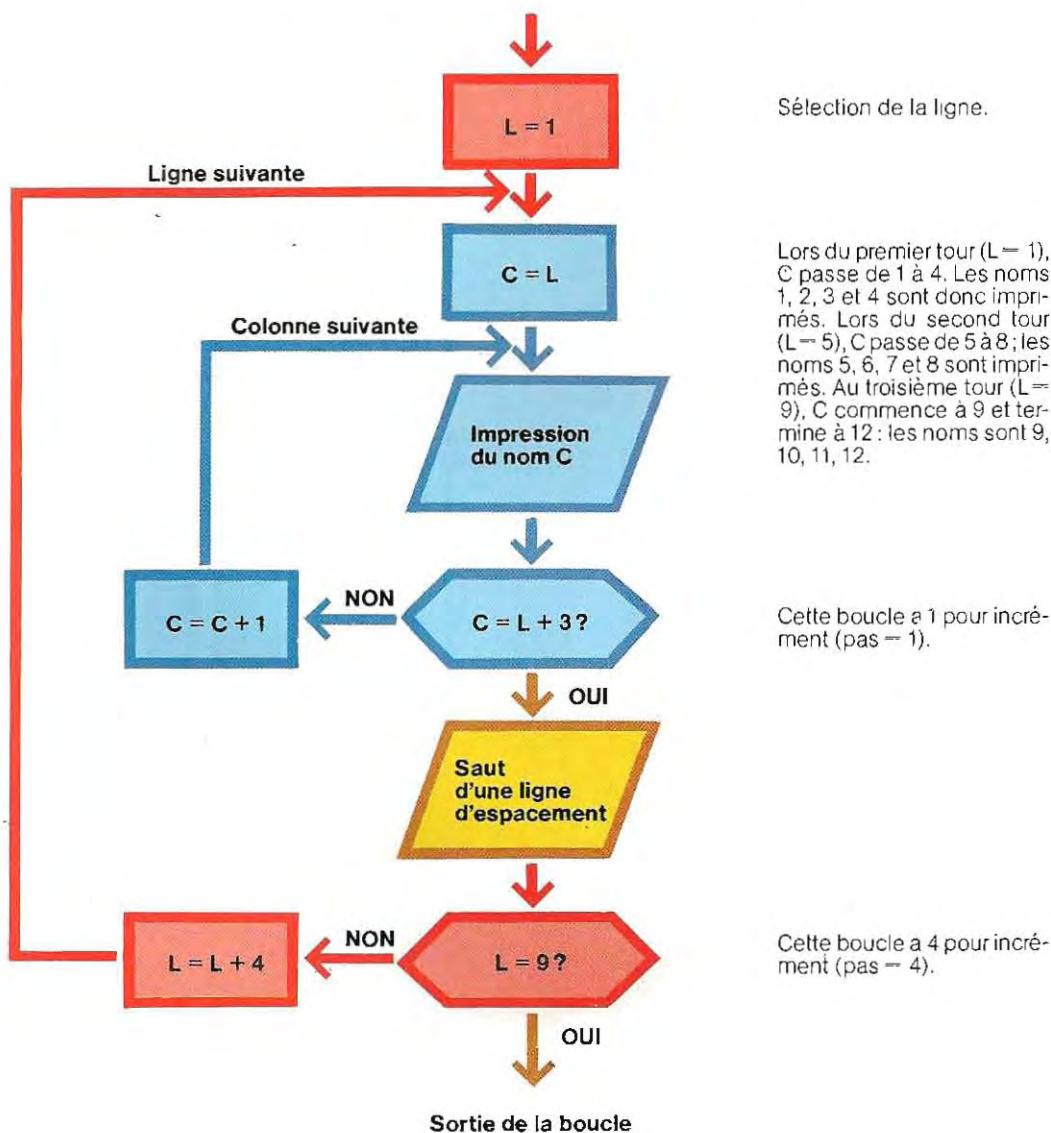
**c** / saut d'une ligne pour aérer le listing.

On peut résoudre le point **a** par une boucle ayant une valeur initiale = 1, une valeur fi-

nale = 9 et un pas = 4. Le point **b** constitue une deuxième boucle dont la valeur initiale est l'indice de la première (1, 5, 9) et la valeur finale ce même indice + 3 (ainsi, pour la première ligne, VI est 1 et VF est 4). Cette boucle a un pas de 1.

Pour exécuter le point **c**, il faut envoyer la commande de saut de ligne (LF, voir code ASCII), entre deux impressions. Le schéma ci-dessous en représente l'organigramme.

### ORGANIGRAMME DE L'IMPRESSION DE 12 DONNEES (NOMS) REPARTIES EN 3 LIGNES DE 4 NOMS CHACUNE



### Sortie forcée d'une boucle

Les boucles sont très employées dans la recherche des données. Supposons que nous avons enregistré les noms et les numéros de téléphone de nos amis et que nous comptons 200 données (nom + numéro). Nous voulons savoir quel est le numéro de M. Martin. Il nous faut écrire une boucle qui accomplisse les opérations suivantes :

- a / lecture d'une donnée en mémoire ;
- b / comparaison entre le nom Martin et le nom contenu dans la donnée ;
- c / en cas d'égalité : écriture du numéro de téléphone ;
- d / en cas de non-égalité : retour au point a et lecture de la donnée suivante.

Ce programme s'écrit très simplement en utilisant une boucle 1 et 200 (première et dernière données contenues dans la mémoire) d'après l'organigramme de la page 191. Toutefois, cette solution présente un inconvénient : elle ne prévoit pas l'interruption de la recherche une fois obtenu le renseignement. Du temps

sera donc perdu. En effet, si les données se trouvaient sur une disquette, le temps de lecture ne serait pas négligeable (bloc « lecture de donnée en position 1 » sur l'organigramme). Dans un programme structuré d'après ce schéma, la lecture se poursuit jusqu'à la fin, même s'il lui était possible de s'arrêter à 3. Pour réduire ce temps au strict nécessaire, on a recours à un **indicateur**, ou **aiguillage** (flag en anglais), qui signale l'aboutissement de la recherche. Les opérations de lecture et de comparaison sont alors interrompues et le programme se termine.

Voici comment les choses se passent.

- Au départ, l'aiguillage est mis à zéro (initialisation), ce qui indique que toutes les opérations prévues doivent être exécutées.
- Une fois obtenu le résultat que l'on recherchait, on donne à l'indicateur une valeur différente (1, par exemple), signifiant que toutes les opérations doivent être sautées.

### Opératrice occupée à suivre le déroulement d'une boucle de recherche.

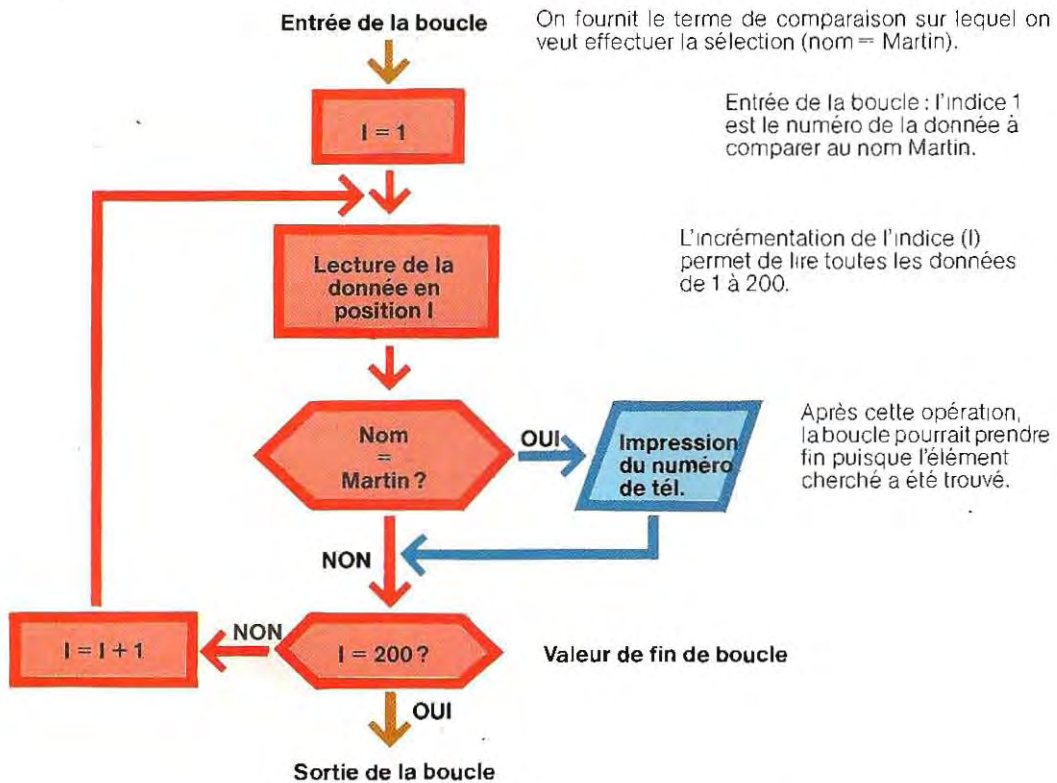
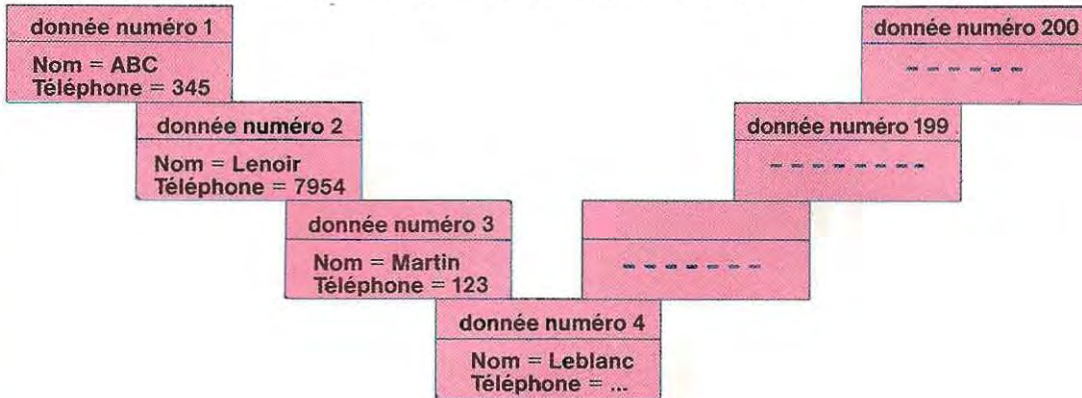


Marka



## ORGANIGRAMME D'UNE BOUCLE DE RECHERCHE DE DONNEE

Structure des données utilisée dans l'organigramme. La donnée recherchée (Martin, tél. = 123) porte le numéro 3.

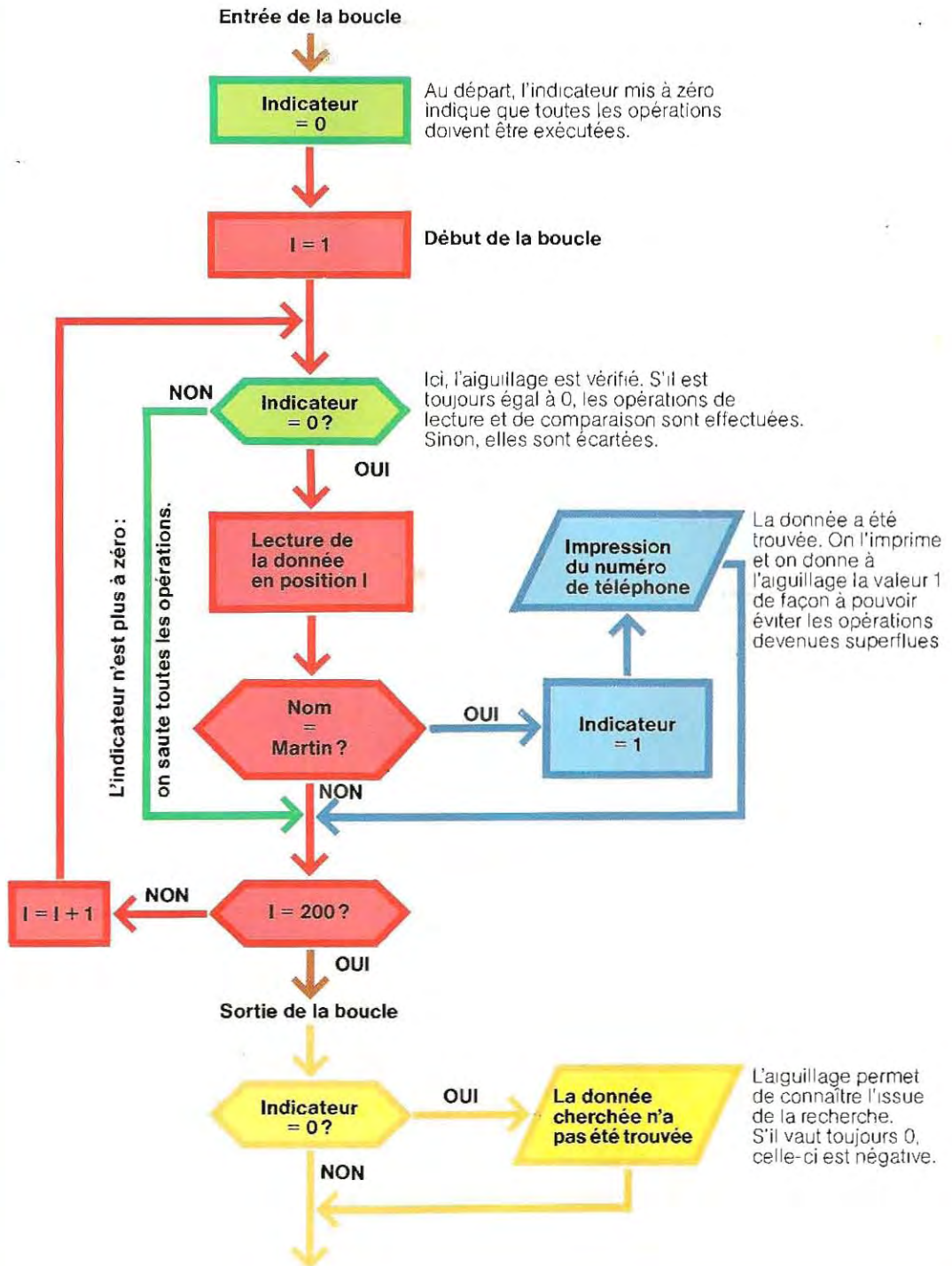


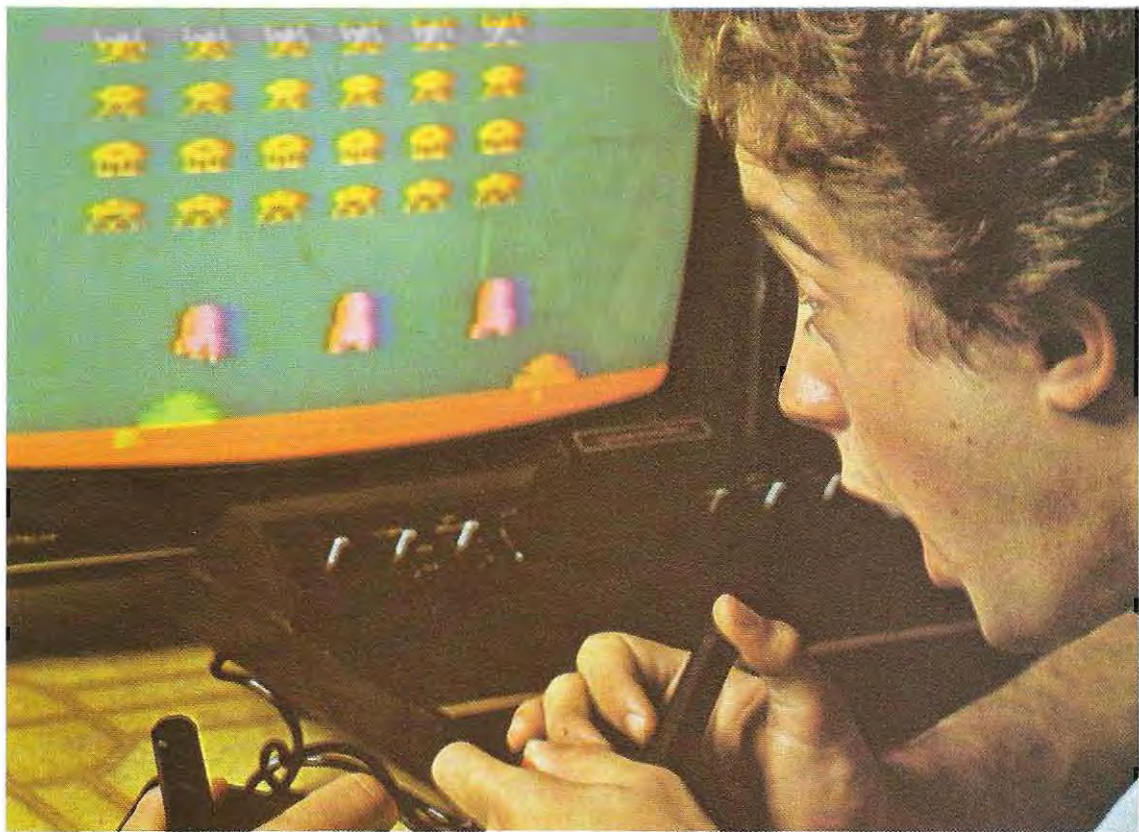
Dans cette organisation de boucle, on ne peut modifier l'enchaînement des opérations : la recherche se poursuit donc jusqu'à la fin, même si la donnée a déjà été trouvée.

L'organigramme de la page 192 illustre les modifications apportées par l'introduction d'un aiguillage. L'issue de la recherche est tout de suite connue grâce à l'impression (sur l'écran, ou sur papier à la sortie de l'imprimante) du numéro de téléphone. Si un pro-

gramme ne prévoit pas cette impression, des erreurs peuvent se produire dans l'exécution des calculs suivants, en cas d'insuccès de la recherche. L'indicateur permet donc de subordonner les opérations suivantes au succès de la recherche.

## ORGANIGRAMME D'UNE BOUCLE DE RECHERCHE DANS LAQUELLE UN INDICATEUR REDUIT LES TEMPS EN ELIMINANT LES OPERATIONS SUPERFLUES





E. Ferrelli/Life-Grazia Neri

**C'est souvent grâce aux jeux vidéo que l'ordinateur a fait son apparition dans les foyers.**

En d'autres termes, l'aiguillage sert d'indicateur et permet de choisir le sens du déroulement du programme.

### **Utilisation des organigrammes**

Les méthodes d'analyse et l'emploi de l'organigramme pour la résolution de problèmes spécifiques ne concernent pas uniquement les questions de gestion. On y a également recours pour des applications mineures, telles que l'emploi du micro-ordinateur comme adversaire de jeu.

Ces applications peuvent d'ailleurs poser des problèmes fort complexes et qui ne seront résolus qu'au prix d'une analyse très poussée. Avec l'organigramme qui clôt cette analyse, la rédaction des instructions constituant le programme n'est plus qu'un travail de traduction, sans grand risque d'échec.

A titre d'exemple, nous allons étudier la mise au point d'un programme permettant à la machine de jouer à Strike and Ball, jeu proche du célèbre Mastermind.

Cette étude nous fournira un intéressant modèle d'application.

Avant d'aborder la programmation proprement dite, énonçons les règles du jeu.

Le partenaire électronique (l'ordinateur) génère un nombre constitué de quatre chiffres aléatoires et que l'adversaire doit deviner en dix essais au maximum.

Après chaque essai, l'ordinateur donne l'une des trois réponses suivantes :

NUL, si le joueur n'a pas deviné un seul chiffre.

STRIKE, si le joueur a deviné des chiffres, mais pas leur position. Dans ce cas la machine doit également dire combien de chiffres ont été devinés.

BALL, si le joueur a deviné à la fois la valeur et la position de certains chiffres. Dans ce cas, la machine doit également donner le décompte de BALL.

La réponse peut être deux STRIKE et un BALL, ce qui veut dire que sur les quatre chiffres générés par l'ordinateur, le joueur a deviné la valeur de deux d'entre eux, mais non leur position, et la valeur et la position d'un troisième.

## Les circuits intégrés

Aujourd'hui, les circuits intégrés sont presque exclusivement fabriqués avec du silicium, mais leur construction fait appel au moins à huit technologies fondamentales, présentant chacune, à côté de nombreux avantages, des inconvénients inévitables.

Ces huit technologies se répartissent en trois groupes principaux, selon qu'il s'agit de circuits intégrés bipolaires, MOS, ou au contraire de pastilles dans lesquelles le silicium est remplacé par l'arséniure de gallium. Les semi-conducteurs les plus répandus aujourd'hui sont au silicium, et leur construction se fait selon deux principes essentiels. D'une part, nous trouvons les composants bipolaires, dans lesquels le courant est divisé en deux flux dont chacun circule vers un pôle électrique différent. D'autre part, nous avons les composants MOS, ou plus précisément MOSFET (Metal Oxide Semiconductor Field Effect Transistor), ce qui signifie transistor en métal oxyde semi-conducteur à effet de champ : ce sont des composants dans lesquels un seul flux de courant est exploité pour obtenir l'effet transistor.

Les puces, ou pastilles, bipolaires peuvent être construites selon trois techniques différentes, mais très comparables : TTL, ECL,  $I_2L$ .

• Les semi-conducteurs de la famille TTL (Transistor-Transistor Logic) sont peut-être les plus anciens et les plus largement utilisés ; offrant une vitesse de commutation\* élevée (comprise entre 1,5 et 3 nanosecondes\*\*), ils présentent l'inconvénient d'une forte consommation d'énergie, ce qui provoque aussi un échauffement notable. Dans les circuits complexes, quand des centaines ou des milliers de puces fonctionnent en même temps, ce phénomène peut nécessiter des installations spéciales de refroidissement des machines.

Dan Silversmith, responsable du développement des composants au silicium auprès du Lincoln Laboratory du Massachusetts Institute of Technology, a qualifié les composants TTL de « chevaux de trait » de la technologie

bipolaire, tout en reconnaissant que la demande du marché est encore élevée pour des dispositifs qui remontent à une quinzaine d'années, bien que leurs constructeurs soient aujourd'hui peu nombreux.

Le principe TTL présente l'inconvénient d'une technique de fabrication compliquée. Pour mettre au point un circuit intégré TTL, il faut construire en moyenne dix à douze masques superposés (les masques sont la reproduction photographique des schémas de liaison entre les diverses parties du semi-conducteur). La microplaquette est construite par superposition de plusieurs masques. Plus le nombre de masques est grand, plus la construction est complexe, et plus la probabilité d'erreur, ou de défaut du produit fini, augmenté, entraînant un rendement plus faible et une élévation des coûts.

• La technologie ECL (Emitter Coupled Logic, logique à couplage par émetteurs) est voisine de la TTL, mais les puces ECL ont une vitesse de commutation sensiblement plus élevée, avec des valeurs comprises entre 0,4 et 1 nanoseconde. Elle convient donc aux composants, tels que les mémoires, qui exigent des vitesses de fonctionnement très élevées. Un autre avantage de la technique ECL est la possibilité de miniaturisation qu'elle offre. C'est en fait la seule technologie bipolaire qui puisse être utilisée pour les circuits à très forte densité d'intégration (VLSI, Very Large Scale Integration) de la toute dernière génération, où le nombre de composants par unité de surface est considérable. La famille TTL ne convient que pour les puces à densité d'intégration moyenne MSI (Medium Scale Integration), deux niveaux au-dessous du VLSI. Toutefois, la consommation des puces ECL reste plus élevée que celle des autres familles.

•  $I_2L$ . Cette technologie bipolaire très récente a permis de réduire en partie le problème de la consommation. Mais en gagnant sur le plan de la consommation et des dimensions, on a perdu sur celui de la vitesse : les puces  $I_2L$  sont les plus lentes, avec des vitesses de commutation comprises entre 2,5 et 10 nanosecondes. Cet inconvénient est en partie compensé par une construction plus facile. La technique  $I_2L$  exige de huit à dix masques, soit moins que TTL et ECL (9 à 11).

La technologie MOS (ou MOSFET) est plus simple que la technologie bipolaire et se

\* La vitesse de commutation détermine la vitesse maximale à laquelle le composant peut fonctionner (par exemple la vitesse maximale d'exécution d'une opération logique).

\*\* 1 nanoseconde =  $10^{-9}$  seconde = 0,000000001 seconde.

contente d'un nombre de masques toujours inférieur à dix. De plus, elle consomme moins d'énergie. Il en existe trois types fondamentaux : NMOS, PMOS et CMOS. Dans les deux premiers, les lettres N et P indiquent le signe des charges électriques (respectivement négatives et positives) qui circulent dans le canal.

La technologie CMOS, où C signifie complémentaire, utilise des circuits doubles, dont chaque composant est formé d'un transistor PMOS et d'un transistor NMOS. Les charges étant de signes différents, lorsqu'un transistor fonctionne, l'autre est inactif, de sorte qu'en condition statique, la consommation est pratiquement nulle.

Les NMOS présentent l'avantage des dimensions. Cet avantage leur donne un très bon rapport nombre de circuits intégrés - surface de la pastille (puce) et permet de réunir beaucoup plus de fonctions que dans les autres technologies. De ce fait, les circuits NMOS sont moins coûteux, et sont aussi les plus rapides de la famille MOS, ce qui les rend particulièrement attrayants sur le marché de la grande distribution. L'inconvénient est leur consommation, supérieure à celle des CMOS, car ils exigent une plus forte alimentation statique. Mais ces consommations restent très largement inférieures à celles des puces bipolaires.

Les PMOS ont été les premiers circuits MOS produits, et sont aujourd'hui moins chers que tous les autres, car le nombre de masques nécessaires à leur production est plus réduit : six à huit en moyenne. Leurs dimensions, supérieures à celles des NMOS, et leur vitesse sont leur principal inconvénient. Les PMOS sont sans aucun doute les puces les plus lentes, et leur consommation ne vient pas contrebalancer cette caractéristique.

Les circuits CMOS sont certainement les plus intéressants de la famille, car les plus rapides et les plus efficaces, mais ils coûtent cher. En raison de leur faible consommation, ils sont utilisés dans des applications spatiales.

Les CMOS présentent un inconvénient technique : dans certaines conditions, les éléments du circuit ont tendance à se bloquer, et on ne peut le faire changer d'état qu'en coupant l'alimentation.

Pour pallier cet inconvénient, il faut améliorer l'isolation entre les transistors PMOS et

NMOS de chaque élément du circuit, d'où une augmentation du coût de production.

L'emploi du silicium comme base et comme élément porteur n'est pas la seule solution pour la construction de semi-conducteurs. Il en existe d'autres, et le remplacement du silicium par l'arséniure de gallium semble bien être une possibilité d'avenir.

Ce composé, de formule chimique GaAs, permet de produire des transistors très rapides, mais une plaquette de GaAs coûte cent fois plus cher qu'une plaquette de silicium ! De plus, on a du mal à obtenir la croissance des cristaux de GaAs sous forme cylindrique, forme naturelle des cristaux de silicium et qui se prête mieux que toute autre à l'utilisation industrielle. Ce produit pourrait d'ailleurs faire naître une autre difficulté. Le GaAs est obtenu par combinaison de deux éléments, le gallium et l'arsenic, dont le second est très toxique. Son utilisation en quantités importantes comporte donc des risques qui n'existent pas avec le silicium.

Une solution permettant d'obtenir les effets désirés (consommation réduite et vitesse élevée) nous est offerte, non par l'application d'une technologie nouvelle comme celle du GaAs, mais par la mise en œuvre dans des conditions de milieu différentes. L'immersion des circuits dans un bain d'hélium liquide permet de les porter à une température très proche du zéro absolu, à laquelle on voit apparaître le phénomène de la supraconductivité (découvert en 1911). Les divers éléments du circuit tendent à perdre à peu près totalement leur résistance passive, et peuvent alors devenir très rapides.

Les circuits capables de fonctionner dans ces conditions sont dits «à effet Josephson», du nom d'un chercheur qui étudiait la supraconductivité en 1962. Un circuit à effet Josephson peut commuter en environ 100 picosecondes (1 picoseconde =  $10^{-12}$  seconde), donc à une vitesse dix à cent fois supérieure à celle des circuits normaux. IBM a déjà réalisé des prototypes de processeurs à circuit Josephson, mais d'après Silversmith, il faudra encore au moins dix ans avant que ces produits n'arrivent sur le marché. Et en dix ans, il peut se passer tant de choses.

(D'après «la Jungle des sigles» dans la revue COMPUTER, n° 53, octobre 1982.)

Exposer les règles du jeu revient à faire l'analyse du problème.

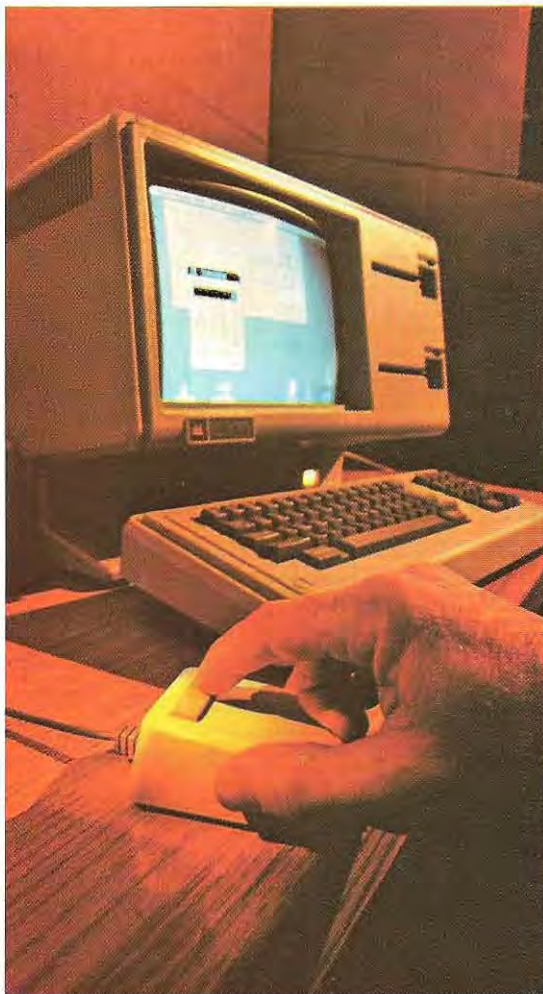
Cette méthode très simple, qui consiste à énumérer de manière claire, détaillée et rigoureusement ordonnée les fonctions à exécuter, permet l'analyse et la résolution de n'importe quel type de problème. C'est le point de départ de la préparation de tout organigramme.

La traduction de cette schématisation en symboles d'organigramme constitue le premier niveau d'analyse. Le schéma page 197 représente l'organigramme correspondant.

Il est très utile, du moins au début, de faire figurer à côté de chaque symbole des «étiquettes» numériques: cela facilite la lecture du schéma et permet d'écrire plus rapidement le programme.

Sur l'organigramme de la page 197 on distingue

**Cet ordinateur personnel affiche sur son écran le menu d'une application.**



C. O'Rear/West Light - Grazia Nen

trois colonnes principales: la première, ponctuée de nombres inférieurs à 1000, est le flux principal; la deuxième, dont la numérotation s'échelonne de 1000 à 1500, est le flux qui termine la partie en cours et en entame une autre; la troisième, dont la numérotation va de 2000 à 2400, est le flux des réponses que la machine doit fournir.

C'est ainsi par exemple que le bloc 600 correspond à une décision. A ce stade, la machine vérifie l'existence de la situation: «nombre de STRIKE = 0 et nombre de BALL = 0». Si telle est la situation, le joueur n'a réalisé aucun point. On notera que la phrase «nombre de STRIKE = 0 et nombre de BALL = 0» équivaut à «STRIKE = 0 ET BALL = 0» où l'opérateur logique ET est adopté pour vérifier la simultanéité de deux situations. Ecrit de cette manière, l'organigramme rassemble bien toutes les fonctions du programme, mais ce programme est «squelettique», peu amusant et inutilisable si le joueur ne connaît pas les règles.

Avant d'entrer dans le corps principal de l'organigramme, qui commence en A et se termine avec FIN (END, bloc 1500), il faut une partie descriptive que l'ordinateur présentera au joueur et sur sa demande, pour l'informer sur les règles du jeu.

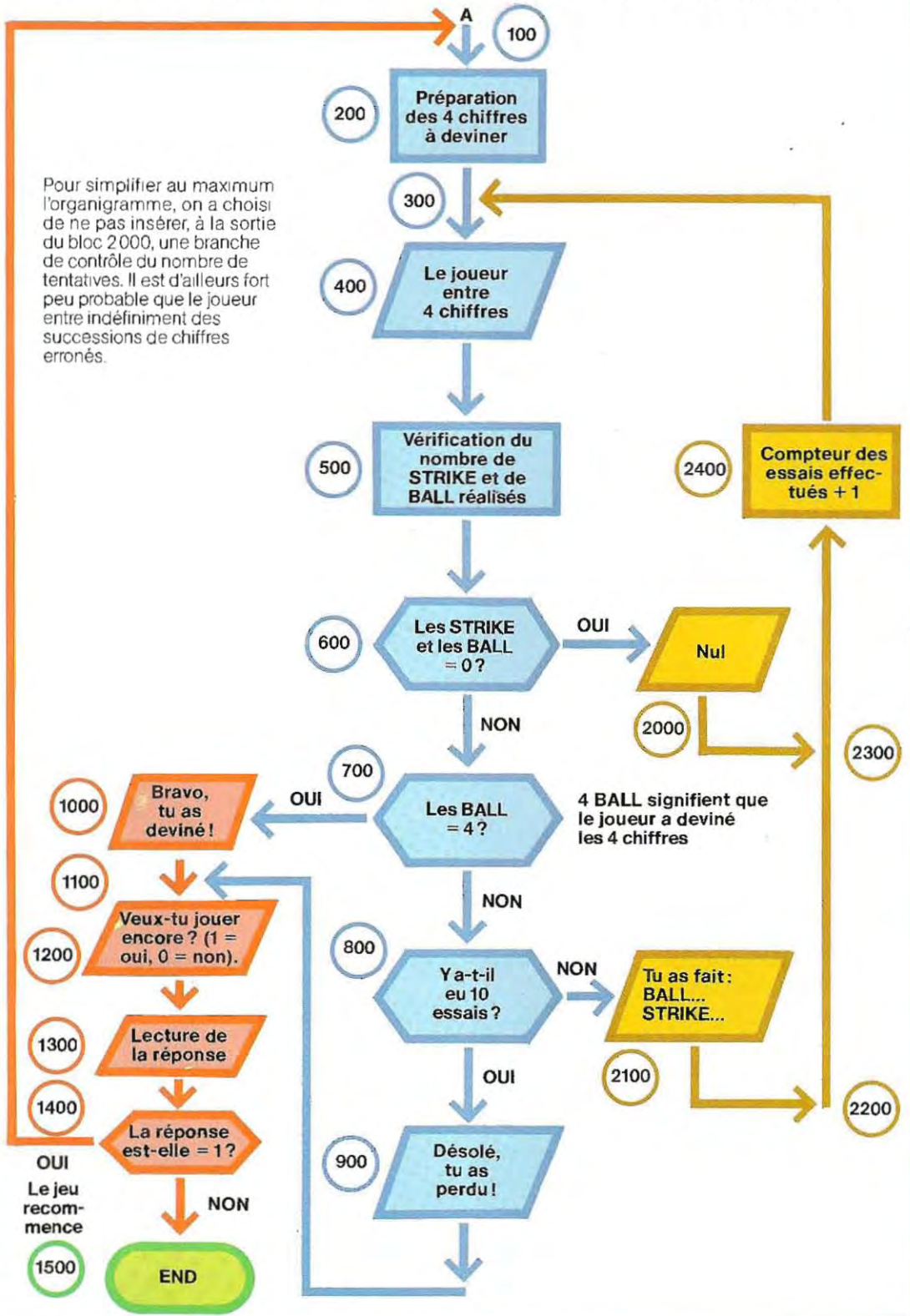
Cette partie d'information préalable sur les objectifs du programme et la façon de l'utiliser est indispensable, quelle que soit l'application envisagée.

Il est difficile que l'utilisateur d'un programme (ou le programmeur lui-même) puisse se souvenir exactement de tous les paramètres à fournir et de toutes les fonctions du programme. Cette partie descriptive initiale sert justement d'aide-mémoire et facilite la tâche de quiconque utilise le programme.

Le schéma de la page 197 comprend deux blocs (200 et 500) représentant des fonctions complexes, mais rien n'indique comment se déroulent ces fonctions. C'est ce qui sera explicité dans les organigrammes suivants. En renvoyant ainsi les explications détaillées à une phase ultérieure, on évite de compliquer l'organigramme principal, et surtout on se réserve la possibilité de mieux développer (à part) les fonctions les plus complexes. Cela permet de les écrire en termes généralisés et d'en faciliter l'utilisation pour d'autres applications. Cette technique constitue la règle d'or de la programmation.

# ORGANIGRAMME DE PREMIER NIVEAU POUR LE JEU STRIKE AND BALL

Pour simplifier au maximum l'organigramme, on a choisi de ne pas insérer, à la sortie du bloc 2000, une branche de contrôle du nombre de tentatives. Il est d'ailleurs fort peu probable que le joueur entre indéfiniment des successions de chiffres erronés.



Le schéma ci-dessous représente l'organigramme du module qui compte les points réalisés pour STRIKE et pour BALL (bloc 500 sur l'organigramme de la p. 197).

Toutes les indications numériques de ce bloc doivent être comprises entre 500 (valeur attri-

buée sur l'organigramme principal) et 590. Pour compléter les organigrammes du programme, il ne reste qu'à développer le module désigné par le nombre 200 (voir schéma p. 199) et qui a pour objet de préparer les quatre chiffres à deviner.

