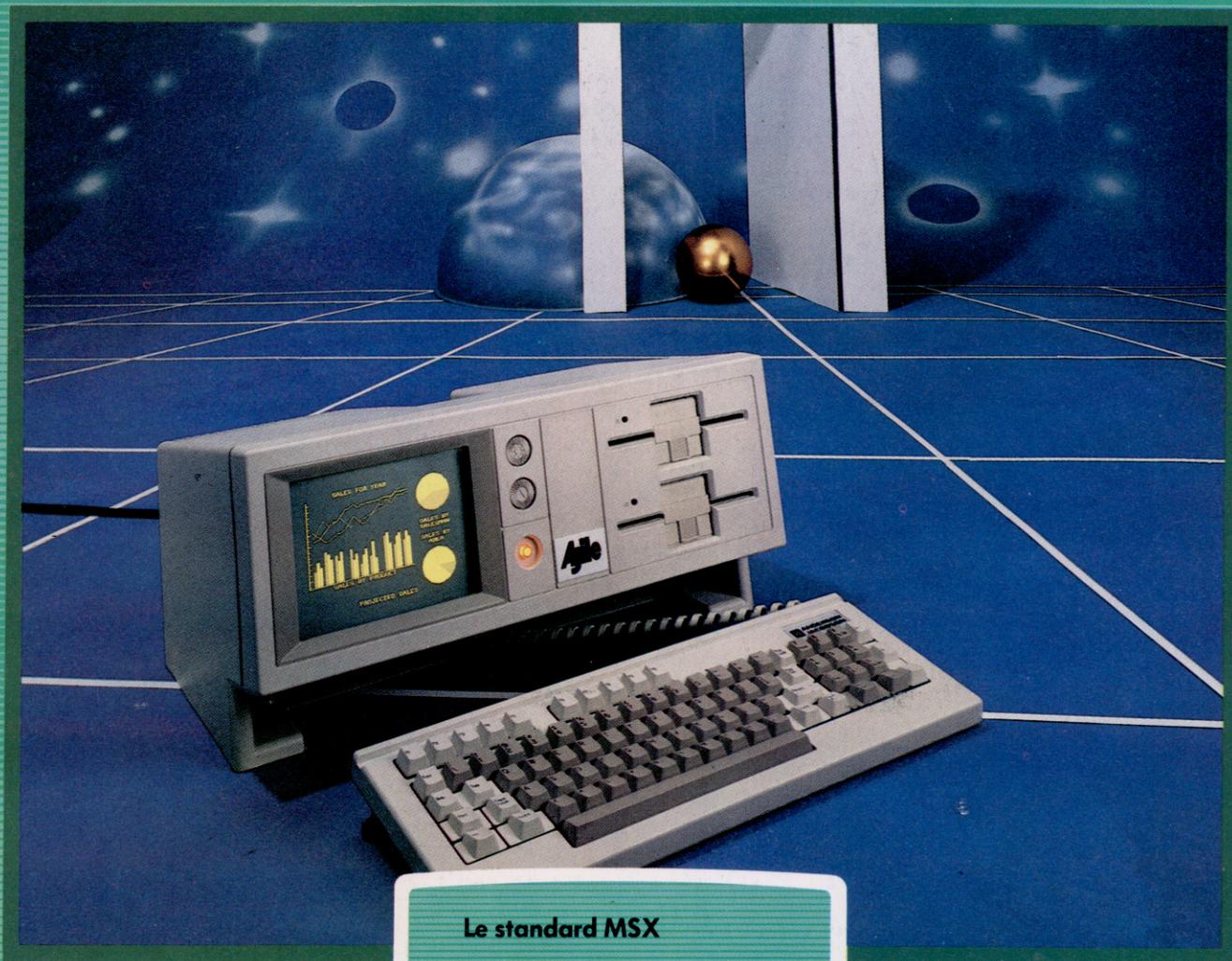


ABC

N° 32

COURS
D'INFORMATIQUE
PRATIQUE
ET FAMILIALE

INFORMATIQUE



Le standard MSX

Les Spectravideo

Logiciel de comptabilité

Leçons de mathématiques

EDITIONS
ATLAS

M6062-32-12F

85FB-3,80FS-51.95



Un pour tous, tous pour un

L'un des gros problèmes auxquels se heurtent les auteurs de logiciels est la compatibilité. Des constructeurs ont adopté le standard MSX dans l'espoir de surmonter cette difficulté.

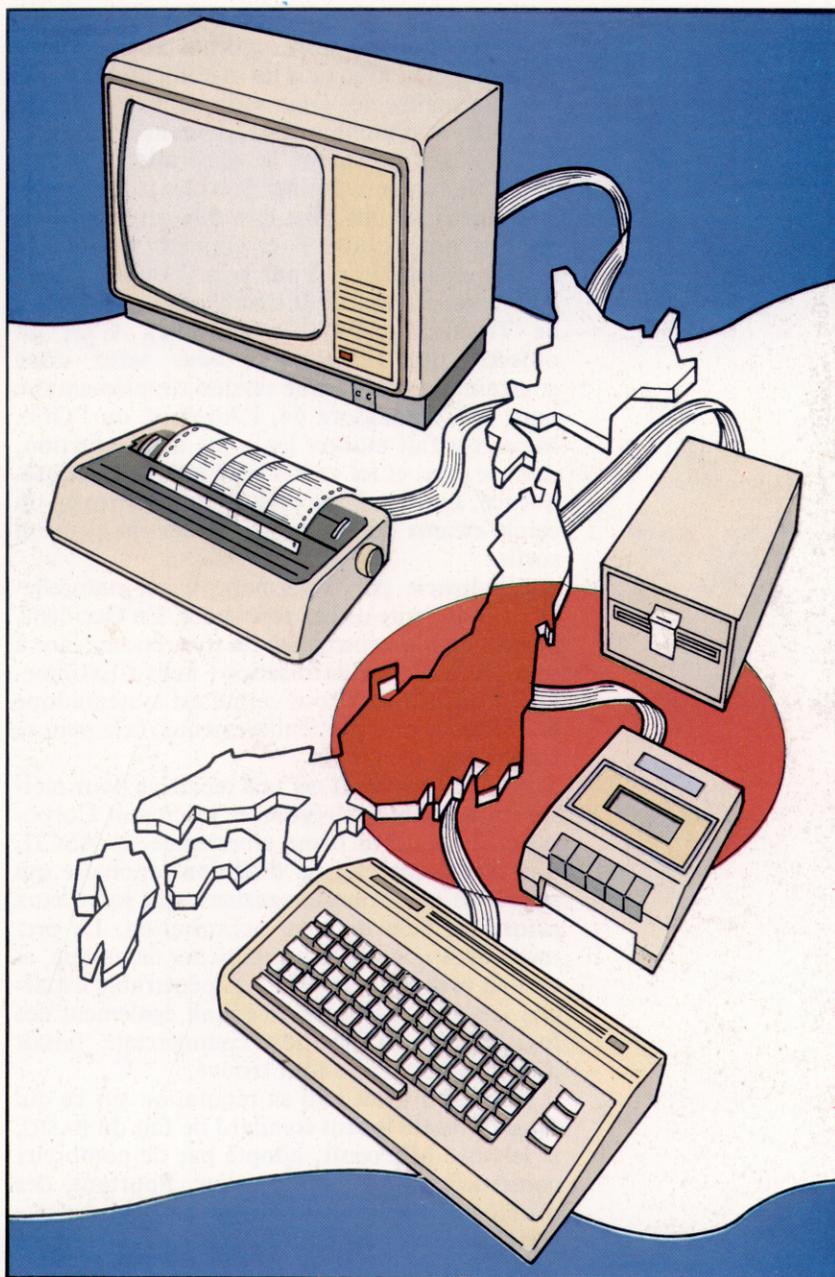
Pour comprendre pourquoi les micro-ordinateurs sont pratiquement incompatibles entre eux, il faut remonter à la naissance même de la micro-informatique. Le 8080 d'Intel et le 6800 de Motorola furent les premiers micro-processeurs à pénétrer réellement le marché de l'informatique individuelle. Leur jeu d'instructions était prédéfini, et les problèmes naquirent de là. Car chacun d'eux avait de grosses capacités : et si envoyer un octet de données vers une sortie quelconque mettait en jeu la même instruction (l'UC étant la même), l'adresse de destination pouvait varier presque à l'infini, selon le micro-ordinateur utilisé. Les premiers constructeurs de ce type d'appareils étaient des passionnés qui travaillaient dans leurs garages, et qui bien entendu ne se souciaient guère d'établir une norme relative aux adresses mémoire. La société A plaçait le port de sortie de l'imprimante en parallèle à l'adresse 255, et la société B en 256.

Cette situation, déjà mauvaise, s'aggrava encore lorsque les effets graphiques et sonores, contrôlés par des puces spécialisées, devinrent accessibles à des prix raisonnables. Il s'ensuivit qu'un programme utilisant au mieux les capacités d'un ordinateur particulier ne pourrait tourner sur un autre sans avoir été presque entièrement réécrit.

Tout se serait passé différemment si une seule compagnie avait eu, dès le départ, la puissance nécessaire pour imposer un standard. Mais il n'en fut rien : l'époque héroïque des premiers ordinateurs familiaux fut celle de petites firmes très nombreuses, et chacune avait ses propres normes. Non seulement les machines différaient entre elles, dans leurs circuits comme dans leur construction, mais elles employaient des langages de programmation incompatibles. Le BASIC qui nous est aujourd'hui si familier ne fut pas pris au sérieux par les professionnels de l'informatique avant les années soixante-dix : on n'y voyait qu'un outil de débutant.

Tout changea vers la fin de la décennie, quand se produisit l'explosion de la micro-informatique. Des pionniers comme Apple se dotèrent de raffinements tels que la couleur, ou un graphisme sophistiqué. Pour faciliter l'emploi de telles innovations, les constructeurs durent développer leurs propres dialectes BASIC, qui ne tardèrent pas à proliférer.

On aurait pu penser qu'un tel état de choses serait profitable à l'utilisateur. Mais il n'en fut que davantage frustré — comme les fabricants



et les auteurs de programmes. Imaginons que l'heureux possesseur d'un SORD veuille se procurer « Kiki le jambon-beurre galactique », un jeu qui fait fureur ; il lui faudra attendre qu'une traduction de ce logiciel, écrit au départ pour le Spectrum, soit réalisée. Et rien ne prouve, en définitive, qu'une version propre à votre appareil sortira !

Le chemin du succès

Le standard MSX est pour le Japon le moyen de s'ouvrir le marché mondial. S'il était adopté partout, les Japonais pourraient dominer la micro-informatique, après la photo et la hi-fi. (Cl. Tony Duncan-Smith.)



Si vous êtes constructeur d'ordinateurs et que vous vous apprêtez à lancer un nouveau modèle sur le marché, vous n'ignorez pas qu'il vous faudra en vendre un certain nombre avant qu'apparaissent des programmes qu'il pourra faire tourner. Mais cette carence aura précisément pour effet de limiter les ventes très étroitement, et vous risquez d'investir à fonds perdus.

Si votre profession consiste à rédiger des programmes vendus commercialement, votre créneau se restreindra, au mieux, aux gens qui possèdent l'ordinateur pour lequel vous avez conçu votre logiciel. Vous avez par exemple mis au point un jeu d'aventures intitulé les « Donjons d'outre-ciel », avec tous les ingrédients nécessaires : un maître des lieux sadique, une présence invisible, et d'innombrables démons d'iniquité ; mille pièges attendent le téméraire. Les responsables du marketing prévoient des ventes énormes si le tout peut être mis sur le marché pour moins de 100 F et franchit le cap des 65 000 exemplaires. Pour cela il faudrait malheureusement attirer de nombreux propriétaires de Thomson TO7 (par exemple). C'est un objectif peu réaliste, et vous serez donc contraint de prévoir une version supplémentaire pour le Commodore 64, l'Apple II ou l'Oric. Mais cela fait monter les coûts de production, donc le prix, et les ventes vont baisser en conséquence. La rentabilité cesse d'être assurée — un cercle vicieux dont bien des auteurs ne peuvent sortir.

L'industrie de l'informatique est naturellement consciente de ces problèmes. En Occident, l'esprit de libre entreprise est très réticent face à toute idée de standardisation ; mais l'Extrême-Orient affectionne tout ce qui est systématique et ordonné, tout particulièrement si cela peut se traduire en bénéfices.

ASCII/Microsoft est une tentative pour mettre fin à l'anarchie régnante. Microsoft Corporation, la célèbre firme américaine, et ASCII, une florissante maison d'édition japonaise qui publie de nombreux magazines, ont joint leurs efforts au sein de cette organisation. La première était riche de son expérience technique, et voulait prendre pied sur l'« impénétrable » marché japonais. La seconde éditait également des logiciels, et sa position commerciale faisait d'elle un partenaire tout trouvé.

Microsoft avait bâti sa réputation sur ce qui est sans doute le seul standard de fait du BASIC, le MBASIC Microsoft, adopté par de nombreux constructeurs du monde entier. Pourtant, des problèmes d'incompatibilité se posaient dès que certaines caractéristiques propres à tel ou tel appareil étaient mises en route. Il était donc impossible de garantir qu'un programme en MBASIC pourrait tourner sur l'un comme sur l'autre.

Ce langage fut cependant vendu à de nombreuses firmes d'informatique japonaises par l'intermédiaire d'ASCII/Microsoft, mais cela ne fournissait pas de véritable solution. Les deux partenaires eurent l'idée d'établir une norme, en association avec les plus grands cons-

tructeurs du Japon, en espérant qu'elle prendrait une valeur internationale. Ce fut le standard MSX, dont les spécifications concernent aussi bien les matériels (construits autour d'un microprocesseur Z-80 et de certaines puces spécialisées) que la programmation elle-même.

MSX : les règles de base

Le BASIC MSX est très semblable au BASIC Microsoft, mais est enrichi d'importantes améliorations, qui permettent de tirer un meilleur parti des ressources sonores et graphiques actuelles. Parmi les instructions, il faut citer SCREEN, qui spécifie le mode écran, la taille des plans-objets, le cliquetis des touches, le rythme (en bauds) de transfert des données sur cassette, l'option imprimante ; LOCATE qui permet le positionnement de caractères sur l'écran ; COLOUR qui sélectionne l'une des seize couleurs de premier plan ou d'arrière-plan ; PUT SPRITE qui définit les attributs des plans-objets ; CIRCLE qui dessine des plans et des ellipses ; DRAW qui trace des lignes et des figures ; LINE qui unit par un trait des coordonnées particulières ; PAINT qui remplit certaines surfaces d'une couleur spécifiée. Une instruction KEY assigne tel ou tel rôle aux touches de fonction. VPOKE inscrit des valeurs dans la RAM vidéo, SOUND fait de même au sein des registres de la puce sonore, et MOTOR contrôle le moteur de la cassette.

MSX définit également les matériels de façon très précise. L'unité centrale doit être un Z-80 tournant à 3,58 MHz. La ROM ne peut être inférieure à 32 K, la RAM à 8 K ; dans les deux cas aucune limite supérieure n'est fixée. Un ordinateur MSX est par ailleurs obligatoirement pourvu d'une puce TMS9918A, fabriquée par Texas Instruments et qui contrôle le signal vidéo. Autre dispositif indispensable : un générateur de sons à trois voies AY-3-8910. L'affichage écran est de 24 lignes, en 32 ou 40 caractères, et pour le moment le passage aux 80 caractères n'est pas prévu. La résolution minimale est fixée à 256 x 192.

Une interface cassette a été choisie, ce moyen étant le plus utilisé pour stocker données et programmes. Le codage se fait suivant le procédé FSK (Frequency Shift Keying, « modulation par déplacement de fréquence »), à 1 200/2 400 bits par seconde. Le clavier, dont la présentation est elle aussi standardisée, et qui comporte plusieurs touches de fonction, n'est pas pour rien d'origine japonaise : on peut au besoin l'utiliser pour les *katakana* (alphabet phonétique aux formes anguleuses et qui sert à transcrire les mots étrangers), ainsi que les *hiragana* (alphabet phonétique aux formes cursives), les caractères graphiques classiques et les *kanji* (les idéogrammes d'origine chinoise).

Chaque appareil de ce type comporte un port standard pour deux joysticks, un port pour cartouches de ROM, et un bus d'entrées/sorties à cinquante broches.

Le système d'exploitation de disquettes a reçu le nom de MSX-DOS. Il est fonctionnellement



Structure d'un système MSX

UC

Le célèbre microprocesseur Z-80 est le cœur de tous les systèmes.

Circuits supplémentaires

Une fois respectées les caractéristiques de base, le constructeur peut ajouter à l'ensemble tel ou tel élément qui lui paraît indispensable.

Générateur de sons

Une puce 8910 (trois voies, effets spéciaux).

RAM Vidéo

Elle est parfois confiée à une puce spécialisée.

Connecteur d'extension

Toutes les machines en sont pourvues.

Affichage écran

Une puce Texas 9918 (ou un équivalent) s'en charge, accordant chichement 16 lignes de 32 caractères. La résolution est de 256 x 192 en 16 couleurs; on peut disposer de 32 plans-objets au maximum.

ROM

Ses 32 K lui permettent d'accueillir un BASIC puissant.

RAM

Elle doit être d'au moins 8 K.

Cassette

Un lecteur de cassette ordinaire, sur lequel les données sont enregistrées au rythme rapide de 1 200 ou 2 400 bauds.

Clavier

Tous les appareils de type MSX ont les mêmes touches, mais leur disposition peut varier.

Port cartouche

Standardisé, il peut accepter n'importe quelle cartouche prévue pour la gamme MSX.

Lecteur de disquettes

Il a lui aussi été défini, tout comme les opérations de formatage.

équivalent au MS-DOS, et peut lire des fichiers de données créés avec ce dernier système. On dit également que le MSX-DOS est compatible avec le DOS CP/M2.2, qui a connu un si grand succès. Le formatage des disquettes de trois pouces (Sony), cinq pouces et demi et huit pouces est aussi défini.

Si toutes ces caractéristiques sont respectées, cela veut dire qu'un programme rédigé pour une machine MSX quelconque pourra tourner sur n'importe quel appareil de type MSX, quelle que soit la disquette sur laquelle il a été stocké, et ne perdra rien de toutes ses ressources graphiques et sonores. Le constructeur comme l'utilisateur semblent devoir y trouver leur compte.

La méthode a pourtant des inconvénients. Le premier est que toute norme, quelle qu'elle soit, tire très difficilement parti de l'innovation technique. Si, par exemple, il apparaît un nouveau circuit contrôlant le signal vidéo, avec des capacités bien supérieures, les programmes destinés au système MSX ne pourront en faire usage, et la concurrence aura donc le champ libre.

Le second est que les microprocesseurs à 8 bits comme le Z-80 (le plus célèbre d'entre eux) n'ont qu'une espérance de survie très limitée. Ils sont par définition incapables de traiter plus de 64 K de mémoire directement, et ne peuvent

manipuler, en une seule fois, des données dont la valeur soit supérieure à 256.

De ce point de vue, le standard est peut-être une ultime tentative de prolonger la vie du Z-80, dont le triomphe sur le marché appartient déjà au passé. Il est difficile d'imaginer sérieusement que les microprocesseurs 8 bits domineront encore la micro-informatique d'ici à dix ou même cinq ans. Sans doute le standard aura-t-il l'utilité de rappeler aux constructeurs que la normalisation doit suivre l'innovation technologique d'aussi près que possible; mais, s'il peut favoriser très certainement la vente d'appareils construits autour du Z-80, on a du mal à imaginer qu'il puisse avoir un impact à long terme. En ce qui concerne les 16-bits, l'IBM PC, devenu une référence de fait, a d'ores et déjà démontré que la raison du plus fort était toujours la meilleure... Qu'en sera-t-il pour les 8-bits? Le standard MSX est à l'heure actuelle adopté par seize fabricants. Au Japon: Yamaha, JVC, Hitachi, Sony, Sanyo, National, Pioneer, Canon, Fujitsu, Mitsubishi. Parmi les autres, on peut citer Spectravideo (États-Unis) et Daewoo (Corée). Aucune firme anglaise ou française, par contre, ne s'y est encore ralliée. Seul l'avenir permettra de dire si le marché a davantage besoin de normes que d'innovations.

Des règles très strictes

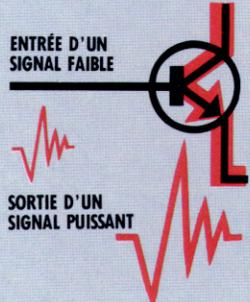
Pour assurer une compatibilité maximale (au niveau du matériel comme des périphériques et des programmes), la conception d'un ordinateur de type MSX obéit à des règles contraignantes. Cependant, une fois respectées les spécifications fondamentales, le constructeur peut décider d'ajouter à son produit des dispositifs qui lui sont propres. (Cl. Steve Cross.)



Tableau de manœuvre

Les transistors : des amplificateurs

Un transistor se compose de trois parties : la base, le collecteur et l'émetteur. La base a pour fonction de contrôler un courant circulant du collecteur à l'émetteur. Ce courant peut passer librement si un certain courant traverse la base. Lorsque ce dernier est modifié, la résistance du trajet collecteur-émetteur varie à l'unisson. En fournissant un courant important au collecteur, et un petit courant à la base, l'émetteur produit un signal, qui varie en fonction du petit signal de la base. Le transistor fonctionne donc comme un amplificateur de ce petit signal.



Les transistors : des commutateurs

Se servir d'un transistor comme commutateur implique l'emploi systématique des propriétés de la base, du collecteur et de l'émetteur. Ce dispositif ne permet que le passage d'une certaine intensité de courant entre le collecteur et l'émetteur, et possède un point de saturation au-delà duquel ce trajet n'est plus affecté par les faibles variations du courant électrique qui traverse la base.



Les diodes, les résistances, les condensateurs et les transistors jouent un rôle fondamental en informatique. Nous allons voir quelles sont leurs fonctions, et construire avec eux des portes logiques très simples.

Les transistors ont deux fonctions principales : agir comme amplificateurs d'un signal électrique, et permettre ou empêcher le passage d'un courant en le contrôlant à l'aide d'un autre courant. C'est cette dernière propriété de commutation électronique qui les rend si utiles dans les ordinateurs. En les regroupant, on peut construire des circuits de type ouvert/fermé, allumé/éteint, que le microprocesseur traitera comme des nombres binaires. Les portes logiques sont des structures au sein desquelles vous rassemblez des circuits de ce genre.

Si vous construisez les portes logiques décrites page suivante, vous vous direz sans doute qu'un ordinateur entièrement composé de transistors doit être à la fois très gros et très coûteux ; et ce fut bel et bien le cas autrefois. Pour diminuer et la taille et les prix, il fallut mettre au point les circuits intégrés, qui ne sont jamais que le rassemblement, sur une minuscule couche de silicium, de centaines de transistors, protégés par un boîtier de plastique noir. Dans la plupart des cas, les circuits intégrés très simples, dits « puces TTL » (à logique transistor-transistor), vous suffiront. Quatre portes ET installées sur une puce ne valent que quelques francs. Pour des tâches plus complexes, la dépense est à peine plus élevée.

On appelle « intégration à très grande échelle » l'opération qui consiste à placer de plus en plus de circuits sur une seule puce (qui peut ainsi comporter des milliers de composants) capable d'effectuer des travaux très complexes. Un microprocesseur est un bon exemple de ces puces dites VLSI (Very Large Scale Integration). On peut également citer les circuits responsables de l'affichage écran, des effets sonores et du fonctionnement des interfaces.

Diverses technologies permettent leur réalisation, d'où des coûts, des performances et une consommation différents. Les puces installées sur la plupart des micro-ordinateurs sont de type MOS (Métal/Oxyde/Silicium), tandis que les portables alimentés par batteries ont recours aux puces de type CMOS (« MOS complémentaire »), plus lentes, mais qui consomment beaucoup moins.

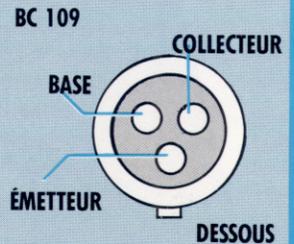
Portes logiques

Ces circuits très simples (page de droite) illustrent la façon dont les transistors, utilisés comme commutateurs, peuvent servir à la construction de portes logiques. Vous pourrez réaliser chacune d'elles successivement, à l'aide des mêmes composants et d'une plaquette d'essai. Les entrées dans

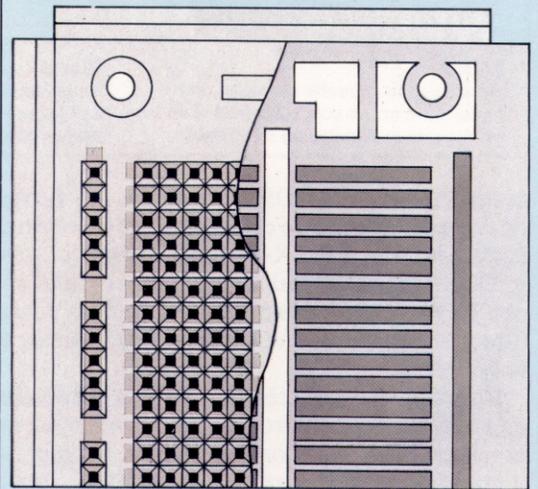
les circuits sont, dans notre exemple, commandées par des boutons-poussoirs raccordés à des diodes électroluminescentes. Mais dans un ordinateur, les opérations de commutation se font grâce à des semi-conducteurs, et il n'y a donc pas de parties mobiles ; les « entrées » sont en fait réglées par d'autres circuits. C'est ce que vous pouvez faire.

Ce qu'il vous faut

- 1 Plaquette d'essai (Experimentator 300 ou autre)
- 2 Transistors BC 109
- 2 DEL rouges
- 1 DEL verte
- 3 Résistances de 500 Ω
- 2 Résistances de 15 K-Ω
- 2 Commutateurs à bouton-poussoir
- 1 Batterie de 9 volts
- 1 Raccord pour batterie
- Un peu de fil électrique



Plaquettes d'essai

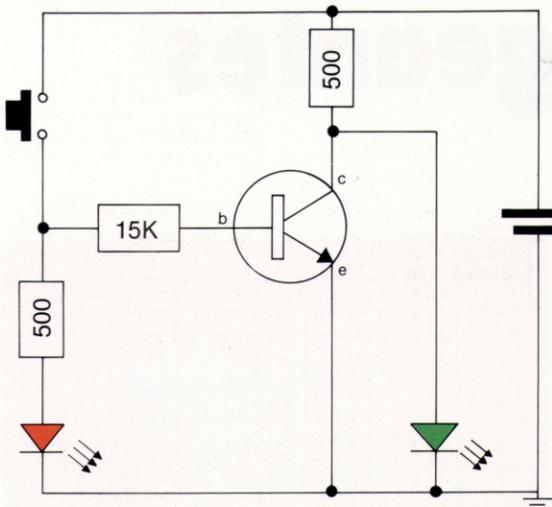


SURFACE

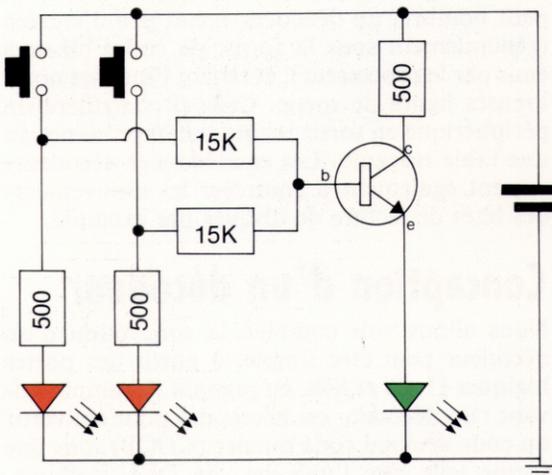
CONNEXIONS INTÉRIEURES

Les plaquettes d'essai permettent de se livrer à des expériences avec des circuits comme ceux-ci, sans être obligé de souder les différents composants. Ce sont des supports réutilisables, sur lesquels on fixe sans effort les éléments. Ils sont installés sur des fixations métalliques, qui servent de conducteurs, et chaque groupe de cinq trous est ainsi relié électriquement. Il est très facile de les connecter entre eux à l'aide d'un peu de fil électrique, et de construire des circuits simples. La plaquette représentée ci-dessus est bien plus large que ce dont vous avez besoin actuellement, mais se révélera utile lors de travaux ultérieurs.

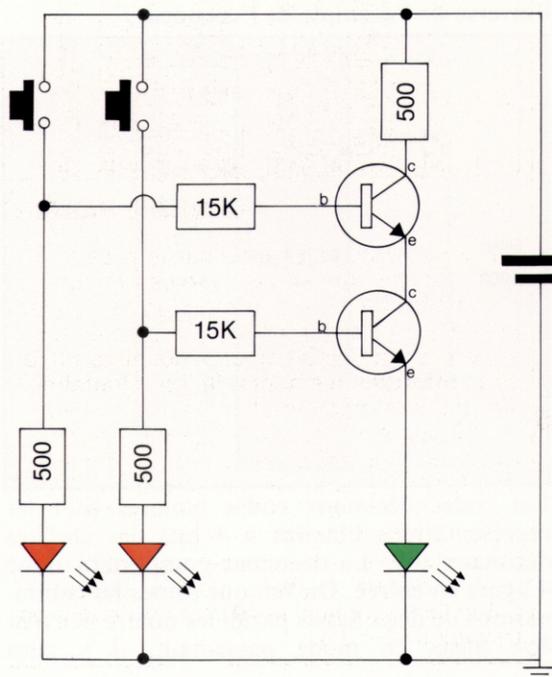
Kevin Jones



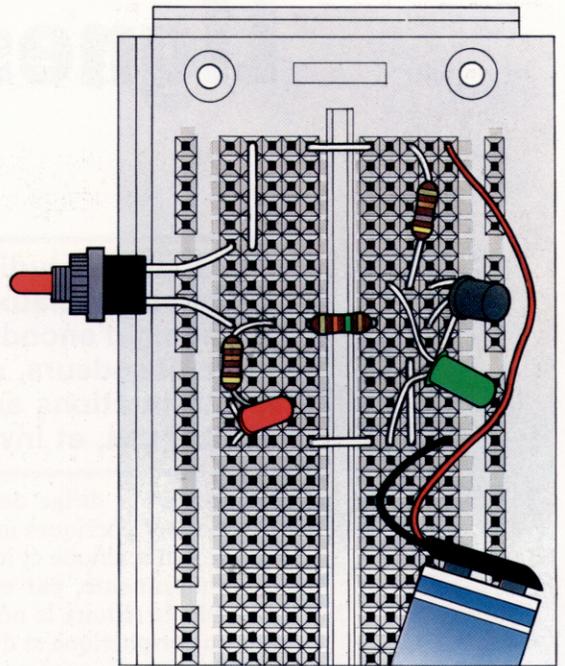
La porte NON
C'est la plus simple de toutes : elle fonctionne avec une seule entrée (indiquée par le commutateur et la diode rouge) et une seule sortie (la diode verte). Lorsque le commutateur reste ouvert, aucun courant ne peut passer de la base à l'émetteur du transistor. Une résistance élevée est donc créée dans le collecteur, aussi le courant empruntera un autre chemin et passera par la diode verte. Ainsi, lorsque le bouton n'est pas enclenché (entrée : 0), la diode verte s'allume (sortie de 1). Manœuvrer le bouton-poussoir fait passer un courant dans la base du transistor, ce qui supprime la résistance du collecteur. Le courant peut désormais traverser le transistor, évitant la diode.



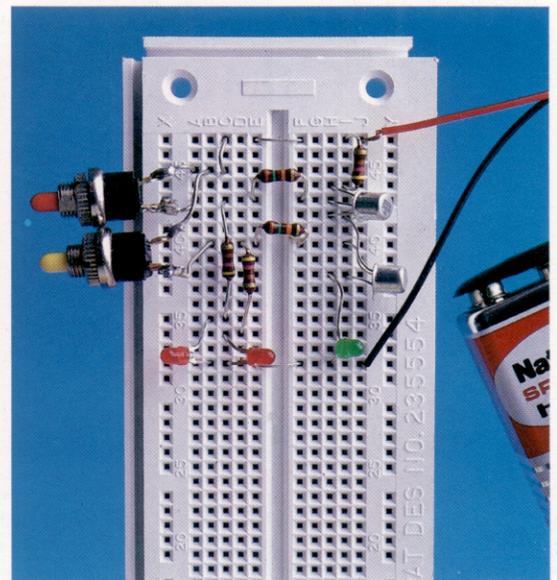
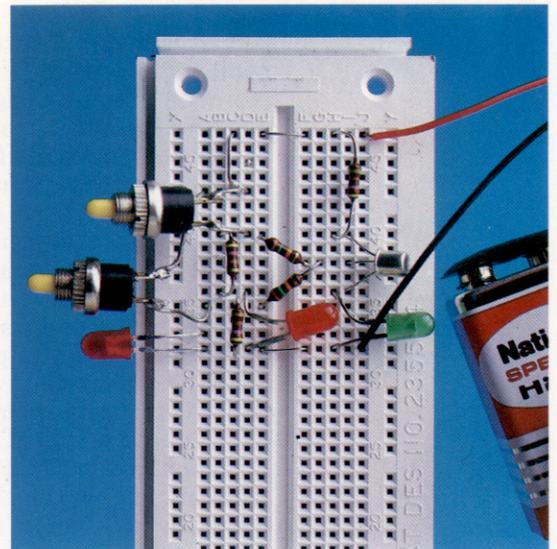
La porte OU
Il suffit, pour la construire, d'adapter la porte NON. Le premier changement consiste à placer la diode de sortie (verte) de telle façon qu'elle soit commandée par un courant passant à travers le transistor. Une porte OU a deux entrées, dont chacune est munie d'un commutateur et d'une diode rouge. Lorsque l'une ou l'autre de ces entrées transmet un signal à la base, le courant peut traverser le transistor pour parvenir à la diode de sortie.



La porte ET
Pour créer une porte ET, il nous faut deux transistors menant à la sortie, chacun d'eux ayant sa propre entrée. C'est seulement lorsque les deux commutateurs sont enclenchés (c'est-à-dire que les deux entrées sont de 1) que les deux transistors permettront le passage du courant jusqu'à la diode verte.



Kevin Jones



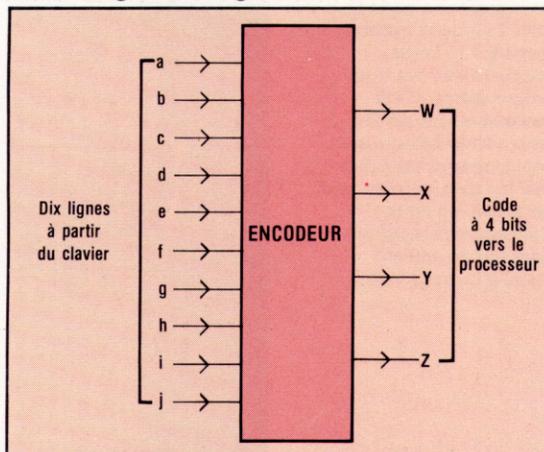
Ian McKinnell

Lignes changeantes

Nous allons étudier les circuits logiques spéciaux connus sous les noms d'encodeurs et de décodeurs, qui transcrivent les instructions en signaux électriques, et inversement.

Le processeur dirige des instructions à la fois vers des périphériques internes (l'accumulateur, l'unité arithmétique et logique, etc.) et externes (une imprimante, par exemple). Il est souvent possible de réduire le nombre de lignes utilisées par un périphérique et d'obtenir une entrée simplifiée pour le processeur. C'est ce que l'on appelle « encoder ». Nous pouvons donner comme exemple d'encodeur un circuit utilisé conjointement avec le clavier. Ce circuit peut présenter en sortie 64 lignes dont l'une produira un signal lorsque la touche appropriée est frappée. Puisqu'une seule touche est généralement frappée à la fois, chacun des 64 signaux possibles en sortie peut être codé sous la forme d'un nombre binaire sur 6 bits. Ce qui signifie qu'il ne faut que 6 lignes pour transmettre au processeur l'information relative à la touche frappée. Le périphérique qui transcrit 64 lignes en 6 est un encodeur. Dans la pratique on ajoute deux lignes supplémentaires, une pour le contrôle de parité, et l'autre pour indiquer l'utilisation d'une touche préfixe (Control ou Shift).

Prenons l'exemple d'un clavier qui ne comporterait que 10 touches, pour les nombres 0 à 9. Un code binaire sur 3 bits ne permettrait que 8 combinaisons. C'est pourquoi nous devons imaginer un encodeur qui accepte 10 lignes en entrée et génère 4 lignes en sortie.



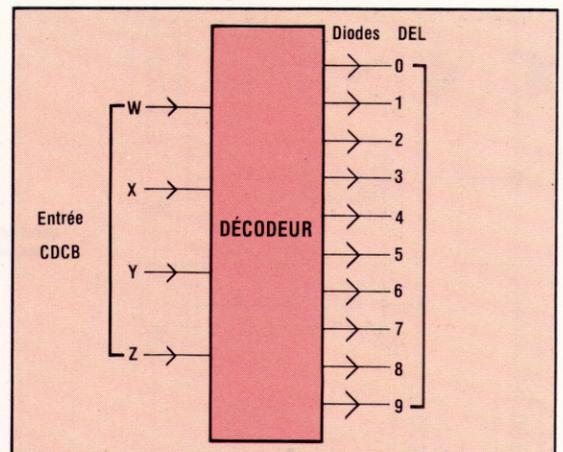
Du fait qu'une seule ligne parmi les 10 peut être activée (ou encore mise en mode « passe-haut ») à la fois, la table de vérité sera la suivante :

Numéros	Entrées										sorties			
	a	b	c	d	e	f	g	h	i	j	W	X	Y	Z
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0	0	1
2	0	0	1	0	0	0	0	0	0	0	0	0	1	0
3	0	0	0	1	0	0	0	0	0	0	0	0	1	1
4	0	0	0	0	1	0	0	0	0	0	0	1	0	0
5	0	0	0	0	0	1	0	0	0	0	0	1	0	1
6	0	0	0	0	0	0	1	0	0	0	0	1	1	0
7	0	0	0	0	0	0	0	1	0	0	0	1	1	1
8	0	0	0	0	0	0	0	0	1	0	1	0	0	0
9	0	0	0	0	0	0	0	0	0	1	1	0	0	1

Le décodage est l'opération inverse de l'encodage. Au lieu de recevoir en entrée de nombreuses lignes pour en produire en sortie un petit nombre, un décodeur reçoit peu d'entrées (généralement sous la forme de codes binaires émis par le processeur), et retient l'une des nombreuses lignes de sortie. Celle-ci contrôlera un périphérique en sortie tel qu'une imprimante ou une table traçante. Les encodeurs et décodeurs servent également à contrôler les mouvements des têtes de lecture de disques par exemple.

Conception d'un décodeur

Nous allons voir combien la conception d'un décodeur peut être simple, à partir des portes logiques ET, OU et NON, en prenant l'exemple suivant : un décodeur est nécessaire pour convertir du code décimal codé binaire (CDCB) sous une forme telle que l'une des dix DEL s'allume, celle qui correspond à la valeur décimale du code. Il s'agit d'un circuit qui doit produire l'inverse de l'exemple de l'encodeur.

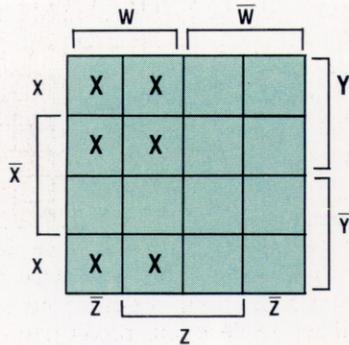


Les codes décimaux codés binaires sont les représentations binaires à 4 bits des chiffres décimaux 0-9. Le décodeur comportera donc 4 lignes en entrée. Du fait que toutes les combinaisons de deux lignes parmi les quatre peuvent être mises en mode passe-haut, il y aura

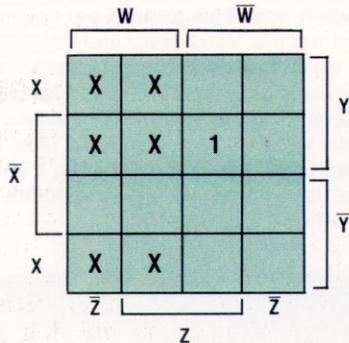
16 entrées possibles. Nous ne retiendrons que les 10 premières et qualifierons les 6 dernières « d'entrées invalides » (X).

La table 1 est la table de vérité. Abordons maintenant séparément les dix sorties. Il apparaît que l'expression booléenne pour chaque sortie doit être constituée de quatre éléments comme pour la table 2.

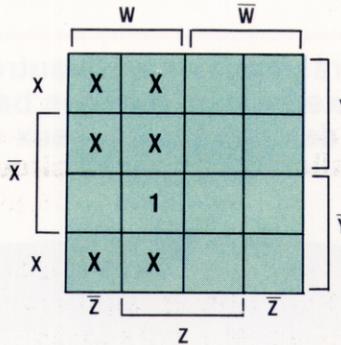
Des simplifications sont possibles : du fait que les conditions d'invalidité pour les données en entrée sont les mêmes pour les 10 sorties.



En étudiant tour à tour chaque sortie, on s'aperçoit qu'il est possible de les simplifier. Prenons par exemple la sortie pour le chiffre 3. La table de Karnaugh montre que cette dernière peut figurer sous la forme d'une boucle.



De la sorte, l'expression pour cette sortie peut être simplifiée en $\bar{X}.Y.Z$. Si l'on prend la sortie 9 comme deuxième exemple :



Il est possible d'écrire une boucle qui regroupe trois des conditions d'invalidité pour les entrées. Elle illustre l'expression booléenne $W.Z$. Vous pouvez vérifier que les termes simplifiés en sortie sont ceux représentés par la table 3.

Il ne nous reste plus qu'à construire le diagramme du circuit logique des dix expressions booléennes. Du fait que chaque entrée est utilisée à la fois sous sa forme normale et sous sa forme négative, il est plus aisé d'établir le circuit à partir de huit lignes parallèles pour les termes et leur négatif. Chacune des dix sorties peut alors être obtenue par branchement sur les lignes appropriées et en utilisant des portes ET. Le diagramme complet du circuit du décodeur est donné ci-dessous.

Exercice 6

1) Vous devez créer un encodeur à trois entrées devant générer un 1 pour les données en entrée : 011, 101, 110 ou 111. Le résultat devra être 0 pour toutes les autres combinaisons d'entrées.

- a) Établissez la table de vérité pour l'encodeur.
- b) Écrivez une expression booléenne pour le résultat en sortie et simplifiez-la.
- c) Tracez le circuit de l'encodeur.

Table 1

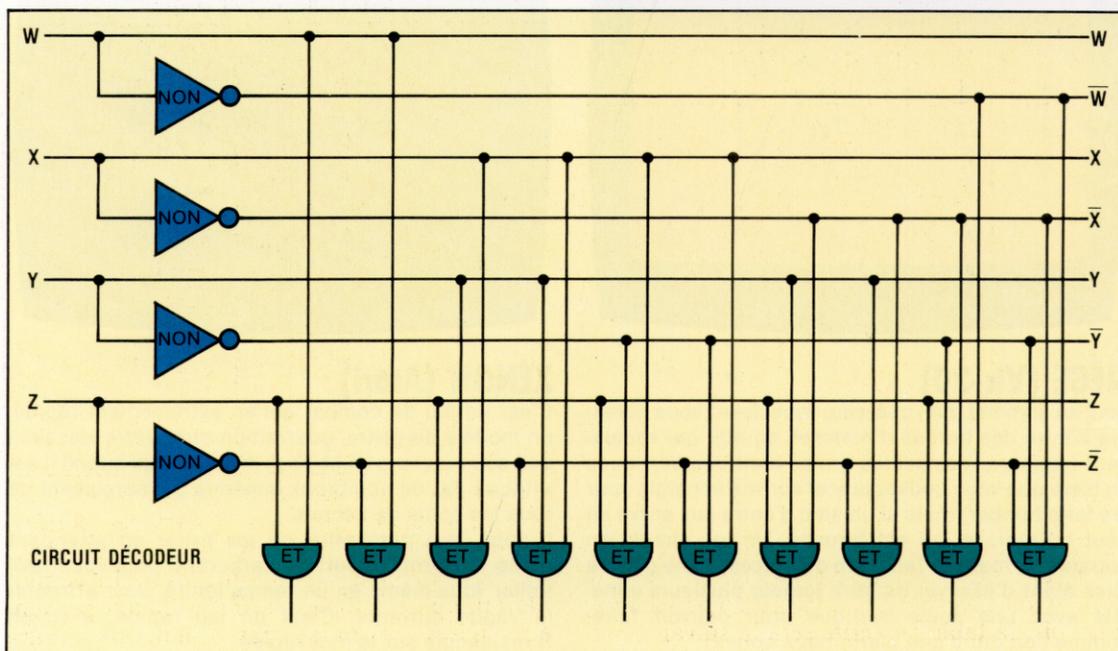
Entrées				Chiffre code
W	X	Y	Z	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X

Table 2

Chiffre code	Expression booléenne
0	$\bar{W}.\bar{X}.\bar{Y}.\bar{Z}$
1	$\bar{W}.\bar{X}.Y.Z$
2	$\bar{W}.X.\bar{Y}.\bar{Z}$
3	$\bar{W}.X.Y.Z$
4	$\bar{W}.\bar{X}.Y.\bar{Z}$
5	$\bar{W}.X.\bar{Y}.Z$
6	$\bar{W}.X.Y.\bar{Z}$
7	$\bar{W}.X.Y.Z$
8	$W.\bar{X}.\bar{Y}.\bar{Z}$
9	$W.\bar{X}.Y.Z$

Table 3

Chiffre code	Expression booléenne
0	$\bar{W}.\bar{X}.\bar{Y}.\bar{Z}$
1	$\bar{W}.\bar{X}.\bar{Y}.Z$
2	$\bar{X}.Y.\bar{Z}$
3	$\bar{X}.Y.Z$
4	$X.\bar{Y}.\bar{Z}$
5	$X.\bar{Y}.Z$
6	$X.Y.\bar{Z}$
7	$X.Y.Z$
8	$W.\bar{Z}$
9	$W.Z$



Liz Dixon

Combats

Nous présentons quatre autres jeux de combat, dont on sait l'attrait qu'ils exercent, à côté des batailles spatiales et des labyrinthes, auprès des mordus des jeux vidéo.



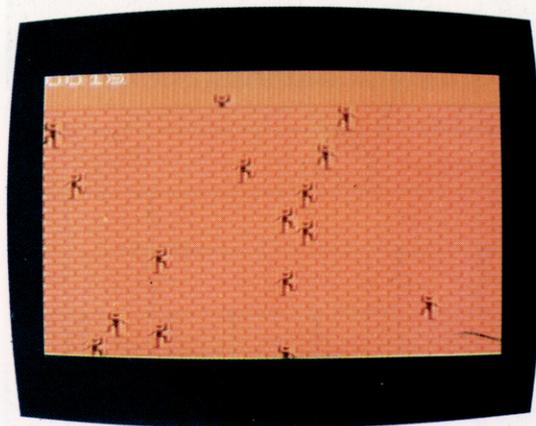
COMBAT ZONE (Spectrum)

Vous conduisez un char et examinez le paysage qui vous entoure par le périscope. Les obstacles sont représentés en perspective par des formes géométriques simples ainsi que les chars de l'adversaire que vous combattez. Un écran radar pour vous permettre de localiser l'ennemi et une mire de visée figurent à l'écran. Le char peut se déplacer à droite, à gauche, en avant et en arrière. Attention aux obus adverses qui sont difficiles à éviter. Un jeu de qualité moyenne, très classique dans sa conception.



ASTROBLAST (Dragon)

Vous êtes en croisière aux limites de la galaxie quand, soudain, vous êtes attaqué par une infinité de vaisseaux étrangers. Vous devez alors détruire quatre escadrilles d'astronefs sans vous ravitailler en carburant et en énergie de protection. Puis vous devez mettre hors d'état de nuire le vaisseau amiral ennemi pour pouvoir vous ravitailler et faire face à quatre autres escadrilles. Pour les débutants, c'est un très bon jeu qui leur permettra de s'introduire dans le monde de la flotte intergalactique.



SIÈGE (Vic-20)

Seul au sommet d'un château, vous devez vous défendre contre des hordes d'hommes en noir qui escaladent les murs. Vos seules armes sont des barriques de bière que vous pouvez lancer sur vos ennemis pour les faire tomber; mais si un seul d'entre eux arrive en haut du mur, le jeu est terminé. Un jeu simple en apparence, mais en fait plein d'astuces; l'une d'entre elles étant d'essayer de faire tomber plusieurs ennemis avec une seule barrique pour pouvoir faire, comme l'on dit, d'une pierre deux coups!



XÉNON (Atari)

C'est un jeu de combat aérien extrêmement rapide, un modèle du genre, pourrait-on dire. Votre vaisseau suit sa route vers un horizon montagneux quand il est attaqué par de nombreux ennemis qui surgissent de tous les coins de l'écran. Il vous faut combattre ou les éviter en attendant qu'ils tombent à court de carburant, puis vous ravitailler vous-même en un temps limité, pour affronter la vague suivante. C'est un jeu rapide, avec un R majuscule sur le mot rapide.



Video star

Les micro-ordinateurs américains Spectravideo possèdent diverses caractéristiques du standard américano-japonais « MSX ». De quoi, peut-être, leur assurer un avenir brillant.

Le Spectravideo 318 est un ordinateur personnel bon marché et certainement de bonne qualité. Il comprend un synthétiseur à trois voix, un graphisme haute résolution avec figures paramétrables, une manette de jeu incorporée et un port cartouche. Le clavier suit de près les spécifications du standard MSX.

Le bloc curseur est ici remplacé par une manette de jeu incorporée qui assume les mêmes fonctions. Lorsque l'utilisateur le veut, il met en place la manette au centre d'un disque de faible diamètre situé en bas à droite du clavier. Celle-ci fonctionne alors comme une manette de jeu positionnant le curseur. Il est également possible de placer le curseur sur une erreur pour corriger un programme. Lorsqu'on retire la manette, le disque assure les mêmes fonctions par le biais d'un simple effleurement. Les points du disque sont ainsi assimilés à des directions pour le curseur.

Le BASIC du Spectravideo est le dernier-né des BASIC Microsoft. Cette version est voisine du

BASIC MSX, lui-même étant une extension du BASIC GW utilisé sur des machines telles que l'IBM PC (Personal Computer). Le développement des programmes est facilité par un éditeur pleine page et par diverses caractéristiques telles que la renumérotation et la numérotation automatiques des lignes. Le BASIC Spectravideo ne comporte pas les commandes « structurées » des nouveaux BASIC telles que WHILE...WEND et REPEAT...UNTIL. Il a néanmoins la commande IF...THEN...ELSE, nécessaire pour écrire des programmes lisibles et efficaces. Enfin, il utilise à plein les capacités graphiques du Spectravideo.

La résolution graphique de l'écran est « honnête », avec 256×192 points en 16 couleurs (même si certains groupes de points prennent nécessairement la même couleur). Cela peut sembler pauvre en comparaison d'autres machines, mais c'est avant tout le degré de contrôle sur l'écran qui fait un bon graphique, et non les seules spécifications. Le 318 comprend toutes les extensions graphiques du BASIC Microsoft,

Le clavier du Spectravideo

Le clavier du modèle 318 suit de près les spécifications du standard MSX.

De conception extraplate, en caoutchouc, il est vaste et permet une bonne qualité de frappe. Il comprend de nombreuses touches, dont les touches habituelles Control, Escape, Tab et Backspace (un caractère en arrière). Il a en outre cinq touches de fonctions (chacune avec deux fonctionnalités), une touche STOP, une touche SELECT, ainsi que des touches d'édition telles que INS, DEL et COPY.

Des figures graphiques telles que des blocs, des pointillés et des lignes sont obtenues en frappant une certaine lettre en conjonction avec la touche LEFT GRAPH ou RIGHT GRAPH. Les formes obtenues sont représentées au clavier.



Ian McKinnell



comme les commandes spécifiques pour tracer des points, des lignes, des pavés, des cercles et des ellipses. La commande PAINT remplit d'une couleur une figure fermée. Si vous voulez intervenir de manière plus fine, les instructions spéciales VPOKE et VPEEK vous permettent d'avoir directement accès à la mémoire d'écran. Il existe également un mini-langage graphique utilisé avec la commande DRAW afin de créer des formes complexes et des dessins. Un rectangle à l'écran peut ainsi être mémorisé en un tableau BASIC par GET, et remis à l'écran par PUT. Ces commandes permettent de créer des motifs, de faire de l'animation, ou d'obtenir des effets spéciaux tels que faire « passer » une image en mode « vidéo inversé » (principe similaire à l'image négative/positive). Enfin, le Spectravideo permet la création de figures programmables. Le composant d'affichage 9929 autorise en effet la création de figures, envahisseurs ou missiles, pour des jeux.

L'instruction ON SPRITE GOSUB vous permet de disposer des « pièges à événements ». Le programme reconnaît alors les occurrences d'un certain « événement » et réagit en conséquence. Dans le cas d'un jeu où deux figures graphiques entrent en collision, le programme passe directement sur un sous-programme traitant précisément de la collision. Cela peut servir aussi à détacher l'impact d'un missile sur un vaisseau spatial. L'utilisation de ces points d'arrêt facilite la tâche du programmeur en lui évitant d'avoir à envisager toutes les occurrences des situations susceptibles de se présenter. La programmation s'en trouve simplifiée, et l'exécution du programme raccourcie.

Un son de bonne qualité est une nécessité absolue pour un micro-ordinateur. Le composant son du Spectravideo comporte trois voix et toute une gamme d'effets spéciaux. Les résultats obtenus peuvent être tout à fait spectaculaires, même si les sons les plus complexes ne peuvent être obtenus à partir du BASIC. Le son est transmis au téléviseur, ce qui donne naturelle-



Manette de jeu

Une manette incorporée remplace les quatre touches fléchées habituelles pour le déplacement du curseur. On peut retirer la manette du disque sensible où elle est fichée. En effleurant un certain point du disque on provoque le mouvement du curseur dans la direction correspondante. L'avantage de ce système disque/manette est de pouvoir susciter un déplacement en diagonale en appuyant sur deux directions à la fois.

Ian McKinnell



Port cartouche

Les cartouches viennent s'enficher sur le dessus du boîtier dans un socle résistant.

ROM

Le BASIC Spectravideo réside sur deux composants ROM de 16 K.

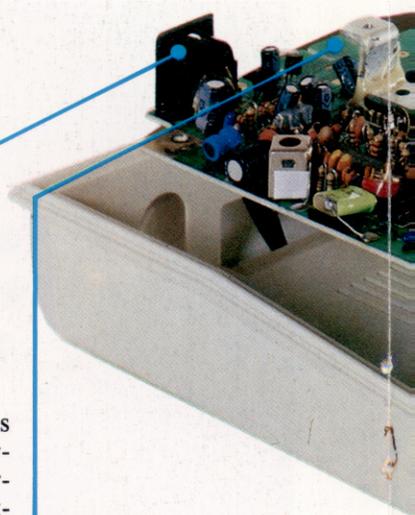
Connexion d'extension

Unité centrale

Le microprocesseur est le Z80.

Sortie moniteur

Un modulateur distinct vient se connecter ici pour piloter un téléviseur. Cela permet au Spectravideo d'accepter divers standards de télévision en logeant des adaptateurs spécifiques.



Port cassette

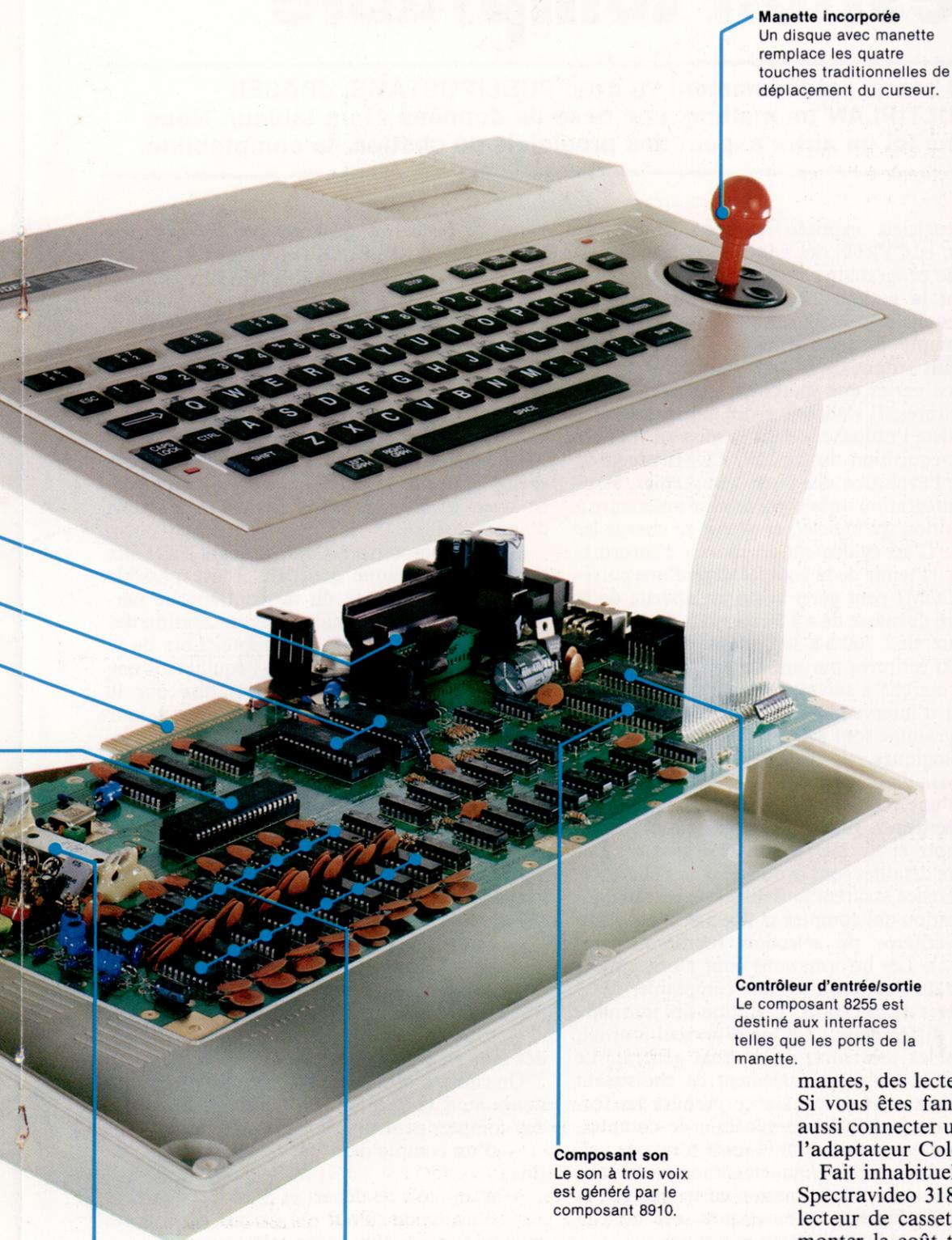
Un connecteur latéral fait office d'interface pour le lecteur de cassettes Spectravideo.

ment un contrôle du son plus efficace et plus fiable. Le Spectravideo dispose de peu d'interfaces : deux ports manette de jeu, un port cartouche, un port cassette et une connexion d'extensions. Il existe des périphériques intéressants, mais ils sont vendus à part et sont chers. Pour connecter un périphérique, vous aurez besoin en outre d'une « mini-boîte d'extension ». Cette dernière vous permettra d'ajouter une option, généralement une extension mémoire de 16 ou 64 K. La « super-boîte d'extension » vous permettra de connecter jusqu'à sept périphériques supplémentaires, par l'intermédiaire de fentes à cartes imprimées, de manière similaire au système Apple. Vous pourrez alors connecter une autre carte mémoire, des interfaces d'impri-



Le grand frère

Le Spectravideo 328 est une version plus sophistiquée du 318, avec clavier indépendant, 80 K de RAM et logiciel de traitement de texte incorporé. Plus cher, il s'adresse aux débutants désireux d'aller plus loin en améliorant ultérieurement leur système.

**Manette incorporée**

Un disque avec manette remplace les quatre touches traditionnelles de déplacement du curseur.

Contrôleur d'entrée/sortie

Le composant 8255 est destiné aux interfaces telles que les ports de la manette.

Composant son

Le son à trois voix est généré par le composant 8910.

Composant d'affichage

Un déperditeur de chaleur, ajouté au dernier moment, cache le composant d'affichage 9929. Ce genre de substitut est souvent nécessaire sur les premières versions d'une machine.

RAM

16 composants RAM fournissent les 32 K de RAM.

SPECTRAVIDEO 318

PRIX

**

DIMENSIONS

410 x 220 x 80 mm.

UC

Z-80.

MÉMOIRE

32 K de RAM, dont 12 pour les programmes BASIC.
32 K de ROM.

AFFICHAGE

Texte sur 24 lignes de 40 colonnes, mode graphique haute résolution 256 x 192 sur 16 couleurs avec figures programmables.

INTERFACES

Connecteur d'extensions, port cartouche, deux ports manette.

LANGAGE DISPONIBLE

BASIC.

CLAVIER

Boîtier caoutchouc à pression avec touches de fonction et d'édition et manette incorporée pour contrôler le curseur.

DOCUMENTATION

Un peu sommaire et sujette aux erreurs, mais l'arrivée d'autres machines MSX devrait encourager des versions indépendantes.

FORCES

Sa version BASIC, ses caractères graphiques programmables, un clavier type machine à écrire complet et la manette incorporée; sa capacité à recevoir de nombreuses extensions et à évoluer vers un système complet de gestion sous CP/M.

FAIBLESSES

Petit nombre de logiciels, absence d'interface d'imprimante en standard.

mantes, des lecteurs de disques et des modems. Si vous êtes fanatique des jeux, vous pourrez aussi connecter un périphérique très intéressant, l'adaptateur Coleco de jeux.

Fait inhabituel pour un ordinateur récent, le Spectravideo 318 requiert sa propre marque de lecteur de cassettes. Cela a pour effet de faire monter le coût total du système mais présente l'avantage de l'homogénéité de conception et donc d'une plus grande fiabilité : l'accès est ainsi plus rapide.

Un des atouts de cet ordinateur est ses vastes possibilités d'extensions. Lorsque vous lui ajoutez la super-boîte d'extension, 64 K de RAM, la carte « 80 colonnes » et un lecteur de disque, vous disposez d'un petit système de gestion sous CP/M, avec accès aux logiciels professionnels.



Gestion comptable

Nous avons précédemment vu avec PUBLIPOSTAGE, dBASEII et MULTIPLAN un mailing, une base de données et un tableur. Nous voyons ici un autre aspect des progiciels de gestion, la comptabilité.

Les progiciels exposés ici sont COMI de Birdy's, et CYRUS des éditions du Logiciel. Il s'agit de programmes de comptabilité générale assurant la gestion comptable et la tenue de comptes. Ils sont établis sur la base du dernier plan comptable, celui de 1982. COMI propose des écrans ordonnancés s'adaptant à des domaines aussi variés que journaux, sociétés, comptes, écritures. Il s'adresse à des non-spécialistes et il assiste l'utilisateur dans la mise en route et dans l'acquisition du logiciel. COMI prend en compte l'évolution des plans comptables, assurant l'intégration de la comptabilité antérieure à l'acquisition du logiciel, et prend en charge les reports. C'est également un moyen d'information sur la tenue de la comptabilité d'une entreprise. COMI peut gérer selon la capacité de la mémoire de masse de 4 à 50 sociétés, de 10 à 100 journaux, de 1 700 à 8 300 comptes et de 31 200 à 91 200 écritures par an. Le logiciel n'impose pas de marche à suivre, de trajets de consultation ou d'intervention. Les différentes parties du programme sont accessibles sans contraintes chronologiques.

La base de la comptabilité est la création en saisie d'une société, des journaux, des comptes et des écritures. Elle permet l'édition d'extraits de compte et du plan comptable général. Des tableaux détaillés sont proposés et des contrôles de cohérence assurent la fiabilité de la saisie. La consultation des comptes se fait librement selon divers critères de sélection (numéro, date, cumul...). Les informations sont alors traitées permettant l'édition des états comptables selon les critères de son choix. L'édition des journaux est très différenciée : édition d'un seul journal, de tous les journaux, du journal général. Le grand livre s'obtient également en choisissant ses critères. Enfin, la balance prendra les formes suivantes : balance générale de comptes, balance des comptes dont le solde n'est pas nul, balance des comptes uniquement mouvementés.

COMI assure l'information en temps réel, la gestion des libellés automatiques, des mouvements comptables mensuels et annuels.

Il est chaînable à d'autres logiciels Birdy's, de comptabilité analytique, de facturation stock-clients, d'achats (stock-fournisseurs), de gestion du personnel. Un autre logiciel de la même source est destiné au traitement et à l'édition du bilan, du compte de résultat, ainsi que de la clôture de l'exercice.

COMI fonctionne sous CP/M, MS/DOS et est implémenté sur de nombreux ordinateurs,

parmi lesquels on peut relever les marques IBM, DEC, APPLE, OLIVETTI, H.P., NCR, VICTOR, SHARP, CANON, LEANORD, TANDY...

CYRUS des éditions du Logiciel s'adresse plus particulièrement à des commerçants, des artisans, des professions libérales, des petites entreprises et des cabinets comptables. Il est destiné à Apple et couvre tous les aspects du suivi comptable pour des besoins moyens.

Une disquette de démonstration présente des exemples de comptabilité que l'on mettra à profit avec le manuel de référence et le guide d'apprentissage.

Le plan comptable est listé donnant accès aux comptes. A chaque journal, il est possible d'associer un compte dit de contrepartie permettant en fin de journal une centralisation des mouvements passés sur ce compte. Lors de la saisie des écritures comptables, l'équilibre d'une écriture est obligatoire. On peut dire que le système vérifie ainsi la saisie. La mise à jour pour les comptes a lieu en temps réel. L'impression de la liste des écritures se fait dans l'ordre de saisie.

Mise à jour en temps réel

Lors de la consultation des comptes, les mouvements peuvent apparaître selon les critères suivants :

- compte de départ et compte terminal;
- date de départ et date de fin;
- numéro de pièce de départ et numéro de pièce de fin;
- tous les mouvements, les mouvements soldés seulement, ou non soldés.

Un compte peut être examiné pour un ou plusieurs mois et avec ou sans cumul. La balance des comptes peut se faire avec sélection :

- d'un compte de départ et d'un compte de fin;
- d'un mois de départ et d'un mois de fin;
- d'un mode cumul qui permet d'éditer les sous-totaux par classe, sous-classe...;
- de tous les comptes ou des comptes mouvementés seulement.

Pour l'édition des journaux, la sélection porte sur le journal de début et le journal de fin, le mois à éditer compte tenu des reports antérieurs, du classement des écritures selon la date ou le numéro de la pièce.

La situation d'un journal peut être obtenue pour un ou plusieurs mois, avec ou sans cumul.

Point de vue

Les amateurs d'informatique ne sont pas toujours très à l'aise face aux mathématiques. Programmeur enthousiaste, les maths vous ont laissé de pénibles souvenirs. Lisez ce qui suit.

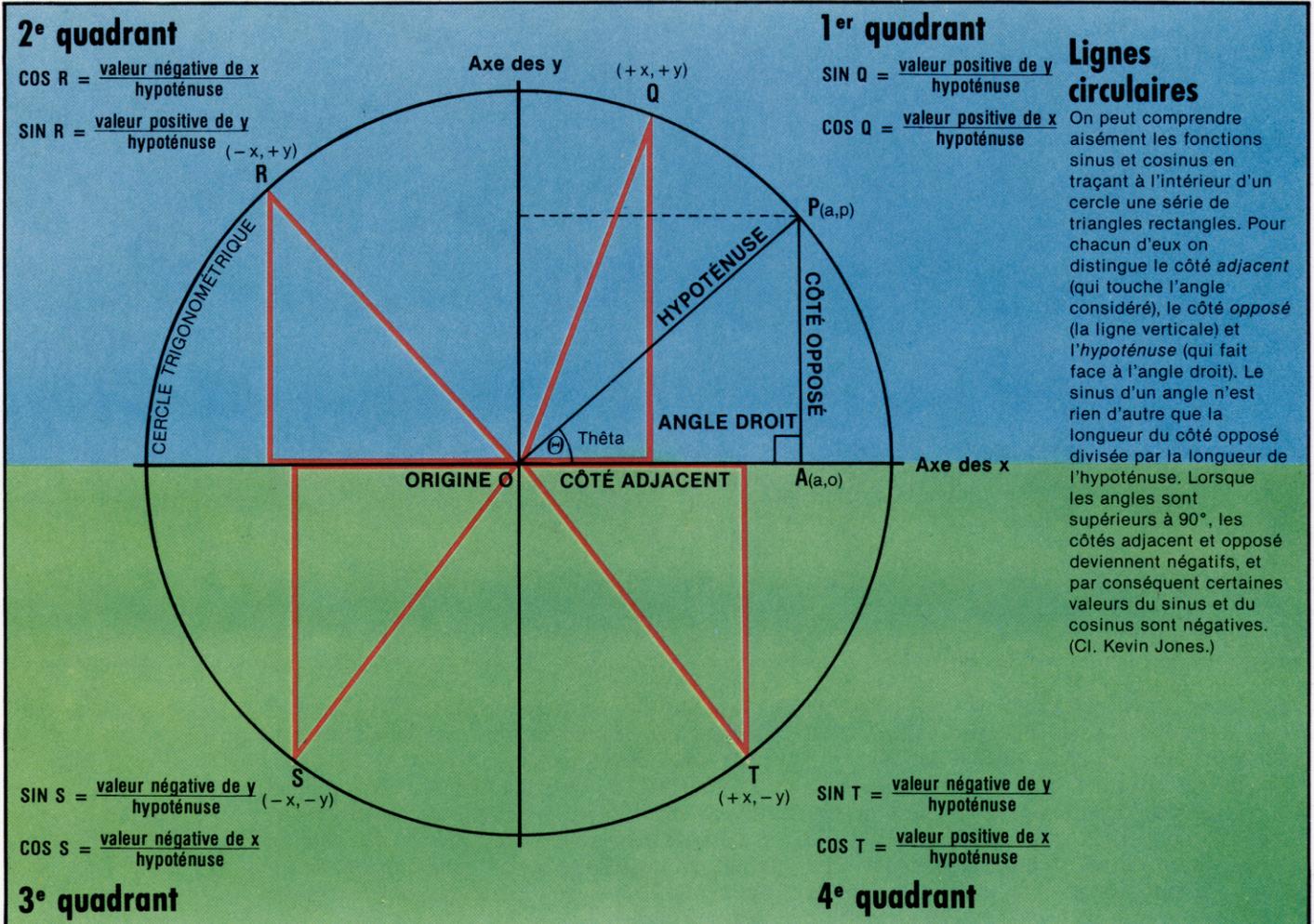
De quelles mathématiques les programmeurs ont-ils réellement besoin? Tout dépend, on s'en doute, du type de programme sur lequel ils travaillent. Le BASIC utilisé par de nombreux micro-ordinateurs comporte des instructions relatives au graphisme : PLOT, FILL, CIRCLE, LINE, COLOUR, INK, PAPER, etc.; le déplacement ou la rotation sur l'écran de figures simples n'est donc pas un véritable problème. Il arrive cependant qu'il soit nécessaire d'employer des fonctions trigonométriques comme Cos, Sin et Tang : mais fort heureusement le BASIC en est largement pourvu. Peut-être tous ces termes n'ont-ils aucun sens pour vous. Patience — tout cela sera expliqué très bientôt.

Les choses se gâtent quand on en arrive aux statistiques : le BASIC n'est généralement pas équipé de fonctions particulières qui facilite-

raient la manipulation des données. Si vous écrivez un programme devant déterminer le résultat des prochaines élections, ou la nature exacte du rapport entre les yeux bleus et la réussite aux examens, vous devrez mettre au point vous-même les procédés de calcul indispensables.

Dans le même ordre d'idées, si vous créez des jeux, ou un cours de dactylographie, vous vous heurterez au problème du délai de réponse, qui est fondamental dans ces deux domaines. Là encore le BASIC, à de rares exceptions près ne peut vous venir en aide. Il ne dispose pas de méthodes fiables de calcul du temps. Notre série d'articles consacrés aux mathématiques abordera donc trois grands domaines : la trigonométrie, la statistique, la mesure du temps.

Les étudiants en mathématiques eux-mêmes se demandent parfois quel rapport la trigono-



métrie peut bien avoir avec le monde réel. Mais la « trigo », comme on l'appelle, est le lien entre la géométrie euclidienne, qui manipule des points, des lignes et des courbes, et l'algèbre, qui brasse des variables aux valeurs inconnues afin de parvenir à des solutions mathématiques. Prenons l'exemple de la parabole : c'est une courbe aux propriétés très utiles, qu'on peut découvrir à l'aide de papier millimétré, d'une règle, d'un crayon et d'un rapporteur. Il est pourtant bien plus simple de partir du fait qu'elle peut être représentée par la formule algébrique $y = x^2$.

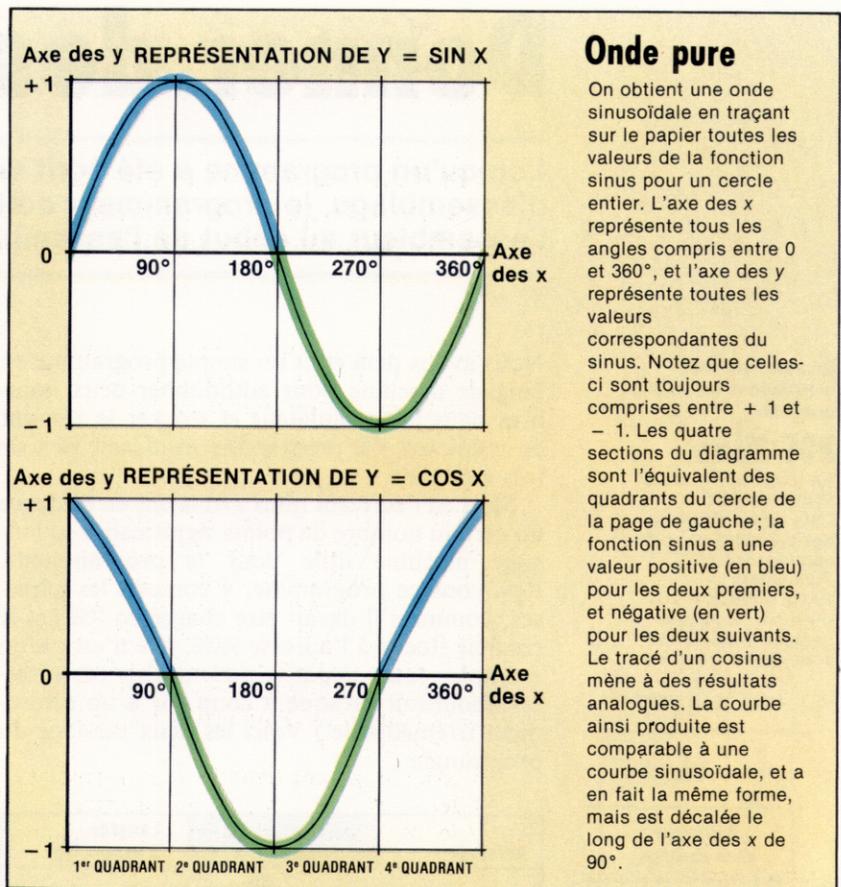
Nous pouvons ainsi calculer la valeur de chacun des points de la courbe, sans jamais être obligés de tracer celle-ci. Une solution de type algébrique se prête infiniment mieux à un traitement informatisé que le dessin de figures ou de diagrammes.

Le but des deux premiers articles de cette série est de vous donner un bref aperçu des concepts trigonométriques, et de vous montrer comment il est très facile de les intégrer dans des programmes en BASIC.

Calcul de rapports

Cosinus et sinus sont deux moyens de déterminer les relations entre les côtés d'un triangle rectangle; ils sont aussi étroitement liés au cercle. En effet tout triangle rectangle peut être dessiné de façon à être contenu dans un cercle, dit « cercle trigonométrique », dont le rayon est égal à l'hypoténuse du triangle considéré. Les dimensions réelles n'ont pas d'importance; ce sont les rapports qui comptent. L'illustration de gauche montre une ligne qui a subi une « rotation » de 35° . Par convention le point de départ de cette translation est l'axe horizontal, qu'on appelle *axe des x*, tandis que l'angle de rotation est appelé *thêta*, et noté à l'aide de la lettre grecque θ . Il suffit de tracer une ligne projetant un point de la circonférence sur l'axe des *x* pour obtenir un triangle rectangle. La rotation se fait dans le sens inverse des aiguilles d'une montre.

On définit le cosinus de θ comme étant le rapport entre le côté *adjacent* du triangle (celui qui est placé sur l'axe des *x*), et l'*hypoténuse* (le rayon du cercle). Si l'on prend par exemple un cercle dont le rayon est de 15 mm, et que l'on mesure avec précision le côté adjacent du triangle qui s'y inscrit, on obtiendra un résultat approché de 12,287 28 mm. Divisée par 15, cette valeur nous permet d'obtenir 0,819 152, qui est la valeur du cosinus d'un angle de 35° . Si vous disposez d'une calculatrice pourvue d'une touche COS, essayez donc d'entrer 35 COS. Vous devriez obtenir une valeur de 0,819 152 204 4. Vous pouvez au besoin vous reporter à des tables trigonométriques. Ce rapport reste constant quelle que soit la dimension du cercle; que son rayon soit de un centimètre, de un mètre ou de une année-lumière, le côté du triangle placé sur l'axe des *x* sera toujours, approximativement, d'une longueur égale à 0,82 fois celle du rayon.



Onde pure

On obtient une onde sinusoïdale en traçant sur le papier toutes les valeurs de la fonction sinus pour un cercle entier. L'axe des *x* représente tous les angles compris entre 0 et 360° , et l'axe des *y* représente toutes les valeurs correspondantes du sinus. Notez que celles-ci sont toujours comprises entre + 1 et - 1. Les quatre sections du diagramme sont l'équivalent des quadrants du cercle de la page de gauche; la fonction sinus a une valeur positive (en bleu) pour les deux premiers, et négative (en vert) pour les deux suivants. Le tracé d'un cosinus mène à des résultats analogues. La courbe ainsi produite est comparable à une courbe sinusoïdale, et a en fait la même forme, mais est décalée le long de l'axe des *x* de 90° .

On peut bien entendu donner à θ d'autres valeurs, et il apparaît alors que $\cos \theta$ (cos est l'abréviation de cosinus) est toujours compris entre 0 et 1. Si toutefois θ dépasse 90° , $\cos \theta$ aura une valeur négative. En effet sa détermination est liée à la valeur de la coordonnée *x* (projection sur l'axe des *x* du point d'intersection de l'hypoténuse et de la circonférence). Par convention on déclare que le point d'origine de la rotation a sur l'axe des *x* la valeur 0. Tous les points situés sur la gauche auront donc une valeur négative. Pour la même raison, les cosinus des angles compris entre 180° et 270° seront également négatifs.

Le *sinus* d'un angle est une notion très semblable à celle de cosinus, à ceci près qu'elle met en jeu l'axe des *y* (l'axe vertical). Si θ est égal à 0, le côté adjacent du triangle aura la même longueur que l'hypoténuse. La coordonnée de P sur l'axe des *x* sera toujours égale à 1 (puisque $1/1 = 1$), mais la coordonnée sur l'axe des *y* aura la valeur 0. $\sin \theta$ est compris entre 0 et 1 pour toute valeur de θ allant jusqu'à 90° . Dans le deuxième quadrant du cercle, il reste positif, mais décroît de 1 à 0 à mesure que θ approche de 180° . Il est négatif pour toutes les valeurs de θ supérieures à 180° (sauf pour 360°).

Pour vous amuser, essayez de déterminer les relations qui peuvent exister entre le sinus et le cosinus d'un angle. A titre d'exemple, remarquez que si θ est égal à 45° , $\cos \theta$ est égal à $\sin \theta$! N'oubliez pas que l'hypoténuse a pour valeur 1.



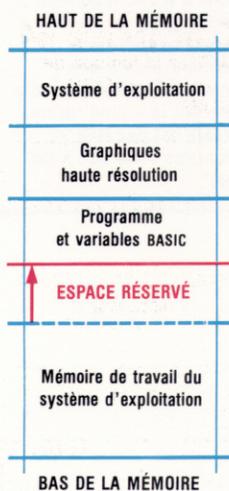
Donner des ordres

Lorsqu'un programme a été écrit sous forme de langage d'assemblage, le programmeur doit donner ses directives à l'assembleur au début de l'assemblage.

Où localiser les instructions en langage machine ?

BBC Micro

En mode direct entrez :
PRINT PAGE
Cela donne l'adresse hex du début de votre espace réservé.
Puis entrez :
PAGE = PAGE + N
où N est le nombre décimal d'octets que vous désirez réserver.



Nous avons déjà écrit un simple programme en langage machine pour additionner deux nombres dans l'accumulateur et stocker le résultat en mémoire. Ce programme ne faisait rien de très étonnant.

Mais en l'écrivant nous avons mis en évidence un certain nombre de points significatifs en langage machine utile pour le programmeur. Revoiyons ce programme, y compris les adresses, comme s'il devait être chargé en \$0000 et le résultat stocké à l'adresse \$0009. (Ce n'est qu'un exemple : toute tentative pour utiliser ces adresses aboutirait presque à coup sûr à un écrasement irrémédiable.) Voici les deux versions du programme :

Adresses	Langage machine	Langage d'assemblage
Z-80		
0000	A7	AND A
0001	3E 42	LD A,\$42
0003	CE 07	ADC A,\$07
0005	32 09 00	LD BYTE1,A
0008	C9	RET
6502		
0000	18	CLC
0001	A9 42	LDA #\$42
0003	69 07	ADC #\$07
0005	8D 09 00	STA BYTE1
0008	60	RTS

Notez que la quatrième instruction (qui stocke le contenu de l'accumulateur en \$0009) dans les deux programmes ne spécifie pas une adresse de destination dans la colonne de langage d'assemblage. Au lieu de cela, elle utilise une adresse symbolique, BYTE1. En code machine, nous voyons cependant que l'opc pour le « transfert du contenu de l'accumulateur » est suivi de 0900, forme à deux octets lo-hi de l'adresse \$0009.

C'est un autre aspect du processus de traduction (ou assemblage) du langage d'assemblage en langage machine. De même que nous utilisons des mnémoniques à peu près compréhensibles (STA et RET, par exemple, au lieu de codes hex tels que 8D et C9) pour rendre les programmes plus faciles à lire et à écrire, nous utilisons souvent des symboles tels que BYTE1 au lieu d'adresses ou de nombres hex comme \$0009. Il en va de même lorsqu'on initialise des variables par des constantes dans un programme BASIC, et le raisonnement est exactement pareil dans les deux cas — le programme est rendu plus lisible, les possibilités d'erreurs à l'écriture sont rédui-

tes et le programme devient plus aisé à manipuler. Par exemple, si l'on change la première instruction où la valeur constante est attribuée à la variable, cela aura pour effet que la nouvelle valeur sera utilisée automatiquement dans tout le programme, sans avoir à réécrire d'autres parties du programme.

C'est facile à comprendre quand il s'agit de programmation BASIC, mais où est l'équivalent, dans notre programme en langage d'assemblage, de l'instruction BASIC LET BYTE1 = \$0009? Actuellement, cette instruction n'existe pas. En fait, lorsque nous assemblons le langage d'assemblage en langage machine, nous devons penser à le faire nous-mêmes. Si toutefois nous utilisons un programme assembleur pour faire l'assemblage, nous pourrions ajouter une instruction d'attribution au début du programme. Voici la version Z-80, bien que de telles directives d'assembleur puissent être employées avec les deux processeurs :

Z-80		
0000		BYTE1 EQU \$0009
0000	A7	AND A
0001	3E 42	LD A,\$42
0003	CE 07	ADC A,\$07
0005	32 09 00	LD BYTE1,A
0008	C9	RET

BYTE1 est placé dans une colonne pour lui tout seul, appelée zone de label, dont nous parlerons plus loin. Dans la zone opc, on utilise un nouveau mnémonique (EQU, pour « égal »); et dans la zone opérande, on donne la valeur à attribuer à BYTE1 (dans ce cas, \$0009).

Il est important de noter que, bien que EQU apparaisse dans la zone opc et ressemble à un mnémonique, ce n'est pas un mnémonique de langage d'assemblage et il n'appartient ni au jeu d'instruction du Z-80, ni à celui du 6502. On l'appelle un *pseudo-op* ou *pseudo-instruction d'assembleur*. EQU dit au programme assembleur que « chaque fois qu'il trouve le symbole alphanumérique précédent (BYTE1, dans ce cas), il doit le remplacer par la valeur suivant la pseudo-instruction (\$0009 ici) ». Rappelons que, lorsque nous utilisons un programme assembleur, nous n'écrivons que le programme en langage d'assemblage soit en fichier sur bande ou disquette, soit directement au clavier, puis appelons le programme assembleur pour le convertir en programme en langage machine. La sortie d'un assembleur est généralement un listage d'assemblage tel que ceux que nous avons pro-



duits, plus le programme en langage machine consistant simplement en une chaîne d'octets hex. Le langage machine peut être sauvegardé pour un usage ultérieur.

L'assembleur peut encore accomplir d'autres tâches que nous avons faites à la main — attribuer des adresses à chaque ligne de programme, par exemple. Un autre pseudo-op, ORG, fait cela pour nous. Voici comment il est ajouté :

Z-80		
0000		ORG \$A000
A000		BYTE1 EQU \$0009
A000	A7	AND A
A001	3E 42	LD A,\$42
A003	CE 07	ADC A,\$07
A005	32 09 00	LD BYTE1,A
A008	C9	RET

Remarquez que l'adresse attribuée à la première ligne du programme est \$0000, mais celle de la ligne suivante est \$A000, ce qui résulte de l'instruction ORG. En outre, aucun octet de langage machine n'apparaît sur les lignes contenant des pseudo-op, précisément parce qu'ils ne font pas partie du programme et ne doivent pas être traduits en langage machine. De ce fait, les pseudo-op différent d'un assembleur à l'autre. EQU, par exemple, est parfois remplacé par « = » et ORG par « . ». Cependant l'effet est le même et nous continuerons à utiliser ORG et EQU comme s'ils étaient standard.

Vous avez pu remarquer, à propos des pseudo-instructions d'assembleur, que nous avons utilisé un pseudo-op presque depuis le début de la série : « \$ », la marque hex. Ce n'est qu'une directive pour dire à l'assembleur que ce qui suit doit être traité comme un nombre hexadécimal. De même, « # » introduit dans le cours précédent est la marque de « donnée immédiate », signifiant que ce qui suit est une quantité absolue plutôt qu'un pointeur ou un symbole. En fait, nous pourrions considérer le langage d'assemblage lui-même comme une série de pseudo-op. Certes, rien ne vous empêche d'inventer vos propres mnémoniques pour les instructions de langage machine, à condition qu'il y ait une correspondance bi-univoque entre les deux ensembles. Un programme assembleur très populaire pour le Vic-20 ne fait que ceci : il utilise une version non standard de langage d'assemblage 6502, essentiellement pour formater l'écran à 22 colonnes du Vic.

Dans ce cours, nous continuerons à utiliser, comme nous l'avons fait jusqu'à présent, les mnémoniques de langage d'assemblage publiés par les fabricants de microprocesseurs — mais il est bon de se rappeler de temps à autre que tout ce que nous appelons langage machine est symbolique. L'UC est indifférente à tout, sauf aux configurations de tension en ses contacts d'entrée/sortie, et donc la manière de décrire ces configurations n'est qu'une question de convention.

Abandonnons pour le moment les pseudo-op et retournons à notre programme. En particulier, comparons les traductions en instructions

LDA et LD A avec leurs traductions dans nos programmes précédents. Nous écrivons :

```
AD ?? ??   LDA $????   (6502)
3A ?? ??   LD A,($????) (Z-80)
```

ce qui signifie « charger l'accumulateur à partir de l'octet d'adresse ???? ». L'opc traduisant « charger l'accumulateur » est \$AD(6502) ou \$3A (Z-80). Comparez cela avec la seconde ligne du programme, qui incorpore l'instruction « charger l'accumulateur avec la valeur immédiate \$42 ». Ici les opc sont \$A9 et \$E3, pour le 6502 et le Z-80 respectivement. Mais pourquoi différents ? Peut-être l'avez-vous deviné. Bien que nous effectuions le même type d'opération (transfert de données dans l'accumulateur) dans les deux programmes, la source de ces données diffère. C'est pourquoi, pour l'UC, ce sont des opérations différentes avec des opc différents.

Dans la première version, la donnée doit être chargée à partir d'un octet dont l'adresse est donnée. Rien n'est précisé sur le contenu de cet octet ; on demande simplement à l'UC de copier ces contenus dans l'accumulateur. Les trois octets de langage machine, AD ?? ?? et 3A ?? ??, sont décodés par l'UC dans le sens d'« interpréter les deux octets suivant cet opc comme l'adresse absolue de la source de données ».

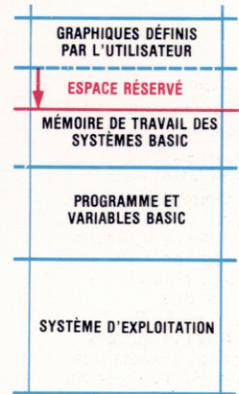
Dans la seconde version, la donnée à charger dans l'accumulateur est en fait dans l'octet suivant l'opc, de sorte que les deux octets de langage machine, A9 42 et 3E 42, sont décodés par l'UC comme « interpréter l'octet suivant cet opc comme la source de données ». Quelque chose dans l'opc (A9 ou 3E) dit à l'UC de charger l'accumulateur à partir de l'octet suivant. Comme son registre d'instruction contient toujours l'adresse de l'instruction suivante à effectuer, l'UC peut calculer l'adresse de l'octet source, puis effectuer une simple opération de « chargement de l'accumulateur à partir d'un octet adressé ».

Cela confirme le fait que les opérations de l'UC sont la plupart du temps des procédures très simples. Toute une classe de ces opérations (environ un cinquième de tout son répertoire) consiste essentiellement à copier une donnée d'un octet adressé dans l'un ou l'autre de ses registres internes. Ces opérations « primitives » impliquent toutes une seule tâche — transférer des données de la mémoire vers un registre de l'UC — et tout ce qui distingue une instruction d'une autre est le format dans lequel l'adresse de l'octet source se présente.

Enfourer cela dans des micro-opérations de l'UC peut prêter à confusion au début, mais permet de donner un aperçu général par la suite. C'est superflu si l'on ne veut qu'écrire des programmes en langage d'assemblage pour aller plus vite. Pour cela, il suffit de s'initier rapidement aux instructions, d'apprendre quelques astuces de programmation et de s'y mettre. Si toutefois vous voulez comprendre ce que vous faites, vous souhaitez faire plus que simplement ajouter un nouveau langage de programmation à votre répertoire.

Spectrum

En mode direct entrez :
LET RTOP = PEEK(23730) +
256 * PEEK(23731)
LET RTOP = RTOP - N
PRINT « RTOP = » ; RTOP
où N est le nombre
d'octets que vous
désirez réserver pour
votre programme. Votre
espace réservé
commence à 1 + RTOP.



Commodore 64

Utilisez le tampon de
cassette de \$033C à
\$03FB



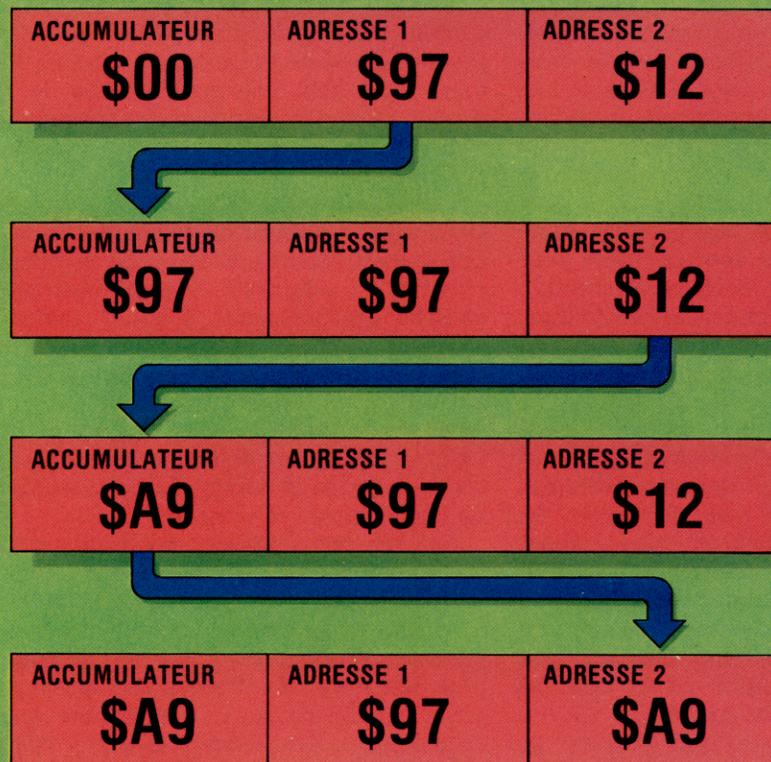
Attention

Vous devez ajuster ces pointeurs de mémoire immédiatement après avoir mis en circuit votre machine lorsqu'il n'y a pas de programme BASIC en mémoire.

**Exercice d'assemblage**

- Dans la case de droite, nous avons donné la version en langage d'assemblage d'un petit programme. Assemblez le programme en langage machine et déterminez les adresses.
 - Quelle instruction manque à ce programme ?
 - Quel est l'effet du programme sur les registres et la RAM concernés ?
 - Que signifie le terme « donnée immédiate » ? Quelles autres sortes de données y a-t-il ?
 - Si **BYTE1** dans le programme est traité comme une adresse, sur quelle page de RAM apparaît-il ?
- N.B.** — Les valeurs données dans ce programme ne sont que des exemples : si vous voulez l'exécuter, vous devrez choisir des emplacements et des valeurs convenant à votre machine.

Adresse	Langage machine	Langage d'assemblage	
6502			
		START	EQU \$A000
		BYTE1	EQU \$45
		BYTE2	EQU \$38
			ORG START
			LDA #BYTE1
			CLC
			ADC #BYTE1
			STA BYTE1
			ADC #BYTE2
			STA BYTE2
Z80			
		START	EQU \$A000
		BYTE1	EQU \$45
		BYTE2	EQU \$38
			ORG START
			LD A, BYTE1
			AND A
			ADC A, BYTE1
			LD (BYTE1), A
			ADC A, BYTE2
			LD (BYTE2), A

L'effet des instructions en langage machine

Instruction 1 : chargez l'accumulateur à partir de l'adresse 1
 Z-80 : LD A,(ADR1)
 6502 : LDA ADR1

Instruction 2 : ajoutez le contenu de l'adresse 2
 Z-80 : ADC A,(ADR2)
 6502 : ADC ADR2

Instruction 3 : stockez le contenu de l'accumulateur à l'adresse 2
 Z-80 : LD (ADR2),A
 6502 : STA ADR2

L'effet des instructions de transfert de données telles que LDA ADR1 ou LD (ADR2), A est toujours de copier les contenus de l'emplacement source dans l'emplacement destination. Les contenus de ces emplacements sont donc écrits par remplacement; l'emplacement source n'est pas affecté par le transfert de données.



Soleil levant

De toutes les sociétés japonaises qui fabriquent des ordinateurs, celle qui est peut-être la plus intéressante compte parmi les moins bien connues : SORD, entreprise de taille moyenne.

Le nom SORD a été formé par contraction des mots SOftware et haRDware; il est révélateur de la vocation de cette société, qui a toujours accordé autant d'importance à la mise au point de logiciels qu'à celle des machines elles-mêmes.

La firme a connu des débuts modestes. Son président, Takayoshii Shiina, quitta l'université pour entrer à Rikei Industries, société relativement prospère, cotée au deuxième rang, à la bourse de Tokyo, en 1967. Il s'employa à redéfinir la politique commerciale de la société.

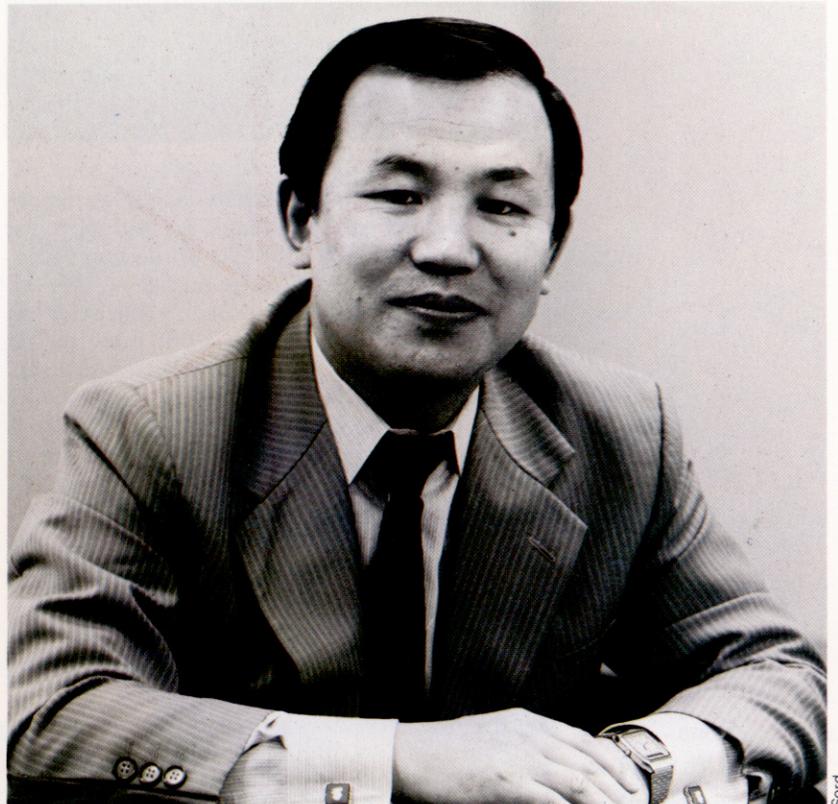
C'est en 1970 que Shiina et un de ses amis furent prêts à créer leur propre société. Ils fondèrent donc SORD, avec un capital de 20 000 F. Shiina n'en continua pas moins à travailler pour Rikei jusqu'au mois de décembre de la même année. Les premières réalisations de SORD furent un testeur logique bon marché, ainsi que des travaux de programmation réalisés sous contrat.

Dès 1971, le chiffre d'affaires de SORD atteignait un niveau enviable, toujours grâce, principalement, à des programmes réalisés à la demande. Deux ans plus tard, la société commença à produire du matériel, et à la fin de 1974, elle commercialisait un lecteur de disquettes avec une interface qu'elle avait elle-même réalisée. Ensuite apparut le SMP-80/20, l'un des premiers ordinateurs japonais dérivés de l'Intel 8080.

Le SMP-80/20 connut un succès remarquable, et les commandes affluèrent. Mais Shiina avait de grands projets d'expansion. En 1977, il vendit 20 % des parts de la société à Toppan, l'une des plus grosses imprimeries du pays. SORD disposa ainsi de moyens financiers qui lui permirent de mettre au point des logiciels adaptés à sa gamme sans cesse plus variée d'ordinateurs, dont, en 1981, PIPS.

PIPS était une gamme de logiciels en avance sur son époque. C'était en effet l'un des premiers logiciels intégrés au monde, et c'est surtout lui qui permit à SORD d'acquérir une position ferme sur le marché. PIPS combine trois fonctions (tableau, traitement de texte, et base de données) sous une forme telle que même des personnes n'ayant jamais utilisé un ordinateur pouvaient apprendre à s'en servir en quelques heures.

Les programmes d'usage courant, dans le domaine des affaires, aussi bien en Europe qu'aux États-Unis, n'avaient pas encore fait leur apparition au Japon, pour deux sortes de raisons : tout d'abord, les Japonais répugnant

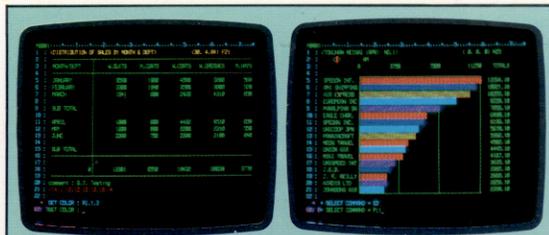


Takayoshii Shiina

généralement à importer quoi que ce soit qu'ils puissent faire eux-mêmes, on ne trouvait au Japon que fort peu de logiciels d'origine américaine. En second lieu, les Japonais sont de piètres linguistes. Si un manuel de Digital Research, en CP/M, constitue un document difficile à déchiffrer pour un lecteur européen moyen, c'est pour un Japonais... du chinois! L'absence de programmes d'applications disponibles sur le marché offrit donc à SORD un créneau qu'elle s'empressa d'exploiter.

SORD commença à développer la série M200 d'ordinateurs dérivés du Z-80, puis la série M23. Ensuite, celle des M343 permit de satisfaire une plus large gamme de besoins. Le M5 fut créé pour le marché national des jeux électroniques, et l'on a aujourd'hui la série M68, qui comporte à la fois des processeurs Z-80 et Motorola 68000, les petits ordinateurs d'affaires M243, et le M285 à 32 bits qui permet l'emploi de programmes VAX-11 et est bien adapté à la conception assistée par ordinateurs.

Aucun autre fabricant au monde ne propose une gamme aussi étendue de matériels sur le



PIPS

PIPS (Pan Information Processing System) est le seul logiciel de gestion de SORD. C'est un programme général destiné à différentes tâches de la gestion des affaires, depuis la comptabilité jusqu'à la représentation graphique. Il démontre une curieuse approche du marketing à la japonaise car il n'est disponible que sur certaines machines de cette société. Cela peut donner une bonne raison pour acheter un ordinateur SORD, mais l'intérêt du produit est limité.

marché, et chaque ordinateur SORD est vendu avec le logiciel correspondant. Le seul reproche que l'on puisse faire à SORD est de ne vendre, pour ainsi dire, que du matériel « clés en mains », sans qu'il soit possible d'acheter un ordinateur seul. Les clients n'ont donc pas la possibilité d'utiliser des logiciels autres que ceux que propose SORD.

La société japonaise a tenté de remédier à cette carence avec le système SB-80, proposé en option avec la série de ses ordinateurs M23, dérivés du Z-80. Le SB-80 est l'équivalent du système CP/M 2.2 de Digital Research, et permet l'utilisation du logiciel CP/M, pour la première fois sur un ordinateur SORD. SORD a fait un autre effort dans la même direction en diffusant le p-Système de l'UCSD (University of California at San Diego).

Aux utilisateurs qui se contentent de ses propres logiciels SORD propose plusieurs BASIC performants, son système SB-80, PIPS, et un système de traitement de texte Wang.

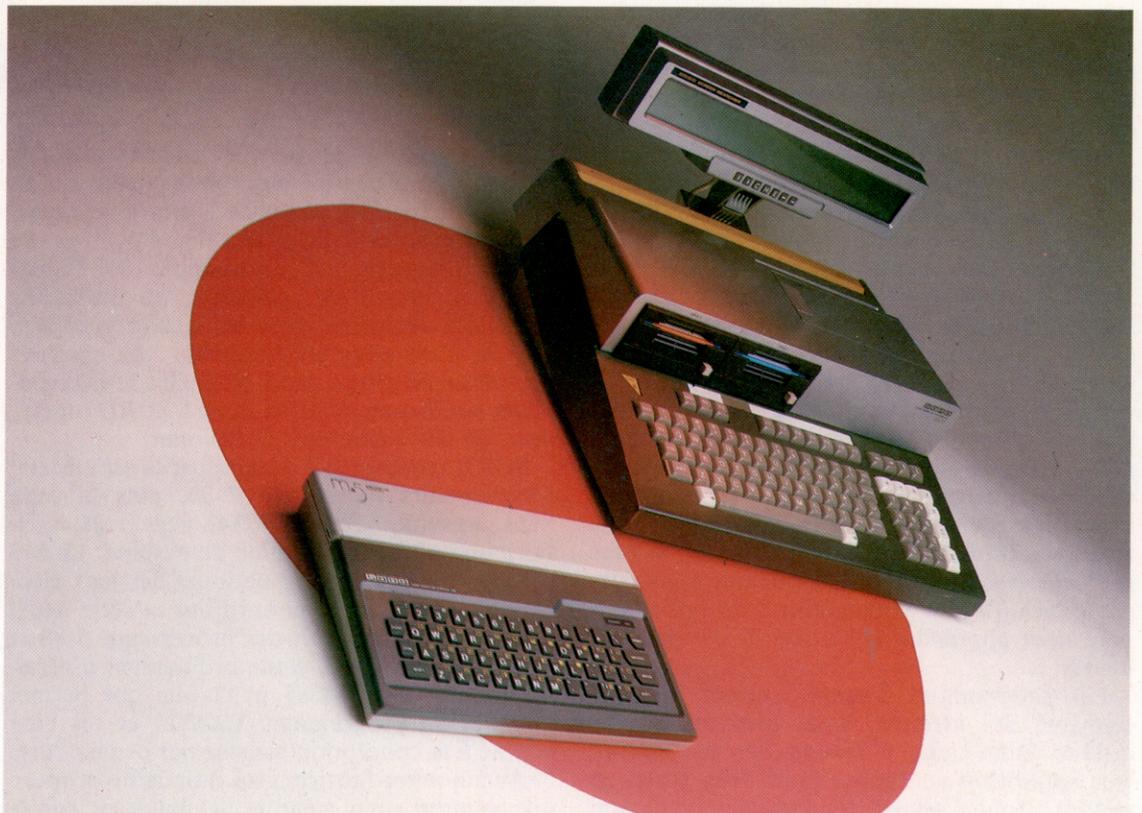
SORD s'est également efforcé d'emboîter le pas aux constructeurs de micro-ordinateurs tels que DEC, en proposant des systèmes dont la mise en œuvre est possible sur du matériel domestique. Dans le même temps, une idée s'imposait chez tous les autres constructeurs d'ordinateurs : la standardisation et la compatibilité entre produits de firmes différentes constituent les meilleurs moyens de développer le marché. SORD a, jusqu'à présent, manifesté une certaine lenteur à s'adapter à cette tendance.

SORD n'en continue pas moins à se signaler par des innovations dans le domaine des logiciels, mais on lui reproche souvent de ne fournir à ses clients que des documentations insuffisantes, concernant ses produits. Ces deux dernières années, il est vrai, SORD n'a pas ménagé ses efforts en vue d'offrir une bonne documentation sur PIPS et sur son système de traitement de texte. Il semble qu'une documentation approfondie sur ses ordinateurs fasse encore défaut, et SORD n'encourage guère les entreprises qui pourraient créer des logiciels utilisables sur ses machines.

Les années qui viennent seront probablement critiques pour SORD, qui doit faire face à la concurrence de puissants rivaux japonais, comme à celle d'IBM. Pourtant Takayoshii Shiina continue à croire à l'avenir des entreprises de taille modeste, et il est vrai qu'il a su donner à SORD une réputation internationale.

Les options SORD

Les machines les plus connues de SORD sont l'ordinateur domestique M5 et le portable d'affaires M23P. Ce dernier met en œuvre de nouveaux standards pour un portable en offrant des disquettes Sony et un affichage à cristaux liquides de 80 colonnes. (Cl. Ian McKinnell.)



ASTRONOMIE

l'encyclopédie ATLAS du ciel



L'astronomie est la plus ancienne des sciences, elle est aussi la plus novatrice dans l'image du monde qu'elle propose aux hommes du XX^e siècle.

De l'exploration des planètes aux hypothèses les plus hardies sur la marche de l'Univers, l'encyclopédie ASTRONOMIE, éditée par ATLAS, fait le point sur l'histoire et les découvertes des professionnels du ciel.

10 volumes - 2408 pages - 1800 photographies en couleurs - 1300 dessins et schémas en couleurs.

ASTRONOMIE s'adresse aussi aux amateurs, et les guide dans leurs travaux : observation d'étoiles variables ou filantes, de comètes, de planètes...

L'encyclopédie ASTRONOMIE fera date dans l'histoire de l'édition, pour avoir rassemblé autant de données sur l'infini qui nous entoure. Elle est rédigée par des hommes de science, qui vous racontent le grand roman du monde.

EDITIONS
ATLAS

Protégés par une élégante reliure, vos numéros d'aBc Informatique seront plus faciles à consulter

Pour classer, répertorier, protéger vos fascicules d'ABC Informatique, les Editions Atlas vous proposent des reliures élégantes, sobres, qui s'insèrent parfaitement dans votre bibliothèque. Chacune contient 12 fascicules,

les maintient, les préserve. Un système simple, résistant, vous permet de les assembler facilement. Elles sont en vente en permanence chez votre marchand de journaux. Demandez-les !



Chaque
reliure:

40 FF
295 FB
18 FS

Cod. N.M.P.P. :
6103

EDITIONS
ATLAS