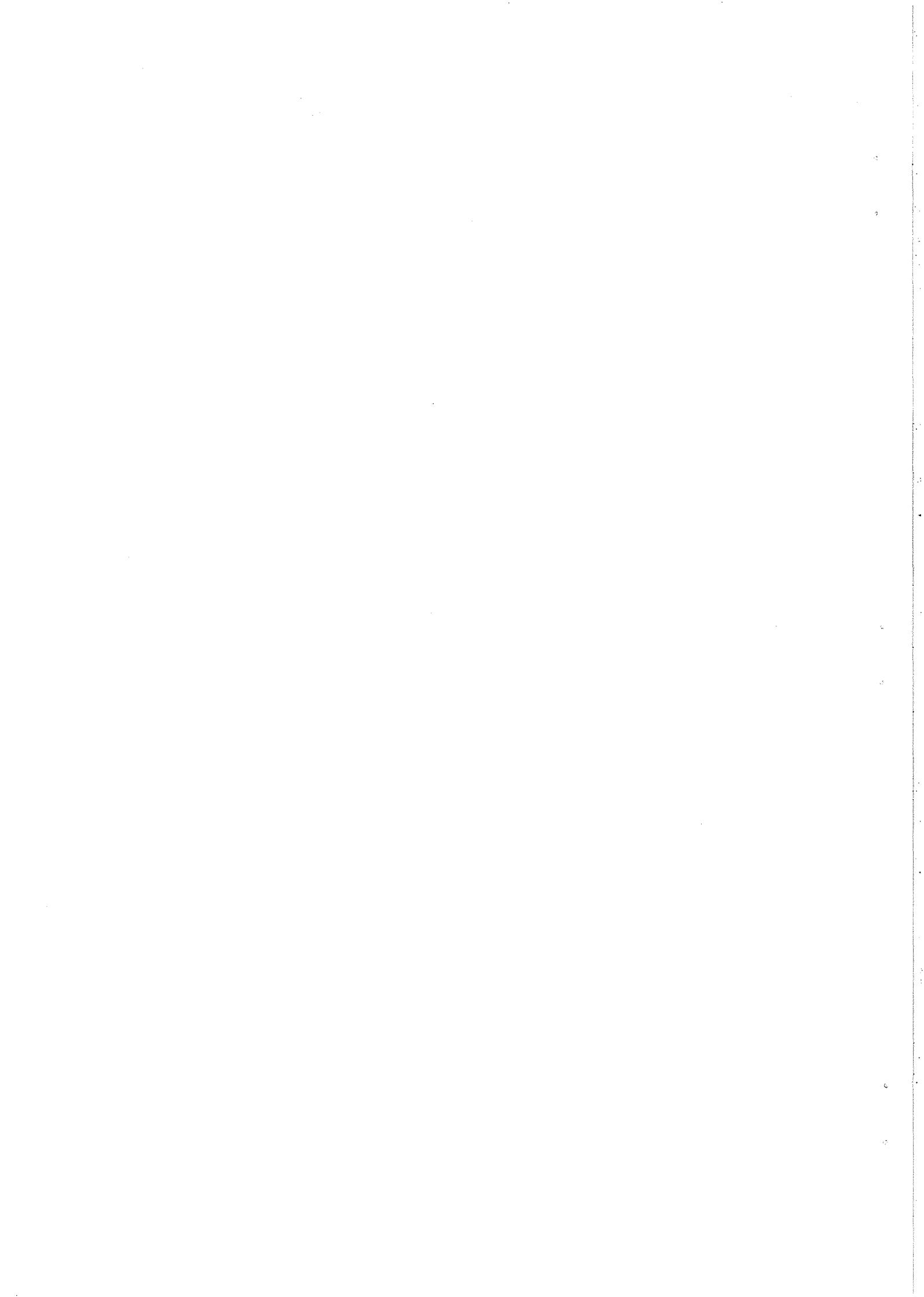


No. 15

avril 1981

**i
n
f
o
r
m
a
t
i
q
u
e
e**
**a
t
i
o
n
s**

**Groupe informatique
de l'enseignement
secondaire**



BILLET DU REDACTEUR

En septembre 1978, désigné par la DES, un groupe d'enseignants crée, dans le cadre du GIDES, une sous-commission ayant pour but l'étude de tout problème gravitant autour des microsystemes (étude des matériels existants, prix, démonstrations, etc.).

Formée actuellement de MM. MOREL, président, ARMICI, ARNOLD, INEICHEN, MESSEILLER, PARISOD, RENAUD, ROSSIER et SERODINO, cette sous-commission "Microsysteme" s'est vue confirmer par la CIDES en janvier 1981 le mandat suivant :

- recenser les besoins en microsystemes dans l'enseignement secondaire;
- donner des préavis techniques dans ce domaine (conseils, faire-part d'expériences, etc.);
- diffuser l'information;
- organiser des démonstrations;
- coordonner les études dans des domaines relatifs à des besoins communs pour divers établissements scolaires.

D'autres questions auraient pu intéresser la sous-commission (assurer la maintenance des microsystemes, établir une liste de tous les produits, développer n'importe quel interface tant du point de vue "hardware" que "software"), mais, faute de moyens suffisants, elle a dû y renoncer pour le moment.

On peut noter que, si à diverses occasions (démonstrations, enquêtes, élaborations de budgets et diffusions de P.V.) la sous-commission a tenu au courant les différentes personnes s'occupant d'informatique dans la République, la publication du présent "Informatique-Informations" constitue une preuve tangible de la quantité et de la qualité du travail fourni.

A côté d'expériences concrètes en matière de développement et d'utilisation des microsystemes, on peut lire une introduction sur l'évolution technologique de l'électronique ainsi que des descriptions diverses concernant les microprocesseurs, les interfaces, le logiciel, etc.

Dans le chapitre "Promesses et réalités", la sous-commission a cherché à montrer qu'il y a encore beaucoup à faire dans les domaines de la documentation, du service après-vente, etc.

Enfin, une importante bibliographie permet à ceux qui n'ont point eu leur content avec ce document d'aller chercher plus amples informations dans les 23 articles de revues ou de livres cités.

En tant que rédacteur d'Informatique-Informations, je me permets de remercier les auteurs des différents articles de ce numéro. Puisqu'ils ont décidé de rester anonymes - bien que, pour les initiés, telle ou telle formule ou tournure de phrase dénonce son auteur -, je les remercie en bloc en souhaitant que le travail accompli par eux serve à tous ceux qui, de près ou de loin, s'occupent et/ou s'intéressent à l'informatique. En conclusion, si la formule du journal et la périodicité de sa parution ont changé, la qualité se veut rester la même.

Aux lecteurs de confirmer d'infirmier cette conclusion. Bonne lecture !

Jean-Pierre STUCKY
ESC Malagnou
7, rue Le Corbusier
1208 GENEVE

1 L'EVOLUTION TECHNOLOGIQUE DE L'ELECTRONIQUE

Depuis la découverte de l'effet diode par Edison en 1883, et de son perfectionnement jusqu'au tube amplificateur, l'électronique a subi deux révolutions technologiques capitales :

En 1948 et 1949, c'est l'invention des transistors, ces éléments semi-conducteurs capables de remplacer les tubes électroniques à vide. Leur faible consommation permet le développement des premières machines digitales réellement commercialisables. A cette époque, et jusque dans les années 60, chaque composant d'un système électronique possède une fonction électrique élémentaire (résistance, condensateur, self, élément actif ou non-linéaire). On parle d'éléments discrets.

Dès 1958, les fabricants de transistors s'avisent qu'il doit être possible de réaliser sur la surface d'une pastille de silicium, plusieurs fonctions électriques élémentaires, l'ensemble de ces fonctions étant interconnectés de manière à réaliser un circuit électronique complet. En 1961, les ingénieurs de Fairchild commercialisent le premier circuit électronique intégré sur une surface de silicium ou circuit intégré (chip).

Ce nouveau composant présente deux caractéristiques importantes :

- a) La fonction électrique réalisée sur la surface du silicium est pratiquement indépendante de ses dimensions. On peut même dire que, lorsque l'on désire traiter de l'information électrique (il n'en est pas de même pour les applications de puissance), plus les dimensions sont réduites et meilleures sont les performances. D'où une tendance vers des intégrations toujours plus grandes. Les premiers circuits intégrés possédaient quelques dizaines d'éléments (circuits SSI ou small scale integration). Puis l'on réalisa quelques centaines d'éléments (circuits MSI ou medium scale integration). L'amélioration des techniques permit quelques milliers d'éléments (circuits LSI ou large scale integration), puis quelques dizaines, voire la centaine de milliers d'éléments sur une pastille de silicium de quelques millimètres carrés (circuits VLSI ou very large scale integration). Nous ignorons quel sera le sigle pour quelques millions d'éléments.
- b) Le prix de fabrication d'un circuit intégré n'est pas fonction du nombre d'éléments contenus sur le chip. D'une manière approximative, on peut dire que le prix d'un circuit intégré double chaque fois que l'on multiplie le nombre d'éléments par dix. Cependant, si le prix de fabrication de circuits de très grande complexité ne varie pas de beaucoup, le prix de l'étude, de la mise en fabrication et de l'élaboration des méthodes de test devient gigantesque. Ce sont des dizaines de millions de dollars.

Si l'on désire que le prix de vente d'un circuit intégré VLSI soit acceptable (<100 fr.), il doit être fabriqué en un très grand nombre d'exemplaires afin de répartir les frais d'étude. Mais plus un circuit devient complexe, plus il devient spécifique à une application particulière, par conséquent, moins il aura de débouchés.

Les domaines d'utilisation des circuits VLSI étant restreints (montres, calculatrices), il était nécessaire de découvrir une structure de circuit électronique qui soit universelle, de manière à augmenter son domaine d'application.

En 1971-1972, les ingénieurs d'Intel développent deux circuits intégrés VLSI pour des applications précises, le 4004 pour un calculateur de bureau et le 8008 pour la gestion d'un écran. Le 8008 est un échec pour l'application envisagée (circuit beaucoup trop lent). Mais le monde de l'électronique se rend rapidement compte que la philosophie de ces deux circuits est la clef de l'électronique moderne. C'est le rush sur ces circuits. Intel redéveloppe alors un circuit mieux adapté à son nouvel usage (le 8080), Motorola entre dans la course avec le 6800, puis les autres fabricants de semi-conducteurs. Le microprocesseur est né.

Quels sont les caractéristiques de ce composant électronique "miracle" ?

En ajoutant quelques circuits intégrés (mémoires, entrée/sortie, logique) autour du microprocesseur, on réalise un ordinateur. Or tout le monde sait (ou devrait savoir) que ce que fait l'ordinateur n'est pas fonction de ce dernier, mais du programme contenu dans ses mémoires. L'application du circuit électronique ne dépend donc plus du circuit lui-même, mais de sa programmation. Il devient ainsi possible de réaliser des circuits VLSI suffisamment universels pour être assurés d'une très grande production.

L'étendue des applications de cette nouvelle génération de circuits électroniques devient pratiquement illimitée. Son introduction pour le remplacement d'organes électroniques, électriques ou mécaniques n'est plus qu'une question de prix. Si par exemple, dans une machine à laver, le programmeur électromécanique est trois fois plus coûteux que son équivalent à microprocesseur, il sera automatiquement éliminé par ce dernier.

Actuellement, il n'est aucun domaine qui peut se prétendre à l'abri de l'attaque des microprocesseurs. En effet, un micro-ordinateur complet, avec microprocesseur, 2 kbytes de mémoire morte, 128 bytes de mémoire vive, 20 lignes d'entrée/sortie, le tout intégré sur une seule pastille de silicium, coûte moins de 10 francs lorsqu'il est produit en grande quantité.

Les applications typiques des circuits électroniques à base de microprocesseurs sont :

- a) Le remplacement de tout dispositif électrique, électronique ou mécanique déterminant une séquence plus ou moins complexe en fonction du temps (machine à laver, machine à coudre, allumage de voiture, montres, jouets, etc.).
- b) Le remplacement de tout dispositif commandant certaines actions en fonction de contraintes extérieures (contrôle de processus, chauffage, alarme, jouets, etc.).
- c) Le remplacement de tout système nécessitant - ou qui est simplifié par - une certaine puissance de calcul (calculatrices, balances de

magasin, caisses enregistreuses, appareils de mesure, jeux, etc.).

d) Le remplacement des ordinateurs de table, voire des mini-ordinateurs.

Les conséquences de cette révolution technologique sont difficiles à estimer. Cependant, il faut voir les points suivants :

Diminution générale de la main d'oeuvre :

Si ces nouveaux composants permettent des baisses de prix spectaculaires sur les produits les plus divers, cette baisse correspond toujours à une diminution importante de la main d'oeuvre. Une montre électronique, par exemple, nécessite en moyenne cinq fois moins de main d'oeuvre que son équivalent mécanique. Cette diminution n'est plus compensée par la main d'oeuvre employée pour la création des machines nécessaires à la fabrication des composants.

Modification radicale des techniques traditionnelles :

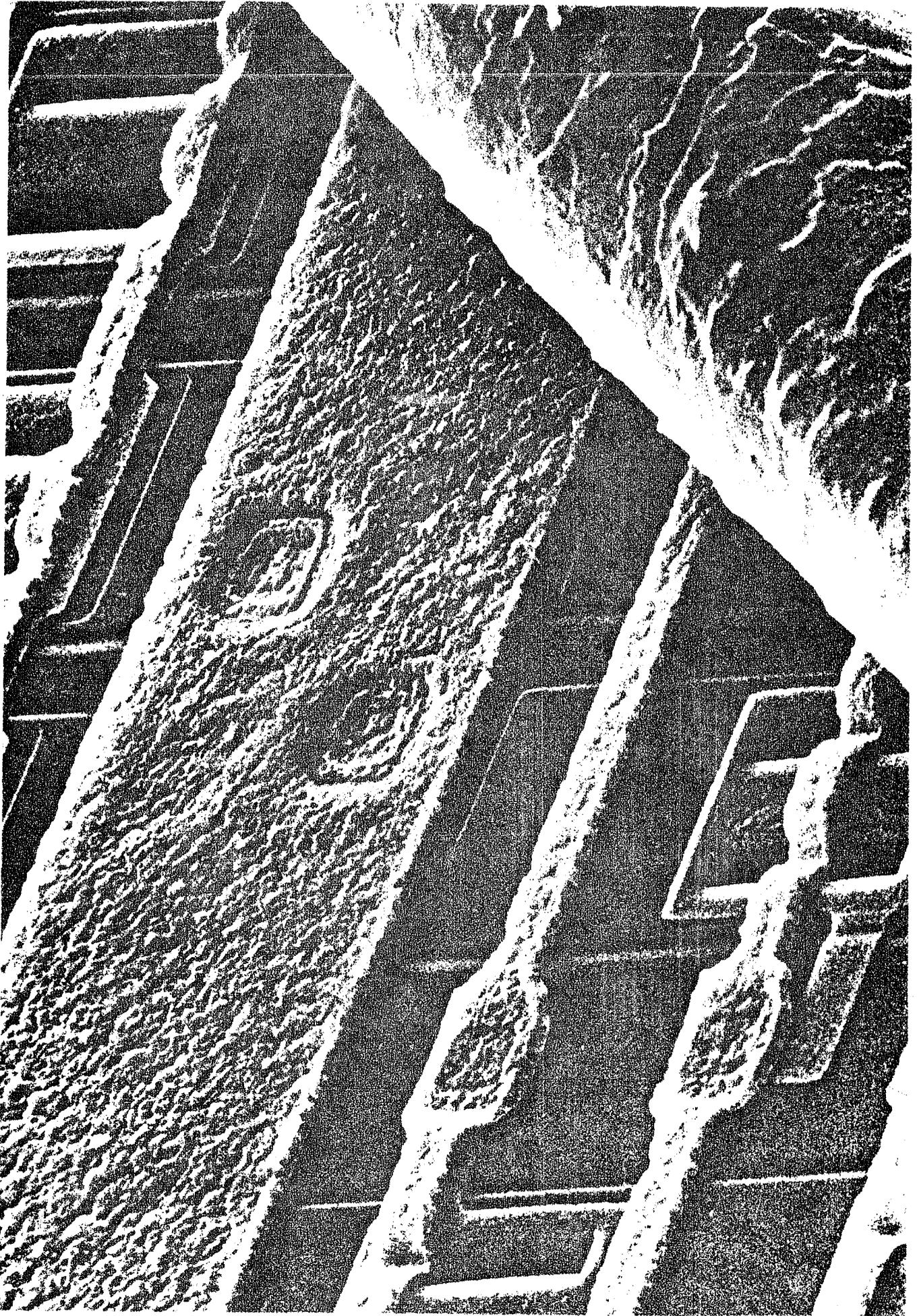
La puissance logique de ces nouveaux composants fait que des techniques qui semblaient immuables sont mises au rang des antiquités.

Le cas le plus spectaculaire est celui de l'horlogerie, mais il n'est pas isolé. Un grand nombre de systèmes mécaniques de très grande ingéniosité sont remplaçables par des systèmes électroniques à base de microprocesseurs. Les calculatrices mécaniques ont été remplacées, les balances commerciales également; les machines à écrire ne garderont pas longtemps leur forme actuelle.

Démocratisation de la puissance d'analyse :

Un système ordinateur de la puissance de l'IBM 1620 des années 60 coûte actuellement moins de 20'000 francs. Cela signifie que tout le monde peut avoir accès à une machine dont la puissance de traitement de l'information est importante.

En résumé, les circuits intégrés VLSI et le microprocesseur sont des composants électroniques extraordinaires, mais qui risquent fort de nous apporter plus de problèmes qu'ils n'en auront résolus !!!



A comparer avec un cheveu

2 STRUCTURE ELECTRONIQUE D'UN SYSTEME A BASE DE MICROPROCESSEUR

Le microprocesseur est un composant électronique dont la fonction est à peu de chose près celle de l'unité centrale d'un ordinateur. Seul, il ne peut strictement rien faire. Il doit pouvoir communiquer avec d'autres composants qui sont des mémoires et des interfaces d'entrée/sortie. L'ensemble de ces circuits formera un micro-ordinateur. (Il faut remarquer qu'actuellement un micro-ordinateur complet peut être intégré sur une seule pastille de silicium de quelques millimètres carrés.)

Un micro-ordinateur - comme du reste les gros ordinateurs - est organisé autour de lignes de communication appelées lignes bus. Ces lignes bus sont subdivisées en trois sous-groupes :

a) Le bus des données :

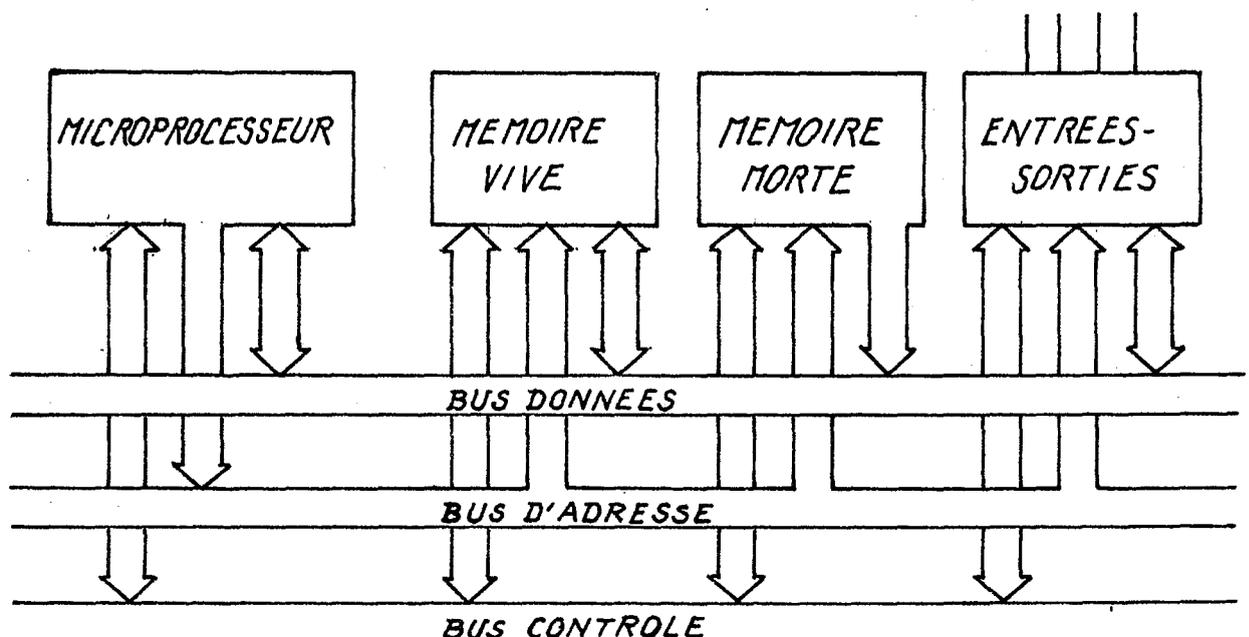
Le nombre de ces lignes est généralement égal à la longueur des mots traités. Ce sont ces lignes qui transporteront les informations entre les circuits associés et le microprocesseur.

b) Le bus d'adresse :

Les lignes transportant les informations étant communes à tous les circuits, le microprocesseur utilise ces lignes d'adresse pour spécifier avec quel élément il désire communiquer. Les microprocesseurs actuels possèdent généralement seize lignes d'adresse, ce qui permet d'adresser $65 \cdot 536$ positions mémoire. Cependant, les microprocesseurs modernes les plus puissants ont jusqu'à 24 lignes d'adresse (adressage de 17×10 puissance 6) et l'on parle de 32 lignes (4×10 puissance 9).

c) Le bus de contrôle :

Ces lignes servent à synchroniser l'ensemble des échanges.



Connectés sur ces lignes de communication, nous trouvons :

2.1 LE MICROPROCESSEUR

C'est l'unité de contrôle principale. Elle va chercher séquentiellement les instructions en mémoire, puis les opérandes, et effectue les opérations logiques entre ces derniers. Trois grandeurs caractérisent la puissance du microprocesseur :

1) la longueur des mots traités.

C'est en quelque sorte le nombre d'informations électriques traitées en parallèle. Dans les bas de gamme, pour les contrôles d'appareils simples, nous avons les microprocesseurs à quatre bits. Les plus courants possèdent huit bits (8080, 6800, Z80, 6500, 6809, etc.). Pour les applications se rapprochant de celles des mini-ordinateurs, nous avons des microprocesseurs à seize bits (8086, Z8000, 9900), voire des 16/32 bits comme le 68000.

2) La richesse des instructions.

La richesse d'instructions d'un microprocesseur est définie d'une part par le nombre d'instructions exécutables, d'autre part par le nombre de modes d'adressage des opérandes. Un jeu d'instructions riche permet d'écrire des programmes concis, faciles à mettre au point et performant lors de l'exécution.

3) La rapidité d'exécution.

C'est le nombre d'instructions exécutées par le microprocesseur en une seconde. Il varie entre quelques centaines de milliers et quelques millions.

2.2 LES MEMOIRES

Ce sont des registres où seront stockés soit des instructions, soit des opérandes. La taille mémoire maximum des microprocesseurs (c'est-à-dire le nombre de registres qui peuvent être adressés directement) est généralement de $65 \cdot 536$ octets (mots de 8 bits). Mais, avec les nouveaux microprocesseurs, elle tend à augmenter jusqu'à plusieurs millions, voire milliards d'octets.

Nous distinguerons les mémoires à accès aléatoire (dans lesquelles le temps d'accès à n'importe quel mot est constant) et les mémoires séquentielles dans lesquelles les informations défilent les unes derrière les autres.

C'est le temps d'accès qui caractérisera les performances d'une mémoire. Les mémoires à accès aléatoire courantes ont des temps d'accès de l'ordre de 250 ns, mais les plus rapides répondent en moins de 40 ns.

Nous distinguerons deux catégories principales de mémoires :

1) Les mémoires mortes (ou à lecture seule ou ROM) :

Ce sont des mémoires qui contiendront un programme spécifique non modifiable, par exemple un interpréteur BASIC, un moniteur ou encore le programme d'application pour un appareil. Leur capacité actuelle maximum par circuit intégré (25 à 100 mm² de silicium) est de huit kilo-octets par chip, mais on parle déjà de 32 koctets.

Nous distinguerons :

- Les mémoires ROM (Read only memory) dans lesquelles les informations sont inscrites à la fabrication du circuit intégré. Pour de très grandes séries, ce sont les plus économiques.
- Les mémoires PROM (Programmable read only memory) dans lesquelles les informations sont inscrites par l'utilisateur, en faisant fondre des fusibles par exemple.
- Les mémoires EPROM (Erasable programmable read memory) dans lesquelles les informations sont inscrites par l'utilisateur en piégeant des charges électrostatiques à l'intérieur d'un isolant. Dans ces dernières, il est possible de les effacer un certain nombre de fois en les exposant à un rayonnement ultraviolet, puis de les reprogrammer. Ce sont les mémoires les plus utilisées pour de petites séries.
- Les mémoires EAROM (Erasable alterable read only memory) semblables aux précédentes, mais effaçables électriquement byte par byte.

2) Les mémoires vives (à lecture et écriture) :

Ce sont des mémoires qui contiendront soit des opérandes, soit un programme temporaire. Leur capacité actuelle maximum par circuit intégré est de 64 kbits par chip, mais on parle du mégabit pour les années 85.

Nous trouverons :

- Les RAM statiques (Random access memory).
Ce sont des mémoires à accès aléatoire, dans lesquelles l'information est contenue dans l'état d'une bascule électronique. Elles perdent l'information mémorisée si la tension d'alimentation disparaît (batteries, condensateurs). Cependant on réalise des RAM dont la consommation est tellement faible que l'on peut les maintenir en permanence sous tension (technologie CMOS).

- Les RAM dynamiques.

Ce sont des mémoires à accès aléatoire, dans lesquelles l'information est contenue dans la charge d'un micro-condensateur. La charge se dégradant en fonction du temps, il est nécessaire de la régénérer périodiquement. Cela complique l'électronique de la mémoire; cependant, elles sont très employées en raison de leur très grande densité d'intégration (quatre fois plus grande que pour les RAM statiques).

- Les mémoires à bulles magnétiques.

Ce sont des mémoires à accès séquentiel, qui gardent l'information, même en l'absence de tension. L'information est contenue dans l'orientation magnétique de domaines microscopiques (quelques microns carrés). Leur capacité actuelle atteint le mégabit et leur vitesse de transfert est de cent kilobits par seconde.

2.3 LES CIRCUITS D'INTERFACE

Pour que le système micro-ordinateur puisse communiquer avec l'extérieur, il est nécessaire d'intercaler un circuit d'interface entre le bus et les lignes externes. Nous distinguerons deux types de communications :

1) Les communications parallèles.

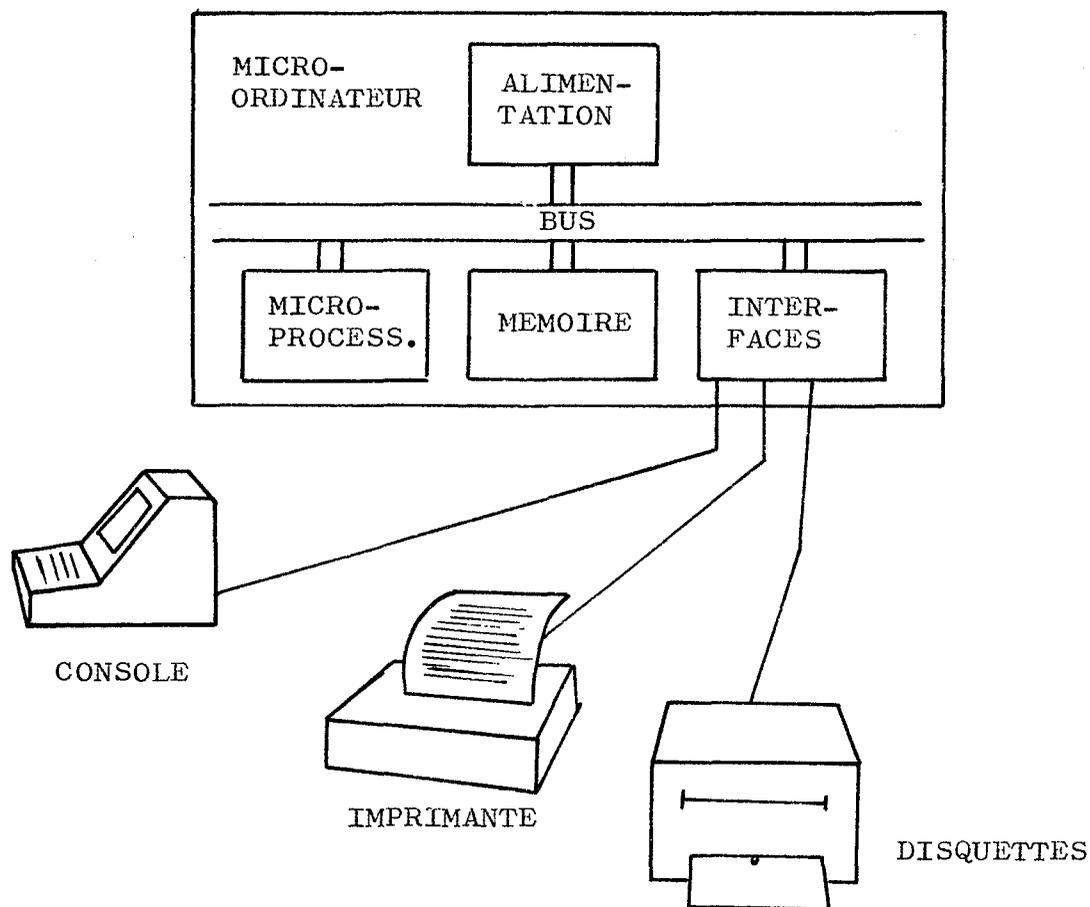
Le circuit d'interface prolonge en quelque sorte les lignes du bus de données. Ce type de transmission est utilisé lorsque le dispositif connecté ne se trouve pas trop éloigné du micro-ordinateur, car le nombre de lignes assurant les liaisons est important (exemple : bus d'instrumentation permettant la connexion d'un certain nombre d'appareils de mesure sur un micro-système).

2) Les communications séries.

Le circuit d'interface transmettra les informations du bus de données les unes à la suite des autres, en série dans le temps. Ce mode de transmission est utilisé lorsque la ligne de transmission atteint une certaine longueur. Elle ne nécessite que deux fils par liaison. C'est le mode de transmission des terminaux, par exemple.

La plupart de ces interfaces sont programmables de manière à pouvoir être adaptées à des applications diverses. Cependant, nous trouvons des interfaces spécialisées. On peut prendre comme exemple les contrôleurs de disques souples qui contiennent pratiquement toute la logique nécessaire à la gestion des enregistrements.

La structure d'un microsysteme complet pourra être la suivante :

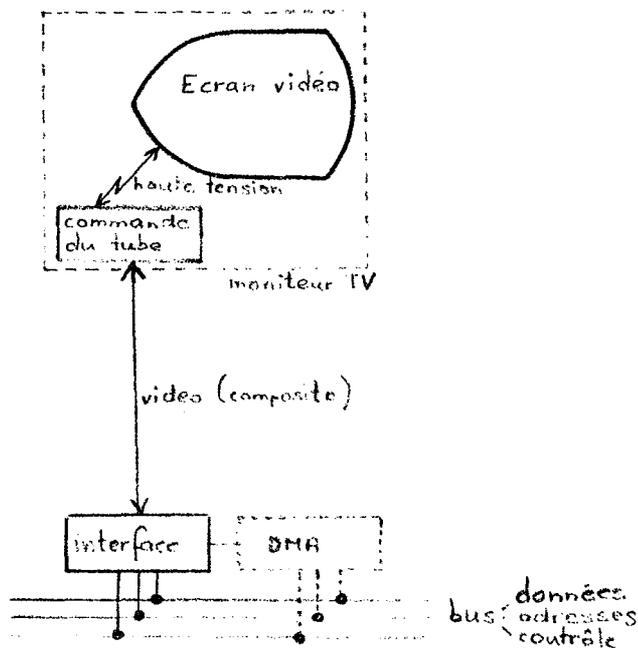


3 LES INTERFACES

Comme nous l'avons vu dans les articles précédents, la structure d'un micro-système est basée sur la notion de bus, espèce de grand boulevard où circulent données, adresses et signaux de contrôle. C'est sur ce bus que viennent se greffer l'unité centrale (microprocesseur), la mémoire et ce que nous avons appelé "périphériques".

C'est précisément ces périphériques que nous allons examiner de plus près dans les quelques lignes qui suivent.

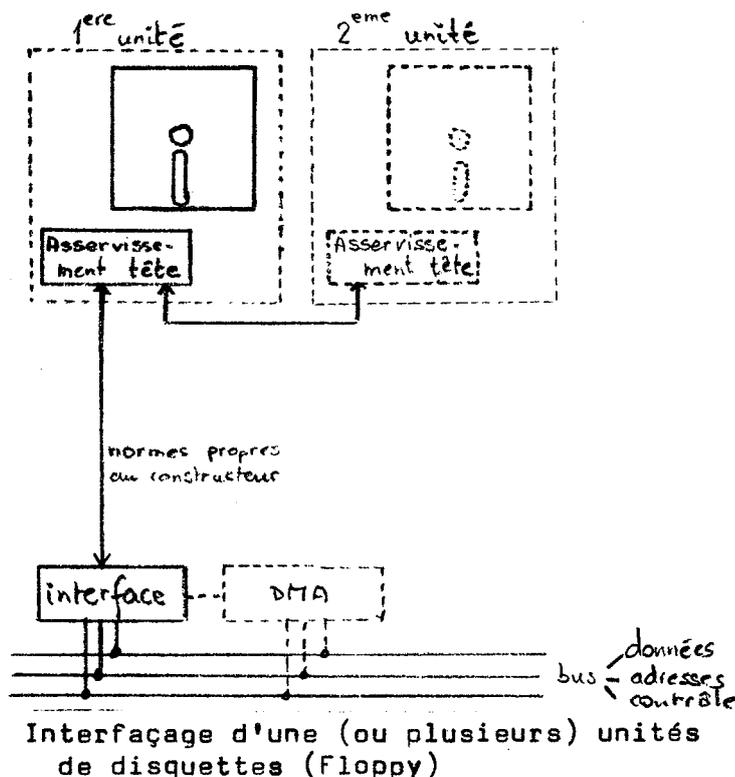
Voyons, à titre d'exemple, le cas d'un terminal avec un clavier et un écran vidéo, et, ensuite, le cas d'un disque (soit "dur", soit souple).



Interfaçage d'un écran vidéo

Il faut noter ici que, lorsque l'on parle d'interface, on ne parle que de l'interface proprement dit, et non de l'électronique se trouvant dans le moniteur de télévision, ni, dans les cas où cela est nécessaire, du module d'accès direct à la mémoire (DMA), qui est souvent considéré comme module séparé.

Le deuxième exemple montre le cas d'une unité de disquettes, qui demande un interface particulier. La première unité que l'on désire connecter va nécessiter un interface, alors que la seconde pourra, en général, être connectée sur le même interface.



"Interfacer" un périphérique semble donc très simple; mais que l'on ne s'y trompe pas : il ne faut pas oublier que même avec un excellent interface, on ne pourra utiliser le périphérique que si, au niveau software, tous les ordres particuliers ont effectivement été donnés, en respectant aussi bien la séquence que les délais. Le sous-programme gérant ces entrées-sorties est appelé "driver".

A priori, il est évident que n'importe quoi, ou presque, peut être connecté à un microprocesseur; mais la connexion d'un périphérique, parfois qualifié d'exotique, laisse supposer que cet appareil permet soit une commande électrique de l'une ou l'autre de ses parties, soit une lecture électrique de son état. Il faudra donc, sur l'appareil même, une première partie d'électronique adaptant les signaux propres à l'appareil aux spécifications de l'interface. Cette électronique se trouvera en général sur l'appareil à commander.

La gestion correcte d'un nouveau périphérique nécessite donc du hardware d'une part, sous forme de plusieurs éléments dans la plupart des cas, et du software pour le gérer, l'utilisateur n'ayant que rarement une idée des commandes exactes pour contrôler ce périphérique. Or, bien souvent, l'adaptation de l'une ou l'autre des parties ne pose que peu de problèmes, mais un bon emploi du tout est relativement rare dès que l'on sort du "standard", pour autant qu'il y en ait un à ce niveau.

4 LE LOGICIEL (SOFTWARE)

4.1 Notion générale

Rappelons qu'un microprocesseur est capable d'aller chercher séquentiellement en mémoire des instructions et leurs opérandes et de les exécuter. Nous appellerons ici "instruction" des tâches simples telles que :

- transférer le contenu d'une position mémoire dans un registre du microprocesseur ou vice-versa
- tester l'état de certains registres, c'est-à-dire examiner leur contenu, et, selon le résultat, exécuter une tâche simple plutôt qu'une autre
- exécuter une opération arithmétique ou logique simple entre deux de ses registres
- "Sauter", de manière conditionnelle ou non, à une adresse mémoire donnée ; c'est grâce à ce genre d'instruction que l'on peut créer des boucles
- tester l'état d'une ligne d'entrée (le microprocesseur doit-il recevoir de l'information ?)
- recevoir de l'information sur une ligne d'entrée
- tester l'état d'une ligne de sortie (le microprocesseur peut-il envoyer de l'information ?)
- envoyer de l'information sur une ligne de sortie.

Il faut remarquer que le jeu d'instructions dépend du microprocesseur et peut être plus ou moins étendu.

En d'autres termes, le microprocesseur peut exécuter automatiquement des séquences d'instructions qui ont été auparavant mises en mémoire. Cela signifie que le même microprocesseur pourra avoir diverses actions selon la séquence d'instructions que l'on aura donnée. Une telle séquence est appelée programme (dans certains cas routines, sous-programme) (on dit alors que le microprocesseur est programmable ; et c'est cette propriété qui le différencie d'autres circuits intégrés en lui permettant d'être utilisé dans des situations très diverses.

Remarque : le microprocesseur exécute un programme. En électronique, le terme de circuit programmable désigne des circuits dont les fonctions peuvent être modifiées en inscrivant diverses valeurs dans des registres internes. Cette modification de fonction étant obligatoirement le résultat de l'exécution d'un programme, dit d'initialisation, le circuit est dit "programmable" (exemple : ACIA, PIA, PIO, etc.).

Lorsqu'un utilisateur se place devant un micro-ordinateur, certaines de ces séquences d'instructions sont déjà en mémoire. La facilité d'emploi du micro-ordinateur, y compris de mémoriser des programmes, dépendra de l'importance des séquences déjà existantes et de leur efficacité. L'ensemble de ces séquences forme le logiciel.

4.2 Principaux éléments du logiciel

4.2.1 Le système

Pour pouvoir être utilisé, le micro-ordinateur devra avoir au moins une entrée (afin d'introduire les programmes et les données) et une sortie (pour qu'il y ait une action !). De plus, on doit pouvoir lui dire s'il doit stocker de l'information (si oui : laquelle ?, où ?, comment ?) ou exécuter; autrement dit, il faut aussi pouvoir introduire des commandes.

Par conséquent, le micro-ordinateur devra disposer, de manière permanente dans des ROM, au moins de séquences d'instructions permettant :

- de comprendre les commandes,
- de "lire" sur une ligne d'entrée,
- de "écrire" sur une ligne de sortie.

Cet ensemble de séquences, qui peut être très sophistiqué (selon la richesse des commandes, le nombre et le type des entrées/sorties) forme le système.

Certains de ses programmes sont appelés des utilitaires (éditeur de texte ou de programme pour faciliter l'entrée de programme et de données, et pour permettre leur correction; traitement de fichiers, pour faciliter la gestion d'informations conservées sur des supports d'informations externes tels que disques et bandes magnétiques; etc.).

Un type d'utilitaire joue un rôle particulièrement important : ceux qui permettent le traitement des erreurs; en effet, il est souhaitable qu'en cas d'erreur de programme, l'ordinateur ne fasse pas absolument n'importe quoi, mais qu'il donne autant de renseignements que possible à l'utilisateur pour permettre à ce dernier une mise au point rapide et efficace de son programme.

4.2.2 Les langages de programmation

a) Origine et rôle des langages de programmation

Si l'on se souvient que l'information est toujours contenue sous forme binaire (suite de 1 et 0, correspondant à "positionné", "non positionné"), un micro-ordinateur très simple à huit bits pourrait avoir, comme entrée, huit interrupteurs, comme sortie,

huit lampes, plus trois touches de commande :

- "aller à la position mémoire indiquée par les interrupteurs"
- "mettre à la position mémoire actuelle l'information définie par les interrupteurs"
- "exécuter depuis la position mémoire actuelle".

Il devrait disposer de programmes pour "lire la position des interrupteurs", "allumer les lampes de manière conforme à telle position mémoire" et interpréter les commandes. Un tel système pourrait être contenu dans quelques dizaines d'octets, mais il ne serait pas très pratique !

Un premier "progrès" consiste à pouvoir représenter les contenus des registres (ou des mémoires) en base huit (code octal) ou en base seize (code hexadécimal; on utilise en général les lettres A, B, C, D, E et F pour les "chiffres" 10, 11, 12, 13, 14 et 15). Par exemple, sur un certain microprocesseur (le 6800), l'instruction "mettre à zéro telle mémoire" a le code

01101111	en binaire
157	en octal
6F	en hexadécimal

Pour pouvoir entrer les instructions et les données en hexadécimal, par exemple, il faudra disposer d'un clavier 16 touches (les 16 "chiffres" hexadécimaux); le logiciel devra savoir interpréter les touches, il sera déjà plus important (de l'ordre de la centaine d'octets).

A supposer que la sortie dont on dispose (les huit lampes) puisse suffire, l'utilisateur, pour pouvoir résoudre un problème à l'aide de cet ordinateur, devrait :

- analyser le problème (savoir à quel type il appartient),
- définir une méthode de résolution,
- définir les données qui sont nécessaires,
- rédiger la liste des choses à faire pour appliquer la méthode de résolution,
- écrire la liste des instructions de l'ordinateur (instructions machine) qui font ces choses,
- coder (en hexadécimal dans notre cas) ces instructions,
- "entrer" ces codes dans l'ordinateur,
- "entrer" les données nécessaires,
- faire exécuter le programme,
- interpréter les résultats "lus" sur les lampes.

Comme on le voit, la tâche de l'utilisateur serait immense et les possibilités qu'il commette une erreur innombrables ! Il faut toutefois remarquer que c'est à peu près ainsi qu'on travaillait avec les premiers ordinateurs !

En fait, selon l'importance du logiciel disponible sur l'ordinateur (c'est-à-dire le nombre et l'efficacité des programmes livrés avec lui), une part plus ou moins grande de ces tâches est prise en charge par l'ordinateur lui-même.

Remarquons d'abord qu'un système ayant un peu plus de commandes et capable de gérer les entrées/sorties de manière satisfaisante, peut assumer l'essentiel de ces quatre dernières tâches.

On s'est attaché alors à faire exécuter autant que possible par l'ordinateur la tâche "écrire la liste des instructions de l'ordinateur". Un premier pas consiste à pouvoir désigner par des mots formés de caractères usuels les instructions machine ("mnémoniques"), les positions mémoire et les données. On parle alors de langage symbolique, que l'ordinateur devra d'abord traduire en ses propres instructions (le programme qui fait cela est un assembleur). Sans entrer dans le détail, on peut dire que cela simplifie déjà bien le travail du programmeur, en particulier pour ce qui concerne les sauts en mémoire, et actuellement on programme pratiquement toujours au moins à ce niveau.

L'étape suivante consiste à utiliser des instructions qui ne correspondent plus aux instructions machine, par exemple donner telles quelles à l'ordinateur des expressions algébriques ou des ordres du genre : "faire n fois telle séquence".

Le premier langage issu de cette démarche (on parle alors de langage évolué) qui s'est généralisé est le FORTRAN (a FORMula TRANslator); il possède des versions standards dès les années soixante.

De nombreux autres langages standardisés sont apparus par la suite (COBOL, BASIC, PL 1, ALGOL, APL, PASCAL, etc.), entre autres pour des raisons d'efficacité dans des domaines particuliers.

b) Les langages de programmation proprement dits

On distinguera donc principalement deux catégories de langages :

1) les langages évolués, dont nous ne citerons que quelques-uns parmi les plus répandus :

- Basic (abréviation de "Beginners All purpose Symbolic Instruction Code"), très proche de la formulation mathématique et simple à utiliser,
- Fortran, (abréviation de "FORMula TRANslator"), premier langage à avoir été plus ou moins standardisé, qui a l'avantage de disposer d'une très vaste librairie de programmes et de sous-programmes,
- Pascal, langage mis au point par Niklaus Wirth, à l'EPFZ, et qui présente un grand attrait pour l'enseignement, vu ses structures de données et ses possibilités de contrôle de l'avancement d'un programme,
- Forth, langage d'astronome présentant un certain intérêt quant à sa structure interne et ses possibilités de redéfinitions des procédures,
- Cobol, (abréviation de COmmon Business Oriented Langage), qui n'est presque pas implémenté sur les microprocesseurs, vu sa structure et la taille des programmes résultants.

2) Les langages machines, ou assembleurs, ou macro-assembleurs, qui sont des langages où le programmeur donne une instruction détaillée pour chaque opération que le processeur devra exécuter ; la programmation, bien que permettant plus, car pouvant utiliser toutes les astuces d'un ordinateur, sera plus fastidieuse et dépendra surtout du système utilisé. C'est pourquoi ces langages sont en général réservés à la programmation système ou à des applications bien particulières.

Il faut remarquer qu'il y a, pour l'ordinateur, plusieurs manières de traduire en ses propres instructions un programme écrit dans un de ces langages : il peut former une suite d'appels à des séquences d'instructions-machine, qui font chacune une part de ce que le langage décrit (en général, une instruction du langage); le programme qui traduit de cette manière est un interpréteur.

L'interpréteur pur est un programme qui stocke en mémoire, ou sur stockage de masse, le code source tapé par l'utilisateur, et, lors de l'exécution, l'analyse, puis saute dans une routine interne pour simuler l'ordre reçu. Il est bien entendu que cette technique n'est, de loin, pas la meilleure en ce qui concerne la vitesse d'exécution et la place mémoire occupée (code source + interpréteur !). Elle présente néanmoins l'énorme avantage de permettre une modification rapide d'un ordre utilisateur, un changement de valeur d'une variable, etc. lors d'une pause, puis de continuer avec ce nouvel état.

Pour des raisons évidentes d'économie de place, la plupart de ces interpréteurs remplacent tous les ordres propres au langage (mots réservés) par un code interne, ce qui a également l'avantage d'accélérer l'exécution par la suite.

L'exemple type, qui est le plus répandu sous forme d'interpréteur, plus ou moins pur, sur les microsystèmes, est le BASIC. Afin de pouvoir, malgré le fait que certains ordres manquent, utiliser toutes les possibilités du microsystème, certains ordres hybrides, tels que PEEK ou POKE, sont apparus; certains devraient même permettre l'appel à des sous-routines écrites en assembleur.

L'ordinateur peut aussi analyser tout le programme écrit en langage évolué et le traduire en une séquence d'instructions-machine équivalente à tout le programme. Le programme qui traduit de cette manière est un compilateur. Le compilateur est un programme qui prend le code source de l'utilisateur, et le traduit en instructions machine directement utilisables, ou bien en code objet. Le temps de compilation peut être plus long que celui d'une interprétation, ennui qui est rapidement supplanté par les avantages d'un code exécutable, donc pouvant être seul résidant en mémoire à un moment donné. A problème égal, le code compilé prend nettement moins de place que le code à interpréter plus le code exécutable de l'interpréteur. L'exemple type du langage compilé, puis exécuté, est le Fortran; d'autres sont le Cobol ou l'assembleur.

De par leur nature, certains langages se prêtent plutôt à être interprétés (par exemple APL) ou compilés (par exemple FORTRAN).

Pour BASIC, les deux traitements existent. Enfin il arrive que l'on traduise d'abord le langage évolu  en un langage interm diaire, puis que l'on interpr te ce dernier, ceci parce que la premi re des deux phases (phase de compilation) ne fait pas intervenir les instructions-machine, donc ne d pend de l'ordinateur sur lequel on veut ex cuter le programme.

Enfin, il est apparu d s les ann es 70 qu'il fallait attacher une attention particuli re au probl me des erreurs : on s'est apercu qu'on  tait incapable de corriger les erreurs de certains programmes (en particulier des syst mes d'exploitation) sans en introduire d'autres ! Cette situation posait aussi le probl me de la maintenance du logiciel (comment faire en sorte qu'un programme reste compr hensible et  ventuellement modifiable, m me apr s une certaine p riode ?).

C'est pourquoi, d s les ann es 70, ont  t  syst matiquement construits des langages qui, d'une part, par leur vocabulaire, leur syntaxe et la forme des donn es qu'ils manipulent, emp chent autant que possible le programmeur de commettre des fautes non d tectables au moment de la traduction et, d'autre part, sont suffisamment "parlants" pour que le programmeur puisse les "lire" sans ambigu t , c'est- -dire pour qu'il sache exactement ce que fait un morceau de programme m me s'il ne l'a pas r dig  lui-m me. Ces langages sont dits "structur s"; le plus connu est PASCAL (d velopp    l'EPFZ par N. WIRTH).

Ces langages évolu s sont de plus en plus disponibles sur les micro-syst mes. Il faut toutefois faire des r serves quant   leur efficacit  sur ce type de mat riel; par contre, on peut utiliser un ordinateur plus puissant pour traduire un programme  crit en langage évolu  en instructions-machine d'un micro-ordinateur; le programme qui fait cette traduction est un cross-compileur.

4.2.3 Les langages (ou programmes) d'application

Si l'on reprend la liste des t ches d finies au d but de ce chapitre, on voit que l'utilisateur disposant d'un bon langage évolu  doit encore faire lui-m me la d marche allant de l'analyse du probl me   la r daction du programme (dans ce langage évolu ). Dans de nombreux domaines (aussi divers que gestion de stock, analyse de circuits  lectriques, planification de travaux, construction d'ouvrages de g nie civil, etc.) il existe des programmes mis au point une fois pour toute; il suffit alors   l'utilisateur d'entrer les donn es qui d crivent le cas qu'il veut traiter (ce qui se fait de plus en plus en r pondant aux questions pos es par l'ordinateur, c'est- -dire de mani re interactive). Ces programmes d'application forment ce que l'on appelle parfois le progiciel.

Actuellement, le progiciel disponible sur un micro-syst me est peu important, souvent peu efficace (mal  crit), voire inutilisable. Toutefois, certaines maisons sp cialis es commencent   s'en occuper un peu plus s rieusement.

4.3 Conclusions et perspectives

En r sum , ce qu'on peut faire avec un ordinateur d pend non seulement des propri t s mat rielles de cet ordinateur (le "hardware"), mais, pour une grande part, de l'ensemble des programmes et sous-programmes formant son logiciel (le "software"). Il faut donc attacher une attention particuli re au logiciel livr  avec un ordinateur,   celui qui est disponible,   celui qui pourrait (ou pourra) l' tre, sans oublier que son co t peut  tre tr s  lev  et risque fort d'augmenter   l'avenir, parall lement   la baisse de prix du mat riel.

Il faut toutefois remarquer que se dessine actuellement une tendance   faire assumer par le mat riel des t ches laiss es jusqu'  pr sent au logiciel; par exemple, "faire n fois telle s quence" devient pratiquement une instruction-machine; plus encore, on projette la production de micro-ordinateur sur un "chip" pour lequel le langage-machine sera un langage structur  ! (peut- tre l'ADA du D partement am ricain de la D fense). Cela signifie d'une part que la compilation de ce langage est faite directement par le mat riel et non par un programme, d'autre part que ce langage contient des instructions traitant des t ches (par exemple la gestion compl te des entr es/sorties) qui ne pouvaient pas  tre assum es jusqu'  pr sent par les langages  volu s. Des langages structur s  tendus de ce type ont aussi  t  d velopp s par les industries pour leurs applications (par exemple, en Suisse, PORTAL : Process-Oriented Real-Time Algorithmic Language).

5 LE MICROSYSTEME SMAKY

Le SMAKY est le système actuellement le plus répandu dans l'enseignement secondaire genevois puisqu'on le trouve - ou trouvera sous peu - dans chaque centre informatique.

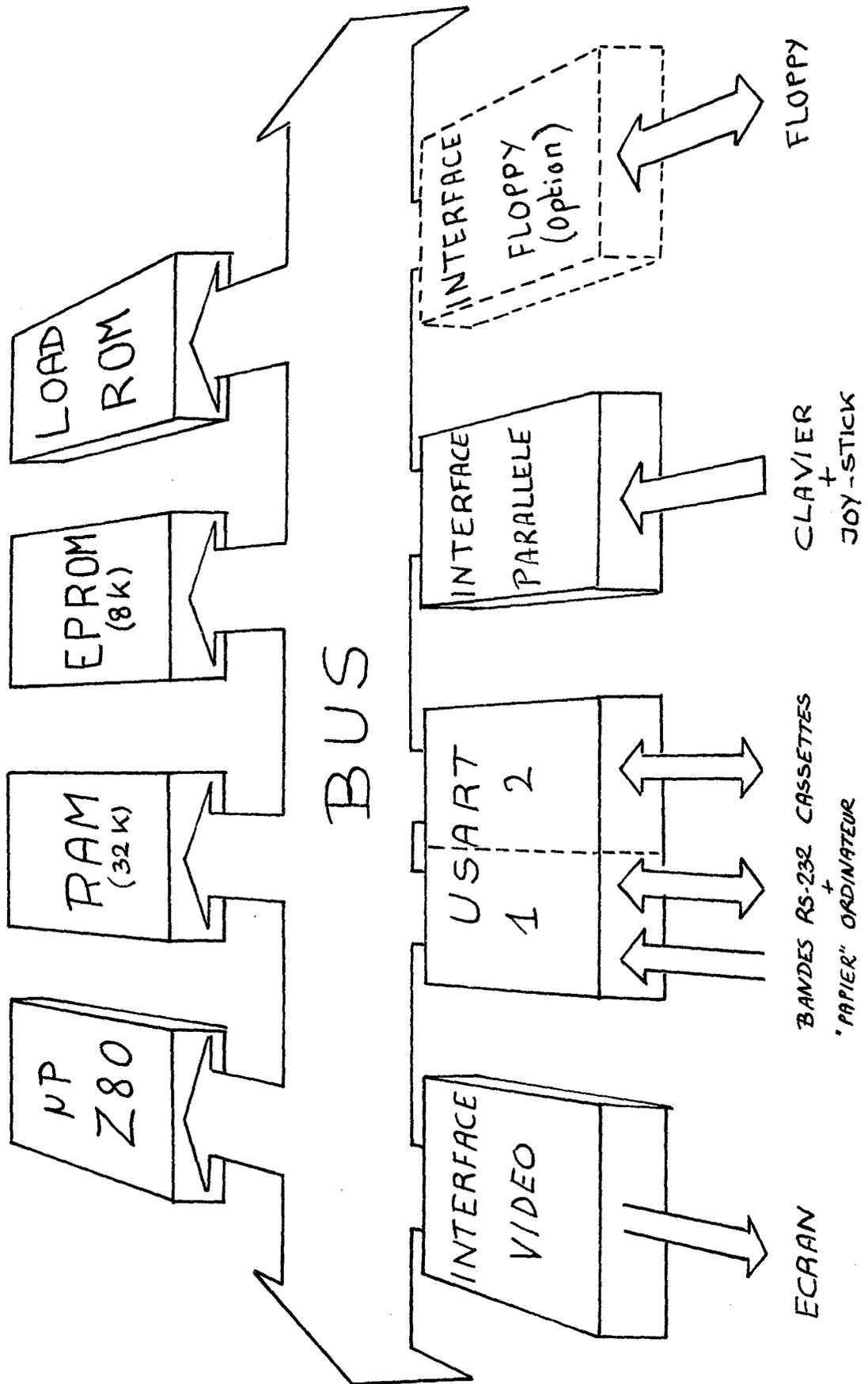
Développé à l'Ecole polytechnique fédérale de Lausanne (EPFL), le SMAKY est un microsystème basé sur le microprocesseur 8 bits Z80 et comprenant 32 kilobytes de mémoire vive (RAM) - extensible jusqu'à 64 kilobytes -, 4 kbytes de mémoire morte (EPROM) - extensible jusqu'à 8 kbytes -, 2 interfaces d'entrées/sorties séries basées sur des USART (Universal Synchronous and Asynchronous Receiver and Transmitter) dont l'un permet la connexion sur un ordinateur via un interface particulier (RS-232), un interface d'entrées/sorties parallèles, et surtout deux écrans superposables, l'un alphanumérique affichant 20 lignes de 64 caractères, l'autre graphique ayant une définition de 256 points en largeur et 120 points en hauteur.

Le noyau de ce microsystème étant réuni d'une façon compacte sur une seule plaque de circuits imprimés, son prix - ainsi que les possibilités d'extensions - est moins élevé que celui d'un système basé sur un principe de cartes enfichables.

Il est possible d'utiliser le SMAKY soit comme ordinateur indépendant (stand-alone), soit comme terminal.

Le SMAKY est un outil très intéressant pour les applications didactiques car il offre aux utilisateurs les facilités suivantes :

- un écran graphique adressable point par point par software (ce qui permet un affichage sélectif et par là une animation de dessins) et superposable à un écran alphanumérique traditionnel;
- une sortie vidéo compatible avec les moniteurs TV disponibles dans chaque école, ce qui permet de faire facilement des démonstrations devant toute une classe;
- des langages de haut niveau, tels le BASIC et le FORTH (langage basé sur la notation polonaise inversée comme les calculatrices Hewlett-Packard);
- divers moyens de charger des programmes, tels :
 - le "LOAD ROM" qui permet d'atteindre des informations stockées dans des EPROM supplémentaires,
 - le lecteur manuel de bandes "papier",
 - l'interface "cassette" permettant de lire ou de stocker de l'information sur des cassettes via un musicassette tout à fait classique,
 - un interface "floppy", livrable en option,
 - la ligne reliant le SMAKY à un ordinateur en faisant de lui un terminal, mais permettant toutefois une communication entre les ressources de l'ordinateur et la mémoire du SMAKY,
- la possibilité d'utiliser, grâce à la liaison décrite ci-dessus, n'importe quel langage de programmation et programmes disponibles sur l'ordinateur,
- la possibilité de servir de banc d'essai et de mise au point pour les dessins que l'on désire tracer avec un plotter.



6 MICRO-SYSTEME DU LABORATOIRE MICROPROCESSEUR DE L'EIG

Le but du laboratoire d'applications industrielles des microprocesseurs de l'EIG est de permettre aux étudiants de se familiariser avec les systèmes électroniques à base de microprocesseurs. Cela signifie que la circuiterie a une importance aussi grande que la programmation.

En 1977, nous n'avons pas trouvé sur le marché des microsystèmes correspondant à nos besoins. C'est la raison qui nous a poussés, lors de travaux de diplômes, à développer un microsystème qui devait présenter la structure suivante :

- 1) Avoir une base constituée par un rack (boîtier) avec alimentation et carte bus imprimée (carte fond de panier) supportant jusqu'à 15 connecteurs à 64 pôles.
- 2) Avoir une électronique constituée par des cartes modules de dimensions et de complexité raisonnables pour pouvoir être développées et réalisées dans le cadre de notre école.

Un tel système présente pour notre laboratoire les avantages suivants :

- a) Il est indépendant d'un microprocesseur déterminé. Ce dernier étant contenu sur une carte, il suffit de l'échanger et d'adapter le logiciel. Cette souplesse se retrouve pour toutes les cartes qui peuvent être adaptées à l'évolution technologique des composants. Le système est donc très évolutif.
- b) Chaque carte représente une fonction déterminée (mémoire RAM, mémoire ROM, interface série, interface parallèle, etc.). On peut donc les étudier séparément. Du point de vue pédagogique, cela représente une subdivision des difficultés. Pour les travaux de diplômes, cette modularité permet un grand choix de sujets de difficultés raisonnables.
- c) La maintenance est simplifiée car on peut localiser les défauts par échange de cartes. D'autre part, comme tous les circuits intégrés sont montés sur socle, les cartes defectueuses peuvent être dépannées par échange de circuits intégrés.
- d) L'architecture du microsystème n'étant pas figée, elle peut s'adapter à des applications variées telles que concentrateur, multiplexeur, acquisition de données, contrôle de processus, etc.

L'inconvénient majeur d'un tel système réside dans le coût. Acheté dans le commerce, on peut dire que son prix correspond au double, environ, de celui d'un microsystème réalisé sur une carte unique (mais qui représente un système généralement figé).

Les caractéristiques générales du micro-système de l'EIG sont les suivantes :

- Système de base prévu pour microprocesseurs à 8 bits.
- Possibilité d'adressage jusqu'à 256 kbytes.
- Possibilité d'extension du système pour des microprocesseurs à 16 bits, avec adressage jusqu'à 512 kbytes. Dans ce cas, nous utilisons deux cartes bus. La carte microprocesseur est une carte double qui effectue la liaison entre les deux bus, mais les cartes mémoires et d'interfaces sont les mêmes que pour le système à huit bits.

Les systèmes qui ont été utilisés jusqu'à aujourd'hui ont été développés lors des travaux de diplômes en 1978. Ils sont constitués par :

- Une carte microprocesseur MC6800.

Le microprocesseur 6800 a été choisi pour la simplicité de la circuiterie qu'il nécessite (pas de multiplexage des adresses et système d'interruptions facile et performant). De plus, nous étions assurés d'un excellent support (la maison Motorola étant à Genève).

- Une carte moniteur avec mémoire de travail propre.
- Une carte interface série pour terminal écran, TTY et cassettes.
- Carte(s) interface parallèle.
- Carte(s) mémoire 4 K RAM statiques.
- Carte(s) mémoire 8 K ROM.

Le logiciel était constitué d'un moniteur de 1 K, d'un assembleur-éditeur, d'un BASIC et d'un programme de "déverminage" ou programme de mise au point (debugging) résident en ROM.

En 1979, des travaux de diplômes permirent la réalisation de cartes mémoires dynamiques de 16 K et l'adaptation de disquettes, avec utilisation du logiciel de la maison TSC.

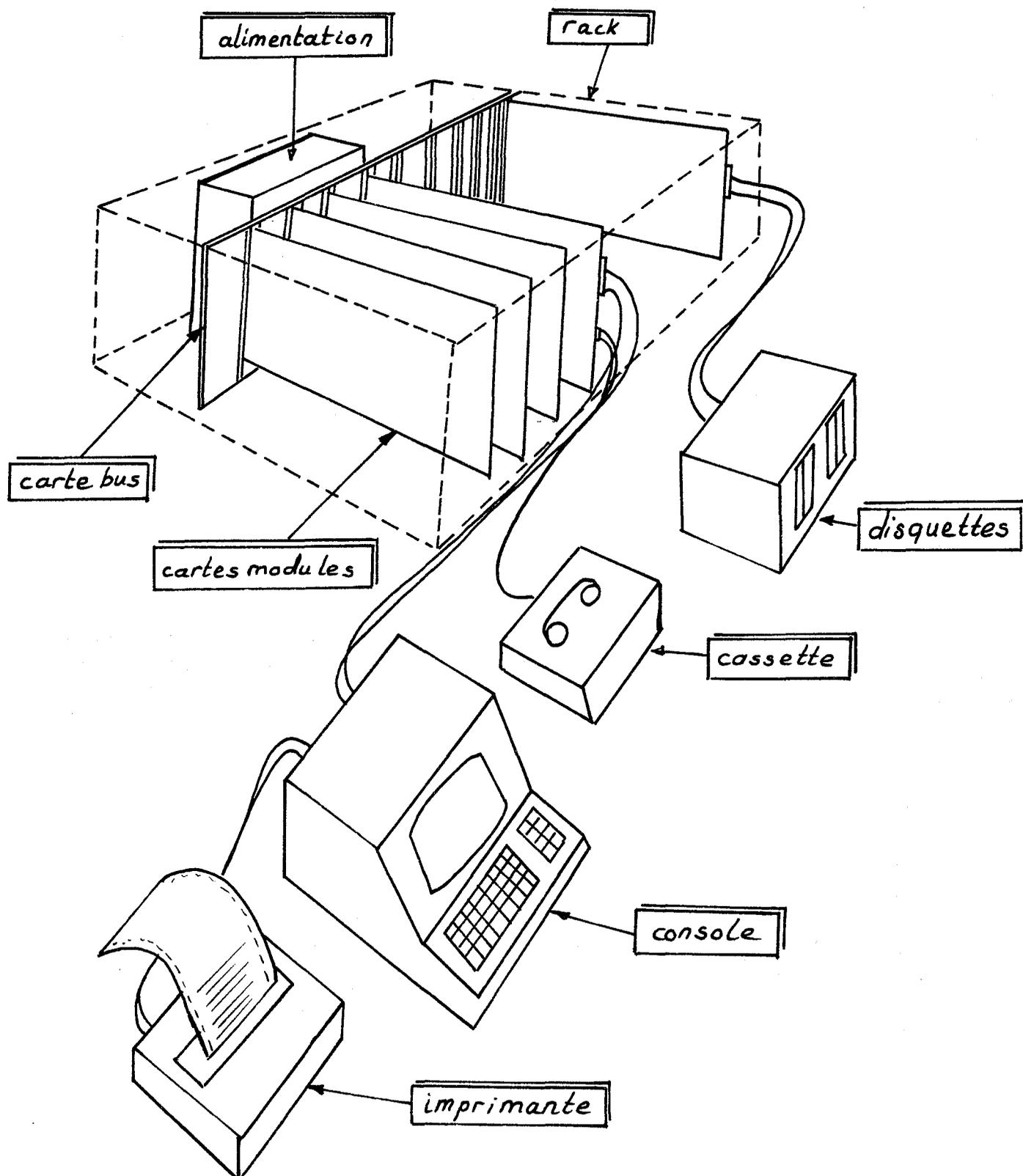
En 1980, deux nouvelles cartes processeurs ont été développées : une 6809 (qui remplace l'ancienne 6800) et une 280. Une interface série double plus performant remplacera l'ancienne version.

Les projets pour 1981 sont : une carte d'interface pour le bus d'instrumentation GPIB, une carte de "debugging" (Breakpoints et pas-à-pas hardware) et une carte mémoire dynamique de 128 Kbytes.

La réalisation de ces systèmes nous a posé un certain nombre de difficultés. Le laboratoire de microprocesseurs n'existant pas (il a été créé officiellement en 1979), nous n'avions ni crédit, ni décharge, ni assistant pour la mise au point définitive des cartes réalisées durant les travaux de diplôme. Ces derniers ayant une durée variant entre six et huit semaines, il n'est pas possible de demander à nos

étudiants de réaliser des circuits entièrement testés avec tous les documents de mise en fabrication. Nous avons donc encore quelques problèmes de fiabilité qui seront résolus durant l'année scolaire 80-81 (un assistant a été attribué au laboratoire en septembre 80).

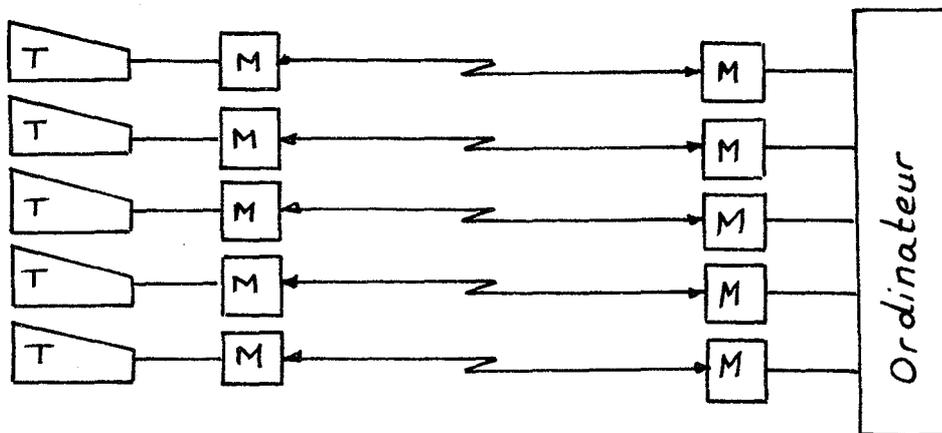
La conception du micro-système de l'EIG a permis aux maîtres enseignant la technique des microprocesseurs d'accroître leur expérience d'une manière beaucoup plus significative que si des systèmes complets avaient été achetés.



7 UNE APPLICATION AU CCEES : LES MULTIPLEXEURS (MUX)

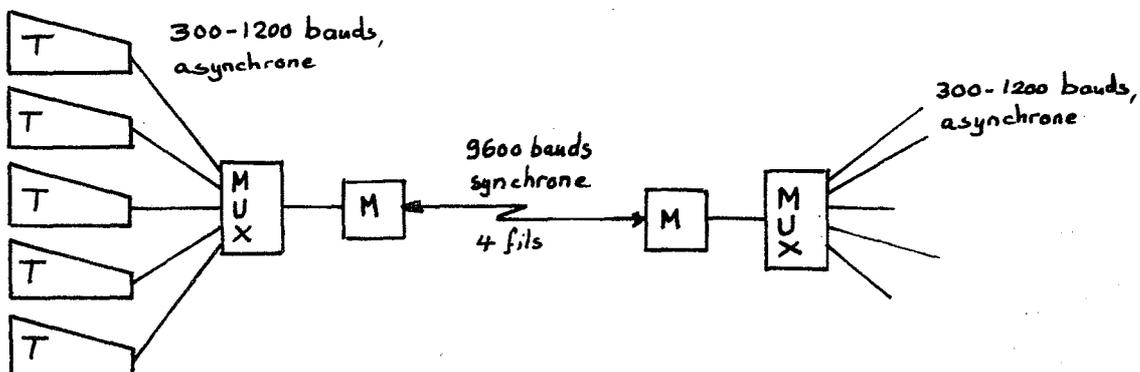
Bien que composé des mêmes éléments qu'un micro-système, à savoir un microprocesseur, des interfaces, de la mémoire vive et de la mémoire morte, le multiplexeur ne peut pas être considéré comme ordinateur personnel; il serait, en effet, surdimensionné du point de vue des entrées/sorties et largement sous-dimensionné quant à la capacité des mémoires et du stockage de masse. Il faut donc considérer le multiplexeur comme faisant partie d'un ensemble "mux-modem".

Jusqu'en 1979, chaque terminal situé à une certaine distance du site central nécessitait un modem à chaque bout d'une ligne PTT, ce qui permettait un travail à des vitesses de 110 à 300 bauds (10 à 30 caractères par seconde). Cette solution n'était pas satisfaisante : vitesse de transmission lente, nombre important de modems et de lignes téléphoniques utilisées, ces dernières étaient disponibles en nombre limité.



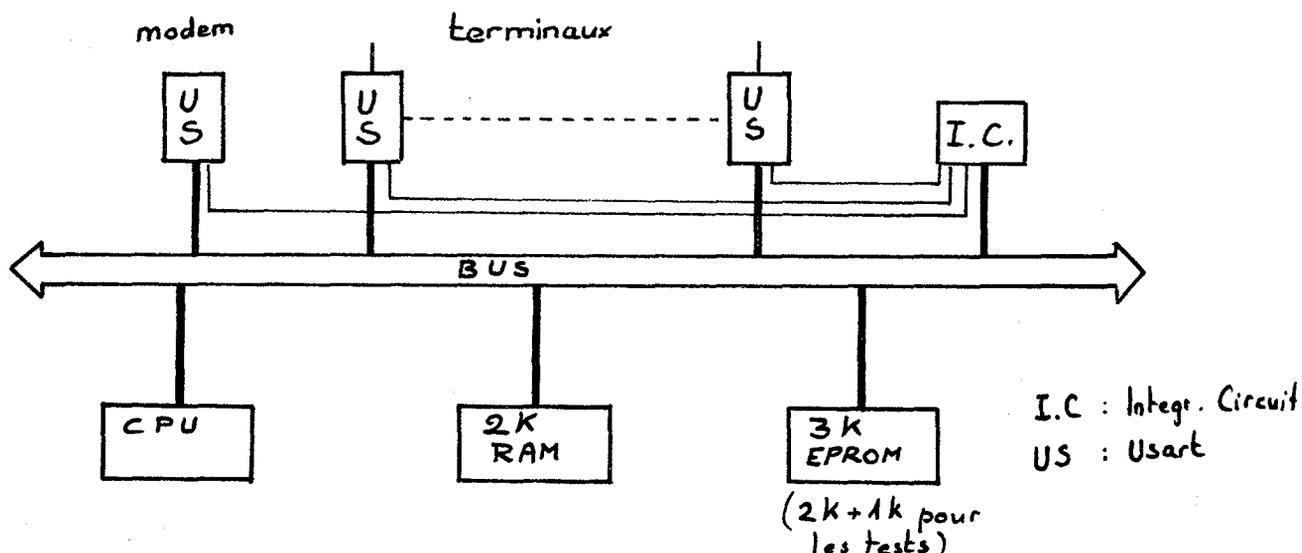
Cas d'une école à 5 terminaux : 5 lignes à deux fils
10 modems.

Il est évident qu'une solution de ce genre est aberrante, l'utilisation des lignes étant peu efficace. On a donc étudié la possibilité de multiplexer les sorties de l'ordinateur pour regrouper sur une ou deux lignes téléphoniques le transfert d'information à destination d'une école.



Après une étude du matériel se trouvant sur la marché et parce qu'on n'a pas trouvé de multiplexeur répondant à certains critères (nombre de terminaux, vitesse, prix, etc.), des contacts ont été pris avec l'Université de Genève qui avait construit ses propres multiplexeurs. Selon ses plans, la fabrication d'une série a pu être entreprise. Ce travail a duré de mai 1979 à juin 1980, les mises en service étant échelonnées entre octobre 79 et mars 80 pour les premiers.

Schéma bloc du multiplexeur :



Principe de fonctionnement :

Tout caractère arrivant de l'un des terminaux est ajouté, avec son adresse, au prochain flot de données en partance. L'autre multiplexeur ressortira ce caractère sur l'interface indiquée par cette adresse dans le message.

Un des critères important était la transparence dans le fonctionnement; c'est-à-dire que l'utilisateur ne doit, à aucun moment, se rendre compte qu'il y a une paire de multiplexeurs entre lui et l'ordinateur.

8 EXEMPLES D'APPLICATIONS PRATIQUES DES MICRO-ORDINATEURS

Comme nous l'avons déjà vu, il est possible d'utiliser les microprocesseurs dans des applications sophistiquées. Les micro-systèmes qui, actuellement, s'apparentent pratiquement aux mini-ordinateurs dans leurs exécutions élaborées, en sont un exemple. Cependant, dans bien des applications, le micro-ordinateur n'apparaît effectivement que dans la publicité du produit (son rôle n'est souvent pas explicite). On peut citer comme exemples les machines à laver, les machines à coudre, les pompes à essence automatiques, les balances, etc.

Pour bien des personnes encore, l'emploi du micro-ordinateur est réservé à des applications complexes. Elles se trompent. Le micro-ordinateur est un COMPOSANT ELECTRONIQUE bon marché s'il entre dans la fabrication de grandes séries.

Pour essayer de le montrer, nous allons décrire sommairement deux produits très simples.

1) "TIMER" (minuterie) de chambre noire Heathkit PI1500

Son but : enclencher une lampe pendant un temps déterminé. Il est possible de préprogrammer jusqu'à neuf intervalles de temps avec, en plus, quelques "gadgets".

Ce "timer" utilise un micro-ordinateur 3870. Il s'agit d'un seul circuit intégré comportant un microprocesseur huit bits de 1 kbyte de ROM (on peut aller jusqu'à 4 kbytes suivant les versions du 3870), de 128 bytes de RAM, de 32 lignes d'entrées/sorties.

Comment justifier le micro-ordinateur dans une application aussi élémentaire ?

Le "timer" digital possède un clavier numérique pour l'introduction du temps et de quelques fonctions auxiliaires, et un affichage digital. Avec de la logique conventionnelle (portes ET, OU etc.), la circuiterie pour la gestion du clavier et des affichages est assez complexe. Avec un micro-ordinateur, le clavier est connecté directement aux entrées/sorties du circuit intégré. L'affichage ainsi que les commandes sont également connectés aux lignes d'entrées/sorties, mais par l'intermédiaire d'amplificateurs (qui auraient également été nécessaires avec de la logique câblée). Le circuit est donc très élémentaire et peu coûteux. La logique de contrôle sera réalisée par le programme dont le coût est divisé par le nombre d'unités vendues.

A remarquer que le programme possède des routines de test qui permettent à l'utilisateur de localiser les pannes du système.

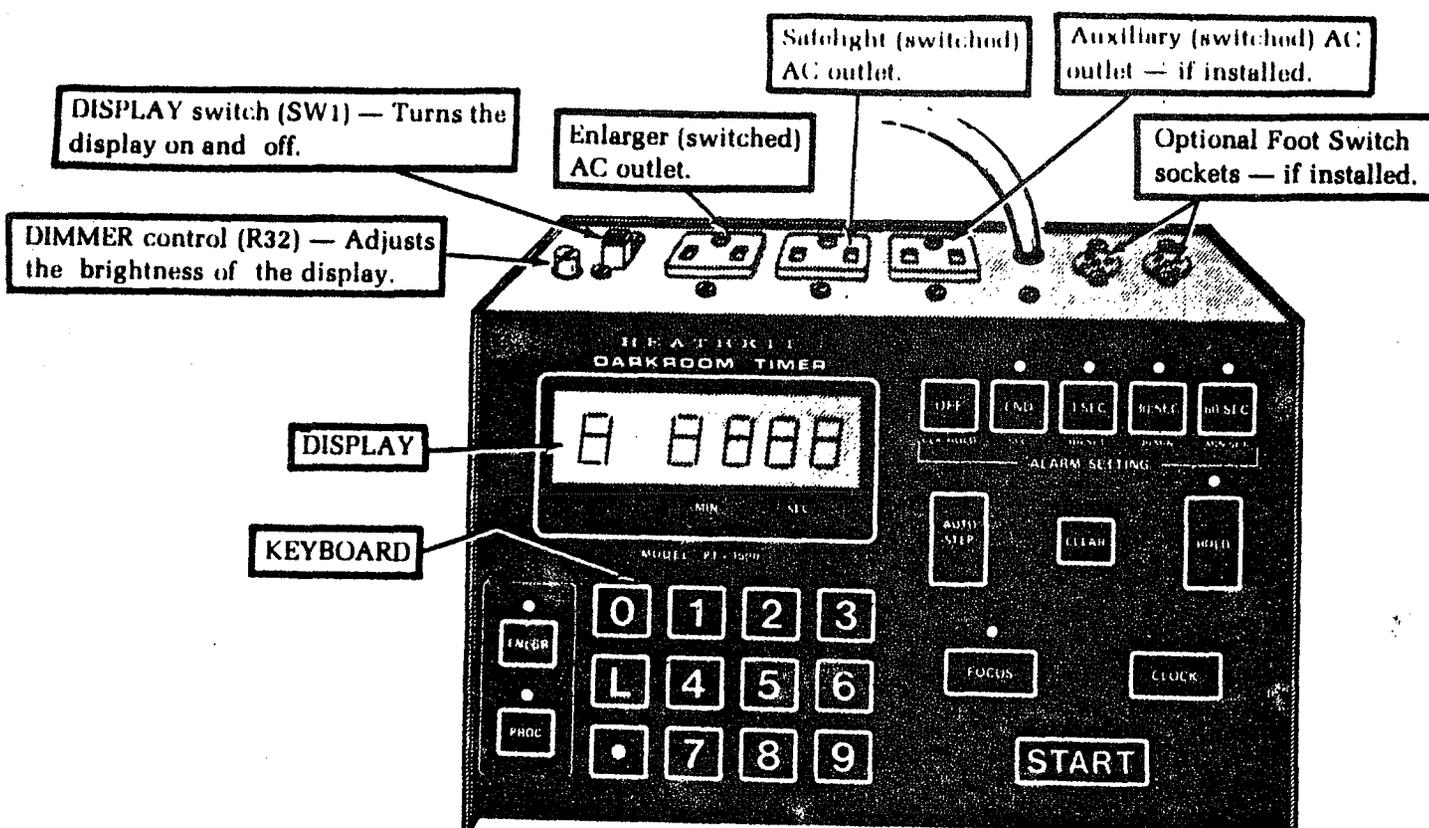
2) Carillon de porte KED 04

Caractéristiques : 12 airs différents sont contenus dans le micro-ordinateur TMS 1000, qui peuvent être déclenchés par deux boutons poussoirs. Dix airs sont affectés au poussoir principal, deux au poussoir secondaire. Soit les airs sont joués selon un mode aléatoire, soit ils sont présélectionnés par un commutateur. Avec amplificateur de sortie et haut-parleur, le prix est de l'ordre de fr. 50.-- en kit.

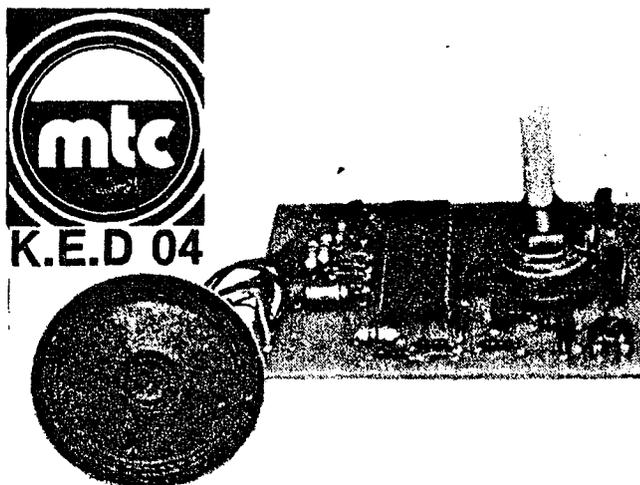
Le micro-ordinateur TMS 1000 de Texas Instruments est le premier à avoir été commercialisé. Il utilise un microprocesseur 4 bits, 1 kbyte de ROM, 64 x 4 bits de RAM et 23 lignes d'entrées/sorties.

Dans cette application, ce n'est pas la possibilité de contrôler directement des lignes d'entrées/sorties qui justifie le micro-ordinateur, c'est la facilité qu'a ce dernier de générer des séquences. Un air est effectivement une séquence de notes, elle-même étant une séquence de 1 et de 0 générés sur une sortie à une fréquence déterminée. Toutes ces séquences tiennent dans la mémoire de programme du TMS 1000.

Bien que l'on puisse être enclin à trouver cette application plutôt idiote, elle montre cependant bien l'étendue et la variété des possibilités d'applications des micro-ordinateurs.



Timer



caractéristiques électriques

alimentation continue 9 v
puissance de sortie maximum sur 50 ohms . 1,5 W
courant consommé au repos $< 0,1 \mu A$
nombre d'airs
contenu dans le microprocesseur 12

Carillon de porte

9. LES PROMESSES ET LA REALITE !

Selon les annonces, dans les journaux et les devantures de magasin, le micro-ordinateur est une merveilleuse machine qui permet :

- de tenir la comptabilité de son ménage
- d'apprendre tout (des mathématiques à la géographie en passant par les langues) sans effort
- de résoudre tous ses problèmes mathématiques et scientifiques
- d'obtenir la réponse à toute question (il adore être interrogé !)
- de stimuler sa créativité (ou celle de ses employés)
- de distraire les enfants (et les amis) grâce à des jeux passionnants
- de gérer sa petite entreprise (salaires, stock, etc.)
- d'automatiser de manière raffinée le matériel technique du ménage, de l'électro-ménager au train électrique et simultanément !

De plus, à celui qui a déjà quelques connaissances, on fait miroiter des facilités de développement de programme : utilisation de langages évolués, souplesse offerte par la mémorisation en RAM, EPROM, ROM, etc. Dans la réalité, les choses se présentent toutefois autrement :

- L'alimentation électrique d'un micro-ordinateur, même sans disque magnétique ou imprimante, exige un matériel assez coûteux (plusieurs tensions nécessaires, bonne stabilité). Les choses s'arrangent un peu actuellement (alimentation incorporée); cependant, en cas d'extension de la mémoire ou d'acquisition de périphériques, il faudra penser à ce point.
- Les divers composants ne peuvent en général pas être connectés sans autre entre eux; il y faut toujours des éléments supplémentaires, soit matériels (circuits d'interface, de contrôle, alimentation supplémentaire), soit logiciels (programmes d'interface, contrôleurs de périphériques), l'un n'excluant pas l'autre, qui peuvent être coûteux pour autant qu'ils existent !
- Le service après-vente est en général mauvais, entre autres parce que la représentation des marques (en principe américaines) change souvent. Cela rend problématique l'entretien du matériel et du logiciel, ainsi que les possibilités d'étendre son installation.
- Le logiciel existant limite beaucoup la souplesse que l'on pourrait attendre de tels systèmes : il est pauvre, souvent mal écrit et peu efficace, et contient en général des erreurs.
- Comme tout dans ce domaine est en pleine évolution, il n'y a pas de standard; d'où, d'une part, incompatibilité entre installations, souvent y compris si elles sont du même type (même matériel de même marque), d'autre part, les logiciels n'ont pas les mêmes modes d'emploi; même les langages évolués sont différents (ils ne respectent pas les normes). Autre point négatif : la documentation; elle ne correspond souvent pas à ce qui a été livré.

- Les protections contre les erreurs de manipulation sont souvent insuffisantes; on détruit facilement des contenus de mémoire ou de disques souples qu'on a mis beaucoup de temps à entrer !
- Les temps d'exécution ne doivent pas être sous-estimés. Par rapport à un "vrai" ordinateur, un micro-ordinateur est très lent; quoi qu'en dise la publicité, il est inutilisable pour du calcul scientifique ou de la gestion commerciale d'une certaine importance. Les mémoires de masse (disques souples ou cassettes magnétiques) sont également très lentes.

Toutefois, si on est bon bricoleur, capable de contrôler, monter et compléter des circuits électriques, si on sait lire un schéma et qu'on comprend quelque chose à l'électronique (y compris les circuits intégrés), si l'on apprend rapidement à programmer en langage machine et si on est capable de bien comprendre la conception générale d'un microprocesseur et du logiciel de base (ce que l'on appelle parfois son architecture), on peut, avec de bons fournisseurs, retrouver une bonne partie des qualités mentionnées au début de ce chapitre ! Mais ...

En résumé, il ne faut pas penser qu'on peut remplacer un "vrai" ordinateur par un micro-ordinateur, et bien savoir ce que l'on veut faire avec le matériel qu'on désire acquérir. Il convient également d'étudier soigneusement les éléments (matériels et logiciels) nécessaires et disponibles ainsi que les possibilités d'entretien.

10 BIBLIOGRAPHIE SOMMAIRE SUR LA MICROINFORMATIQUE

Parmi la masse d'articles, de revues ou de livres qui traitent des microprocesseurs, on peut en citer quelques-uns :

- dans "The Practitioner", vol. VI, no 1 (oct. 1979) :
Microcomputer... The Future is Now
 Rapport très complet sur le sujet : Qu'est-ce qu'un micro-système ? - Un système typique - La sélection d'un micro-système - Description de quelques micro-systèmes actuels (TRS-80, Apple II, Pet, The Sorcerer) - Les sources d'informations (journaux, revues, groupes d'utilisateurs, constructeurs, etc.) - Des exemples d'écoles américaines équipées.
- dans "L'Informatique nouvelle" de mai 1979, page 15 :
Le France-Computing de J.-F. PHELIZON
 "Small is beautiful", mais Small n'est pas vraiment conçu pour tourner 8 heures par jour, 365 jours par année.
- dans "L'Informatique nouvelle" d'avril 1980, p. 37-40 :
Le marché des ordinateurs pour PME/PMI aux Etats-Unis de H. THIRIEZ
 Etude qui est le résultat d'une série d'entrevues aux USA avec des fabricants d'ordinateurs individuels, des distributeurs et des entreprises de conseil spécialisées dans la réalisation de logiciels pour ordinateurs individuels.
- dans "01 Hebdo" no 540, p. 23 (14.5.1979) :
L'introduction de l'informatique dans l'enseignement secondaire
 Décision du gouvernement français du 22.11.1978 pour un programme plurisannuel et progressif d'équipement en micro-ordinateurs (de l'ordre de 10'000) et initiation rapide de 10'000 enseignants.
- dans la revue "01 Informatique" no 139, p. 77-83 (avril 1980) :
Des micros saisis par la gestion de R. KOPEIKIN
 L'année 1979 a été marquée, outre-atlantique, par une exceptionnelle évolution du catalogue du fait de l'arrivée de micros vraiment destinés à la gestion.
- dans "01 Hebdo" no 594, p. 2 (27.5.1980) :
Panorama de la microinformatique européenne
 Evolution du marché mondial (en 1979, quelque 330'000 micro-systèmes) et riposte de l'Europe.
- dans "Microsystèmes", juillet-août 1979, p. 11-17 :
Applications des microprocesseurs de D. J. DAVY
 Les applications des microprocesseurs sont innombrables et encore en grande partie inexplorées. Un essai de classification, avec exemples, est tenté pour différencier les applications informatiques, industrielles et "grand public".
- dans "L'Informatique nouvelle" de juillet 1980, p. 42-45 :
Le point sur la bureautique de E. BENAJEAN
 A l'ère du tertiaire, la bureautique regroupe les activités de saisie, de traitement et de communication d'informations dans l'entreprise, c'est-à-dire la diffusion de documents, textes, notes, graphes, etc. Le terme bureautique recouvre un nombre

important de fonctions telles que le courrier électronique, l'accès aux systèmes documentaires, la réception et la transmission automatiques des messages, la gestion automatique des emplois du temps, mais actuellement l'effort est surtout porté sur l'automatisation progressive des travaux dactylographiques à l'aide de systèmes de traitement de textes souvent basés sur des micro-ordinateurs.

- dans "L'ordinateur individuel" no 17, p. 65 (mai 1980) :
L'ordinateur familial est un leurre de J.-F. POITEVIN
L'informatique individuelle est à volonté un jeu ou un outil, mais nous portons tous la responsabilité du futur de l'informatique familiale.
- dans "L'Informatique nouvelle" de février 1980, p. 38-43 :
La fusion du traitement de texte et de l'informatique annonce le bureau du futur de M. MANDELL
L'association naturelle du traitement de texte et de l'informatique rendra les décisions plus complexes et plus délicates dans les entreprises. Ces dernières, espérant réduire la montée croissante des coûts de la papperasse et des frais généraux, devraient accueillir favorablement cette tendance. La motivation la plus caractéristique des grandes entreprises est d'assumer la responsabilité globale du traitement de l'information : données, texte, image et son.
- dans "01 Informatique" no 140, p. 94-95 (mai 1980) :
Demain, la télédistribution de M.-F. GARDIOL
Cette nouvelle révolution est née d'un mariage à trois : le microprocesseur, le téléphone et le petit écran. De la télématique à la télédistribution.
- dans le cahier spécial du no 37 de "01 Informatique" no 131 (juin-juillet 1979) entièrement consacré à l'informatique individuelle.
- dans "01 Hebdo" no 588, p. 8-9 :
De l'intégration à grande échelle à l'ordinateur au foyer
Aucune découverte n'est une fin en soi. Combien de temps s'écoule-t-il entre l'invention proprement dite et la production permettant, par son volume et le coût, de toucher le grand public ?
- dans "L'Express" du 13.10.1979, p. 84-85 :
L'Ordinateur : l'inconnu dans la maison de D. SIMONNET
Bientôt, l'homme et la femme feront ménage à trois. Un individu de sexe indéterminé va se glisser dans le cocon familial. Comment cohabiterons-nous avec ce troisième sexe ?
- dans les comptes-rendus du symposium CAL-79 d'Exeter, l'article de 5 pages de A. GOLDBERG :
Educational uses of Dynabook
Description de la conception d'un micro-système révolutionnaire basé sur l'idée de réseau de communication et actuellement testé par un groupe de recherche chez Xerox à Palo Alto.
- dans la traduction française du Scientific American parue en 1979 aux Editions "Pour la Science" se trouve un excellent historique de la "Révolution microélectronique".

- dans la revue "Temps réel" no 7 du 9.2.1981, un article de M. ZAOUÏ
Micro-ordinateurs : des petites bêtes à problèmes.
 Après avoir été longtemps considérés comme le "joujou des hobby-istes", les micro-systèmes sont en train de faire la conquête des utilisateurs professionnels. Outils de décentralisation, toutes les difficultés liées à leur emploi ne sont pas encore levées. De plus, la prolifération des modèles disponibles sur le marché rend le choix difficile.

- dans le livre de J.-P. BOUHOT de 335 pages "Un fil d'Ariane" (tome V) :
La micro-informatique à la portée de tous aux Editions d'informatique (1979)
 Tout sur les notions fondamentales hardware et software en relation avec les microprocesseurs.

- dans le numéro spécial no 19 bis de la revue "L'Ordinateur individuel" le guide 1980 de 123 pages sur les micros :
 Description de 85 systèmes avec leurs caractéristiques et leurs prix. Annuaire de l'informatique individuelle.

- dans le livre (à paraître chez North Holland) sur les comptes-rendus de la Working Conference IFIP WG 3.1 ou MICROCOMPUTERS in SECONDARY EDUCATION (Sèrnes 15-18 avril 1980) :
 A partir d'une vingtaine d'exposés d'experts, le résumé des discussions donne de très bonnes indications sur la situation actuelle et les tendances pour les prochaines années dans ce domaine.

- dans le livre de J.-J. SERVAN-SCHREIBER de 477 pages édité chez Fayard (1980) :
Le défi mondial
 Une grande partie de ce livre est consacrée à l'introduction et l'impact de la micro-informatique dans les années à venir.

- dans le livre de A. TOFFLER de 623 pages publié aux Editions Denoel en 1980 :
La 3ème vague Comme pour le livre précédent, l'auteur imagine les conséquences de l'informatique et de la micro-informatique dans la vie de tous les jours. - les revues les plus utiles en ce qui concerne la micro-informatique et son utilisation dans l'enseignement secondaire sont :
 en anglais : BYTE, KILOBAUD et CREATIVE COMPUTER
 en français : L'ORDINATEUR INDIVIDUEL et MICROSYSTEMES.

Toutes les références de cette bibliographie sommaire peuvent être consultées au Centre de Calcul Electronique de l'Enseignement Secondaire (CCEES) à Genève (case postale 172, 1211 Genève 3 ou tél. 022/27.22.28).