

**Zeitschrift:** Bulletin des Schweizerischen Elektrotechnischen Vereins, des Verbandes Schweizerischer Elektrizitätsunternehmen = Bulletin de l'Association suisse des électriciens, de l'Association des entreprises électriques suisses

**Herausgeber:** Schweizerischer Elektrotechnischer Verein ; Verband Schweizerischer Elektrizitätsunternehmen

**Band:** 71 (1980)

**Heft:** 17

**Artikel:** Problèmes posés par la construction d'un système microprocesseur

**Autor:** Gaumain, M.

**DOI:** <https://doi.org/10.5169/seals-905284>

### **Nutzungsbedingungen**

Die ETH-Bibliothek ist die Anbieterin der digitalisierten Zeitschriften. Sie besitzt keine Urheberrechte an den Zeitschriften und ist nicht verantwortlich für deren Inhalte. Die Rechte liegen in der Regel bei den Herausgebern beziehungsweise den externen Rechteinhabern. [Siehe Rechtliche Hinweise.](#)

### **Conditions d'utilisation**

L'ETH Library est le fournisseur des revues numérisées. Elle ne détient aucun droit d'auteur sur les revues et n'est pas responsable de leur contenu. En règle générale, les droits sont détenus par les éditeurs ou les détenteurs de droits externes. [Voir Informations légales.](#)

### **Terms of use**

The ETH Library is the provider of the digitised journals. It does not own any copyrights to the journals and is not responsible for their content. The rights usually lie with the publishers or the external rights holders. [See Legal notice.](#)

**Download PDF:** 16.05.2025

**ETH-Bibliothek Zürich, E-Periodica, <https://www.e-periodica.ch>**

# Problèmes posés par la construction d'un système microprocesseur

Par M. Gaumain

681.325-181.48;

*Pour les besoins de l'enseignement de la construction et de la mise au point de circuits et de programmes, il était nécessaire de disposer d'un matériel adapté aux besoins particuliers de l'école. Dans ce but, un système a été spécialement développé et construit autour du 8080 d'Intel. On expose ici les problèmes rencontrés au point de vue options tant au niveau du matériel qu'au niveau des programmes.*

*Für die Ausbildung im Aufbau und in der Einstellung von Schaltkreisen und Programmen ist es notwendig, über Geräte und Bauelemente zu verfügen, die auf die besonderen Bedürfnisse der Schule abgestimmt sind. Deshalb wurde in Yverdon um einen Intel 8080 Mikroprozessor ein spezielles System entwickelt und aufgebaut. Es werden Probleme erläutert, die bei der Auswahl auf Stufe Material sowie bei den Programmen angetroffen wurden.*

## 1. Introduction

En 1975, l'Ecole d'ingénieurs de l'Etat de Vaud (EINEV) a introduit dans ses plans d'études l'enseignement des microprocesseurs. La vocation d'une école d'ingénieurs étant d'assurer une formation aussi bien pratique que théorique, il devenait nécessaire de faire un choix judicieux du support matériel (les machines à utiliser) et du support logiciel (les programmes).

## 2. Le matériel

Il s'agissait de choisir le type (marque et modèle) du processeur et le niveau d'intervention externe (partie construite, partie achetée montée). A fin 1974, les microprocesseurs à 4 bits (dimension des mots que le processeur peut véhiculer d'un bloc, 8 bits assurant la codification d'un caractère d'imprimerie) cherchaient à se rapprocher de l'architecture plus souple des 8 bits, avec le circuit 4040 d'Intel. Les microprocesseurs à 8 bits alors sur le marché présentaient l'avantage d'avoir une structure assez semblable entre eux. Les microprocesseurs 16 bits n'existant pas encore avec une unité centrale (CPU) en un circuit intégré, la décision de choisir un microprocesseur à 8 bits paraissait raisonnable et le choix du type se résumait à une alternative 8080 d'Intel ou 6800 de Motorola, le 8080 étant déjà un 8008 amélioré. L'option 8080 fut prise principalement à cause de l'expérience d'Intel dans ce domaine et des secondes sources (firmes différentes fabriquant sous licence les mêmes circuits), ce qui garantissait une certaine longévité au produit.

Pour les besoins de l'enseignement, il était souhaitable d'avoir un système le plus modulaire possible, afin de permettre aux étudiants d'effectuer plus facilement le rapport entre les circuits et le schéma-bloc du système (fig. 1). Pour une bonne analyse des modules, il était avantageux que ces derniers puissent fonctionner, suivant le but recherché dans la manipulation, soit à l'intérieur du rac (étude de la régulation), soit hors de celui-ci (étude du module). Dès lors il devenait nécessaire d'entreprendre la construction d'un système simple offrant ces performances, car il n'existait pas dans le commerce d'équipement fournissant ces prestations. Comme chaque carte ne devait représenter qu'un seul module, le format des cartes européennes était suffisant (fig. 2). L'interconnexion rigoureusement parallèle de toutes les cartes posait un problème pour dimensionner le connecteur. Finalement un connecteur à 37 conducteurs s'avéra suffisant pour transmettre tous les signaux plus les alimentations et la masse, ce format ayant l'avantage d'être contenu sur une face de circuit imprimé.

Actuellement le système comporte les modules suivants:

- Unité centrale plus calcul (CPU)
- Mémoire vive
- Mémoire morte
- Sorties digitales  $2 \times 8$  bits
- Entrées digitales  $2 \times 8$  bits
- Sorties analogiques 2 canaux de 8 bits
- Lecteur papier
- Aiguillage des adresses et des commandes
- Aiguillage des données

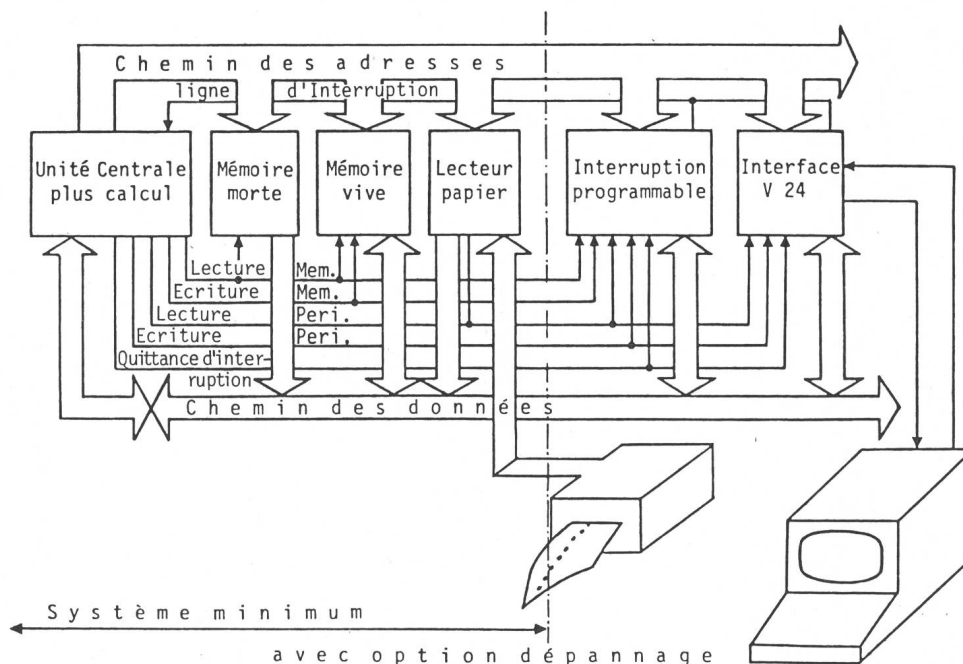


Fig. 1  
Schéma bloc du système complet

Ces deux dernières cartes assurent la réalisation d'un montage DMA<sup>1)</sup> (possibilité d'écrire ou de lire la mémoire sans l'intervention du processeur) hors du rac, ce qui permet dans un premier temps de charger des programmes sans le secours d'aucun moniteur ou chargeur.

Une conséquence du petit format du connecteur est que le montage DMA dans le rac nécessite un câblage frontal des cartes pour piloter le processeur.

### 3. Les programmes

Dès que l'on désire aborder des programmes de moyenne importance, soit des programmes de plus d'une page, tout un support informatique devient nécessaire pour éditer (assistance à l'écriture), assembler (traduire le texte en binaire) et déboguer (contrôler le programme et éliminer les erreurs).

Dans le cas présent, la solution du cross-assembleur (programme exécutable sur un autre ordinateur pour assurer la traduction) s'imposait, car on disposait déjà d'un miniordinateur. On a donc acquis et adapté un programme permettant de générer des bandes binaires pour le 8080 à partir des sources éditées sur le miniordinateur. Cette solution présentait l'avantage de ne pas nécessiter un gros investissement, l'ordinateur étant sur le site. De plus elle permet de changer, à peu de frais, le type de microprocesseur si cela s'avérait nécessaire.

L'édition des programmes se trouvait de ce fait résolue par le miniordinateur qui en assure également le stockage. Il ne restait plus qu'à résoudre le problème du débogage. Deux solutions étaient envisageables :

- Acquérir ou écrire un programme de simulation tournant sur le miniordinateur.
- Implanter sur les microprocesseurs un programme de débogage.

La première solution présente l'inconvénient de ne pas pouvoir tester le mariage entre le programme et l'interface particulier à l'application, c'est pourquoi la seconde paraît préférable.

### 4. Le programme de débogage

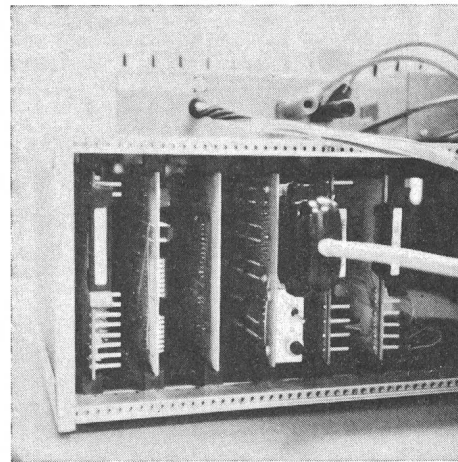
L'implantation du programme de débogage sur le système microprocesseur nécessitait un dialogue aisé entre l'opérateur et le système. Pour assurer cette fonction, un écran et un clavier reliés au microprocesseur par un interface standard (norme V 24) présentait l'avantage d'admettre des périphériques courants (fig. 3, 4).

Pour rendre le dialogue plus performant, la carte d'interface devait permettre l'activation des interruptions du système afin que l'opérateur puisse avoir une action immédiate sur le système.

Pour assurer le partage entre le programme de débogage et le programme utilisateur, une carte d'interruption programmable (fig. 5) fut implantée sur le système. Cette carte se comporte comme un comparateur de l'état du système par rapport à la référence programmée et lors de l'égalité entre les deux, elle génère une interruption qui redonne le contrôle au programme de débogage (p.ex.: interrompre si l'on écrit sur le périphérique d'entrée le numéro 0).

Cette solution, emploi d'une carte d'interruption programmable, présente l'avantage d'éviter de toucher au programme utilisateur pour y mettre des points d'arrêt. De ce fait, le pro-

<sup>1)</sup> DMA Direct Memory Access.

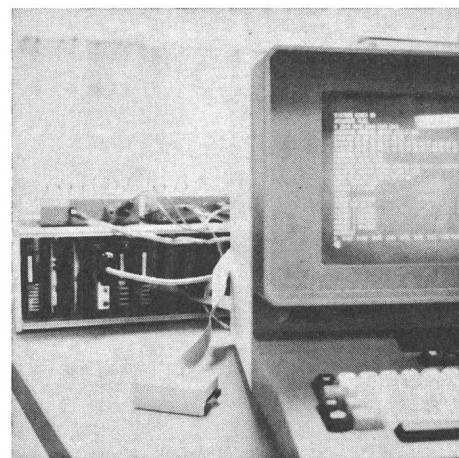


**Fig. 2**  
L'ensemble des cartes nécessaires dans le rac

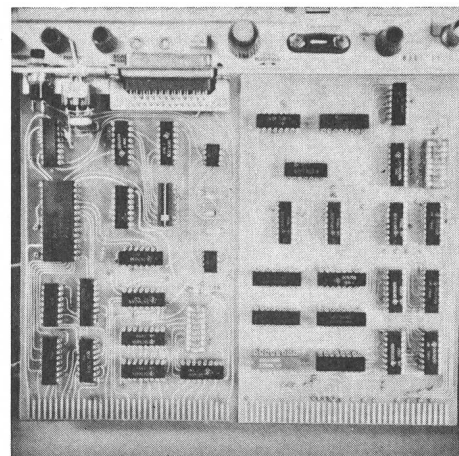
De gauche à droite: unité centrale, interruption programmable, mémoire morte 8 kbytes, interface clavier/écran (V 24), mémoire vive 1 kbyte, lecteur papier

gramme peut se trouver en mémoire morte. Elle permet en plus de poser indifféremment des conditions d'arrêt sur les périphériques ou la mémoire et d'assurer le déroulement du programme utilisateur en temps réel jusqu'à cette condition. On retrouve ici les performances de l'émulateur qui, lui, nécessite un microordinateur pour contrôler le programme utilisateur.

Pour faciliter l'apprentissage de l'utilisation du programme de débogage, une syntaxe des commandes a été imposée. Chaque mot de commande se donne par deux caractères suivis



**Fig. 3**  
Vue de l'ensemble



**Fig. 4**  
A gauche, la carte d'interface (V 24), à droite la carte d'interruption programmable

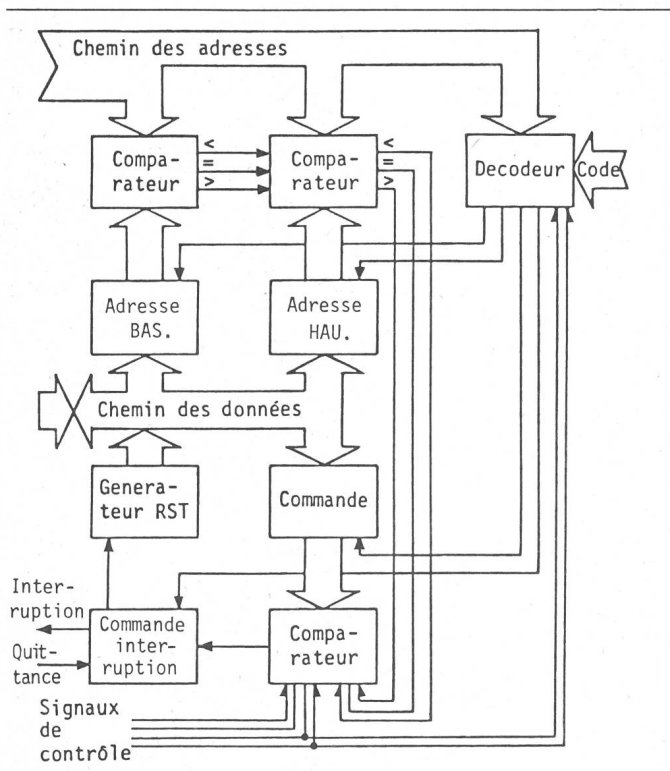


Fig. 5 Carte d'interruption programmable

- d'un espace si la commande est incomplète (passage au mot de commande suivant ou paramètre) ou
- d'un retour de ligne si la commande est complète (passage à l'exécution).

Le programme se charge de compléter les mots de commande sur l'écran, ce qui évite les équivoques. L'avantage de cette structure réside dans le fait que tous les branchements des commandes se font par table et qu'un même module de programme assure leur exécution quel que soit le niveau du paramètre de la commande. Ce module assure la lecture des deux caractères, le contrôle de l'existence de la commande, l'impression de la fin du message sur l'écran, le passage à l'exécution ou le passage sur une autre table de commande, suivant que l'opérateur a commandé «retour de ligne» ou «espace».

Les fonctions à disposition sont :

- Examiner des registres, des indicateurs, de la mémoire, des entrées (avantageux pour le dépannage des circuits).
- Changer des registres, des indicateurs, de la mémoire, des sorties (pour le contrôle des circuits).
- Avancer pas par pas depuis l'adresse courante, depuis une adresse donnée.

- Avancer par bloc de pas, ce qui donne la trace pas par pas sur l'écran.
- Initialiser une condition d'arrêt (lecture ou écriture d'une case mémoire ou d'une entrée/sortie).
- Exécuter jusqu'à la condition initialisée une ou plusieurs fois.

L'exécution d'une commande génère toujours un écho sur l'écran.

Le système avec son programme de dépannage tel qu'il a été décrit présente néanmoins deux lacunes par rapport à un émulateur. L'une est due au fait que le système de dépannage est le même que le système de l'utilisateur, ce qui a pour effet que des erreurs de câblage dans le système peuvent gêner le déroulement du programme de dépannage. L'autre est due à la faiblesse du connecteur qui n'autorise qu'un niveau d'interruption. L'existence d'un seul niveau d'interruption a pour conséquence que son activation par l'utilisateur pendant l'exécution de certaines fonctions du programme de dépannage gêne le déroulement de ces dernières. Pour remédier en partie à ce problème, le programme de dépannage contrôle si l'utilisateur est actif lors des interruptions sur les canaux utilisateur et avise ce dernier en donnant le code de l'interruption généré. Pour les interruptions valides, il assure leur exécution. Il autorise même l'exploitation de celles de la console du système par l'utilisateur sur le même code que celui du programme de dépannage.

## 5. Perspectives

Pour compléter ce système, il manque un moyen d'édition et d'assemblage locaux (sur le système microprocesseur). Pour réaliser ces fonctions, on a le choix entre d'une part ajouter ces programmes au système, et d'autre part connecter le microprocesseur à un système informatique offrant ces prestations.

La seconde solution paraît la plus intéressante car plus riche en possibilités potentielles. Cette phase des travaux nécessitera la réalisation de la liaison tant au point de vue programmes que circuits. Une ligne multiprocesseurs (plusieurs microprocesseurs sur un canal de l'ordinateur) paraît être une structure avantageuse, même si les problèmes de programmation au niveau de l'ordinateur ne paraissent pas simples.

On peut donc conclure qu'avec le programme de dépannage, la deuxième tranche des travaux est terminée et que si la troisième n'est qu'à l'état d'ébauche, elle promet déjà d'être très enrichissante du point de vue technique.

### Adresse de l'auteur

Maurice Gaumain, ingénieur électricien, chargé de l'enseignement des systèmes logiques, Ecole d'Ingénieurs de l'Etat de Vaud EINEV, 1, route de Cheseaux, 1400 Yverdon.

### Corrigendum

Im Heft 15 vom 9. August 1980 hat sich auf Seite 803 im Aufsatz von H. Zumstein: «Datenübertragungsnetze – Angebot und Anforderungen von morgen», ein sinnstörender Druckfehler eingeschlichen. Richtig sollte der letzte Satz des Abschnittes 2.1 lauten: «Zudem müssten vielfach die bisherigen Systemkomponenten durch Komponenten der neuen Architektur ersetzt werden, was oft unwirtschaftlich sein dürfte.»