LABORATOIRE D'INFORMATIQUE THéORIQUE
& APPLICATIONS DE MARSEILLE
L.I.T.A.M
Département de Mathématiques Informatique
LUMINY
UNIVERSITé D'AIX-MARSEILLE II

ISSN 0291 - 5413

INFORMATIQUE FONDAMENTALE ET APPLICATIONS

SOMMAIRE

MARS 1987

Adresse postale : FACULTé DES SCIENCES DE LUMINY
Mathématiques-Informatique.
L.I.T.A.M
Batt TPR 2 9e Etage
case 901 70 route LEON LACHAMP
13 288 MARSEILLE cedex 9
Tel. 91 26 90 81

EDITORIAL

e. bianco

INFORMATIQUE ET COMPLEXITE

Il y a des mots qui sont dangereux. Le mort est de ceux là. Sans jeux de maux. Il faut préciser bien sur, qu'ainsi se manifestent des gradations et des distances dans le danger. Et pour décrire les phénomènes qui suggèrent l'emploi d'un tel mot, on pourrait dire qu'il y a des fluctuations de niveau sémantique. Le niveau se définirait quand il y a objet et qu'on parle de l'objet, quand il y a idée et qu'on parle de l'idée, quand il y a phrase et qu'une phrase parle de la phrase.

La phrase n'est qu'une conséquence d'un acte, qui n'est qu'une conséquence d'une intention, qui n'est qu'une conséquence d'une pression du milieu... Et là encore apparaissent des niveaux. Autant la sémantique sans changement de niveau parait abordable, pas simple pour autant mais abordable, autant la sémantique qui jaillit à la fracture des niveaux se révèle difficile à appréhender, je dirai: complèxe.

En ces longues périodes de grande humidité où glissent des montagnes, le soleil semble nous abandonner, la terre planète frileuse, s'enveloppe chattement d'une ouate nuageuse.

Sur d'autres terrains bourbeux aussi bien s'amorcent des glissements. Prenez le cas, tenez, de la recherche... Avant, la recherche c'était le ciel inébranlable auquel était suspendu l'avenir des vrais peuples. Et puis, petit à petit à cela on a substitué

de la recherche avec innovation. Il est vrai que, toujours les mêmes mots ça lasse, et puis ce substantif: "innovation" a un petit air jovial et frais, un peu coquin même.

L'idée de mort et le mauvais emploi du mot "Mort" ne peuvent nuire que dans des registres éloignés, entre lesquels on peut placer une ou plusieurs fractures de niveau. Sur le plan du code pénal l'idée de mort peut entrainer la mort physique, considérer la Mort comme un personnage peut entrainer à des élucubrations imaginatives débridées, seulement reliées par des liens ténus au cas précédent.

Imaginons alors qu'on se heurte à un tel concept qui aurait été en quelque sorte sublimé par l'emploi un peu manipulé d'un mot dont le pouvoir dépasserait ainsi son but d'honnête descripteur.

Comment endiguer le déferlement de sémantique qui, à l'instar de la Calomnie menace même le pauvre Diable, et nous enroule dans ses anneaux mouvants.

On peut toujours essayer, en construisant une fable, de mettre en évidence le ridicule des excès de ces débordements et éventuellement souligner le danger des aberrations qui peuvent en procéder.

Et à l'époque où chaque **Homme Important** soigne son look à la télévision, même les mots les plus anodins doivent soigner le leur.

Et **innovation** n'est pas un mot anodin. Surtout à l'heure où dans un grand concert de modernisme nos Universités s'américanisent. Faire une réforme importante exige de la faire raisonnable et de l'adapter à l'esprit du milieu. L'Université ce n'est pas l'EDF et laisser se rendre indépendantes d'énormes entreprises qui risquent de devenir rapidement ingouvernables ne vaut pas la conservation d'une administration centrale étatique dans laquelle pourraient se développer une multitude de laboratoires destinés à devenir des collecteurs de ressources.

Il y a un effet de la substantivation cela parait indéniable. Une action est locale, elle est décrite par un verbe. Une classe d'actions est

rassemblée sous la bannière d'un substantif, au départ: commodité d'expression. Puis il y a tout naturellement sécrétion d'une entité attachée au substantif qui peut, selon les circonstances, se gonfler d'une substance qui n'appartient plus à l'action de départ. Et plus cela déborde sur du merveilleux, de l'extraordinaire, même du fantastique, plus cela se répand dans l'imagination où se dessine le mythe.

Peut-être est-il encore possible de réfréner l'envahissement en ramenant le débordement à des considérations terre à terre. L'illusion meurt quelquefois dans les pierres du chemin.

Quel serait alors le rôle de cette administration ? Eh bien tout simple, et c'est là que se manifeste le génie de la société ambiante. Le Monde nous envie notre organisation fiscale, et l'administration pourra faire la preuve ainsi de toute son efficacité en prélevant la quote part sur les rentrées des laboratoires. Imaginons le mouvement bien lancé: l'administration pourra même reverser une partie de ses gains à l'état.

La question peut ainsi se poser à propos de ces substantifs qui se révèlent capables de "gonfler". Alors que tous ne le font pas. Pourquoi gonflent-ils ? Peut-être sont-ils liés à des questions profondes qui agitent nos civilisations , comme la Mort pour le monde chrétien. Ou encore à des questions de pouvoir qui sous-tendent fortement le monde hiérarchique.

Non seulement la recherche ne coûte plus rien, mais elle rapporte. Si ce n'est pas là manifestation de Génie, que le Dieu tout puissant des administrations me frappe de ses foudres.

Mais tout ce bel édifice tient debout en reposant sur l'idée que chaque laboratoire puisse vendre sa camelotte à d'éventuels preneurs.

L'idée justificatrice est simple, l'Université est coupée de la société vivante, que voila un bon moyen de lui remettre les pieds sur terre au lieu de planer dans des sphères éthérées autant qu'inutiles.

Un petit tour de la question ne saurait faire de mal si vous soulignez quelques points amusants. En remarquant toutefois que quelquechose d'amusant n'amuse pas forcément tout le monde - pourvu que ça amuse l'auteur.

A partir de ce moment vous pouvez entrer en relation contractuelle avec qui bon vous semble. En fait, avec qui veut bien vous acheter. Mais c'est qu'ils sont méfiants les entrepreneurs. Vous ne leur vendriez pas du Fortran pour du Cobol. Ils ont l'oeil.

Alors ma foi, à vous de jouer et de sentir ce qui a quelque chance de se vendre bien, mais n'imaginez pas qu'on vous en donnera des fortunes. N'imaginez pas qu'une petite entreprise peut payer le prix de la recherche, et les grosses s'y refusent absolument d'ailleurs. De toute façon si c'est vous qui proposez, on vous voit venir de loin...

Alors on invente des subventions mais qu'on donne aux entreprises, et aux grosses de préférence, peut-être parceque ça fait plus sérieux. Et si vous ne vous débrouillez pas trop mal vous récupèrerez bien quelques miettes. A condition, bien sur de *faire le beau*.

Duneton a expliqué très en profondeur ce phénomène, qu'il a relié avec beaucoup d'humour à la notion de "génie de la langue".

Dire que les "soldats défilent" n'évoque pas les mêmes images que de parler du "défilé des soldats". Dans le premier cas on évoque volontiers les ampoules au pieds, dans le second on est plus proche des grands rassemblements organisés, liés par une grande idée, chers à Brasillac qui les a sublimés dans ses "sept couleurs". Quelle fracture de niveaux !

L'ennui, il y a toujours un petit ennui, c'est que le côut de la recherche n'est pas linéaire. Prenons l'exemple de la construction aéronautique. Quand on a voulu dépasser une certaine plage de vitesses, on s'est apperçu qu'on se heurtait à une sorte de mur qui brisait les avions, et il a fallu la convergence d'études théoriques fondamentales et d'expérimentations côuteuses sur les propriétés des gaz pour pouvoir absorber les difficultés causées par les ondes de choc. De nos jours, à part l'armée, peut-être, qui accepterait de financer de telles recherches ?

Il n'est pas mauvais de pousser le bouchon ostensiblement un peu loin, mais surtout que l'exagération ait plutôt tendance à faire rire. Le dragueur sait bien que s'il parvient à amuser l'objet de son désir, il a déjà presque gagné.

J'ai un peu peur que l'ouverture d'un tel marché soit essentiellement un débouché pour des briquets à ranimer la flamme sacrée, des brosses à cirer les pneus des voitures, des plaques électriques pour réchauffer les convictions un peu tièdes, voire des vaccins pour soigner les blessures des paroles mordantes.

Bref, que petit à petit les patrons de laboratoires pris dans l'engrenage de la rentabilité, deviennent les Guy Lux de la pensée scientifique. J'ai déjà pu connaître une grande entreprise de logiciel qui emploie plusieurs centaines d'ingénieurs, mais pas un seul programmeur, et qui continue à écrire des compilateurs comme on le faisait il y a trente ans. Parcequ'ils ne peuvent détacher deux ingénieurs pour apprendre des méthodes plus élaborées.

Et puis ma foi, la chute, comme un grand pont qui réussirait à franchir toutes les fractures. La chute doit être brève pour être efficace, mais cette qualité à elle seule ne peut suffire.

Aux Etats-Unis ce sont les petits constructeurs qui ont forcé les gros à la micro-informatique, ce sont encore les petits qui ont obligé les gros à commercialiser des logiciels pratiques à la place de leurs traditionnels et lamentables opérating systems.

Peut-on imaginer pareil effet chez nous ?

Il ne me reste plus qu'à conclure en donnant le titre:

L'informatique pure ou le rêve français.

Vers l'aide à l'accès à un thésaurus. Représentation des connaissances néphrologiques: systèmes de production et réseaux de Pétri.

J.M.KNIPPEL

C.R.: Subject classification informatics : H.3.1, F.1.1, I.2.4

Résumé:

Dans les années soixante, C.A. Pétri suggère une représentation générale des processus sous la forme de graphes. Dans des articles précédents du bulletin d'informatique approfondie et applications J.C.FUMANAL, P.ISOARDI, A.PANARIELLO et J.M.KNIPPEL ont déjà exploité ce type de formalisme à différents niveaux: automate programmable, gestion des entreprises, gestion des connaissances bibliographiques.

Nous faisons ici le point sur divers formalismes de représentation des connaissances dans le contexte d'application choisie qui est l'aide au diagnostic médical par l'étude approfondie du répertoire de KENT. Nous nous proposons, étape supplémentaire, d'automatiser l'accès à un tel thésaurus à l'aide du formalisme général des réseaux de Pétri. Cette démarche a pour but d'aider le praticien dans l'art d'interroger le patient, afin de faciliter le passage au thésaurus permettant de trouver une prescription utile et efficace.

REPRESENTATION DES CONNAISSANCES NEPHROLOGIQUES: SYSTEMES DE PRODUCTION ET RESEAUX DE PETRI.

J.M.KNIPPEL (°)

I.INTRODUCTION:

De nos jours de nombreux malades ont déjà dépassé dix ans du traitement qui consiste à pallier la défaillance des fonctions d'excrétion et de régulation hydro-électrolytique du rein par des méthodes d'épuration extracorporelle du sang. Un échange discontinu, au travers d'une membrane semi-perméable entre le sang du malade et une solution de dialyse proche de celle du sang normal, permet l'élimination des substances de déchets (urée, créatinine, ...), la normalisation des éléctrolytes plasmatiques du sang (sodium, potassium,.....) et l'élimination de l'eau retenue dans le sang. Les variables de réglage, régissant cet échange sont imposées par la machine de dialyse et définissent le point de fonctionnement. La connaissance du système homme-machine de dialyse est basée sur l'observation des malades. Nos travaux ont permis une clarification, une description et un résumé de la structure des données par l'approche statistique qui est essentiellement numérique /I/, nous nous proposons d'ajouter une approche de nature symbolique provenant de l'expérience des experts du domaine.

2.METHODOLOGIE:

2.I.Définition des réseaux de Pétri:

En I962, C.A.PETRI proposa une représentation générale de processus sous forme de graphes, qui fut étendue par A.HOLT en I9

(°) Exposé des séminaires . Quinzième session de l'école internationale d'informatique (AFCET).

Ce dernier lui donna le nom de "réseaux de Pétri".

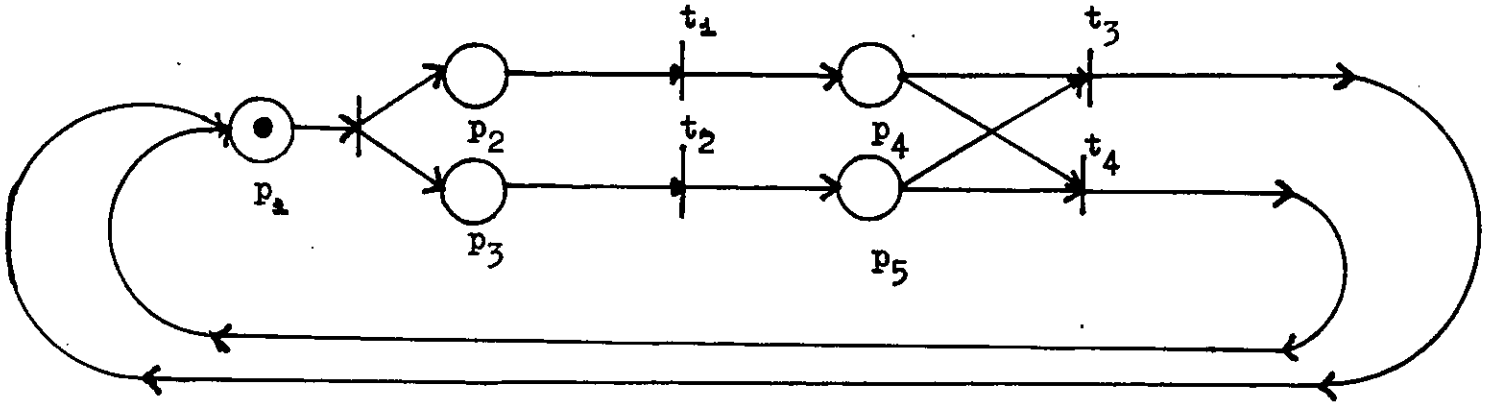
Un réseau de Pétri est un graphe orienté défini par <T,P,A,Mo> où:

$T=\{t_1,.....,t_m\}$ est un ensemble fini de transitions. } Noeuds du
$P=\{p_1,.....,p_n\}$ est un ensemble fini de places. } graphe
$A=\{a_1,.....,a_k\}$ est un ensemble d'arcs orientés (x,y) qui assurent la liaison d'une place vers une transition ou inversement: $x\varepsilon T$, $y\varepsilon P$ ou $x\varepsilon P$, $y\varepsilon T$.

Mo est la distribution initiale des marqueurs (ensemble des places qui ont initialement un marqueur).

Une place (représentée par un cercle) peut posséder un, zéro, (des) marqueur(s); elle est dite marquée ou non. L'ensemble des places possédant un (des) marqueur(s) caractérise la fonction de marquage du réseau (c'est à dire son état à un instant donné).

Une transition (représentée par un trait) est validée si toutes ses places d'entrée (places d'où sont issus des arcs orientés vers la transition) sont marquées (figure I). Seules les transitions validées peuvent être franchies. Franchir une transition revient à enlever un marqueur de chaque place d'entrée de t_i et à placer un marqueur supplémentaire dans chaque place de sortie (place où aboutissent des arcs issus de la transition).



figureI. Exemple de réseau de Pétri /2/.

2.2.Intérêt des réseaux de Pétri:

Restant proche du cahier des charges (on peut rattacher au réseau de Pétri une signification physique), ils permettent de représenter les fonctionnements parallèles. Les systèmes chimiques sont un bon exemple de système qui peut être modélisé par les réseaux de Pétri /3/. Par exemple des réactions catalytiques peuvent être ainsi représentées. La combinaison d'hydrogène et d'éthylène,

pour former de l'éthane ($H_2+C_2H_4 \rightarrow C_2H_6$) seulement en présence de platine, se formalise dans le diagramme de la figure 2.

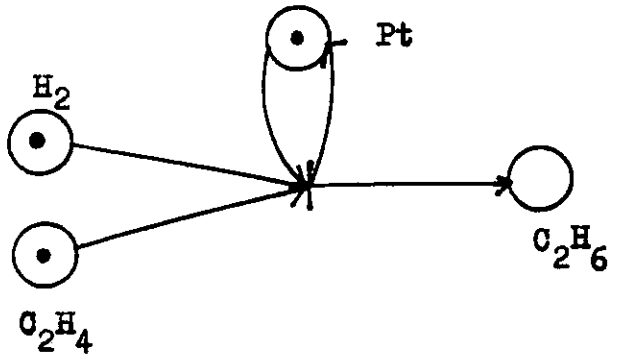


figure 2. Production d'éthane à partir d'hydrogène et d'éthylène, en présence de platine.

3.ORGANISATION DE LA CONNAISSANCE:

Initialement nos travaux s'intéressaient à la recherche d'un modèle du sous-système humain lié à la néphropatie chronique. Après avoir déterminé le domaine temporel où les données cliniques, observées sur les malades, étaient les plus caractéristiques de la maladie et avoir adapté sur un calculateur des programmes d'analyse de données, nous avons isolé les variables essentielles. Par réduction des variables leur nombre a été porté à dix sept qui entre dans trois relations essentielles. L'existence d'un modèle étant ainsi réalisée, l'analyse par les nuées dynamiques a permis de découvrir cinq classes de modèles. la recherche des liens possibles entre ces classes et l'origine de la néphropatie chronique a pu caractériser nettement deux classes comme liées à une origine glomérulaire de la maladie pour l'une, et à une origine héréditaire pour l'autre /I/.

De plus, nous nous attachons à introduire une approche subjectiviste et objectiviste. L'organisation des variables se complètera de signes subjectifs, objectifs, lésionnels ou particuliers à l'organe /4/. L'étude de cette réorganisation peut-être un élément complémentaire d'aide au diagnostic. Le tableau de la figure 3 donne un aperçu de la signification des signes subjectifs objectifs, lésionnels ou particuliers à l'organe.

Signes	subjectifs	I-bizarres, 2-constriction, 3-douleurs 4-douleurs, type clinique, 5-faiblesse, 6-plénitude, 7-spasmes, 8-ténèsme vésical.
	objectifs	9-besoin pressant d'uriner, I0-catarrhe, écoulement muscopurulent, II-hémorragies, I2-inertie, inactivité vésicale, I3-miction I4-prurit, I5-sensible.
	lésionnels	I6-calculs vésicaux, I7-hypertrophie, enflure, distension, I8-hémorroïdes, I9-inflammation, cystites, 20-paralysie, 2I-polypes, 22-rétention d'urines, 23-suppurations, 24-tumeurs et fongoïdes, 25-ulcérations.

figure 3. Organisation par signes sujectifs, objectifs, lésionnels ou particuliers à l'organe: vessie.

4. REPRESENTATION ET UTILISATION DES CONNAISSANCES:

Un champ immense du traitement de l'information échappe à la programmation classique: il s'agit de tous les domaines où l'homme manipule des informations factuelles, des règles isolées, des stratégies et des savoir-faire partiels. Toute cette connaissance est de nature non algorithmique /2/.

4.I.Réseaux sémantiques:

R.DAVIS et J.KING /5/ ont souligné que les règles de production sont un formalisme intéressant pour représenter une connaissance énonçant des jugements, maisqu'il n'est pas naturel d'utiliser ce formalisme pour représenter des connaissances déclaratives telles que des relations entre différents objets d'un domaine. Ainsi par exemple, une connaissance anatomique peut se traduire par un graphe comme celui de la figure 4. Les liens ne sont pas tous de même nature /6/, et il peut être utile d'utiliser des relations causales, des relations temporelles, taxonomiques, associatives...

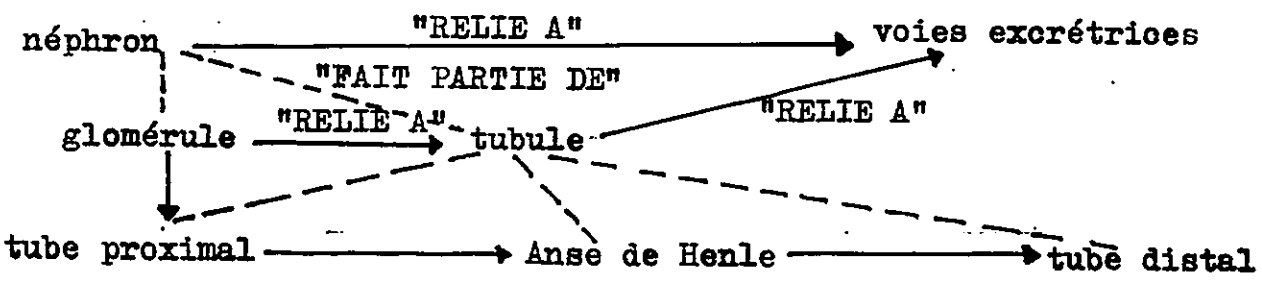


figure 4. Connaissance anatomique /5/.

4.2.Frames:

M.MINSKY est parti de l'idée que nous avons des structures stéréotypées d'informations et que nous sélectionnons chaque fois qu'une situation nouvelle se présente, l'un de ces stéréotypes en tentant de le faire coïncider avec les données de la situation. Le système PIP (Present Illness Program) est un système qui contient environ soixante -dix schémas de maladies rénales et qui a été conçu sur ce type de structures, à l'aide du comportement des médecins.

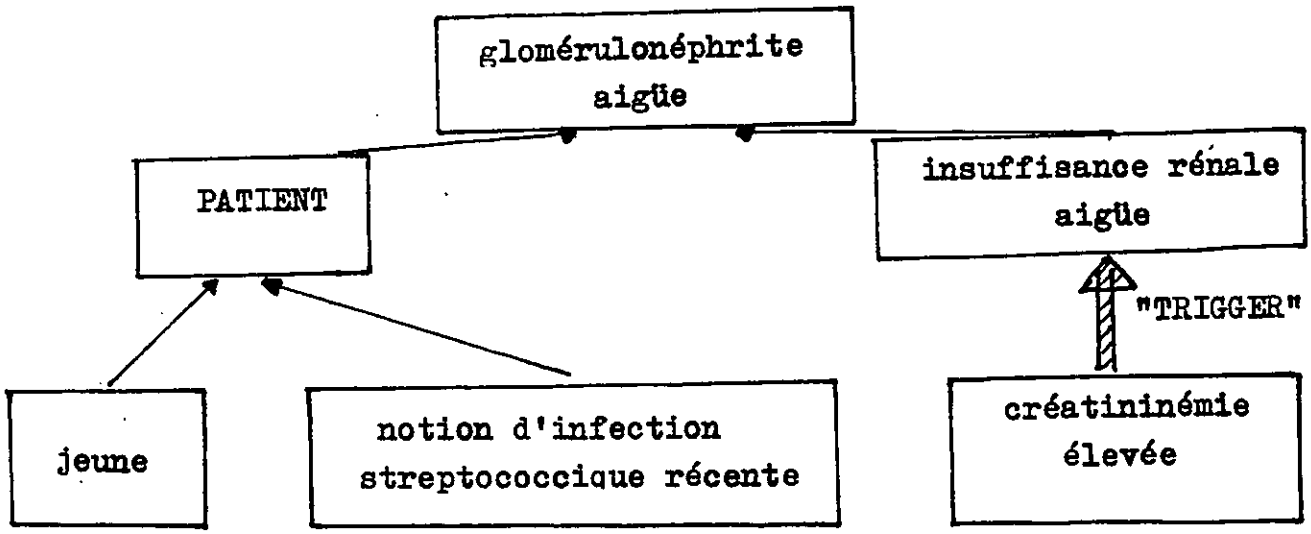


figure 5. Exemple de PIP /7/.

4.3.Systèmes de production et réseaux de Pétri:

Les systèmes de production ont été proposés par E.POST /8/, comme mécanisme général de calcul. La connaissance de l'expert est représentée par un grand nombre de règles simples utilisées pour guider le dialogue entre le système et l'utilisateur afin de déduire des conclusions. Les réseaux de Pétri ont l'avantage de représenter, non seulement le processus lui-même, mais aussi son contrôle: les jetons et la règle de déclenchement. Ils permettent ainsi de modéliser les problèmes de parallélisme et synchronisation qui sont au coeur des processus de traitement de l'information et en particulier des systèmes de règles de production/2/.

Nous proposons ici de représenter la connaissance sous forme de réseaux de Pétri généralisés, dont la figure 6 donne un extrait de l'équilibre acido-basique.

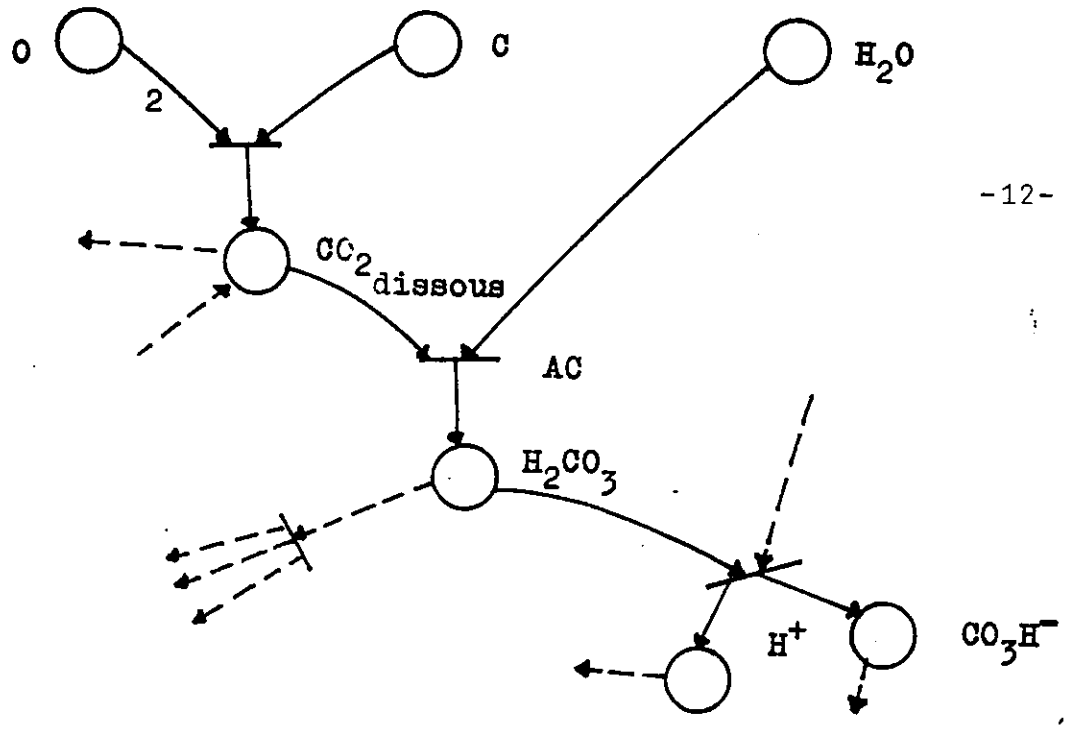


figure 6. Extrait du réseau de Pétri du contrôle de l'équilibre acido-basique.

Nous rappelons qu'un réseau de Pétri généralisé est un doublet $\langle R, \lambda \rangle$ dans lequel :
-R est un réseau de Pétri ordinaire,
-λ est une application qui associe à chaque arc i un entier positif $\lambda_i > o$ appelé poids de l'arc.

Sur le plan graphique les poids des arcs sont indiqués directement sur les arcs (sauf s'il s'agit de poids unitaire, ainsi si aucun poids ne figure sur l'arc implicitement ce poids est égal à un). Une transition est sensibilisée si et seulement si chacune de ses places d'entrée possède un nombre de marques au moins égal au poids de l'arc qui la relie à t_i. La mise à feu d'une transition t_i ne peut s'effectuer que si et seulement si cette transition est sensibilisée. L'opération de mise à feu consiste à prélever dans chacune des places d'entrée de t_i un nombre de marques égal au poids de l'arc qui la connecte à t_i et à placer dans chacune des places de sortie de t_i un nombre de marques égal au poids de l'arc qui la connecte à t_i.

5.CONCLUSION:

Dans cet article, nous avons tenu à mettre en avant l'utilisation de la représentation des connaissances dans une unique structure graphique, celle d'un réseau de Pétri des relations entre éléments de

nature chimique. L'étude de ce système d'information est complétée par une approche objectiviste et subjectiviste permettant d'affiner la représentation des connaissances par des règles qui pouvaient paraître isolées de ce contexte. Le passage de ce système d'information en application informatique est à l'étude à l'aide du langage GAP qui à la particularité grâce à un cycle préetabli de générer des programmes /9/.

6.REFERENCES:

/I/ J.M.KNIPPEL Contribution à l'étude comportementale du système homme-machine de dialyse.
Thèse pour obtenir le Diplôme de Docteur Ingénieur. Université d'Aix-Marseille III. I98I.

/2/ J.L.LAURIERE Représentation des connaissances.
T.S.I. N°2. DUNOD. I98I.

/3/ J.L.PETERSON Petri net theory and the modeling of systems.
Prentice-Hall,inc. I98I.

/4/ G.BROUSSALIAN Répertoire de Kent.
Imprimerie du Vercors. I966.

/5/ R.DAVIS, J.KING An overview of production systems.
In E.W.Elcock and D.Mackie Eds. Machine intelligence 8.
Wiley and Sons. New-York. I977.

/6/ M.FIESCHI Intelligence artificielle en médecine. Des systèmes experts.
MASSON. I984.

/7/ S.G.PAUKER, P.SZOLOVITS Analysing and simulating taking the history of the present illness: context formation.
SCHNEIDER and SAGAWALL HEIN Eds. Computational linguistics in medecine . North-Holland. I977.

/8/ E.POST Formal reduction of the general combinatorial decision problem.
American journal math. I943. Vol 65. N°2.

/9/ M.REMY Le générateur automatique de programme. GAP.
MASSON. 2°édition. I984.

LE MICROPROCESSEUR 8040

S. HILALA

C.R. Subject Classification informatics : B4. C1. C5.

Résumé:

Notre but est de réaliser un système dans lequel il sera possible d'entrer un programme ou des programmes en langage machine, dans une mémoire vive, de les tester, de les corriger et de les faire tourner.

Nous faisons, à priori, le choix du microprocesseur 8 bits le 8040 d'Intel.

Cet article commence par la description de ce microprocesseur. Nous présenterons ensuite l'interfaçage matériel microprocesseur - mémoire morte EPROM, puis l'interfaçage matériel/logiciel 8040 - E/S clavier.

Ces interfaçages seront l'outil de développement de notre système(U.C).

LE MICROPROCESSEUR 8040

DESCRIPTION GENERALE

Le 8040 est un microprocesseur 8 bits sur une seule puce dans un boîtier "dual-in-line" à 40 broches.

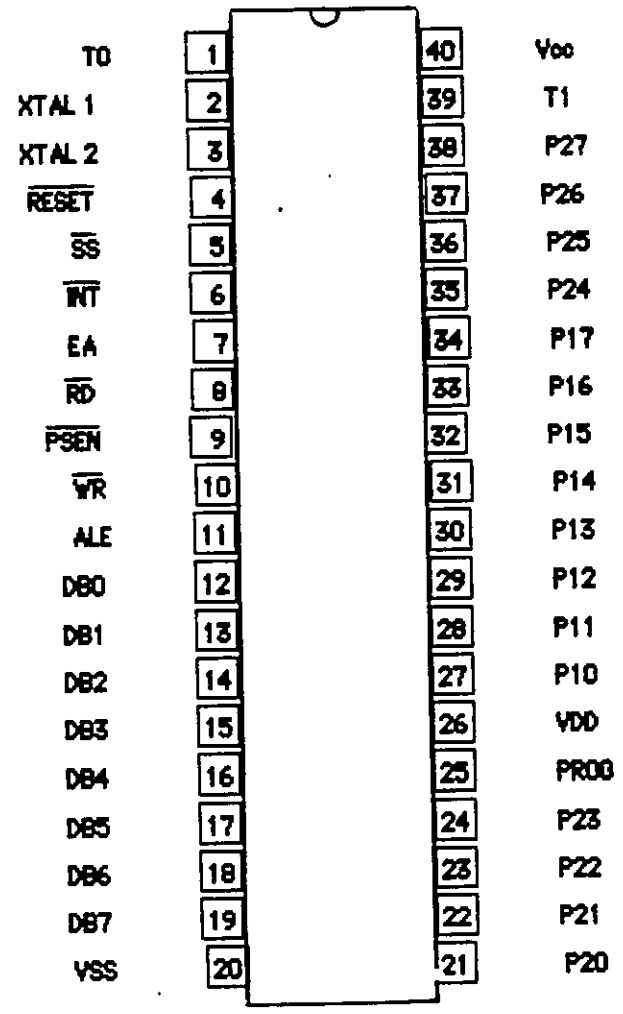
Il incorpore sur la puce :

- Une Unité Centrale "CPU" ,
- Une RAM de 256 octets,
- 27 lignes d' E/S,
- Un système d'horloge (sauf le quartz et 2 capa.).

Il dispose :

- Un mono-bus articulé autour de l'accumulateur,
- 17 registres internes : A + 16 registres adressables,
- 96 instructions de 1 ou 2 cycles (2.5 - 1.36 micro-s),
- 2 ports d' E/S à 8 bits (P1 et P2),
- Bus de contrôle à 4 bits,
- Registre Compteur / Horloge 8 bits " T " .

REPRESENTATION DU 8040



PRINCIPE DE FONCTIONNEMENT :

Le 8040 est un microprocesseur universel complet permettant de réaliser facilement des applications comme :

- interfaçages et contrôle de périphériques,
- instruments de mesures et de test,
- ...

Les échanges entre le 8040 et les dispositifs extérieurs, mémoires ou périphériques, se font à l'aide d'un bus de 8 bits et 2 ports E/S de 8 bits qui remplacent le bus de données (8 bits) et le bus d'adresses (16 bits) habituel.

Le bus est géré selon le diagramme temporel :

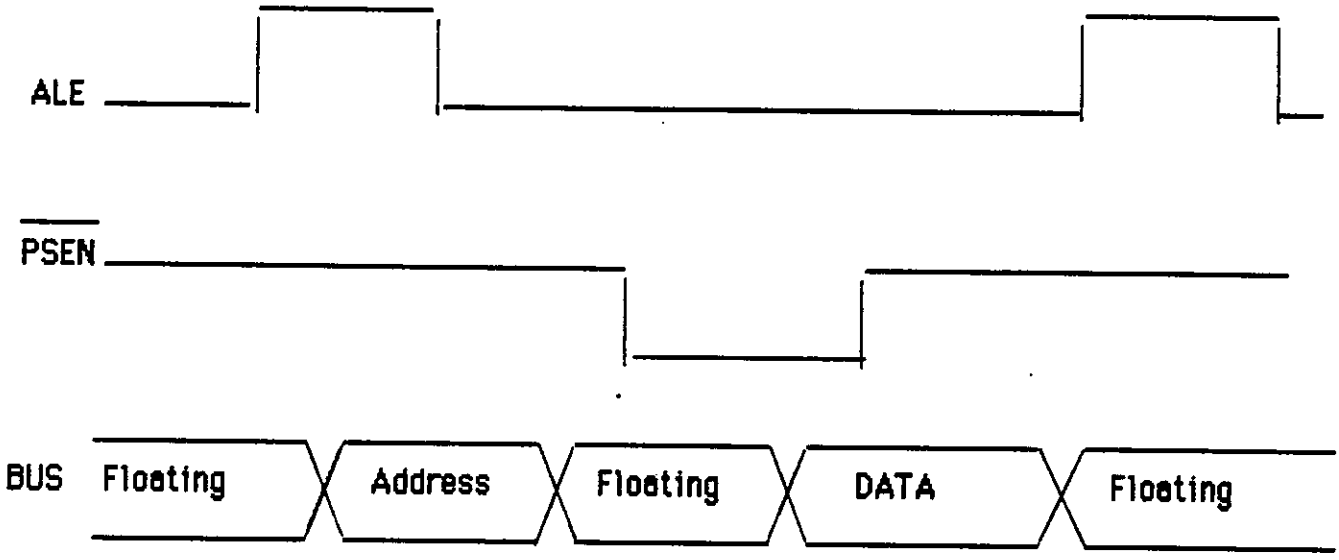


Fig.1. Lecture d'une instruction programme en mémoire externe

Cet artifice permet d'économiser 7 pattes : les 8 du bus d'adresses, mais il en faut une de plus pour le sigal ALE .

Le 8040 a une capacité d'adressage de 4 K octets, ce qui nécessite 12 fils d'adresses répartis sur les 8 fils du bus pour les poids faibles, et les 4 bits : P20 à P23 qui font office de poids forts. ___ ___ ___

Le 8040 fournit des signaux de contrôle (ALE, PSEN, RD et WR) qui indiquent lorsqu'une adresse validée est présente et lorsque le 8040 lit ou écrit une donnée dans la mémoire externe.

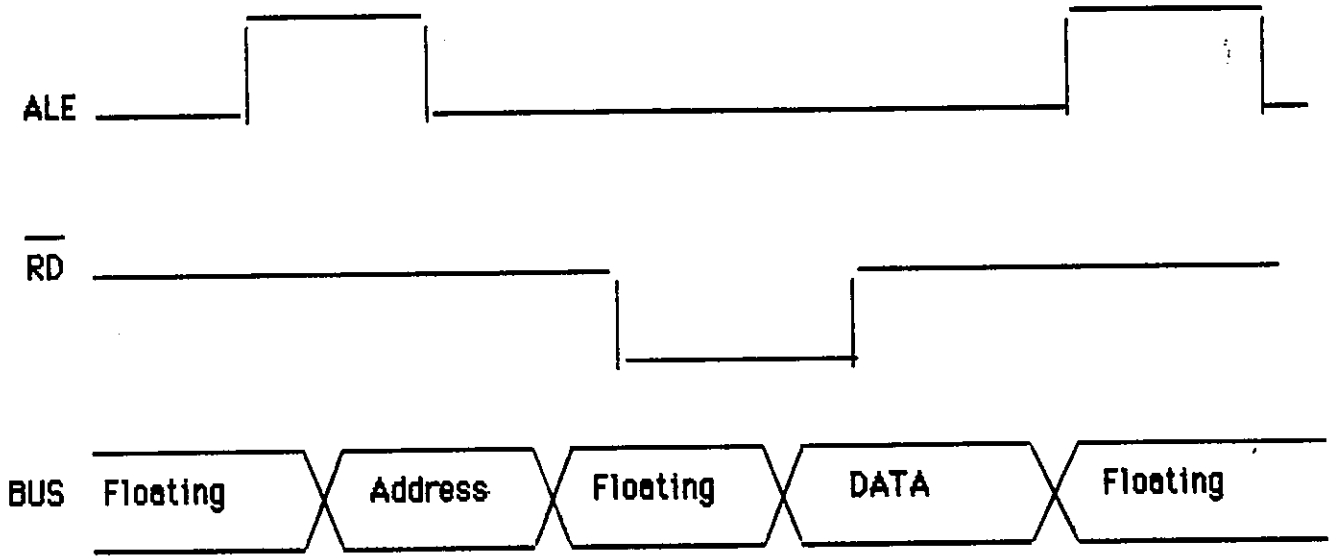


Fig.2. Lecture d'une donnée externe

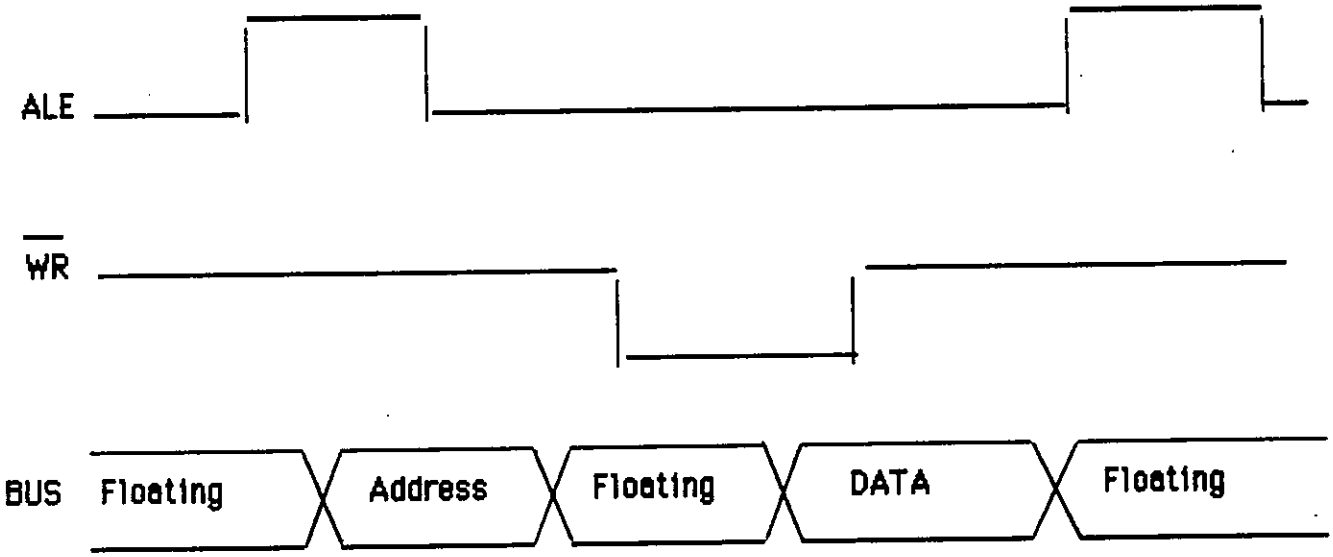


Fig.3. Ecriture dans la mémoire externe

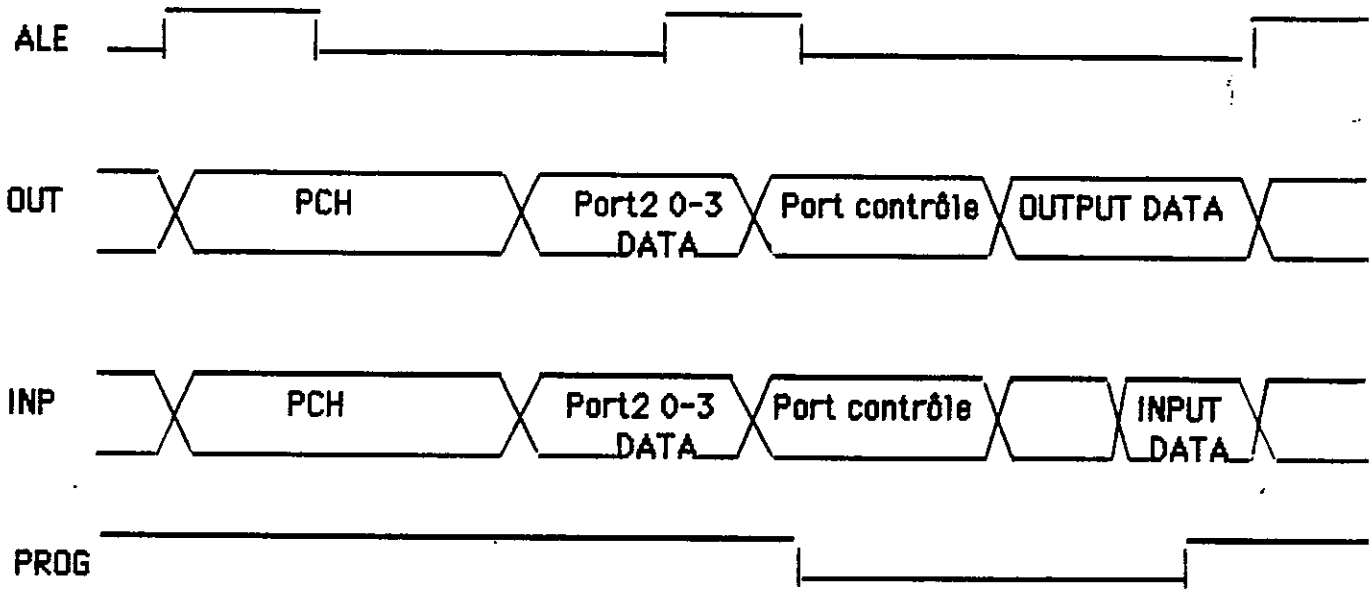


Fig.4. Gestion de P20-23 par le 8243 de ALE et PROG

Le bus de contrôle est composé de 4 signaux :

* ALE : Address Latch Enable
  L'adresse est validée sur son front descendant : elle doit être présente sur le bus.

* $\overline{\text{PSEN}}$: Program Store ENable
  Signale que le 8040 lit une instruction dans la mémoire programme.

* $\overline{\text{RD}}$: ReaD
  Prévient les composants extérieurs que le 8040 effectue la lecture d'une donnée (Fig.2).

* $\overline{\text{WR}}$: WRite
  Signale que le 8040 écrit une donnée dans une mémoire externe (Fig. 3) .

Les autres signaux d'E/S sont destinés à des fonctions générales .

Une description détaillée de chaque signal est donnée en table 1 :

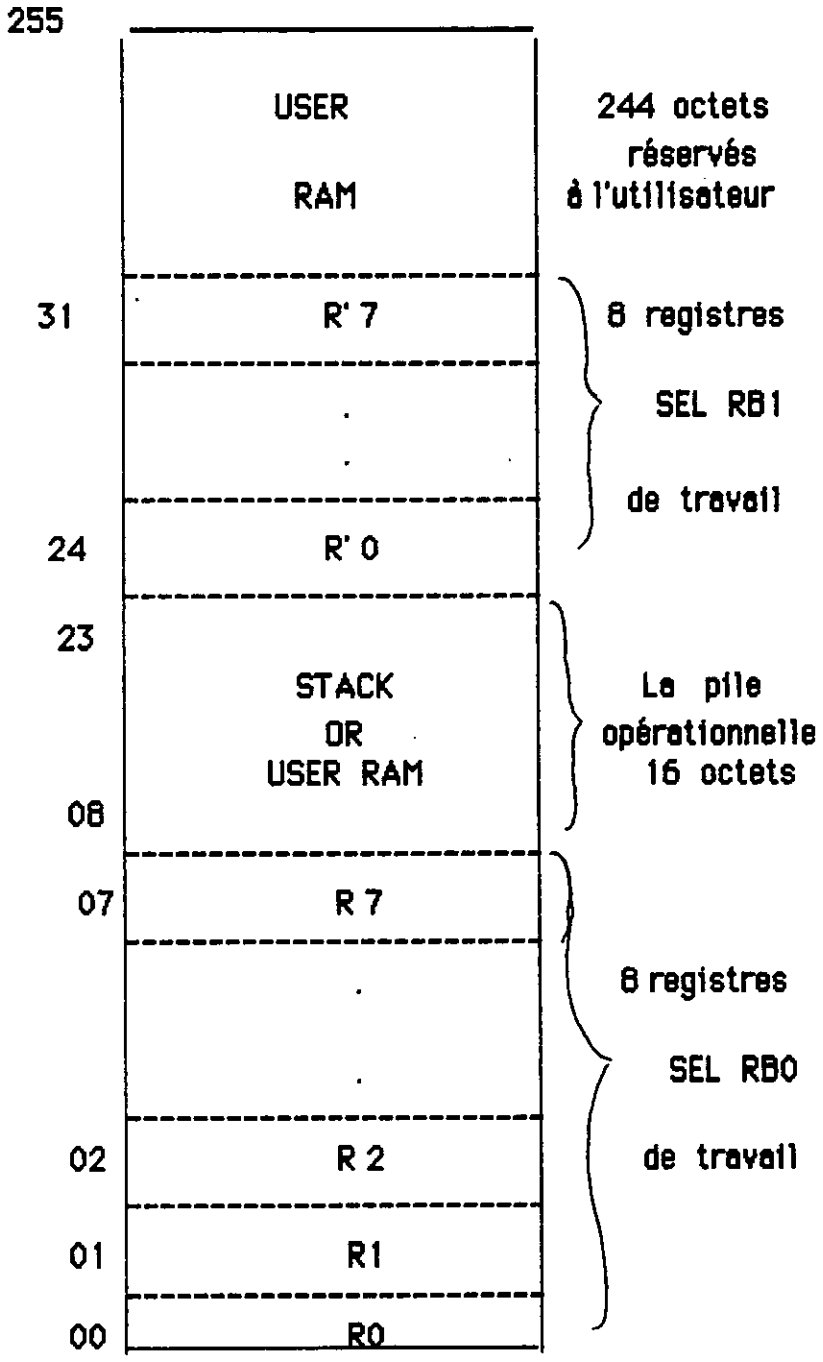
table 1 :

NOM	FONCTION	DESCRIPTION
$\overline{\text{RESET}}$	Entrée Initialisation	Lorsque cette entrée est 0, le 8040 arrête toutes les opérations en cours. quand elle passe de 0 à 1, un circuit de commande interne met à zéro tous les registres accessibles par programme ; ensuite, la première instruction est appelée à l'adresse mémoire (000 H).
$\overline{\text{INT}}$	Entrée interruption	Lorsque cette entrée est à 0, elle provoque une interruption dans le programme courant, si l'interruption externe est permise (ENI). Cette entrée peut être aussi utilisée en entrée consultable permettant au programmeur d'utiliser directement dans le programme des instruction telles que (saut si $\overline{\text{INT}}$ = 0). L'interruption est interdite pendant et après un $\overline{\text{RESET}}$.
$\overline{\text{SS}}$	Entrée pas à pas	Permet l'execution pas à pas .
EA	E. adresse Exterieure	Pour lire et exécuter une instruction de Si EA=0V, le 8040 doit utiliser sa mémoire morte interne. EA=1, s'il doit utiliser une ROM externe.
$\overline{\text{PSEN}}$, ALE, $\overline{\text{RD}}$, $\overline{\text{WR}}$	Bus de contrôle	Voir page 18 .
XTAL1,2	Horloge	Le 8040 possède de manière interne toute l'électronique du compteur d'horloge sauf le quartz et 2 capacités. La sortie de l'oscillateur est divisée par 3 puis par 5 pour définir un Cycle Machine qui sera disponible à ALE .

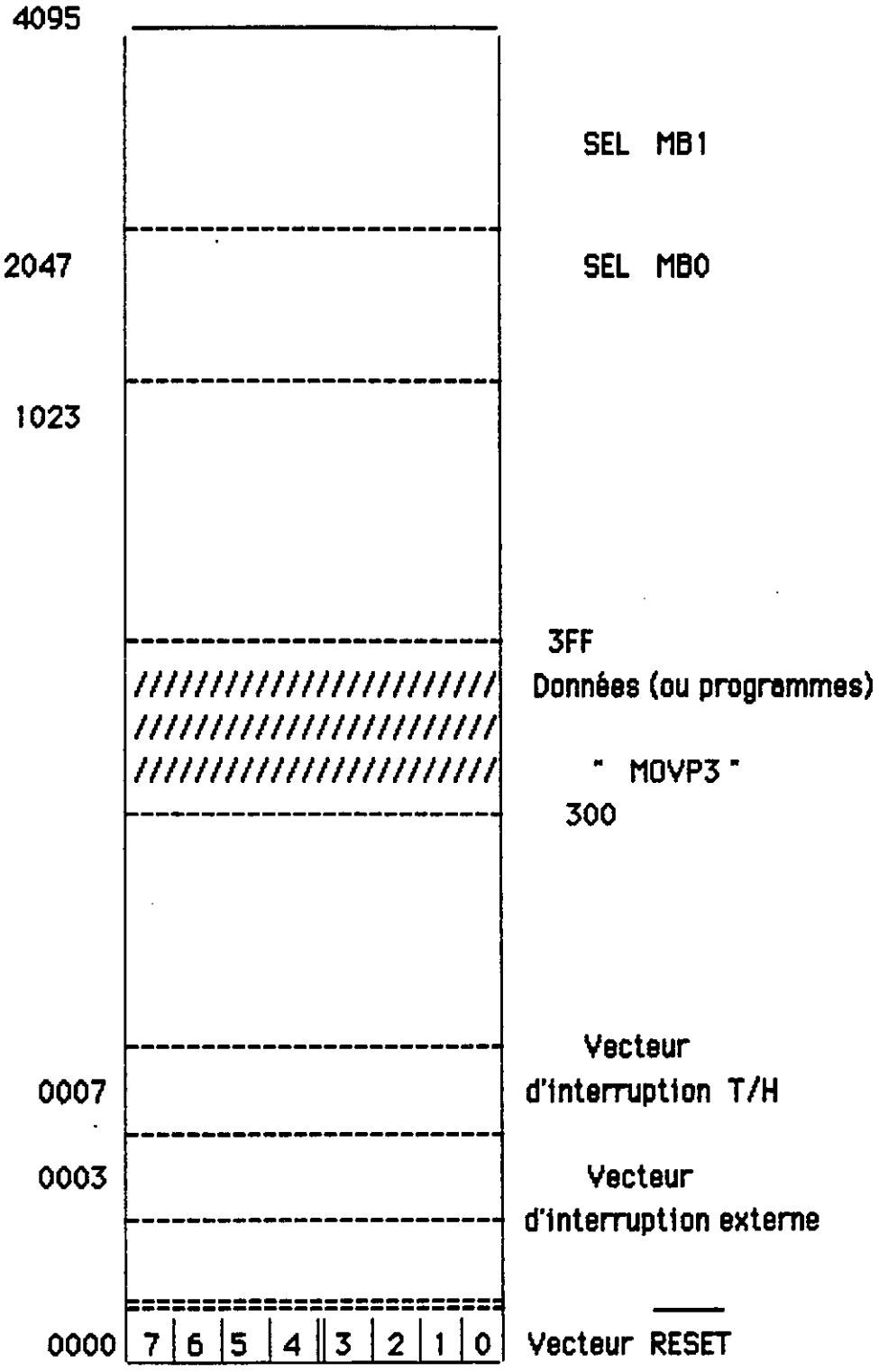
Suite table 1 :

NOM	FONCTION	DESCRIPTION
T0, T1	Ports d'E/S	Peut être utilisé en entrée testable par le programmeur par les instructions de branchement : JT0, JT1, JNT0, JNT1 ou en sortie horloge pour T0 à l'aide de l'instruction ENT0 CLK. La fréquence issue de XTAL est divisée par 3 et est disponible à l'entrée T0. On peut utiliser T1 comme compteur ou comme horloge (chronométreur) par l'instruction STRT CNT (STRT T).
P1 , P2	Ports d'E/S	Outre leur rôle de port d'E/S, les bits P20 à P23 jouent également le rôle de bus d'adresse pour la lecture des instructions de programme (A8 à A11). P20-P23 jouent successivement le rôle de port, de bus d'adresses pour le programme, de port de contrôle et de bus de donnée pour le 8243. Les fronts de ALE et de PROG permettent le démultiplexage (Fig. 4).
PROG	Sortie Expander	Utilisée pour étendre le nombre de ports d'E/S.
DB 0 - 7	BUS E/S	Port d'E/S bidirectionnel à 8 bits est dans l'état haute impédance sauf lorsqu'il est réellement utilisé par le 8040 ($\overline{RD}$, $\overline{WR}$, ALE, $\overline{PSEN}$). (Fig. 1,2 et 3)

[illegible]

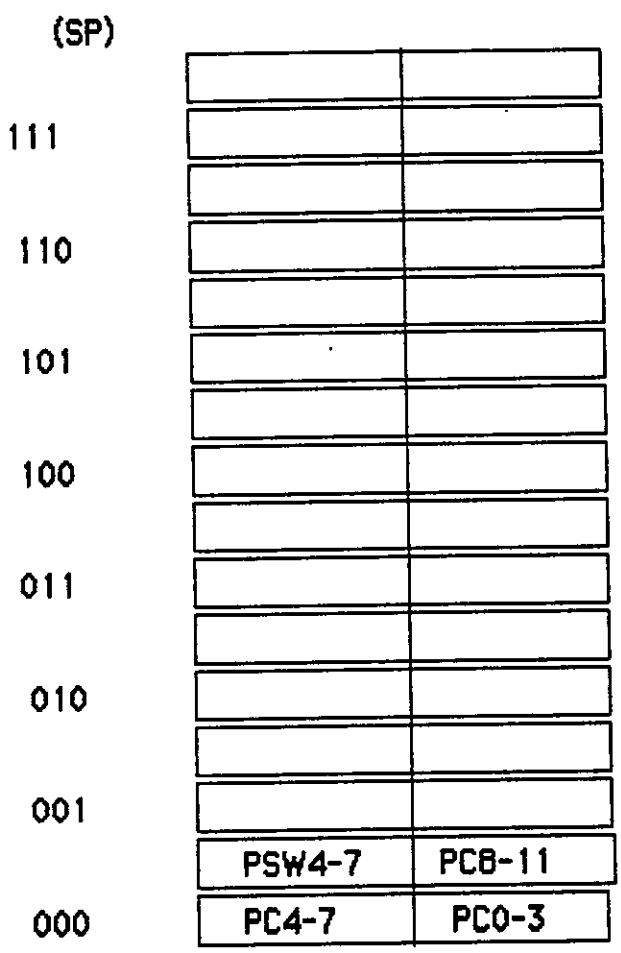

255
USER
RAM
244 octets
réservés
à l'utilisateur
31
R' 7
8 registres
SEL RB1
de travail
24
R' 0
23
STACK
OR
USER RAM
La pile
opérationnelle
16 octets
08
07
R 7
8 registres
SEL RB0
de travail
02
R 2
01
R1
00
R0

Désassemblage de la mémoire PROM externe à 4 K octets :



La Pile Fonctionnelle

Le pointeur de pile (SP) contient l'adresse de la première case libre dans la pile .



PSW : Program Status Word

Le registre d'état (8 bits) mémorise certains critères que le programme peut tester par instruction. Chacun d'eux est représenté par un bit que l'on nomme FLAG.

MBS ---> B7 B6 B5 B4 B3 B2 B1 B0 <---- LSB

PSW : | CY | AC | F0 | BS | 1 | S2 | S1 | S0 |

S0,S1,S2 : Constituent le pointeur de pile (SP)

CY : CARRY (retenue)

AC : Retenue Auxiliaire (demi retenue)

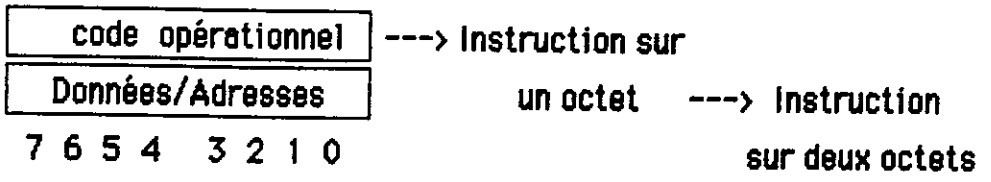
F0 : FLAG0 (Logiciel "branchement")

BS : Registre BANK SELECT " Selection des registres R ", Si BS = 0 Alors ce sont les registres R qui seront concernés .

La valeur de BS est contrôlée par le programme .

Format des instructions :

Le répertoire d'instructions du 8040 comprend des instructions codées sur simple et double octets. Une instruction simple octet spécifie l'opération exécutée par le 8040 sans référence ultérieure à la mémoire. Une instruction double octets comporte un code à 8 bits et 8 bits de données ou de valeur de déplacement :



Stockage des informations :

Le 8040 possède 21 registres internes, parmi lesquels un seul "Ri" n'est pas accessible par programme. Ces registres sont décrits ci-dessous :

Compteur Ordinal "PC" :

C'est un registre à 12 bits qui contient l'adresse de l'instruction en cours d'exécution. Le contenu de ce registre est auomatiquement augmenté de 1 juste avant l'appel de chaque instruction.

Le 8040 ne gère pas PC11, en particulier, lorsque PC = 7FFH, l'incrémentation automatique de PC ne donne pas 800H mais 000H, il travail modulo 800H. Le 8040 n'adresse directement que 2 KO. Dans ce cas, c'est le programmeur qui impose la valeur de PC11 à l'aide des instructions SEL,MB0,SEM,MB1.

Registre d'état "PSW" :

Le rôle du registre d'état est de mémoriser les situations exceptionnelles qui peuvent survenir pendant le fonctionnement du microprocesseur. Le contenu du registre d'état peut être testé par des instructions spécialisées .

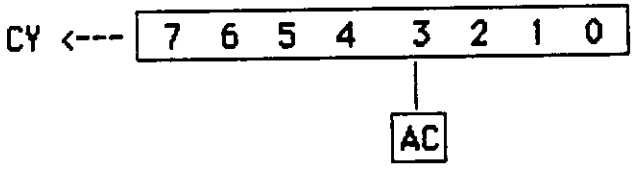
Pour plus de détails sur ce registre voir page 23.

Registre d'instruction :

Le registre d'instruction, à 8 bits, n'est pas accessible par programme.Pendant la phase de chaque instruction ce registre est chargé avec les 8 bits du code d'instructon en provenance de la memoire; (Pour une instruction à un seul octet ou le premier d'une instructin double octets).

[illegible]

L'accumulateur à 8 bits est le registre de travail principal du 8040.Il est utilisé pour exécuter et stocker les résultats des opérations arithmétiques et logiques ainsi que pour les transferts de données, les décalages ,les rotations et les échanges d'informations avec le compteur ordinal, le registre d'état, les ports d'E/S et les autres registres de travail "R0-7, R'0-7". En plus, les operations à 4 bits Decimal Codé Binaire (DCB) peuvent être accomplies de sorte que la retenue auxiliaire (AC) soit maintenue.



AC : utilisé seulement quand on convertit le contenu de l'accumulateur d'un binaire à un DCB.

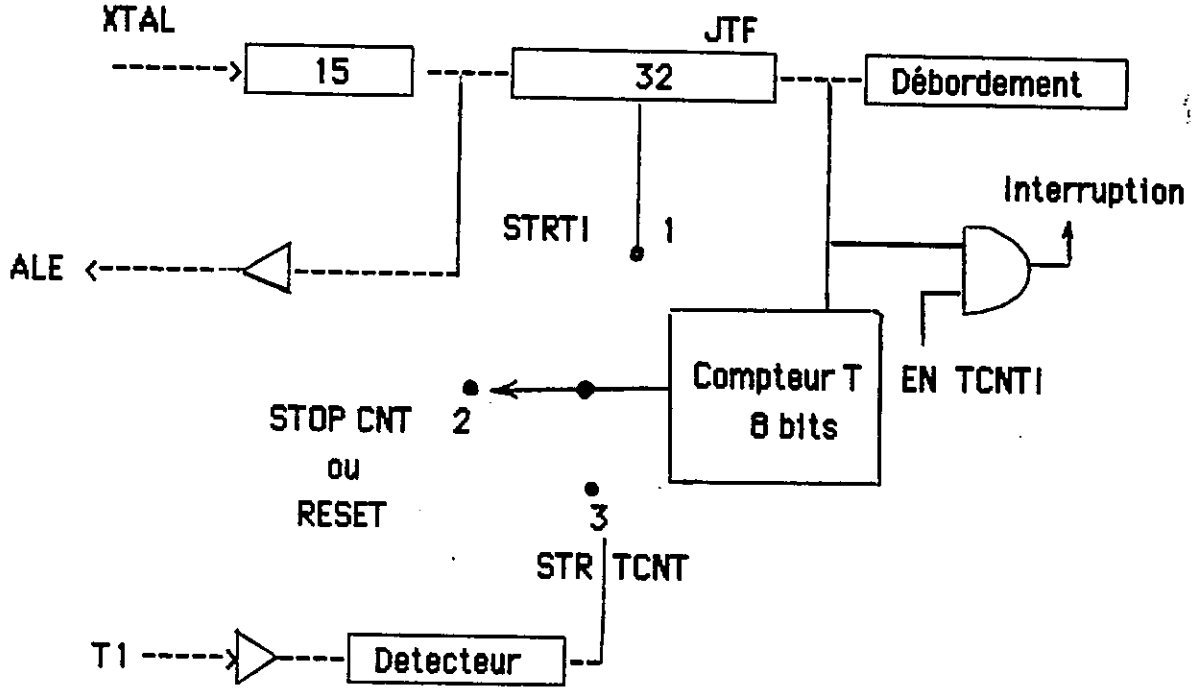
[illegible]

Le 8040 dispose de deux séries de registres de travail, notés R0 à R7 et R'0 à R' 7. Ces registres jouent un rôle particulier au niveau de la programmation.

[illegible]

Un compteur interne est disponible dans le microprocesseur 8040 qui peut compter soit les évènements externes soit les cycles machines. Les cycles machines sont divisés par 32 avant qu'ils ne soient appliqués à l'entrée d'un compteur à 8 bits.

Les évènements externes sont appliqués directement à l'entrée du compteur T. Le débordement du compteur peut interrompre le déroulement normal du programme.



1. Le compteur T compte le temps toutes les 110 micro-secondes "pour un quartz 4 MHZ", le débordement sera signalé après 28160 micro-secondes.

2. Le compteur ne fait rien.

3. Le compteur compte les fronts montants des évènements externes issus de l'entrée T1.

L'Unité Arithmétique et Logique (U A L) :

L'unité arithmétique et logique (UAL) offre la possibilité de manipulation des données , indispensable à tout microprocesseur.

Les opérations peuvent être faites par l'UAL comprenant le OU (OR), le OU exclusif (XOR), le ET (AND), le complément (CPL), le CLEAR (CLR), la rotation avec ou sans débordement (RL,RLC,RR,RRC), incrément, décrément, addition binaire et addition décimale.

Pour l'addition décimale, les données présentées à l'UAL sont traitées comme deux chiffres BCD à 4 bits.

L'instruction DAA permet de faire des opérations en DCB.

Le nombre décimal 43 sera representé par 43H, et non par 2BH.

L'addition de 43 H et de 49 H ne donne pas 92 H mais 8C H.

L'instruction DAA "Decimal Adjust" effectue la transformation

8C H -----> 92 H.

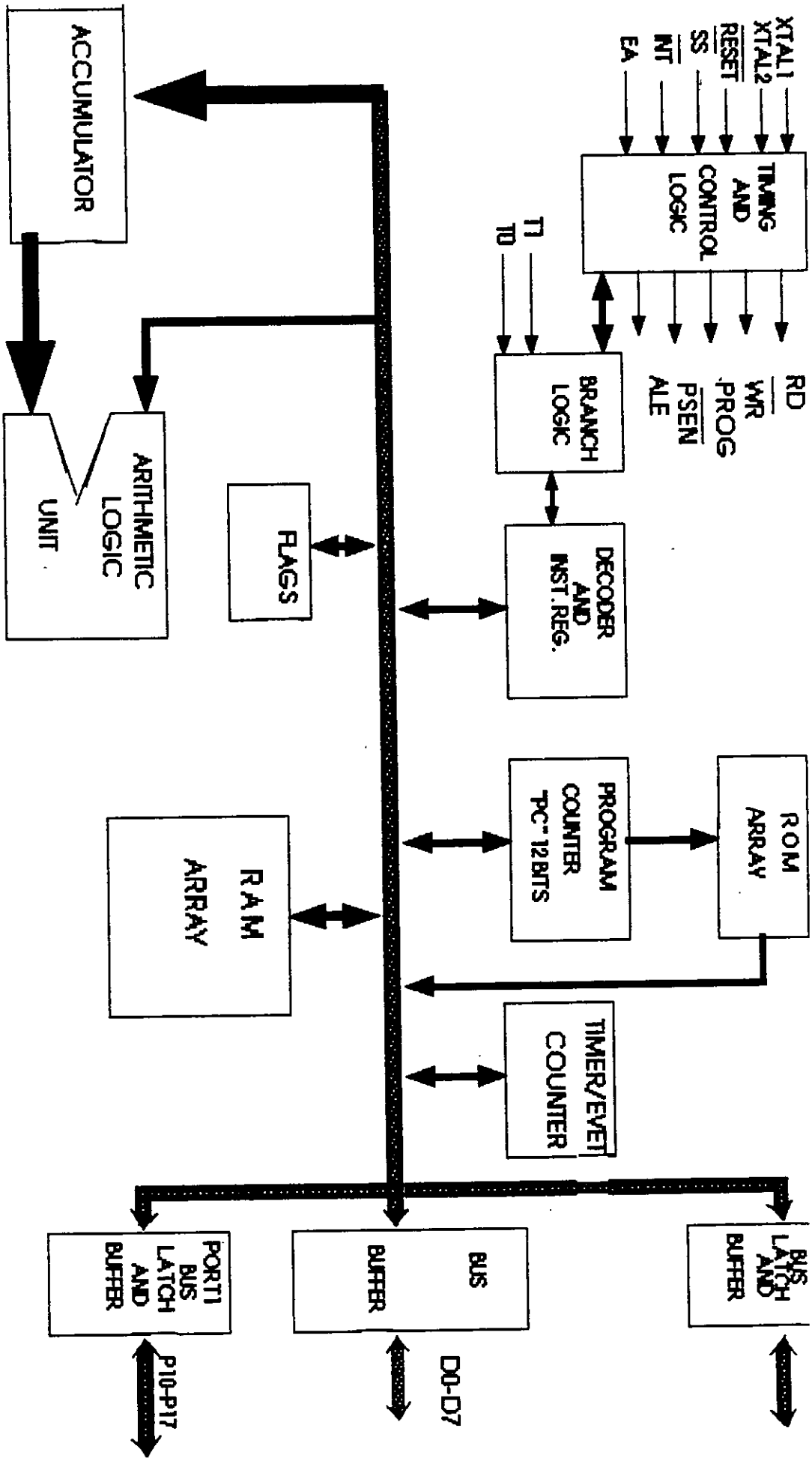


Fig.5. Organisation interne du 8040

Les interruptions du 8040 :

Une interruption consiste en un appel à un sous programme spécifique sans que figure une instruction CALL dans le programme en cours d'exécution. une interruption du 8040 peut être générée par, soit une entrée externe ($\overline{INT}$,patte6),soit le débordement du compteur/horloge interne (T) (Flag TF = 1). Ces interruptions sont dites "masquables" c-à-d que le microprocesseur ne les prendra en compte que si le programme en donne l'autorisation grâce à l'instuction EN.

Dans chaque cas le microprocesseur complète l'exécution de l'instruction en cours, ceci permettant au 8040 de terminer complètement une séquence d'instructions avant toute prise en compte de l'interruption, et il masque automatiquement les autres sources d'interruption.

La première action qui se passe pendant une opération d'interruption est que le registre PC (compteur de programme) est mis en pile. Ceci sauvegarde en mémoire l'adresse à laquelle le 8040 devra retourner après avoir terminé le traitement de l'interruption à l'aide de l'instruction de fin de sous-programme RETR ou RET.

L'emplacement 003 H de la mémoire du système contient le début sous-programme qui doit être déroulé pendant l'exécution de l'interruption $\overline{INT}$, alors que l'emplacement 007 H contient le début du sous-programme de l'interruption de débordement du compteur/horloge T.

Le programmeur doit sauvegarder les registres utilisés pendant le sous-programme d'interruption.

LE JEU D'INSTRUCTIONS DU 8040 :

Le jeu d'instruction du 8040 offre une grande puissance de programmation, de même qu'une bonne vitesse et une bonne souplesse.

Ce jeu d'instructions comprend 96 instructions réparties en 10 catégories principales :

1. Instruction à référence mémoire :

MOV A, @Rj

Format : 1 1 1 1 0 0 0 j — Copie l'octet pointé par Rj dans A.

Interprétation : PC := PC + 1 ; A := (Rj) ;
Cycle Machine : 1

MOV @Rj, A

Format : 1 0 1 0 0 0 0 j — Copie A dans l'octet pointé par Rj .

Interprétation : PC := PC + 1 ; (Rj) := A ;
Cycle Machine : 1

MOVX A, @Rj

Format : 1 0 0 0 0 0 0 j — Copie dans A l'octet pointé par Rj (dans la mémoire externe).

Interprétation : PC := PC + 1 ; BUS-A 0-7 := Rj ;
BUS-A 8-11 := PC 8-11; $\overline{RD}$ est active ; A := BUS-D ;
Cycle Machine : 2

MOVX @Rj, A

Format : 1 0 0 1 0 0 0 j — Copie A dans l'octet pointé par Rj (dans la mémoire externe).

Interprétation : PC := PC + 1 ; BUS-A 0-7 := Rj ; BUS-A 8-11 := PC 8-11 ;
BUS-D := A ; $\overline{WR}$ est active ; (Rj) := BUS-D ;
Cycle Machine : 2

MOVP A, @A

Format : 1 0 0 1 0 0 1 1 — Copie dans A l'octet pointé par A (dans la page courante , même si externe).

Interprétation : PC := PC + 1 ; BUS-A 0-8 := A ;
BUS-A 8-11 := PC 8-11; A := BUS-D;
Cycle Machine : 2

MOVP3 A, @A

Format : 1 1 1 0 0 0 1 1

Copie dans A l'octet pointé par A (dans la page trois même si externe).

Interprétation : PC := PC + 1 ; BUS-A 0-8 := A ;
BUS-A 8-11 := 0 0 1 1 ; A := BUS-D;

Cycle Machine : 2

XCH A, @Rj

Format : 0 0 1 0 0 0 0 j

Echange A et l'octet pointé par Rj dans A.

Interprétation : PC := PC + 1 ; A <-----> (Rj) ;

Cycle Machine : 1

XCHD A, @Rj

Format : 0 0 1 1 0 0 0 j

Echange le poids faible de A et le poids faible de l'octet pointé par A.

Interprétation : PC := PC + 1 ; A 0-3<-----> (Rj)0-3 ;

Cycle Machine : 1

ADD A, @Rj

Format : 0 1 1 0 0 0 0 j

Addition de l'octet pointé par Rj à A.

Interprétation : PC := PC + 1 ; A := A + (Rj); CY := Carry; AC := AUX-Carry;

Cycle Machine : 1

ADDC A, @Rj

Format : 0 1 1 0 0 0 0 j

Addition de l'octet pointé par Rj et de la retenue courante (CY) à A.

Interprétation : PC := PC + 1; A := A + (Rj) + CY;
CY := Carry; AC := AUX-Carry;

Cycle Machine : 1

ANL A, @Rj

Format : 0 1 0 1 0 0 0 j

Intersection bit à bit de l'octet pointé par Rj à A.

Interprétation : PC := PC + 1 ; A := A ET (Rj) ;

Cycle Machine : 1

ORL A, @Rj

Format : `0 1 0 0  0 0 0 j`

Union bit à bit de l'octet pointé par Rj à A.

Interprétation : PC := PC + 1 ; A := A OU (Rj) ;
Cycle Machine : 1

XRL A, @Rj

Format : `1 1 0 1  0 0 0 j`

OU-eXclusif bit à bit de l'octet pointé par Rj à A.

Interprétation : PC := PC + 1 ; A := A OUX (Rj) ;
Cycle Machine : 1

INC @Rj

Format : `0 0 0 1  0 0 0 j`

Incrémentation de l'octet pointé par Rj .

Interprétation : PC := PC + 1 ; (Rj) := (Rj) + 1 ;
Cycle Machine : 1

2. Instructions immédiates :

MOV A, #data

Format : `0 0 1 0  0 0 1 1`
`data`

Copie la valeur "data" sur 8 bits dans A .

Interprétation : PC := PC + 2 ; A := " data " ;
Cycle Machine : 2

MOV Ri, #data

Format : `1 1 0 1  0 " i "`
`data`

Copie la valeur "data" sur 8 bits dans la registre Ri .

Interprétation : PC := PC + 2 ; Ri := " data " ;
Cycle Machine : 2

MOV @Rj, #data

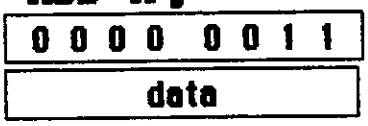
Format : `1 0 1 1  0 0 0 j`
`data`

Copie la valeur "data" sur 8 bits dans l'octet pointé par Rj .

Interprétation : PC := PC + 2 ; (Rj) := " data " ;
Cycle Machine : 2

ADD A, # data

Format :

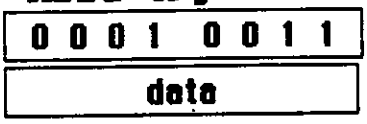


Addition la valeur "data" sur 8 bits à A .

Interprétation : PC := PC + 2; A := A + " data ";
CY := Carry ; AC := AUX-Carry;
Cycle Machine : 2

ADC A, # data

Format :

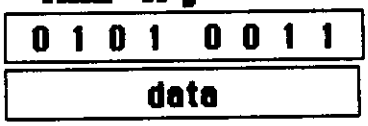


Addition la valeur "data" sur 8 bits et la retenue courante (CY) à A .

Interprétation : PC := PC +2; A := A+CY+" data "; CY := Carry; AC := AUX-Carry;
Cycle Machine : 2

ANL A, # data

Format :

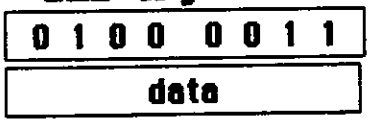


Intersection bit à bit de la valeur "data" sur 8 bits avec A .

Interprétation : PC := PC + 2 ; A := A [illegible] " data " ;
Cycle Machine : 2

[illegible] A, # data

Format :

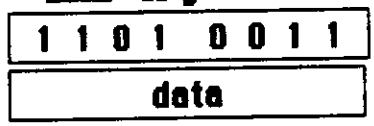


Union bit à bit de la valeur "data" sur 8 bits avec A .

Interprétation : PC := PC + 2 ; A := A [illegible] " data " ;
Cycle Machine : 2

[illegible] A, # data

Format :



OU-exclusif bit à bit de la valeur "data" sur 8 bits avec A .

Interprétation : PC := PC + 2 ; A := A [illegible] " data " ;
Cycle Machine : 2

3. Instructions à référence registre :

MOV A , Ri

Format : 1 1 1 1 1 " i "

Copie le registre Ri dans A.

Interprétation : PC := PC + 1 ; A := Ri ;
Cycle Machine : 1

MOV Ri , A

Format : 1 0 1 0 1 " i "

Copie A dans le registre Ri .

Interprétation : PC := PC + 1 ; Ri := A ;
Cycle Machine : 1

XCH A , Ri

Format : 0 0 1 0 0 " i "

Echange le registre Ri avec A .

Interprétation : PC := PC + 1 ; A <----> Ri ;
Cycle Machine : 1

MOV A , PSW

Format : 1 1 0 0 0 1 1 1

Copie le registre d'état PSW dans A .

Interprétation : PC := PC + 1 ; A := PSW ;
Cycle Machine : 1

MOV PSW , A

Format : 1 1 0 1 0 1 1 1

Copie A dans le registre d'état PSW .

Interprétation : PC := PC + 1 ; PSW := A ;
Cycle Machine : 1

MOV T , A

Format : 0 1 1 0 0 0 1 0

Copie A dans le registre T .

Interprétation : PC := PC + 1 ; T := A ;
Cycle Machine : 1

MOV A , T

Format : 0 1 0 0 0 0 1 0

Copie dans A le registre T .

Interprétation : PC := PC + 1 ; A := T ;
Cycle Machine : 1

INC Ri

Format : 0 0 0 1 0 " i " Incrémentation du registre Ri .

Interprétation : PC := PC + 1 ; Ri := Ri + 1 ;
Cycle Machine : 1

DEC Ri

Format : 1 1 0 0 1 " i " Decrémentation du registre Ri .

Interprétation : PC := PC + 1 ; Ri := Ri - 1 ;

Cycle Machine : 1

ADD A, Ri

Format : 0 1 1 0 1 " i " Additionne à A le registre Ri .

Interprétation : PC := PC + 1 ; A := A + Ri ; CY := Carry ;
Cycle Machine : 1

ADDC A, Ri

Format : 0 1 0 1 1 " i " Additionne à A le registre Ri et la retenue courante (CY) .

Interprétation : PC := PC + 1 ; A := A + Ri + CY ; CY := Carry;
Cycle Machine : 1

ANL A, Ri

Format : 0 1 1 0 1 " i " Intersection bit à bit du registre Ri avec A .

Interprétation : PC := PC + 1 ; A := A ET Ri ;
Cycle Machine : 1

ORL A, Ri

Format : 0 1 0 0 1 " i " Union bit à bit du registre Ri avec A .

Interprétation : PC := PC + 1 ; A := A OU Ri ;
Cycle Machine : 1

XRL A, Ri

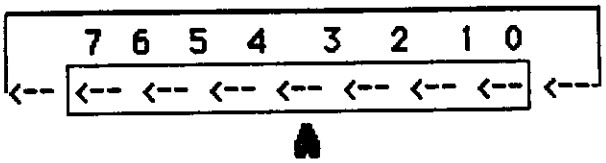
Format : 1 0 1 1 1 " i " OU-eXclusif bit à bit du registre Ri avec A .

Interprétation : PC := PC + 1 ; A := A OUX Ri ;
Cycle Machine : 1

4. Instructions sur l'accumulateur:

RL A

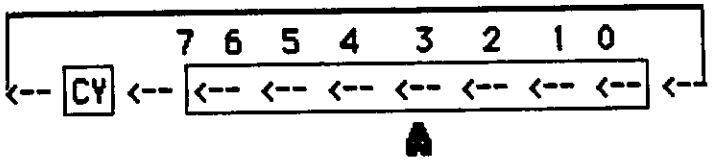
Format : 1 1 1 0 0 1 1 1 Décalage à gauche de A .



Interprétation : PC := PC + 1 ; A 7-1 := A 6-0 ; A 0 := A 7 ;
Cycle Machine : 1

RLC A

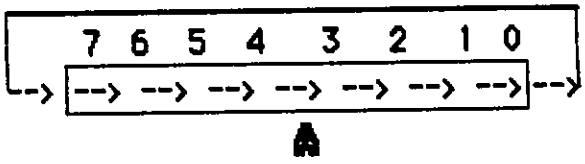
Format : 1 1 1 1 0 1 1 1 Décalage à gauche de A avec CY .



Interprétation : PC := PC + 1 ; A 7-1 := A 6-0 ; A 0 := CY ; CY := A 7 ;
Cycle Machine : 1

RR A

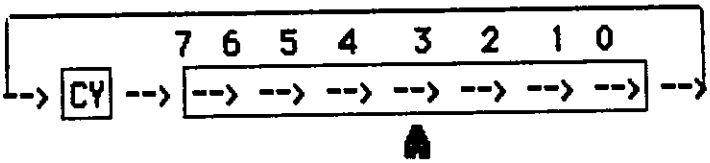
Format : 0 1 1 1 0 1 1 1 Décalage à droite de A .



Interprétation : PC := PC + 1 ; A 6-0 := A 7-1 ; A 7 := A 0 ;
Cycle Machine : 1

RRC A

Format : 1 1 1 1 0 1 1 1 Décalage à gauche de A avec CY .



Interprétation : PC := PC + 1 ; A 6-0 := A 7-1 ; A 7 := CY ; CY := A 0 ;
Cycle Machine : 1

SWAP A

Format : 0 1 0 0 0 1 1 1 — Echange A 0-3 et A 4-7 .

Interprétation : PC := PC + 1 ; A 0-3 <----> A 4-7 ;

Cycle Machine : 1

INC A

Format : 0 0 0 1 0 1 1 1 — Incrémentation de A .

Interprétation : PC := PC + 1 ; A := A + 1 ;

Cycle Machine : 1

DEC A

Format : 0 0 0 0 1 1 1 — Décrémentation de A .

Interprétation : PC := PC + 1 ; A := A - 1 ;

Cycle Machine : 1

CLR A

Format : 0 0 1 0 0 1 1 1 — Remise à zéro de A .

Interprétation : PC := PC + 1 ; A := 0 ;

Cycle Machine : 1

CPL A

Format : 0 0 1 1 0 1 1 1 — Complémentation bit à bit de A .

Interprétation : PC := PC + 1 ; $A := \overline{A}$;

Cycle Machine : 1

DA A

Format : 0 1 0 1 0 1 1 1 — Ajustement décimal de A .

Interprétation : PC := PC + 1 ;

Cycle Machine : 1

5. Instructions sur les ports P1 et P2 :

IN A, Pp

Format : 0 0 0 0 1 0 "p"

Réceptionne dans A l'information prise sur le port Pp (p=1-2).

Interprétation : PC := PC + 1 ; A := Pp ;

Cycle Machine : 2

OUTL Pp, A

Format : 0 0 1 1 1 0 "p"

Envoie l'information de A sur le port Pp (p=1-2).

Interprétation : PC := PC + 1 ; Pp := A ;

Cycle Machine : 2

ANL Pp, # data

Format : 1 0 0 1 1 0 "p"
data

Intersection bit à bit de la valeur "data" sur 8 bits avec Pp (p=1-2).

Interprétation : PC := PC + 2 ; Pp := Pp ET " data " ;
Cycle Machine : 2

ORL Pp, # data

Format : 1 0 0 0 1 0 "p"
data

Union bit à bit de la valeur "data" sur 8 bits avec Pp (p=1-2).

Interprétation : PC := PC + 2 ; Pp := Pp OU " data " ;
Cycle Machine : 2

6. Instructions qui utilisent le bus de données comme port :

INS A, BUS

Format : 0 0 0 0 1 0 0 0

Réceptionne dans A l'information prise sur le bus de données .

Interprétation : PC := PC + 1 ; A := BUS ; $\overline{RD}$ est active ;
Cycle Machine : 2

OUTL BUS , A

Format : 0 0 0 0 1 0 0 0

Réceptionne l'information de A prise sur le bus de données .

Interprétation : PC := PC + 1 ; BUS := A ; $\overline{WR}$ est active ;
Cycle Machine : 2

ANL BUS , # data

Format : 1 0 0 1 1 0 0 0
data

Intersection bit à bit de la valeur "data" sur 8 bits avec l'information sur le bus de données.

Interprétation : PC := PC + 2 ; BUS := BUS ET " data " ;
Cycle Machine : 2

ORL BUS , # data

Format : 1 0 0 0 1 0 0 0
data

Union bit à bit de la valeur "data" sur 8 bits avec l'information du bus de données.

Interprétation : PC := PC + 2 ; BUS := BUS OU " data " ;
Cycle Machine : 2

7. Instructions d'extension d'entrées/sorties 8243 :

MOVD A , Pe

Format : 0 0 0 0 1 0 "e"

Copie dans A l'information du port de 8243 Pe (e=4-7).

Interprétation : PC := PC + 1 ; A 0-3 := Pe ; A 4-7 := 0 ;

Cycle Machine : 2

MOVD Pe , A

Format : 0 0 1 1 1 0 "e"

Copie l'information de A sur le port de 8243 Pe (e=4-7).

Interprétation : PC := PC + 1 ; Pe := A 0-3 ;

Cycle Machine : 2

ANLD Pe , A

Format : 1 0 0 1 1 0 "e" Intersection bit à bit de A avec l'information issue du port de 8243 Pe (e=4-7).

Interprétation : PC := PC + 1 ; Pe := Pe ET A 0-3 ;

Cycle Machine : 2

ORLD Pe , A

Format : 1 0 0 0 1 0 "e" Union bit à bit de A avec l'information issue du port de 8243 Pe (e=4-7).

Interprétation : PC := PC + 1 ; Pe := Pe OU A 0-3 ;

Cycle Machine : 2

8. Instructions sur les flags :

CLR C

Format : 1 0 0 1 0 1 1 1 Remise à zéro de la retenue CY.

Interprétation : PC := PC + 1 ; CY := 0 ;

Cycle Machine : 1

CLR F0

Format : 1 0 0 0 0 1 0 1 Remise à zéro du flag F0.

Interprétation : PC := PC + 1 ; F0 := 0 ;

Cycle Machine : 1

CLR F1

Format : 1 0 1 0 0 1 0 1 Remise à zéro du flag F1.

Interprétation : PC := PC + 1 ; F1 := 0 ;

Cycle Machine : 1

CPL C

Format : 1 0 1 0 0 1 1 1 Complémentation de la retenue CY .

Interprétation : PC := PC + 1 ; CY := $\overline{CY}$;

Cycle Machine : 1

CPL F0

Format : 1 0 0 1 0 1 0 1 Complémentation du flag F0 .

Interprétation : PC := PC + 1 ; F0 := $\overline{F0}$;

Cycle Machine : 1

CPL F1

Format : 1 0 0 1 0 1 0 1 Complémentation du flag F1 .

Interprétation : PC := PC + 1 ; F1 := $\overline{F1}$;

Cycle Machine : 1

9. Instructions de contrôle :

SEL RB0

Format : 1 1 0 0 0 1 0 1 Sélection des registres R .

Interprétation : PC := PC + 1 ; BS := 0 ;

Cycle Machine : 1

SEL RB1

Format : 1 1 0 1 0 1 0 1 Sélection des registres R' .

Interprétation : PC := PC + 1 ; BS := 1 ;

Cycle Machine : 1

SEL MB0

Format : 1 1 1 0 0 1 0 1

Sélection du banc 0 de la mémoire programme "adresses 0 - 2047" .

Interprétation : PC := PC + 1 ; PC 11 := 0 ;

Cycle Machine : 1

SEL MB1

Format : 1 1 1 1 0 1 0 1

Sélection du banc 1 de la mémoire programme "adresses 2048 - 4095" .

Interprétation : PC := PC + 1 ; PC 11 := 1 ;

Cycle Machine : 1

EN I

Format : 0 0 0 0 0 1 0 1

Autorisation de la prise en compte d'interruption externe " $\overline{\text{INT}}$ " .

Interprétation : PC := PC + 1 ;

Cycle Machine : 1

DIS I

Format : 0 0 0 0 0 1 0 1

Interdiction de la prise en compte d'interruption externe " $\overline{\text{INT}}$ " .

Interprétation : PC := PC + 1 ;

Cycle Machine : 1

EN T0 CLK

Format : 0 1 1 1 0 1 0 1

Autorisation T0 en sortie horloge .

Interprétation : PC := PC + 1 ;

Cycle Machine : 1

EN TCNTI

Format : 0 0 1 0 0 1 0 1

Autorisation de l'interruption compteur / horloge .

Interprétation : PC := PC + 1 ;

Cycle Machine : 1

DIS TCNTI

Format : 0 0 1 1 0 1 0 1 — Interdiction de l'interruption compteur / horloge .

Interprétation : PC := PC + 1 ;

Cycle Machine : 1

STRT T

Format : 0 1 0 1 0 1 0 1 — Démarrage horloge .

Interprétation : PC := PC + 1 ;

Cycle Machine : 1

STRT CNT

Format : 0 1 0 0 0 1 0 1 — Démarrage compteur .

Interprétation : PC := PC + 1 ;

Cycle Machine : 1

STOP TCNT

Format : 0 1 1 0 0 1 0 1 — Arrêter compteur / horloge .

Interprétation : PC := PC + 1 ;
Cycle Machine : 1

IDLE

Format : 0 0 0 0 0 0 0 1 — Le mode IDLE est terminé par le signal RESET, ou par un des deux interruptions ($\overline{INT}$,TF) si elles sont autorisées.

Interprétation : PC := PC + 1 ;
IF $\overline{INT}$ = 0 THEN PC := 003 H
ELSE IF TF = 1 THEN PC := 007 H
ELSE RAM, R, R', P1, P2 et BUS sont maintenus ;

Cycle Machine : 1

NOP

Format : 0 0 0 0 0 0 0 0 — Ne rien faire .

Interprétation : PC := PC + 1 ;
Cycle Machine : 1

10. Instructions de branchements et appel des sous programmes :

JMP @ adr

Format :

a10	a9	a8	0	0	1	1	0
a7	a6	a5	a4	a3	a2	a1	a0

Saut direct à l'adresse adr "a l'interieur du bloc de 2K octets " .

Interprétation : PC 0-7 := adr 0-7 ; PC 8-10 := adr 8-10;
PC 11 := BS ;

Cycle Machine : 2

JNC @ adr

Format :

1	1	1	0	0	1	1	0
a7	a6	a5	a4	a3	a2	a1	a0

Saut conditionnel à l'adresse adr (si CY=0) .

Interprétation : IF CY = 0 THEN PC 0-7 := adr 0-7
ELSE PC := PC + 2 ;

Cycle Machine : 2

JC @ adr

Format :

1	1	1	1	0	1	1	0
a7	a6	a5	a4	a3	a2	a1	a0

Saut conditionnel à l'adresse adr (si CY=1).

Interprétation : IF CY = 1 THEN PC 0-7 := adr 0-7
ELSE PC := PC + 2 ;

Cycle Machine : 2

JZ @ adr

Format :

1	1	0	0	0	1	1	0
a7	a6	a5	a4	a3	a2	a1	a0

Saut conditionnel à l'adresse adr (si A=0) .

Interprétation : IF A = 0 THEN PC 0-7 := adr 0-7
ELSE PC := PC + 2 ;

Cycle Machine : 2

JNZ @ adr

Format :

1	0	0	1	0	1	1	0
a7	a6	a5	a4	a3	a2	a1	a0

Saut conditionnel à l'adresse adr (si A ≠ 0).

Interprétation : IF A ≠ 0 THEN PC 0-7 := adr 0-7
ELSE PC := PC + 2 ;

Cycle Machine : 2

JT1 @ adr

Format :

0	1	0	1	0	1	1	0
a7	a6	a5	a4	a3	a2	a1	a0

Saut conditionnel à l'adresse adr (si T1=1).

Interprétation : IF T1 = 1 THEN PC 0-7 := adr 0-7
ELSE PC := PC + 2 ;

Cycle Machine : 2

JNT1 @ adr

Format :

0	1	0	0	0	1	1	0
a7	a6	a5	a4	a3	a2	a1	a0

Saut conditionnel à l'adresse adr (si T1=0).

Interprétation : IF T1 = 0 THEN PC 0-7 := adr 0-7
ELSE PC := PC + 2 ;

Cycle Machine : 2

JT0 @ adr

Format :

0	0	1	1	0	1	1	0
a7	a6	a5	a4	a3	a2	a1	a0

Saut conditionnel à l'adresse adr (si T0=1).

Interprétation : IF T0 = 1 THEN PC 0-7 := adr 0-7
ELSE PC := PC + 2 ;

Cycle Machine : 2

JBT0 @ adr

Format :

0	0	1	0	0	1	1	0
a7	a6	a5	a4	a3	a2	a1	a0

Saut conditionnel à l'adresse adr (si T0=0).

Interprétation : IF T0 = 0 THEN PC 0-7 := adr 0-7
ELSE PC := PC + 2 ;

Cycle Machine : 2

JF0 @ adr

Format :

1	0	1	1	0	1	1	0
a7	a6	a5	a4	a3	a2	a1	a0

Saut conditionnel à l'adresse adr (si F0=1).

Interprétation : IF F0 = 1 THEN PC 0-7 := adr 0-7
ELSE PC := PC + 2 ;

Cycle Machine : 2

JF1 @ adr

Format :

0	1	1	1	0	1	1	0
a7	a6	a5	a4	a3	a2	a1	a0

Saut conditionnel à l'adresse adr (si F1=0).

Interprétation : IF F1 = 0 THEN PC 0-7 := adr 0-7
ELSE PC := PC + 2 ;

Cycle Machine : 2

JTF @ adr

Format :

0	0	0	1	0	1	1	0
a7	a6	a5	a4	a3	a2	a1	a0

Saut conditionnel à l'adresse adr (si TF=1).

Interprétation : IF TF = 1 THEN PC 0-7 := adr 0-7
ELSE PC := PC + 2 ;

Cycle Machine : 2

JBI @ adr

Format :

1	0	0	0	0	1	1	0
a7	a6	a5	a4	a3	a2	a1	a0

Saut conditionnel à l'adresse adr (si $\overline{INT}$=0).

Interprétation : IF $\overline{INT}$ = 0 THEN PC 0-7 := adr 0-7
ELSE PC := PC + 2 ;

Cycle Machine : 2

JBI @ adr

Format :

0	0	0	1	0	1	1	0
a7	a6	a5	a4	a3	a2	a1	a0

Saut conditionnel à l'adresse adr (si le bit n° i de A est à 1).

Interprétation : IF Ai = 1 THEN PC 0-7 := adr 0-7
ELSE PC := PC + 2 ;

Cycle Machine : 2

DJNZ Ri , @ adr

Format :

0	0	0	1	0	"	i	"
a7	a6	a5	a4	a3	a2	a1	a0

Décrementation de Ri, si Ri ≠ 0 saut à l'adresse adr .

Interprétation : Ri := Ri - 1 ; IF Ri ≠ 0 THEN PC 0-7 := adr 0-7
ELSE PC := PC + 2 ;

Cycle Machine : 2

CALL @ adr

Format :

a10	a9	a8	1	0	1	0	0
a7	a6	a5	a4	a3	a2	a1	a0

Appel de sous programme à l'addresse adr .

Interprétation : PC := PC + 2 ; (X) := PC 0-7; " où X = 8+2*SP "
(X+1) 0-3 := PC 8-11 ; (X) 4-7 := PSW 4-7 ; SP := SP + 1 ;
PC 0-7 := adr 0-7 ; PC 8-10 := adr 8-10;

Cycle Machine : 2

RET

Format :

1	0	0	0	0	0	1	1

Retour de sous programme sans restitution du PSW.

Interprétation : SP := SP - 1 ; PC 0-7 := (X); " où X = 8+2*SP "
PC 8-11 := (X+1) 0-3 ;

Cycle Machine : 2

RETR

Format : 1 0 0 1 0 0 1 1 — Retour de sous programme avec restitution du PSW.

Interprétation : SP := SP - 1 ; PC 0-7 := (X); " où X = 8+2*SP "
PC 8-11 := (X+1) 0-3 ; PSW 4-7 := (X+1) 4-7 ;

Cycle Machine : 2

APPLICATIONS DU 8040 :

I. Connexion du 8040 à une EPROM :

Une EPROM 2716 possède 2048 cellules mémoire de 8 bits (2 K Octets) . La liaison entre les lignes d'adresse, les données du 8040 et une EPROM 2716 s'effectue en reliant les pattes de données du 2716 (01-08) au bus de données du 8040 (DB0 - DB7), les pattes d'adresses du 2716 (A0 -A10) au bus d'adresses du 8040 (D01 - D08,P20 - P22) comme le montre la figure 6 .

- La patte 18 ($\overline{CS}$) est à 0 : l'EPROM est active .
- La patte 20 ($\overline{G}$) est contrôlé par PSEN pour la lecture des instructios de programme dans l'EPROM.
- La patte 21 ($\overline{W}$) est à 1 : l'EPROM est en mode lecture .

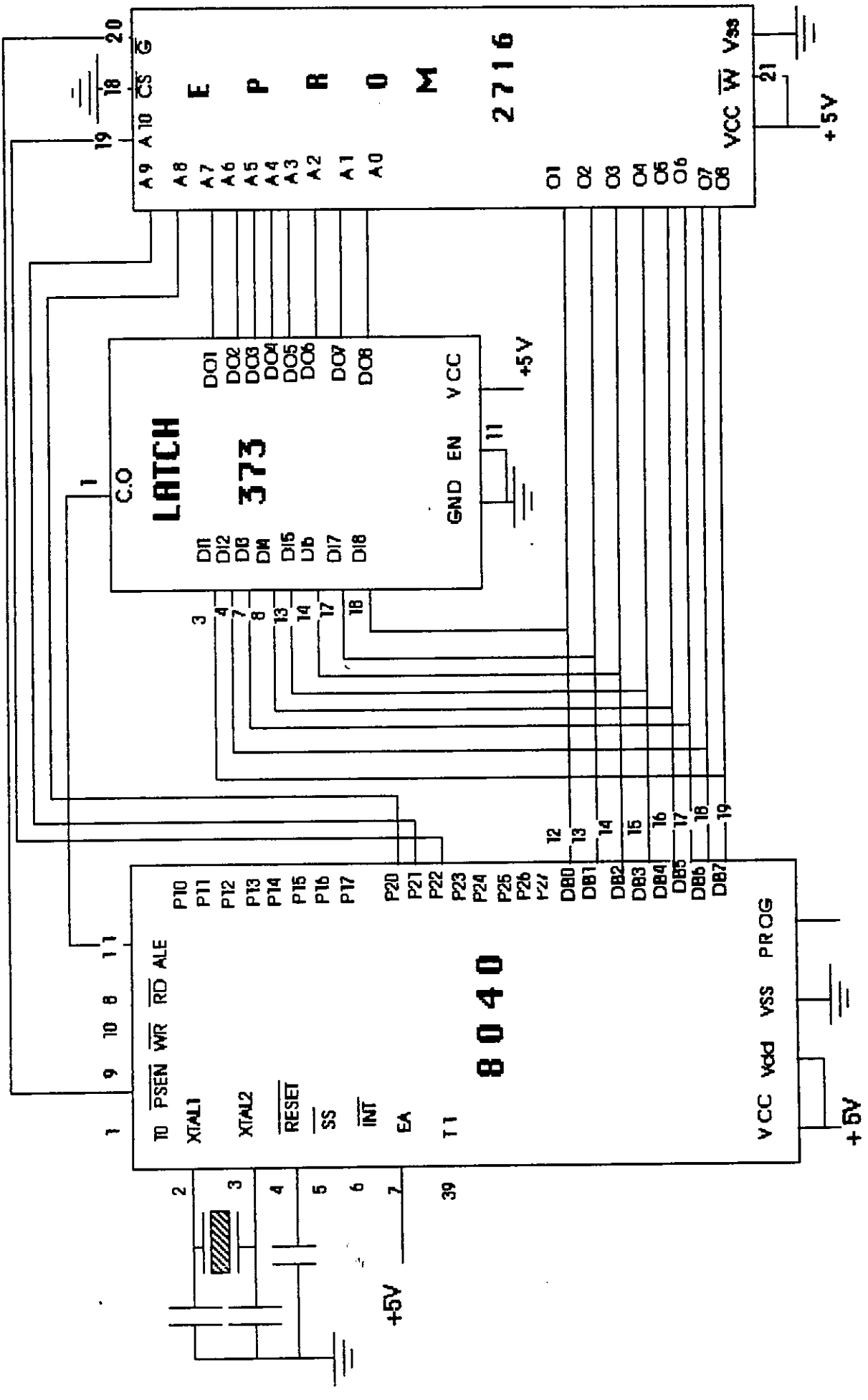


Fig.6. Connexion du 8040 à une EPROM 2716

II. Le clavier :

1. Analyse du clavier:

Le clavier qu'on va connecter au 8040, est un clavier non codé, ayant 20 touches organisées en matrice (4,5) .

Pour identifier une touche on utilise la méthode de balayage des lignes :

On sélectionne une ligne par les bits A0 - A7, puis on lit la donnée correspondante sur le bus DB0 - DB7 :

Si elle contient un ou plusieurs 0, cela signifie qu'il y a des touches enfoncées; les bits à 0 indiquent le numéro de la colonne.

Si aucune touche n'est enfoncée, le 8040 ne lit que des 1 logique sur le bus de données (les colonnes étant reliées au 5V par des résistances).

Les inverseurs ont trois rôles :

- Bloquer le retour du 5V sur le bus d'adresse .
- Amplifier le courant disponible sur le bus d'adresse .
- Inverser le signal des fils d'adresse pour le rendre compatible avec la logique utilisée dans l'analyse du clavier.

Les lignes du clavier sont contrôlées par les bits A0 - A3 du bus d'adresse. Les colonnes sont lues grâce au port d'entrée du buffeur sélectionné par le signal C ; [C = (A4*A5*A6*A7)*(P23=0)*($\overline{RD}$=0)] .

Les informations venant du clavier et les instructions référencées à la mémoire externe sont véhiculées sur le même bus (DB); on ne peut autoriser l'accès au bus par le clavier que si celui-ci est sélectionné. Ici, lorsque C=0.

Pour supprimer le rebond, on a choisi la solution "logiciel" qui consiste à utiliser un sous-programme de temporisation de durée d'environ 20 millisecondes (ms) .

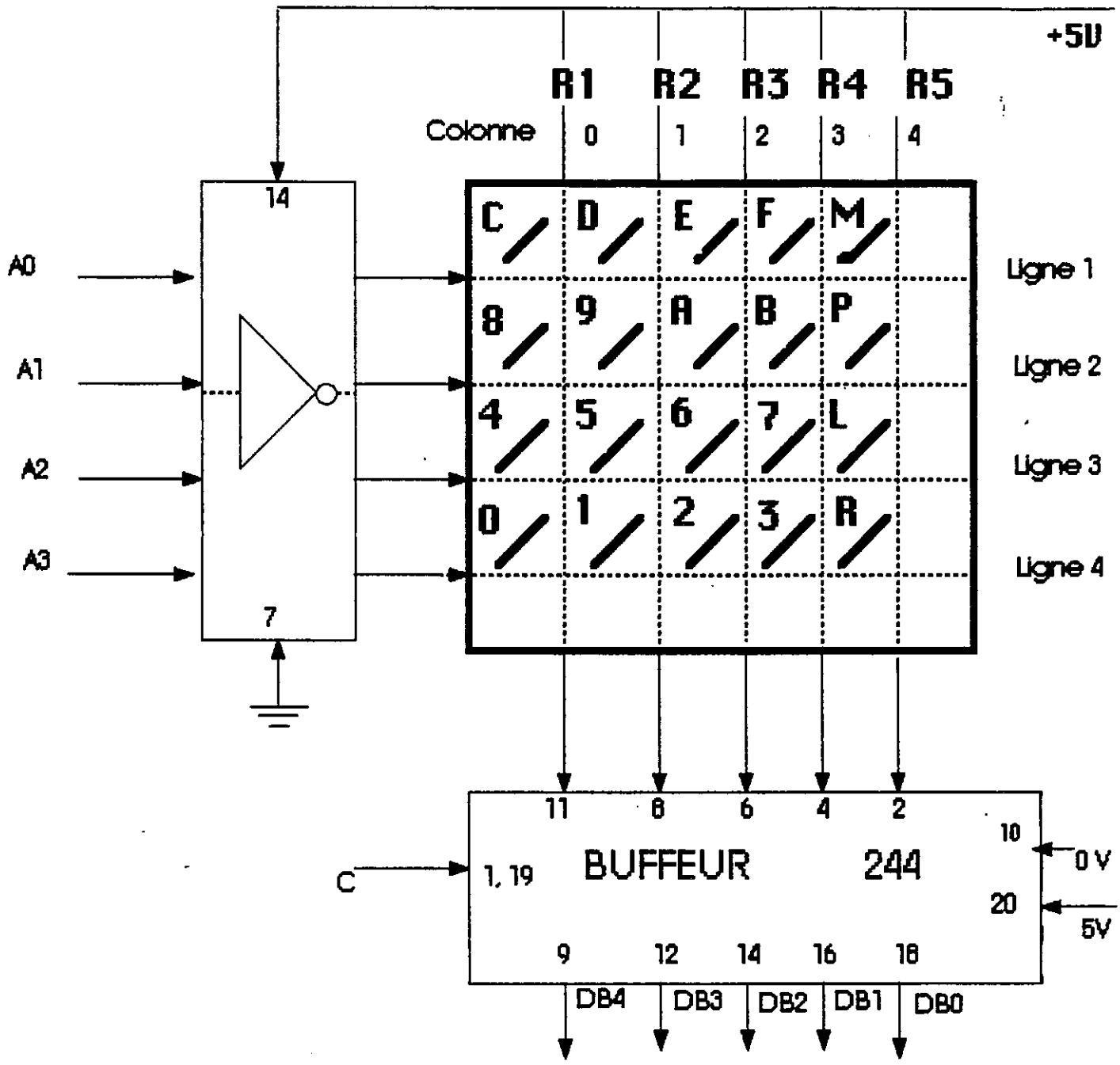


Fig.7. Schéma du clavier

2. Programme :

```
ch_touche   SEL   RB1              ; sélection des registres R'
            ANL   P2 , # FE H      ;      P20 := 0 pour
                                   ; sélectionner le clavier.
            CALL ch_ligne          ;      appel à   ch_ligne .

            CALL  attente          ;   appel  à     attente.
            CALL  ch_ligne         ;   appel  à     ch_ligne .

ch_colonne  MOV R1 , #05H          ; R1 := nombre de colonnes
col_suiv    RRC  A                 ;  décalage à droite de A
                                   ;          avec CY.
            JNC   col_trouver      ; si CY = 0 vers col_trouver.
            DJNZ R1, col_suiv      ; si R1≠ 1  vers  col_suiv .
col_trouver MOV A , R1
            DEC   A
            RL     A               ; Décalage   de A à gauche
            RL     A               ; pour que le n° de la colonne
                                   ;  appuyée soit dans les
                                   ; 3  bits de A " A2-A4 ".
            DEC    R2              ; R2 := n°  de  la  ligne .
            ADD  A , R2            ;  A0-A1 := n° de la  ligne
                                   ; A2-A4 := n° de la colonne .
            ADD  A , #tab_clavier  ;   A := adresse  de  la
                                   ;        touche appuyée .
            MOVP A , @A            ;  A := code  de  la
                                   ;   touche  appuyée
            MOV  R3 , A
relâcher    MOVX A , @R1
            ORL   A , #E0 H        ;        attente
            INC   A                ;      du relâchement
            JNZ    relâcher        ;   de la touche appuyée.
            CALL afficher
```

```
                SEL   RB0
                RETR

ch_ligne        MOV  R2 , # 05 H   ; R2 := nombre des lignes +1.
ligne_précéd    MOV  A , R2        ; A := numéro de la ligne à
                                   ;       sélectionner .
                ADD A , # tab_select ; A := adresse de la ligne
                                   ;       à séletionner .
                MOVP A , @ A       ; A := code de sélection
                                   ;       de la ligne .
                MOV  R1 , A
                MOVX A , @ R1      ;       lecture du clavier .
                ORL   A , # E0 H   ; masquer les bits inutilisés.
                INC   A            ; si A=FF alors pas de touche
                                   ;    appuyée.
                JNZ  lig-trouver   ; Si  A≠ 0  vers lig-trouver
                DJNZ R5 , lig-précéd ; Sinon Si  R2 ≠ 1  vers
                                   ;       lig-précéd.
                JMP   ch_ligne
lig_trouver     DEC A
                RETR


afficher        MOV   A , R3       ;          affichage
                OUTL P1 , A        ; sur le port 1 le code ASCII
                RETR               ;  de la touche appuyée .
attente         MOV  R0 , #50 H    ;    sous-programme
    T1          MOV  R1 , #52 H    ;          de
    T2          DJNZ R0 , T2       ;      temporisation
                DJNZ R1 , T1       ;    d'environ 20 ms .
                RETR

tab_select        XX               ;  table des codes de
                  F1 H             ;  sélection des lignes .
                  F2 H
                  F3 H
                  F4 H
                  F8 H
```

```
tab_clavier     0C H            ;   table des codes
                08 H            ;     du clavier.
                04 H
                00 H
                0D H
                09 H
                05 H
                01 H
                0E H
                0A H
                06 H
                02 H
                0F H
                0B H
                07 H
                03 H
                10 H    ; le code de la touche M .
                20 H    ; le code de la touche P .
                30 H    ; le code de la touche L .
                40 H    ; le code de la touche R .
```

CONCLUSION :

Pour faire une étude de contrôle de processus, nous avons choisi un modèle de processeur adapté : le 8040, qui présente de nombreuses qualités .

Dans une premiére partie nous avons voulu souligner ces qualités et exploiter les possiblités de les étendre. En particulier nous avons étudié la possibilité d'étendre la mémoire externe autant pour le programme que pour les données. Nous présentons deux applications d'interfaçage Microprocesseur-EPROM, Microprocesseur-E/S clavier .

Ce sont les composants qui vont nous permettre de construire une Unité Centrale de contrôle de processus, qui peut-être une Unité Centrale embarquée, et qui doit nous permettre également de développer un langage de programmation adapté au traitement que permet ce type de processeur.

BIBLIOGRAPHIE :

RODNAY ZAKS : DU COMPOSANT AU SYSTEME
Introduction aux microprocesseurs

JAMES W. COFFRON : APPLICATIONS DU Z80

EUROTECHNIQUE : - LE 8040
(MF/08 OCT. 81)

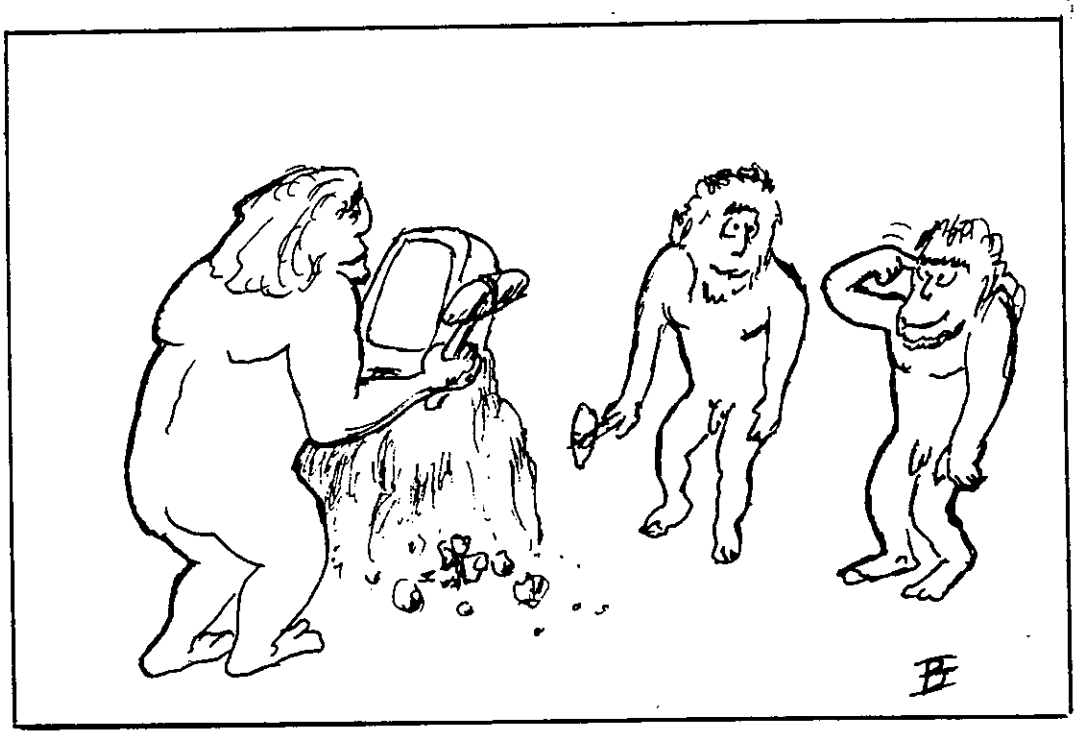
- ET 8048 - Series Microcomputer/
Microprocessor Family Jan 1982

DEVELOPMENT SAMPLE DATA : SINGLE-CHIP 8-BIT CMOS
MICROCONTROLLERS PCB0CXX Family
Novembre 1983

G. LELARGE et J. L. PLAGNOL : Applications des microprocesseur et de la micro-informatique
articles parus, dans les revues RADIO-PLANS et ELCOTRONIQUE APPLICATION Janvier 1977

B. HENRY : APPRENEZ L'ORDINATEUR
" Le Microprocesseur "
article parus, dans la revue MICRO-SYSTEME Avril 1986

VOUZZAVEDIBISAR



L'art est visionnaire