الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE
« HOUARI BOUMEDIENNE »

ECOLE NATIONALE POLYTECHNIQUE

Département d'Electronique

PROJET DE FIN D'ETUDES

»o«

INGENIORAT D'ETAT EN ELECTRONIQUE

SUJET

ENSEIGNEMENT DIRIGE DES MICRO - ORDINATEURS

AIDE A LA CONCEPTION DE SYSTEMES MICRO-ORDINATEURS

Proposé par : **Mr F. BRIKCI**
Docteur en Physique

Etudié par :
Mr BENKAHOUL Djamil

et :
Mr HASSANI Abdelouahab

Promotion Juin 1983

الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE
« HOUARI BOUMEDIENNE »

ECOLE NATIONALE POLYTECHNIQUE

Département d'Electronique

PROJET DE FIN D'ETUDES

»o«

INGENIORAT D'ETAT EN ELECTRONIQUE

SUJET

ENSEIGNEMENT DIRIGE DES MICRO - ORDINATEURS

AIDE A LA CONCEPTION DE SYSTEMES MICRO-ORDINATEURS

Proposé par : **Mr F. BRIKCI**
Docteur en Physique

Etudié par :
Mr BENKAHOUL Djamil

et :
Mr HASSANI Abdelouahab

Promotion Juin 1983

-- (II) E D I C A C E S --

A la mémoire de mon père

A Mes Frères et surtout " HADJ AISSA "

A toute ma FAMILLE

A tous mes AMIS

A. H A S S A N I

A la memoire de mes grands pères " RABAH " et " KHALED "

A mon Frère " ANTAR "

A toute ma FAMILLE

A tous mes AZIZIS

D. B E N K A H O U L

R E M E R C I E M E N T S

Nos vifs remerciements s'adressent à notre promoteur Mr F.BRIKCI pour les conseils qu'il n'a cèssè de nous prodiguer tout au long de notre etude,ainsi qué pour le suivi attentif qu'il nous a accorde.Nos remerciements s'en vont egalement à monsieur et madame BOUDRAA pour leurs esprits cooperatifs.

Que Mme DJEGHRI MALIKA,Melle NORI KARIMA et Mr MEKAKDA MUSTAPHA trouvent ici l'expression de notre sincere gratitude,qui sans leurs aimables concours ce fascicule n'aurait pu être mis au point à temps.

S O M M A I R E

--

-- INTRODUCTION --

D'une logique cablée difficile à mettre en oeuvre et très spcifique d'application, l'electronique cherchant la diversité et la souplesse dans ses réalisations est passée à la logique programmée en élaborant le " MICROPROCESSEUR ".

Cette revolution technique voit son champ d'application s'etendre de jour en jour. Sa necessité semblerait être dans un proche avenir celle d'une ampoule electrique.

L'importance des microprocesseurs de nos jours est telle qu'il est necessaire de mettre en place une structure d'enseignement dirigé en vue de faciliter leur utilisation.

Pour répondre à ce souci, il nous a été demandé de travailler sur un fascicule permettant de fournir aux étudiants un certain nombre d'informations générales ou specifiques en vue de la conception et de la réalisation de systèmes microordinateurs.

Ces travaux s'inscrivent en continuité des travaux élaborés par Mlles : N. Souag et F. Loumi concernant " l'enseignement dirigé des microprocesseurs, T.P. SUR LA CARTE T M S 990/189 " qui étaient chargées de répondre essentiellement à un souci de compréhension des microprocesseurs et des microordinateurs. Dans cette optique nous proposons le plan de travail ci-dessous :

La première partie etudie les problèmes relatifs à l'adressage et la connexion des mémoires et des interfaces à l'unité centrale. Ce sont les principaux problèmes auxquels est confronté le concepteur lors de la réalisation de son système minimum (microprocesseur + mémoires + interfaces)

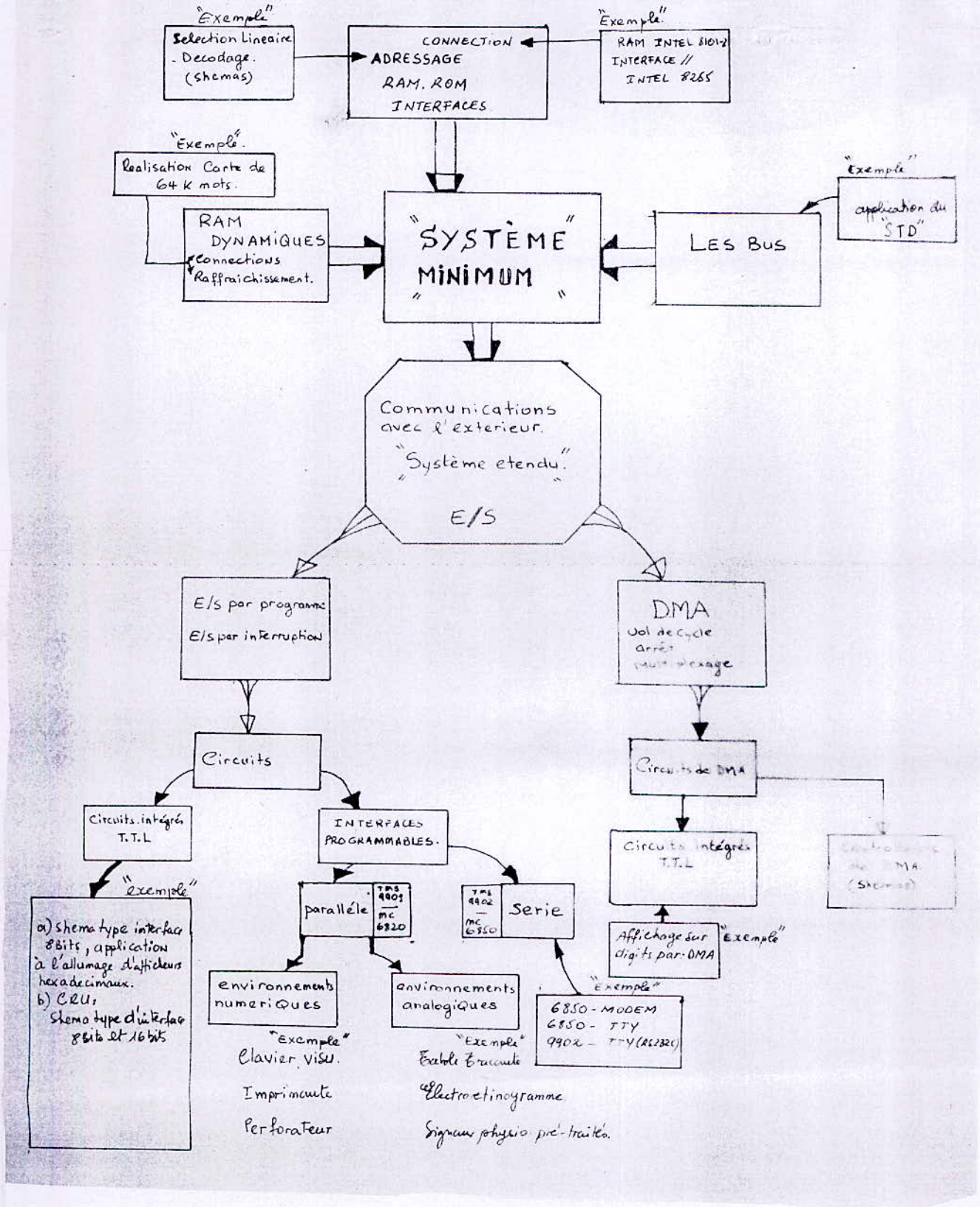
Les co[illegible]rti

Dans la seconde partie nous nous sommes penchés sur les problèmes d'entrées/sorties et d'interfaçage des microordinateurs. Notre étude se base sur des exemples precis et de leurs schémas electronique afin d'eviter l'exposé de principe théoriques vagues et schémas synoptiques arides d'informations.

La troisième partie traite des applications des microordinateurs au traitement des données analogiques. Trois exemples typiques y sont integrés afin de sensibiliser le lecteur dans ce domaine.

De façon générale, le lecteur trouvera dans ce document des informations relevant du domaine materiel ou " HARDWARE "; celles du logiciel ou " SOFTWARE "pouvant faire l'objet d'une troisième étude.

Pour guider son choix, il est fourni un organigramme précisant la demarche élaborée.


"Exemple"
Selection Lineaire
- Decodage.
(shemas)
CONNECTION
ADRESSAGE
RAM. ROM
INTERFACES
"Exemple"
RAM INTEL 8101-2
INTERFACE //
INTEL 8255
"Exemple"
Realisation Carte de
64 K mots.
RAM
DYNAMIQUES
connections
Raffraichissement.
"SYSTÈME"
MINIMUM
"
LES BUS
"Exemple"
application du
"STD"
Communications
avec l'exterieur.
"Système etendu"
E/S
E/S par programme
E/S par interruption
DMA
Vol de cycle
Arrêt
Circuits
Circuits de DMA
Circuits intégrés
T.T.L
INTERFACES
PROGRAMMABLES.
Circuits intégrés
T.T.L
"exemple"
a) Shema type interface
8Bits, application
à l'allumage d'afficheurs
hexadecimaux.
b) CRU;
Shema type d'interface
8bits et 16bits
paralléle
TMS 9901 mc 6820
TMS 9902 mc 6850
Serie
Affichage sur
digits par DMA
"Exemple"
environnements
numeriques
environnements
analogiques
"exemple"
6850 - MODEM
6850 - TTY
9902 - TTY (RS232C)
"exemple"
Clavier VISU.
Imprimante
Perforateur
"Exemple"
Electroencephalogramme
Signaux physio. pré-traités.

PREMIÈRE PARTIE

CHAPITRE I "L'ADRESSAGE ET LA CONNECTION DES MEMOIRES ET DES INTERFACES"

I) LES MEMOIRES : (Branchement et Connection).

Les mémoires servent à stocker les informations et les données manipulées ; elles se distinguent par la nature des programmes qu'elles peuvent contenir.

A) MEMOIRES MORTES :

Ce sont des mémoires non volatiles, elles conservent l'information une fois l'alimentation coupée ; ces memoires sont à lecture seulement.

a) LES ROM : (READ ONLY MEMORY-.

Lorsque de petits programmes sont souvent utilisés il est préférable de les stocker en ROM où ils resteront figés et seront donc considérés comme sous-programmes aux quels on fera appel si besoin est. Ces memoires sont livrées programmées par le constructeur, les programmes qu'elles contiennent ne pourront être ni modifiés, ni changés, ce qui n'est pas souple d'où en retrouve.

b) LES PROM : (PROM. programmable).

Ce sont des mémoires livrées vièrges pour être programmées par l'utilisateur (Notons que toute programmation est définitive).

c) EPROM : (ERASABLE PROM).

Ces mémoires peuvent être livrées programmées par le constructeur cependant, l'utilisateur pourra les effacer pour les programmer à nouveau.

d) LES REPROM : (REPROGRAMMABLE EPROM).

B) MEMOIRES VIVES : "LES RAM".

Ce sont des mémoires volatiles, elles perdent toute information dès que l'alimentation est coupée. Elles sont des mémoires à lecture et écriture. L'utilisateur du microordinateur y rangera les instructions et les données du programme qu'il voudra executer. On distingue 2 types de RAM :

- RAM STATIQUE : L'information contenue dans une RAM statique demeure memorisée jusqu'à ce que la cellule considérée soit selectée pour être effacée on ecrasée par une autre information.

- RAM DYNAMIQUE : Contrairement au precedent, ce type de RAM necessite un raffraichissement permanent pour garder telles qu'elles les informations qu'elles contiennent.

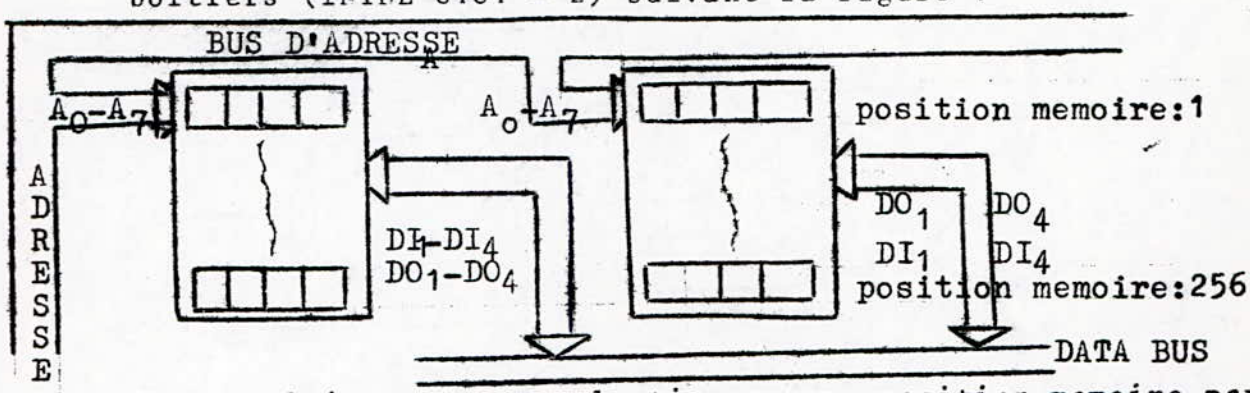
Les mémoires RAM et ROM se presentent sous forme de boitiers, sous différentes organisations dont nous citerons quelques unes ici bas :

Description	Organisation
Boitier ROM de 1024 Bits	256 x 4 Bits
Boitier PROM de 512 Bits	64 x 8 Bits
Boitier RAM Statique de 256 Bits	256 x 1 Bit
Boitier RAM Statique de 1024 Bits	256 x 4 Bits

DI1 - DI4 (DATA INPUTS) : lors d'une ecriture, ils serviront à acheminer l'information du bus de données vers la ligne sollicitée.

DO1 - DO4 : (DATA OUTPUTS) : Serviront quant à eux à acheminer vers le bus de données le contenu de ligne sollicitée si commande de lecture il y a.

Une position memoire n'est autre qu'une ligne de 8 bits (ou de 16 bits dans certains cas) ainsi, pour obtenir des positions memoires de 8 bits on doit associer en cascade deux boitiers (INTEL 8101 - 2) suivant la figure :

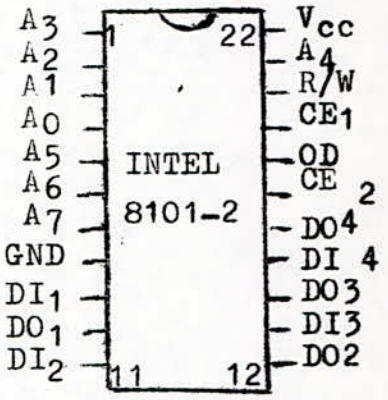


Il est clair que pour selectionner une position memoire parmi les 256 ainsi reunies on doit valider les 2 boitiers en même temps, de plus l'adresse de la position memoire sollicitée ainsi que la commande d'ecriture ou lecture doivent parvenir en même temps sur les entrées correspondantes des 2 boitiers.

Voyons maintenant comment se fait la validation, l'adressage et la connection de ces boitiers avec les bus du Système.

Prenons le cas de la RAM statique organisé en 256 x 4 Bits.

INTEL 8101 - 2 de 1024 Bits



A0 - A7 : entrées adresse du boitier

DI1 - DI4 : ENTREES données

DO1 - DO4 : Sorties données

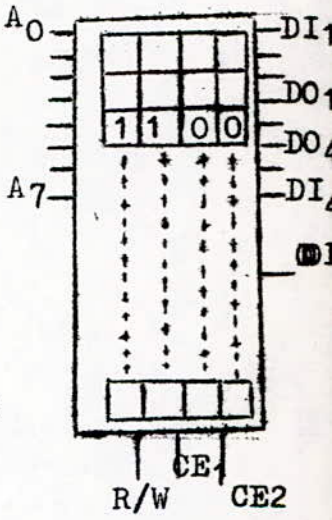
CE1, CE2 : Chip Enable

R/W : Commande de lect/Ecrit

OD : OUTPUT DISABLE.

Pour bien fixer les idées, on assimilera ce boitier à une matrice de 256 lignes de 4 bits chacune (256 lignes x 4 colonnes)

Pour lire de contenu de la 3ème ligne par exemple (voir figure ci-dessous) le microprocesseur doit en même temps, valider le boitier, specifier l'adresse de la ligne du boitier où il desire faire une lecture ensuite ordonner la lecture.



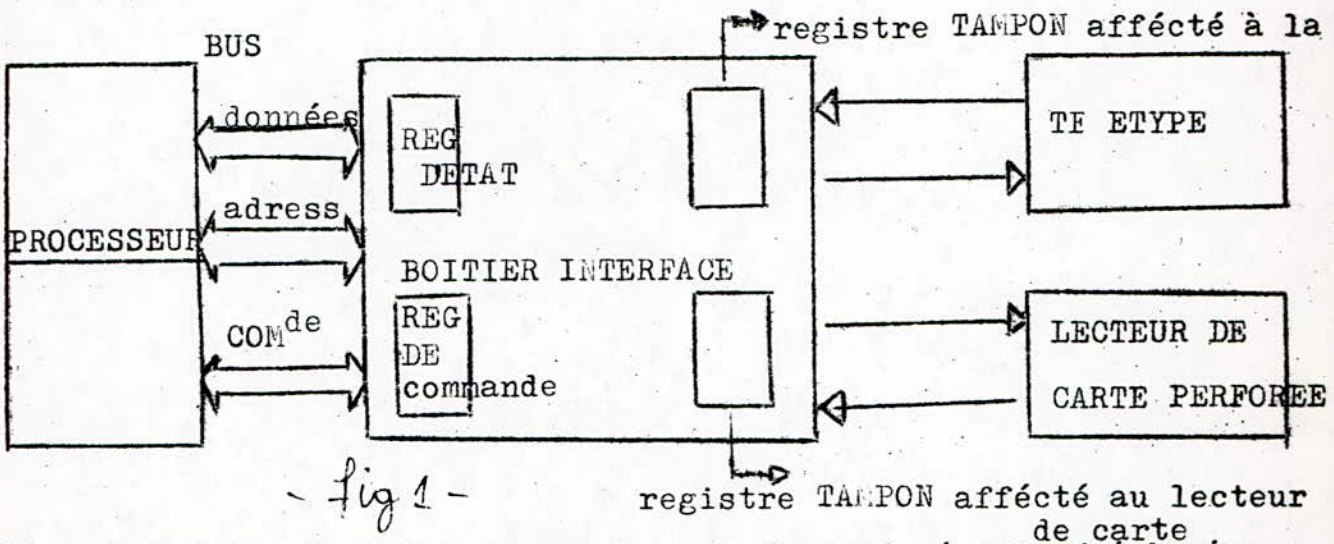
Le Boitier est validé si $\overline{CE}_1$ est à l'état bas et CE2 est à l'état haut ; donc l'une ou les deux de ces broches doit être reliée à un fil du bus d'adresse, si on désire que le boitier soit validé en permanence on mettra $\overline{CE}_1$ et CE2 à leur état actif par cablage.

A0 - A7 : Seront connectés au bus d'adresse du système (elles serviront donc à selecter une ligne quelconque parmi les 256 que comprend le boitier).

R/W sera connecté au un fil du bus de commande ordonnant la lecture ou l'ecriture mémoire.

II) LES INTERFACES.

Un microprocesseur ne peut pas commander directement les peripheriques (à quelques exceptions prés) Il est necessaire de réaliser une adaptation entre le processeur et le peripherique assurant ainsi une compatibilité entre les E/S du processeur et celles du peripherique .
Cette adaptation se fera par un circuit intégré qu'on appelle circuit "Interface".
Tout interface possède 1 ou plusieurs registres TAMPONS chacun affecté a un peripherique (voir fig ci-dessous). Il possède également un ou plusieurs registres de commande et un registre d'état.



- fig 1 -

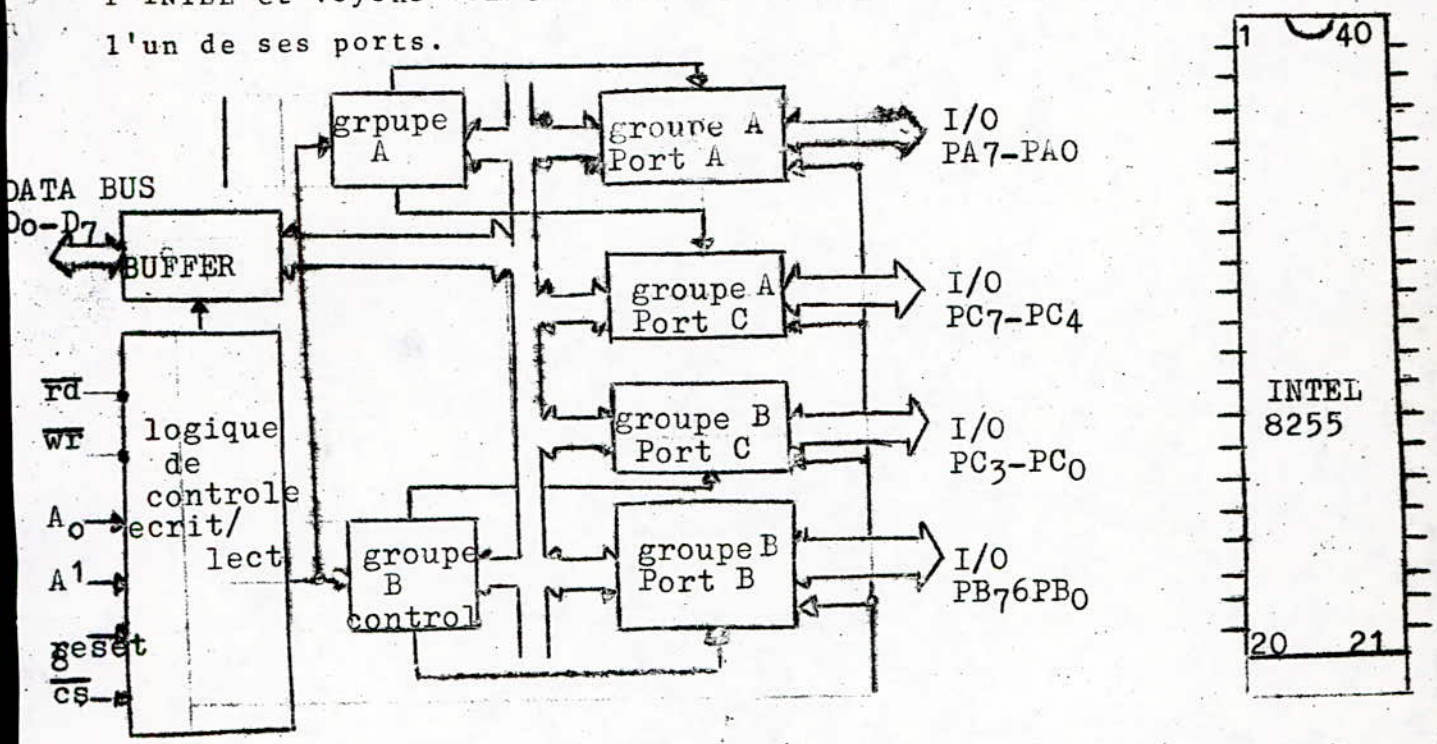
Un seul boitier interface peut adresser 1 ou plusieurs périphériques différents. Le 8255 de INTEL peut adresser 3 périphériques différents par Exemple : une teletype, une imprimante, et un lecteur de carte perforée pour adresser un périphérique donné on doit d'une part valider le boitier interface (qui s'occupe de son adressage et de son adaptation avec le processeur), et d'autre part sellecter un de ses ports c. à d le registre TAMPON affecté à ce péripherique (voir fig 1).
Le boitier interface sera validé par l'envoi d'un niveau logique 1(0) sur son entrée : CS ($\overline{CS}$). Le registre TAMPON sera selectionné en envoyant son adresse sur les entrées adresse de l'interface.
L'adressage de l'interface désiré et la selection de l'un de ses ports se fera bien entendu par le bus d'adresse cependant 2 possibilités s'offrent à nous :

.../...

-Périphériques adressés comme une memoire (E/S par instruction memoire) cette structure est offerte par tous les processeurs; le registre TAMPON affecté à chaque périphérique est adressé et commandé (par le processeur) comme une position mémoire.

-Périphériques adressés séparement : (E/S par instruction E/S) contrairement a l structure précedente, les registres TAMPONS ne sont plus adressés et commandés comme des positions mémoires, dans le cas les transferts d'E/S se feront pour des instructions spécifiques aux périphériques.

-Prenons l'exemple de l'interface programmable parallèle le 8255 de l'INTEL et voyons comment se fait sa validation et la selection de l'un de ses ports.



$\overline{CS}$: La validation du boitier se fait en mettant cette broche au niveau logique zero.

$\overline{RD}$: Quand elle est à l'état bas, cette broche signifie qu'il ya commande de lecture d'un des registres de l'interface.

$\overline{WR}$: A l'état bas elle signifie qu'il ya commande d'écriture dans l'un des registres.

RESET : A l'état haut cette broche remet à zero le contenu de tous les registres de l'interface (TAMPON, COMMANDE, ETAT).

.../...

A0.A1 : Servent à la selection d'un des registres de l'interface

D0.D7 : Connectés au bus de données.

PA0.PA7 : Bits d'E/S du Port A.

PB0-PB7 : Bits d'E/S du Port B.

PC0.PC7 : Bits d'E/S du Port C.

Validation, adressage, et connection de cet interface.

Le 8255 sera validé en mettant son entrée $\overline{CS}$ à l'état bas. Cette entrée sera alors reliée a un bit du bus d'adresse (elle peut être connectée à la sortie d'un decodeur d'adresse).

Pour selecter un de ses ports (A,B, ou C) il faut specifier l'adresse du port à travers les bits A0 et A1 qui seront donc connectés aux bits de plus faible poids du bus d'adresse.

L'Interface condideré étant parallèle, il recevra donc les données en parallèle à travers les bits D0-D7 via le bus de données.
Les broches RESET, $\overline{RD}$, $\overline{WR}$ sont des commandes de l'interface elles seront alors reliées au bus de commande du systhème.

(Notre but dans cette partie est de montrer la connection d'un interface avec les bus d'un systhème, sa connection avec un péripherique sera vue dans le chapitre traitant les E/S).

III - Les modes et les methodes d'adressage des memoires et des interfaces.

Les Facteurs essentiels qui influent sur l'adressage sont :

- le nombre de bits du bus d'adresse du microprocesseur utilisé
- la ou les structures d'E/S qu'il offre
- la capacité mémoire RAM et ROM qu'on désire réaliser
- le nombre et l'organisation des boitiers qu'on utilisera pour ce faire
- le nombre et la nature des interfaces qu'on désire connecter au systèmes

Avant d'aborder l'adressage proprement dit on doit commencer par les opérations suivantes :

- Fixer le nombre de positions memoires RAM et ROM qu'on désire réaliser, ensuite definir le nombre et l'organisation des boitiers utilisés.
- Fixer le nombre et définir la nature des interfaces qu'on désir connecter au système
- Choisir la structure d'E/S qu'on utilisera pour l'adressage (si le microprocesseur en offre plusieurs).
- Etablir le mode d'adressage qui nous convient (par selection linéaire* (RAM ou ROM) il faut d'une d part selectionner une position memoire parmi celles qu'il dispose et d'autre part, selectionner le bloc memoire lui même ; de même par la selection d'un peripherique, il faut d'une part selectionner un registre donné de l'interface (qui s'occupe de lui) et d'autre part selectionner l'interface lui-même.

L'adressage se fait en général suivant 2 Modes :

- Par selection lineaire
- Par decodage.

1) ADRESSAGE PAR SELECTION LINEAIRE /

Ce mode est utilisé lorsqu'on désire réaliser une capacité memoire relativement reduite et quelques peripheriques seulement (pour un bus d'adresse de 16 bits on arriverai juste à adresser 8000 mots memoires et quelques 2 à 3 peripheriques). Dans ce mode on affecte un bit exclusif du bus d'adresse à chaque peripherique.

* ou par decodage). pour selectionner une memoire →

Voyons maintenant lesprocédures de ce type d'adressage On suppose qu'on désire realiser . N position memoires RAM, M positions memoires ROM et connecter aux bus de système I interfaces serie à P entrées adresses et J interfaces parallèles à Q entrées adresses.

1.a) selection d'une position memoire RAM et ROM et d'un registre d'interface serie et //:

- affecter "n" bits du bus d'adresse (offrant 2^n combinaisons) pour la selection individuelle de 2^n = N positions memoires. RAM qu'on désire realiser (ces bits seront prisde poids faible).

- affecter "m" bits du bus d'adresse (offrant 2^m combinaisons) pour la selection individuelle de 2^m = M positions memoires ROM qu'on désire realiser (ces bits seront aussi pris de poids faible).

- affecter P bits du bus d'adresse pour la selection d'un des registres des interfaces serie (les bits seront pris de poids faible)

- affecter Q bits du bus d'adresse pour la selection d'un registres des interfaces paralele (les bits seront pris de poids faible).

A' ce stade une position memoire RAM ou ROM ainsi qu'un registre d'interface serie ou parallèle peuvent avoir la même adresse, il est maintenant necessaire d'en faire la distinction c'est ce qu'on appelera "distinction par fonction"

1) b- selection d'un bloc memoire RAM ou ROM et d'un interface serie ou parallèle parmi ceux que comprend le système :

- distinction entre RAM et ROM /

- affecter un bit du bus d'adresse (de poids le plus fort) pour valider le bloc RAM

- affecter ce même bit mais complementé pour valider le bloc ROM (par exemple si A15 valide la RAM, $\overline{A15}$ doit valider la ROM).

- Selection des Interfaces :

Pour selecter un interface serie donné parmi ceux qu'on dispose on valide chacun d'eux par 1 bit du bus d'adresse. Ces bits seront pris dans un ordre croissant à partir du dernier bit affecté à la selection d'un des registre des interfaces parallèles (parceque ceux ci possèdent généralement plus d'entrée adresse que les interfa ces serie) : de même pour la selection d'un interface parallèle parmi les J dont on dispose, on affecte 1 bit du bus d'adresse pour valider chacun d'eux ces bits seront pris dans un ordre croissant à partir du dernier bit du bus d'adresse affecté à la validation du dernier interface serie.

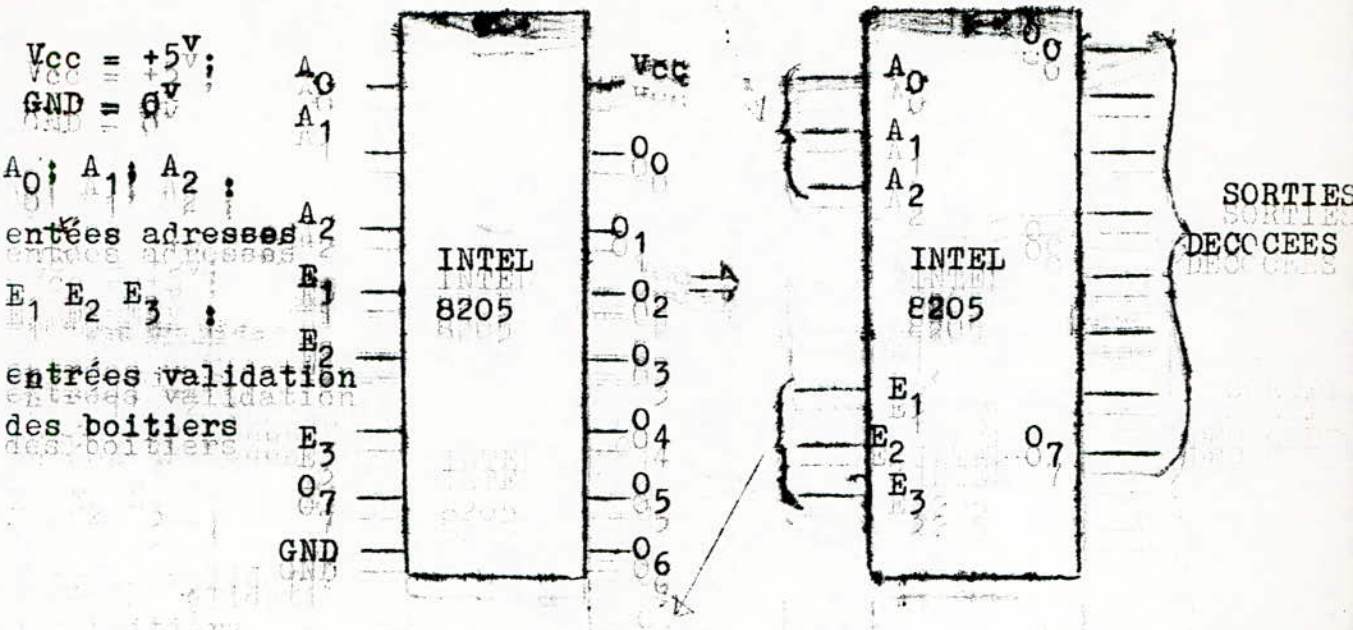
Ainsi on a fait la selection d'un interface serie ou parallèle parmi ceux qu'on dispose et en même temps, la distinction entre les 2 types d'interfaces (serie parallèle).
Il nous reste maintenant pour terminer l'adressage veiller à faire la distinction entre les Blocs memoires et les interfaces.
Ceci se fera en affectant un autre bit du bus d'adresse (on le prendra de poids fort) pour la validation du Bloc RAM ou ROM, et valider les interfaces serie et parallèle par ce même bit mais complémenté. Notons que si on utilise la structure d'E/S par instruction E/S cette distiction est completement inutile (ce qui nous librera un bit du bus d'adresse augmentant ainsi nos capacités d'adressage), car même si une position memoire RAM ou ROM et un periphérique aient la même adresse un, ou une seule d'entre eux seulement recevra un ordre de lecture ou d'ecriture.

2) ADRESSAGE PAR DECODAGE :

Lorsqu'un système neçéssite une grande capacité mémoire et un grand nombre d'interfaçes, leur adressage devient impossible en utilisant la selection linéaire . On fait alors appel à des décodeurs .

Ce mode aux dépens de sa complexité permet une occupation mémoire trés grande (relativement au 1er mode).Il consiste à décoder (par un décodeur) les bits d'adresse pour selectionner un module mémoire ou un interface (par module mémoire on entend un bloc mémoire RAM ou ROM qu'on validera par une sortie décodée du décodeur ;le lecteur comprendra cette notion en se referant à l'éxercice proposé en fin de cette partie) . Ce mode permet l'occupation de la plus grande des mots mémoire théoriquement disponibles avec notre bus d'adresse .

Commençons d'abord par définir les décodeurs: Les décodeurs se présentent sous forme de boitiers à 1,2,ou 3 entie s donnant respectivement 2,4,ou 8 sorties . LES entrées adresses du décodeur doivent être reliées à des fils du bus d'adresse, elles permettent de sélectionner un module RAM, ROM ou un interface parmi les différents modules RAM ou ROM et interfaces dont on dispose . Les entrées validation du boîtier peuvent être reliées au bus d'adresse ,l'une ou plusieurs d'entres elles peuvent aussi être mises à leur état valide par cablage .Les sorties décodées serviront à sélecter différent modules mémoires (RAMou ROM) et des interfaces (voir figure ci-dessous) .



- Les différentes procédures de l'adressage par décodage :

On supposera comme pour le 1er mode qu'on désire réaliser :

N positions mémoires RAM , M positions mémoires ROM , et connecter au système I interfaces séries à P entrées adresse et J interfaces parallèles à Q entrées adresses .

Il faut d'abord commencer par regrouper les N(M) positions mémoires RAM(ROM) qu'on a en modules de façon à ce que chaque sortie du décodeur en validera Un (1) . Si on utilise un décodeur mémoire à 8 sorties (décodeur 1 parmi 8), on doit diviser l'ensemble de nos positions mémoires RAM et ROM en 8 modules pour que dans notre cas chaque sortie décodée validera un module mémoire de $\frac{N+M}{8}$ positions mémoires ;

Pour la selection d'un interface parmi les interfaces séries et parallèles qu'on a ,on utilisera un décodeur périphérique, chaque sortie du décodeur périphérique validera un interface .

Pour selectionner une position mémoire dans un module

et d'un registre dans un interface, on operera de la même façon que pour la sélection linéaire .

Si on utilise la structure d'E/S par instruction mémoire, on doit veiller à faire la distinction entre le bloc mémoire et les interfaces ; Ceci peut se faire par exemple en validant le décodeur périphérique par la dernière sortie décodée du décodeur mémoire (Ceci revient dans notre cas à diviser l'ensemble des positions mémoires qu'on a en 7 modules laissant ainsi la dernière sortie du décodeur mémoire libre pour valider le décodeur périphérique) . Nous signalons aussi que diverses autres solutions peuvent être utilisées .

EXEMPLE: (SELECTION LINEAIRE)

Le microprocesseur 8080 -A de Intel posséde un bus d'adresse de 16 bits ,un bus de données de 8bits et reconnait les 2 structures d'E/S. On veut connecter aux bus de ce processeur:

-2 boitiers ROM de 1024 bits organisés en 256 bits 4 bits chacun

- 16 boitiers RAM de 256 bits organisés en 256 x 1 bit chacun.

- 2 interfaces paralléles avec 2 entrées adresse chacun (type 8255 de Intel)

- 1 interface serie avec 1 entrée adresse (type 8251 de Intel)

SOLUTION/

a) Selection d'une position memoire RAM :

On dispose de 16 boitiers de 256 x 1 bit;pour former des positions memoires de 8 bits, les boitiers doivent être arrangés en 2 lignes de 8 boitiers chacune formant ainsi 512 positions memoires.

Pour selectionner une position memoire parmi ces 512, il nous faut 9 fils du bus d'adresse ($2^n=512 \Rightarrow n=9$);onles prendra de poids faibles soit: $A_0, A_1, A_2, A_3, A_4, A_5, A_6, A_7, A_8$,

b) Selection d'une position memoire ROM:

On posséde 2 boitiers ROM de 256 4 bits ,on les arrangera en cascade de facon à obtenir 256 positions ROM .

Pour selectionner une de ces positions il nous faudra 8 fils du bus d'adresse ($2^n=256 \Rightarrow n=8$) ; on les prendra de poids faibles soit $A_0, A_1, A_2, A_3, A_4, A_5, A_6, A_7$..

c) Selection d'un registre d'un interface paralléle:

Les interfaces parallé les ont 2 entrées adresse ,onleur connecte les bits A_0 et A_1 du bus d'adresse.

d) Selection d'un registre de l'interface serie:

L'interface serie posséde une entrée adresse,le bit A_0 suffira pour la selection .

e) Distinction entre les interfaces:

On validera :

-l'interface serie par A_2.

-l'interface paralléle n°1 par A_3.

-l'interface paralléle n°2 par A_4.

f) Distintion entre RAM et ROM :

On validera :

La ROM par $\overline{A_{15}}$ (car début d'implantation des ROM à l'adresse 0000)

La RAM par A_{15}

Pour la distinction entre memoire et interface : on validera:

- Les ROM par $\overline{A_{14}}$
- Les interfaces par $\overline{A_{15}}$ et A_{14}

(Pour la structure d'E/S par instruction d'E/S cette dernière distinction est inutile).

Recapitulons :

A_{15} valide les RAM

$\overline{A_{15}}$, $\overline{A_{14}}$ valident les ROM

$\overline{A_{15}} \cdot \overline{A_{14}} \cdot A_2$ valident l'interface serie

$\overline{A_{15}} \cdot \overline{A_{14}} \cdot A_3$ valident l'interface parallèle n°1

$\overline{A_{15}} \cdot \overline{A_{14}} \cdot A_4$ valident l'interface parallèle n°2.

Pour faciliter la mise au point d'un schéma de branchement il sera toujours très utile de dresser un tableau d'adressage comme le suivant :

	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
RAM	1							X	X	X	X	X	X	X	X	X
ROM	0	0							X	X	X	X	X	X	X	X
INT serie	0	1												1		X
INT parallèle 1	0	1											1		X	X
INT parallèle 2	0	1										1			X	X

Les case non remplies sont insignificatives. X : peut-être 0 ou 1. Par exemple pour selectionner un registre de l'interface para llèle n° 1 on doit envoyer sa combinaison à travers A_0 et A_1 et mettre : A_3 et A_{14} à "1" et A_{15} à 0
(Voir schéma de branchement)

Pour cet exercice l'utilsation de l'adressage par decodage est vraiment deconseillér(elle ne ferai que compliquer les choses davantage). Nous verrons par contre dans l'exercice après , un exemple où l'adressage par decodage s'impose.

Quelques remarques concernant le schéma de branchement :

-- Connexion des RAM :

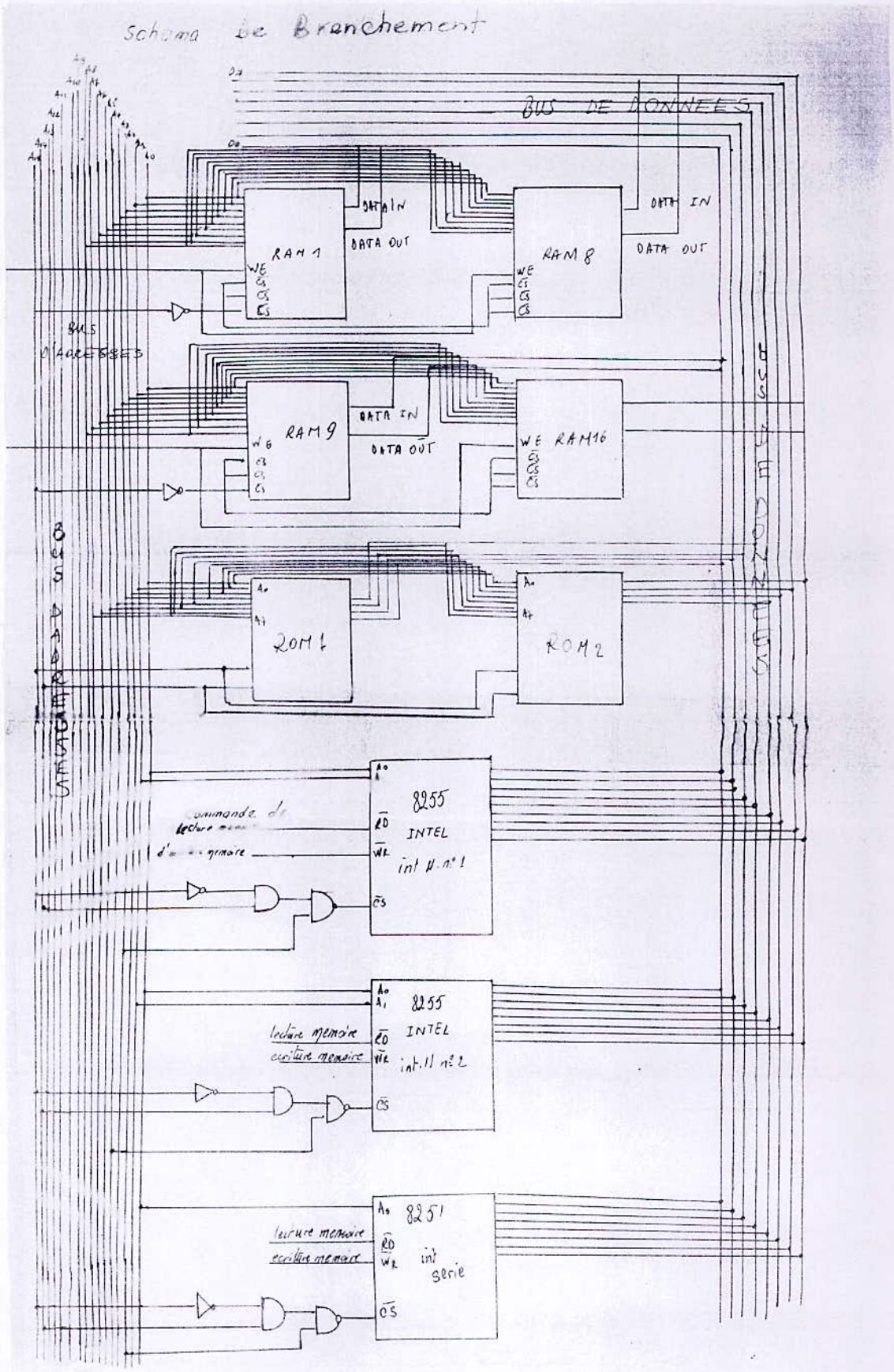
En presence d'une instruction de lecture le WE est mis à l'etat haut par le bus de commande, s'il s'agit d'un ordre d'ecriture ce bit est mis à l'etat bas (le memory Enable est toujours à l'etat bas).

FUNCTION	INPUTS ME	INPUTS WE	OUTPUT
Write	0	0	1
Read	0	1	Stored DATA
INHISIT	1	X	1

A_{15} peut être relié à un seul CS et les duex autres seront mis à l'etat bas par cablage.

-- Connexion des interfaces :

On a affecté 3 bits du bus d'adresse pour selectionner un interface. Seulement les interfaces choisis ne comporte qu'un bit CS (de validation). On a alors fait appel à des circuits logiques combinés et connectés aux bits du bus d'adresse qui sont affectés à leursvalidation.


Schema de Branchement
BUS DE DONNEES
RAM 1
RAM 8
RAM 9
RAM 16
ROM 1
ROM 2
DATA IN
DATA OUT
WE
CS
BUS D'ADRESSES
BUS DE CONTROLE
8255
INTEL
int II n° 1
int II n° 2
lecture memoire
ecriture memoire
8251
int serie

Exemple : ADRESSAGE PAR DECODAGE.

On est toujours en présence du microprocesseur 8080A de INTEL

Bus de données : 8 fils.

Bus d'adresses : 16 fils,

On desire connecter au bus de ce processeur

48 Kilo RAM

8 Kilo ROM

2 interfaces parallèles (INTEL 8255).

1 interface serie (INTEL 8251)

On utilisera des boitiérs RAM du type : TMS 4016 JL de 2Kx8 bits chacun, et des boitiérs ROM du type : INTEL 81316-A de 2Kx8 bits chacun. On aurait donc besoin de 24 boitiers RAM et 4 boitiers ROM.

Il est ici hors de question d'utiliser l'adressage par selection lineaire ainsi pour selectionner 1 position memoire RAM parmi les 48 Kilo qu'on desire realiser il nous faudra (2^n = 49152 donne n=15) 16 fils du bus d'adresse. On est donc dans l'obligation de proceder par decodage.

On reunira alors 4 boitiers RAM pour former un module, on aura donc 6 modules RAM. Les 4 boitiers ROM réunis formeront le moduleRO

1 module RAM= 4 boitiers RAM = 8 kilo = 8192 positions memoire RAM.

Pour selectionner une position memoire RAM parmi les 8192 du module Il nous faudra affecter (2^n = 8192 $\Rightarrow$ n = 13) 13 fils du bus d'adresse, on les prend de poids faible soit :

A_0, A_1, A_2, A_3, A_4, A_5, A_6, A_7, A_8, A_9, A_{10}, A_{11}, A_{12}.

On utilisera un decodeur mémoire " un parmi huit " chaque sortie de ce dernier validera un module mémoire, les trois entrées de c celui-ci seront reliées respectivement aux bits A_{13} - A_{14} - A_{15} du bus d'adresses (ce sont des bits de poids fort qui sont libres)

Si on utilise la stucture d'entrée / sortie par instruction mémoire, l'unndes deux décodeurs seulement doit être validé à un moment donné. Nous avons choisi de valider le décodeur péripherique par une sortie du décodeur mémoire (plusieurs solutions sont possibles).

Par contre, si on utilise la structure E/S par instruction E/S. les décodeurs mémoire et péripherique doivent être validés en permanence, on veillera donc à le faire par cablage.

On remarque aussi pour cette structure la possibilité d'adresser 8.K mémoires RAM ou RAM supplémentaires, étendant ainsi la capacité mémoire à 64.K mémoires. Cela se fera en connectant la sortie S_7 du décodeur mémoire (non connecté dans notre exemple) à un autre module mémoire.

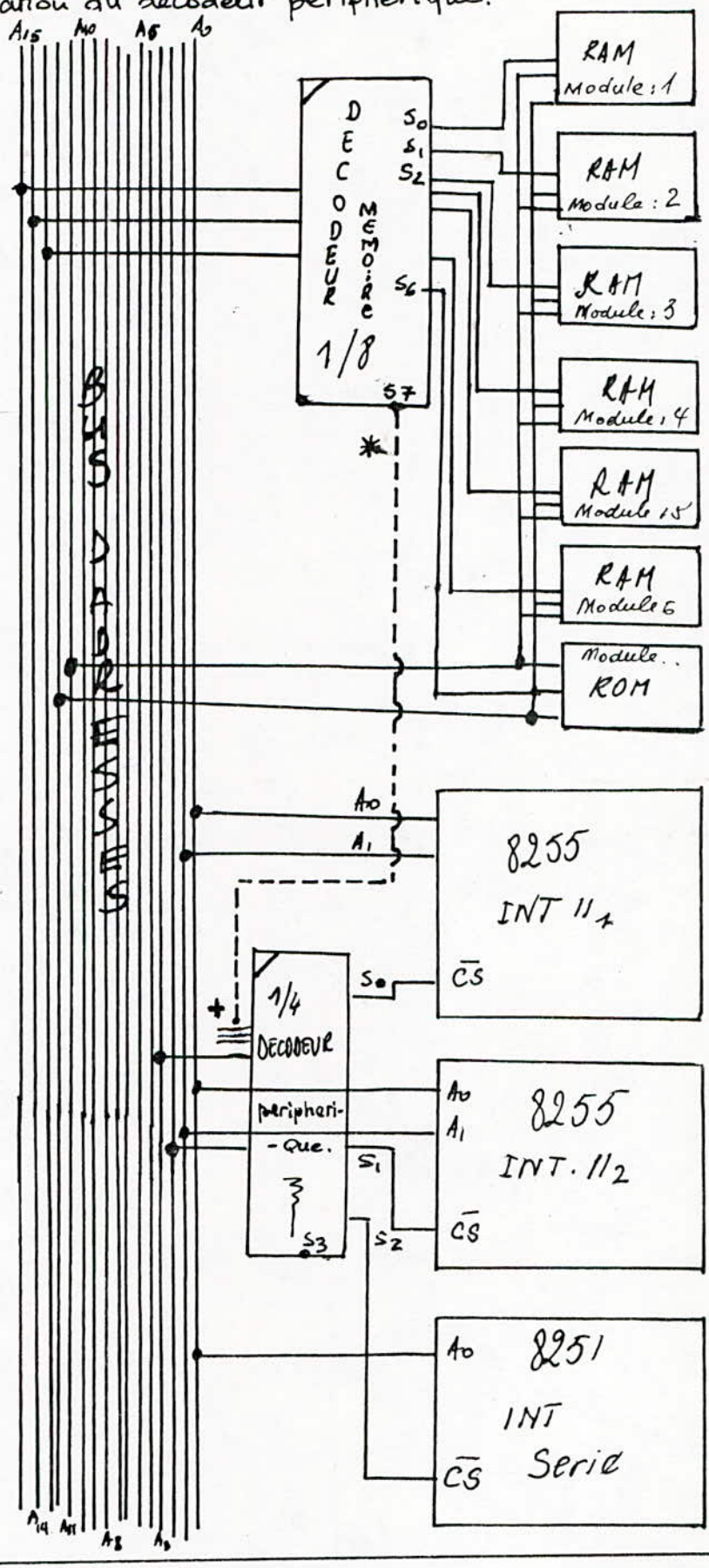
Table de vérité du décodeur memoire " 1 parmi 8 "

ADRESSE			VALIDATION			SORTIES DECODEES							
A_0	A_1	A_2	$\overline{CS}_1$	$\overline{CS}_2$	CS_3	0	1	2	3	4	5	6	7
0	0	0	0	0	1	0	1	1	1	1	1	1	1
1	0	0	0	0	1	1	0	1	1	1	1	1	1
0	1	0	0	0	1	1	1	0	1	1	1	1	1
1	1	0	0	0	1	1	1	1	0	1	1	1	1
0	0	1	0	0	1	1	1	1	1	0	1	1	1
1	0	1	0	0	1	1	1	1	1	1	0	1	1
0	1	1	0	0	1	1	1	1	1	1	1	0	1
1	1	1	0	0	1	1	1	1	1	1	1	1	0
X	X	X	0	0	0	1	1	1	1	1	1	1	1
X	X	X	1	0	0	1	1	1	1	1	1	1	1
X	X	X	0	1	0	1	1	1	1	1	1	1	1
X	X	X	1	1	0	1	1	1	1	1	1	1	1
X	X	X	1	0	1	1	1	1	1	1	1	1	1
X	X	X	0	1	1	1	1	1	1	1	1	1	1
X	X	X	1	1	1	1	1	1	1	1	1	1	1

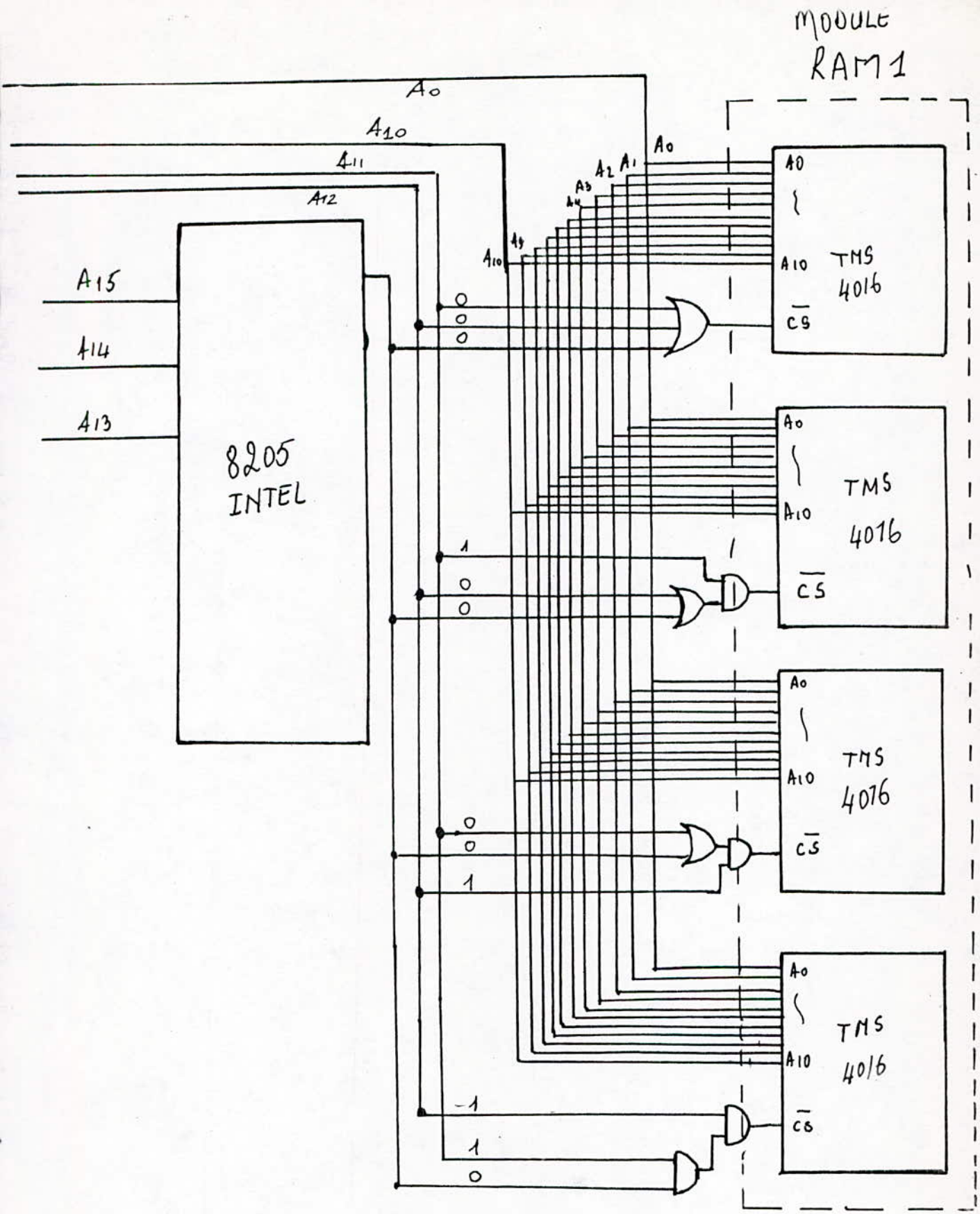
X peut être 0 ou 1

schema de branchement :

* En structure d'E/S par Instruction memoire
la sortie S_7 du decodeur memoire sert d'entrée validation du decodeur peripherique.



†: les entrées validation du decodeur peripherique doivent être mises à leurs etats actifs par calculage pour une structure d'E/S par instruction E/S



Detail de branchement du module RAM 1

IV) LES BUS DU SYSTEME

Les bus sont les voies qui relient les differents organes d'un systeme microordinateur. Ces organessont selon les cas,descartes des circuits,des peripheriques etc....

Pratiquement,chaque famille de microprocesseur propose son propre bus, incompatible avec ceux des autres familles; Certains construc--teurs se sont alors penchés sur leurs normalisations en elaborant. Le Multibus, Le STD, L'unibus, Le S100, Le versabus, Le Q-bus, Le Z-bus,etc .....

Ces bus bien que normalisés sont incompatibles entre eux, de ce fait des dispositifs et des circuits conçus pour être reliés à un bus d'un type donné ne peuvent pas être connectés à un autre.

La normalisation d'un bus porte sur :

- Son organisation materielle (nombre de fils;de voies qu'il comprend)
- Son fonctionnement (limite de vitesse de transmission, possibilité de multitraitement, sens de circulation des informations, etc...)
- Ses procedures d'echanges.

Le S T D (mis au point par PROLOG.)est le bus qui jouit d'une normalisation effective multifamille, il est destiné à plusieurs microprocesseurs 8 bits : 8080, 8088, 6800, 6809; Z80,...

Ce bus se compose de 56 fils répartis en :

- 6 pour les alimentations
- 8 pour les données
- 16 pour les adresses
- 22 pour les commandes
- 4 pour les alimentations de puissance.

On verra sur le schéma de la page suivante une application type du bus S T D .

Le multibus (élaboré par INTEL) présente quant à lui 86 fils

répartis en :

- 12 pour les alimentations
- 12 pour les commandes
- 10 pour le contrôle et les adresses
- 8 pour les interruptions
- 16 pour les adresses
- 16 pour les données

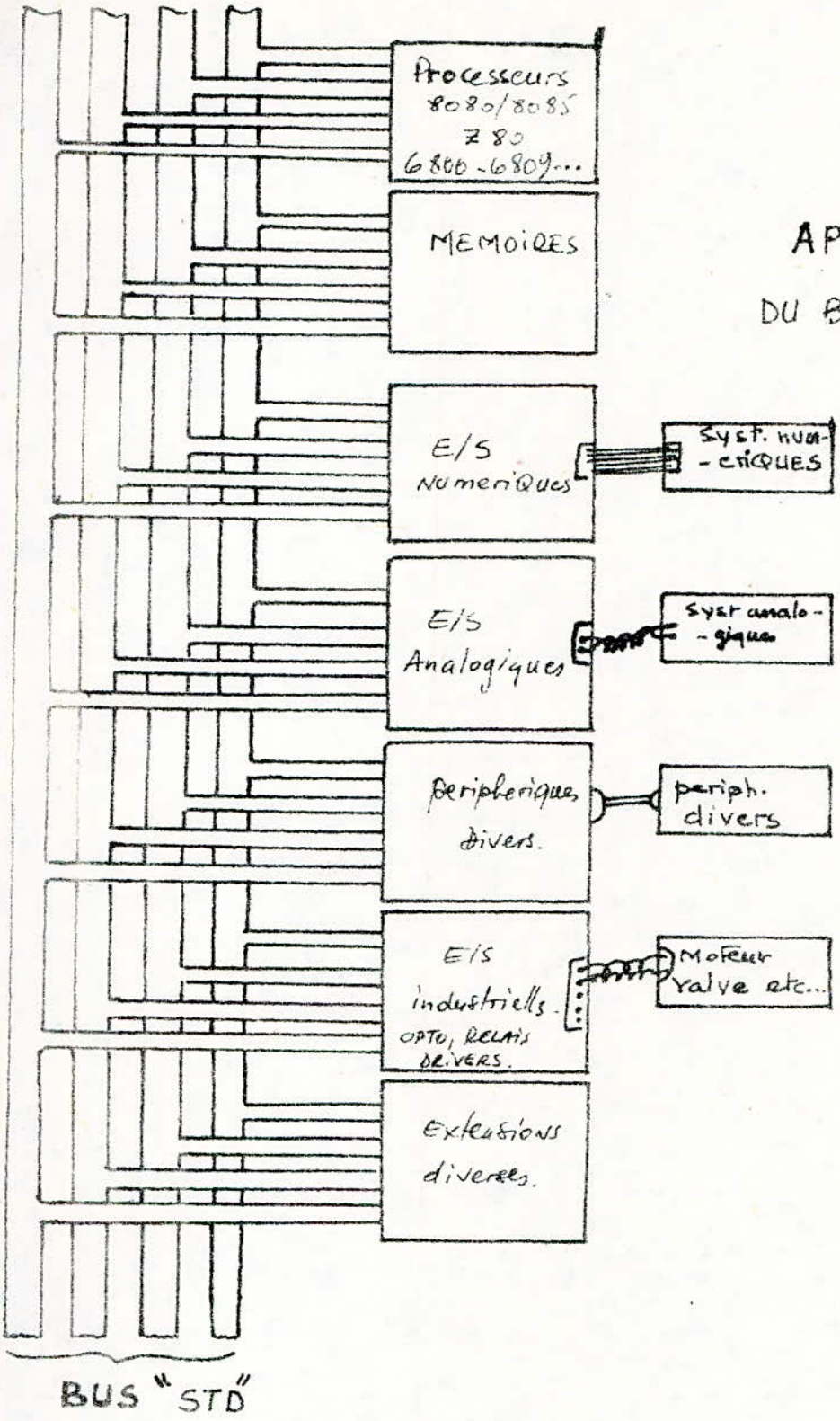
Il est destiné aux microprocesseurs INTEL 8 et 6 bits (8080 8085, 8086, ...) il peut adresser sur 20 bits 1 mega-octets en mémoire et peut transferer des informations jusqu'à 5 MHz et supporte des systèmes en multitraitement .

On donne dans le tableau ci-dessous quelques bus (pour MP) normalisés.

B U S	SOCIÊTE	micro-processeur	adressage sur	Notes
Versabus ..	Motorola	68000	32 bits	pour µP 8, 16 et 32 bits.
Multibus ..	INTEL	8080, 8085 8086...	20 BITS	pour µP 8 et 16 bits
Qbus	DEC	LSI-11	18 bits	pour µP 16bits
Z-bus	ZILOG	Z8000	32 bits	pour µP8,16 et 32 Bits
STD.....	PROLOG	8080, 8085 6800, Z80	16 bits	pour µP 8 bits
STD-Z80	MOSTEK	Z80	16 Bits	pour le Z80
S-100....	MITS	8080	16 Bits	pour l'information"personnelle "

Nous n'avons donné ici que de petits aperçus concernant les bus d'un système. Le lecteur devra consulter des ouvrages plus spécialisés pour recueillir de plus amples informations.

APPLICATION DU BUS "STANDARD" (STD)



CHAPITRE II

LES RAM DYNAMIQUES ET LEUR RAFRAICHISSEMENT

UTILISATION DES RAM DYNAMIQUES :

Dans les systèmes necessitant une occupation memoire importante (48 K octets par exemple) il est préférable d'utiliser des circuits à grande capacité tels que les RAM dynamiques.

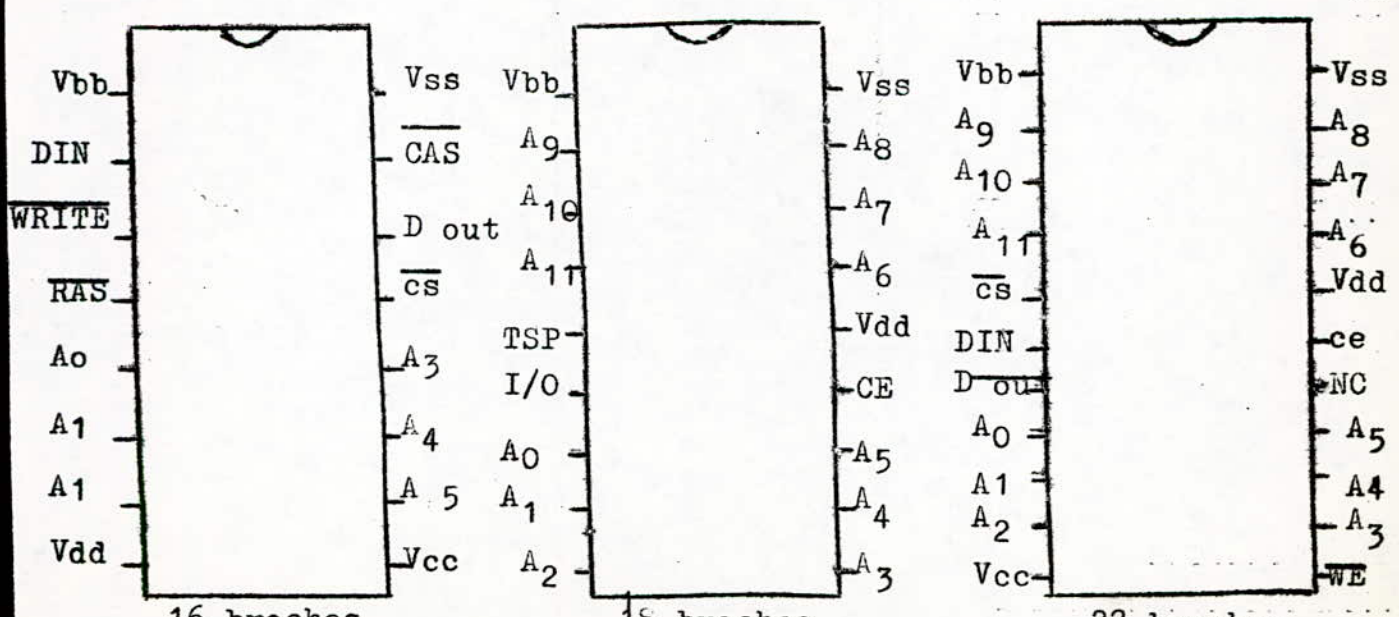
Le principal problème posé au concepteur de circuits par ces memoires est que ces dernieres necessitent un rafraîchissement matérialisé par une lecture complète de toute la mémoire. Pour rafraîchir toute la mémoire, il faudra adresser et rafraîchir successivement chacune des lignes. Il en résulte que les entrées de la mémoire recevront :

- soit des fils adresse desélection de lignes en provenance du bus d'adresse
- soit des fils de selection de lignes en provenance d'un circuit spécial de rafraîchissement ayant sa propre horloge d'où la necessité d'un multiplexage des fils d'adresse. Le multiplexeur étant commandé par le signal de rafraîchissement.

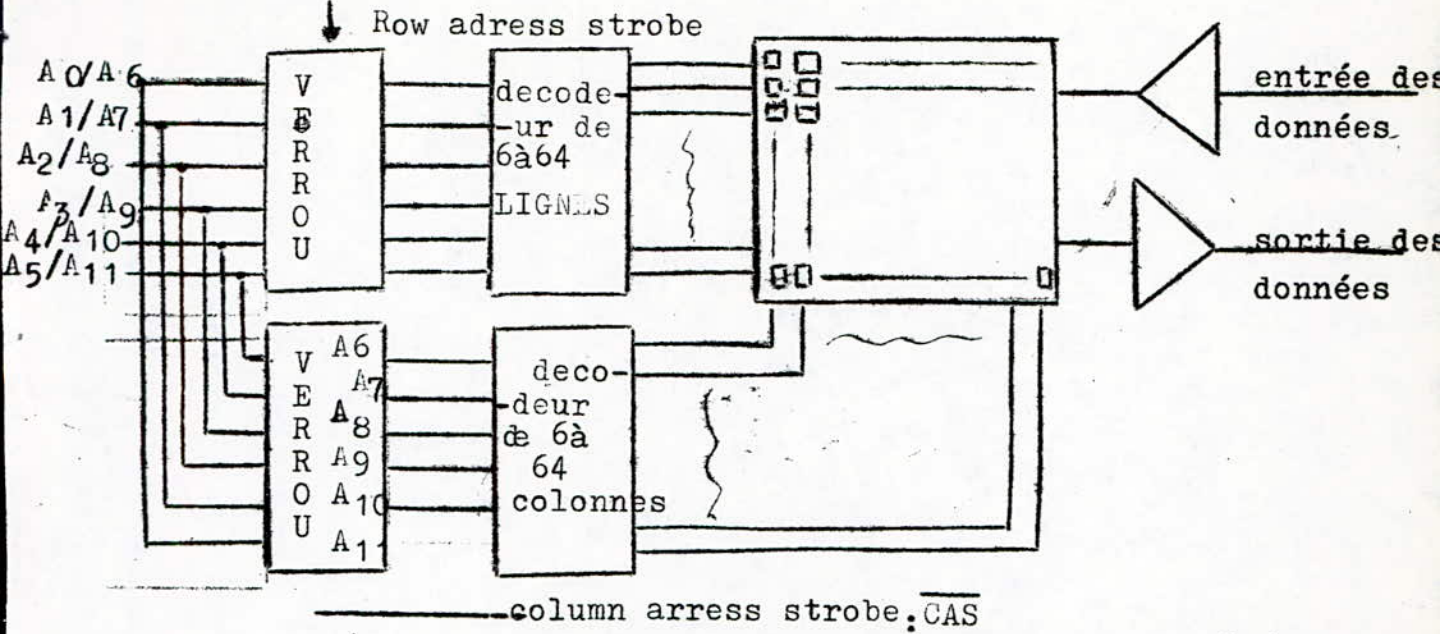
ADRESSAGE DES RAM DYNAMIQUES

L'adressage d'une mémoire de 4K bits nécessite 12 fils du bus d'adresse pour une mémoire de 16 K bits, il faut 14 fils.
La limitation du nombre de fils du bus d'adresse et du nombre de broches par boitier a conduit les constructeurs à supprimer les entrées cs (CHIP SELECT) et à multiplexer les adresses.

On peut avoir des mémoires 4 K bits sous plusieurs types de boitiers (16, 18 ou 22 broches) :

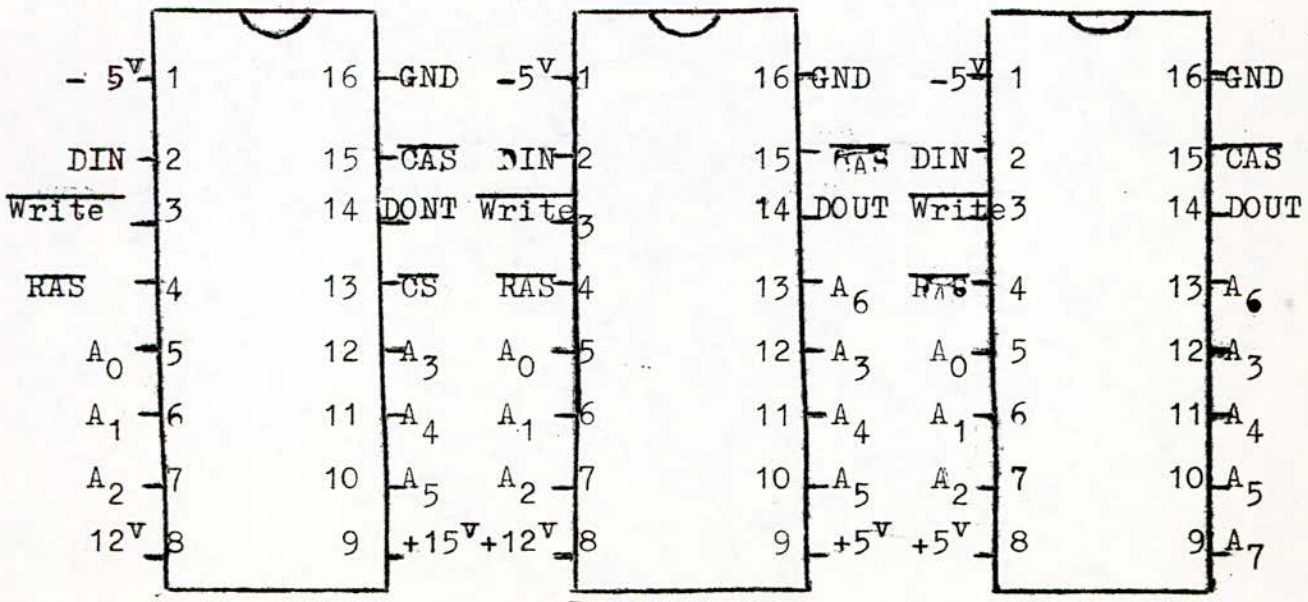


Voyons maintenant comment utiliser l'adressage multiplexé pour une RAM à 4096 bits.



Le signal appliqué sur la broche RAS valide les 6 bits d'adresse lignes (Ao, A 1, A2, A3, A4, A5) pour les ranger dans le verrou du decodeur de ligne. Ensuite les 6 bits (A6, A7, A8, A9, A10, A11) qui sont multiplexés avec (A0,....., A5) seront rangés dans le verrou du decodeur de colonne à l'aide d'un signal sur CAS. Des deux decodeurs ou reconnait le bit sollicité parmi les 4096 bits.

Certaines cartes de circuits imprimés sont réalisées de manière qu'elles puissent recevoir des RAM dynamiques de 4,16 ou 64 K bits. Pour cela de nombreuses RAM dynamiques sont conçues pour être interchangeables. Cette conception permet de choisir la configuration adaptée aux besoins particuliers de chaque utilisateur. Nous donnons ci-dessous 3 RAM dynamiques interchangeables organisées par boitier de 16 broches.



DIN : Entrée des données
DOUT : Sortie des données
RAS : ROW ADRESS STROBE) validation d'adresse ligne
CAS : (COLUMN ADRESS STROBE) validation d'adresse colonne.

Pour remplacer une une RAM à 4 K par une autre à 16 K ; il suffit de changer la broche (CS) par le bit A6 du bus d'adresse.

Pour remplacer la RAM à 4 K ou à 16 K eu une autre à 16 K il faut faire les changements suivants :

Broche 8 : (+12V)_____(+5V)
Broche 13 : (CS de 4K)_____(A6)
Broche : 9 (+5V)_____(A7).

REMARQUES :

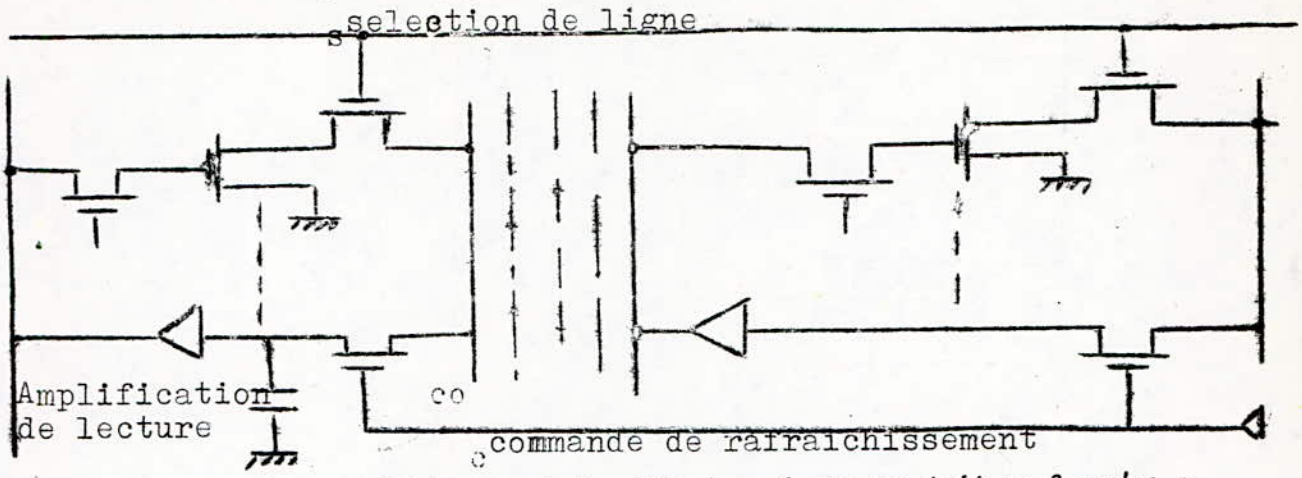
1) Dans le cas des RAM à 16 et 64 K, la broche libre 13 (CS) est remplacée par A6. Si l'on a besoin d'une fonction CS il suffit de faire appel à une logique extérieure.

2) La RAM 64 K possède une broche libre (broche 1). L'utilisateur peut obtenir une RAM dynamique à 262144 bits (256K) en connectant cette broche libre à A8.

RAFRAICHISSEMENT DES RAM DYNAMIQUES.

Le rafraichissement n'est qu'une simple lecture pour laquelle les données n'apparaitront pas en sortie. Chaque position memoire est rafraichie chaque fois que l'on y accède. Comme on ne lit ou on écrit qu'une seule ligne à la fois, il suffira d'un seul emplificateur de rafraîchissement par colonne (soit 8 pour une memoire organisée en mots de 8 Bits).

Le principe de rafraîchissement d'une RAM dynamique en technologie MOS est illustré par la figure suivante :



Le rafraîchissement des RAM dynamiques se fait en 3 modes :

1) <u>Rafraichissement par arrêt du micrprocesseur</u> :

Ce mode de rafraîchissement met le microprocesseur à l'état HALT tous les 2ms pour le sequencement de rafraîchissement.
En effet on profite des entrées de maintien "HOLD" prévues sur les boitiers d'horloge pour effectuer le rafraîchissement. Le principe consiste à "geler" l'état de l'horloge lors de l'arrivée de la demande de rafraîchissement (REFRECH REQUEST) et à effectuer pendant ce temps de "gel" un cycle de rafraîchissement des memoires. Ce mode peut être appliqué sur des mémoires 4K ou 16 K, mais il est moins utilisé parcqu'il ralentit le système.

2) <u>Rafraîchissement par vol de cycle</u> :

Dans ce mode, un circuit spécialisé vole périodiquement un cycle au microprocesseur pour effectuer le rafraîchissement d'uneligne. L'intervalle de rafraîchissement est determiné par le temps maximum de rafraîchissement et le nombre de lignes à rafraichir.

On reprend la RAM 4K de 64 lignes pour calculer la période de rafraîchissement comme chacune des lignes necessite 2ms de rafraîchissement la période sera donné par Traf :

$$Traf = \frac{2ms}{64} = 31,2\ MS$$

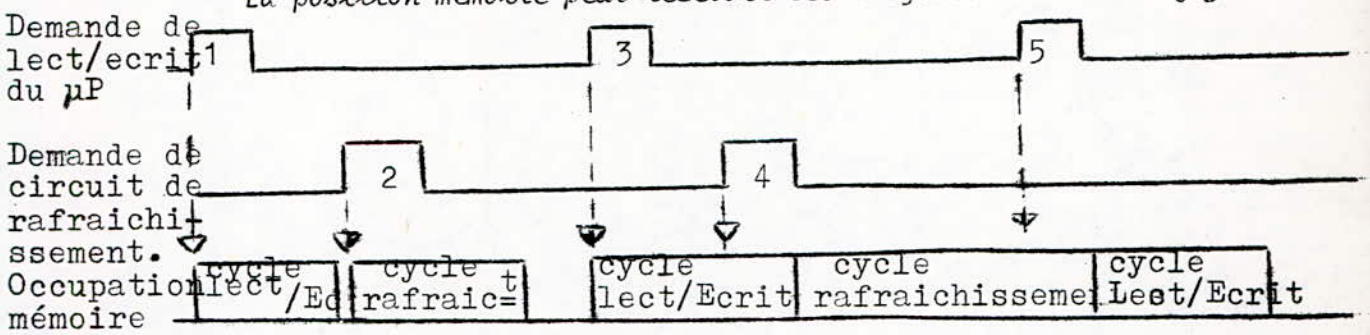
On risque des fois d'avoir des interferences entre les 2 cycles lorsque l'intervalle de rafraîchissement n'est pas respecté.

3) Rafraîchissement transparent :

Ce mode elémine cette interference en synchronisant le cyle de rafraîchissement avec le cycle mémoire du microprocesseur. La synchronisation consiste àutiliser une circuiterie qui réalise le rafraîchissement d'une manière transparente vis-à-vis du microprocesseur ; par exemple si les échanges des données se font sur le bus quand l'horloge Ø 2 est à l'état haut, le raffraichissement pourrait avoir lieu pendant l'état bas de Ø 2.

Les systèmes de mémoire utilisant la technique des RAM dynamiques sont caractérisés par unelogique de rafraîchissement. Cette logique accède automatiquement à chaque colonne toutes les 2ms au moins (la logique de rafraichissement doit être coordonnée avec les autres activités du microprocesseur) Une position mémoire peut être sollicitée d'une part le microprocesseur et d'autre part par le circuit de rafraichissement. Quelques contraintes peuvent apparaître sur l'occupation mémoire.

La position mémoire peut recevoir les 2 cycles suivant la figure :



1°) Au front montant de (1) le microprocesseur demande un cycle de lecture:écriture. Celà ne pose aucun problème puisque la mémoire n'est pas occupée par l'autre cycle de rafraîchissement.

2°) Même chose puisque seul le circuit de rafraichissement sollicite la mémoire.

3°) Si pendant la demande de lecture/ecriture du microprocesseur survient une demande rafraîchissement (4), ce dernier sera validé dès que le cycle de lecture/écriture soit terminé.
4°) Situation inverse, si pendant le cycle de rafraîchissement, le microprocesseur demande un cycle de lecture/écriture (5) il sera exécuté à la fin du cycle de rafraîchissement.

REMARQUE :

Si les deux demandes arrivent simultanément, la logique de rafraichissement devra laisser la priorité au microprocesseur, mais dans tous les cas les deux cycles seront executés.

EXEMPLE 1

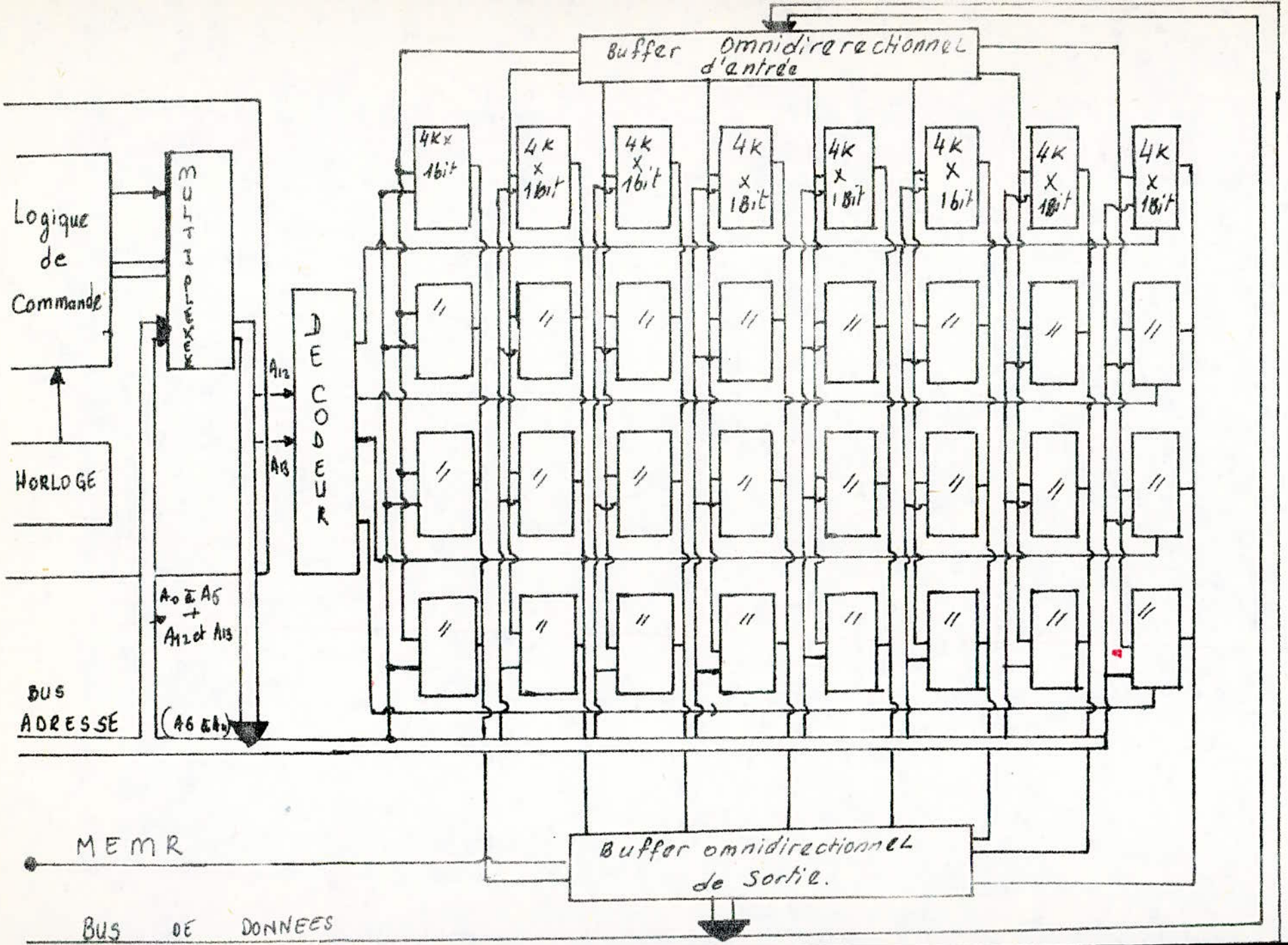
A l'aide des modules RAM de 4096 bits on peut réaliser une mémoire dynamique de 16 K mots de 8 bits. les boitiers RAM seront disposés en 4 rangées avec 8 modules chacune.

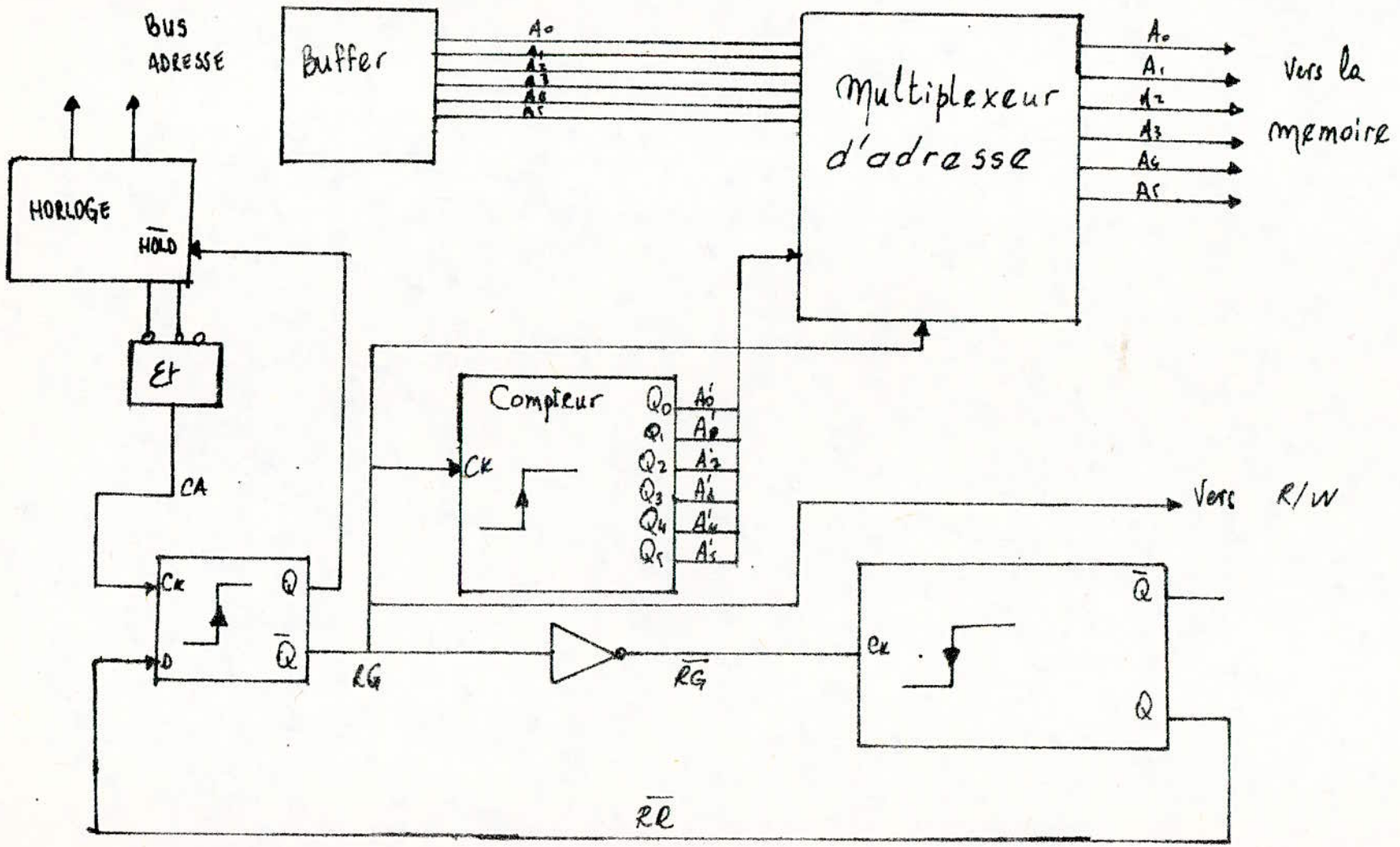
- Pour la sélection des 4K mots, il faut 6 bits du bus d'adresses Ao à A5
- Les bits A6 à A11 sont multiplexés avec Aà à A5 pour sélectionner une position mémoire.
- Pour valider une des 4 rangées, il faut decader 2 autres bits du bus d'adresses (on prend A12 et A13 pour cet exemple).

Le rafraichissement se fait donc en selectionnant :

- une ligne en envoyant une adresse sur Ao à A5,
- une rangée de modules RAM en activant une seule sortie du decodeur. Le rafraichissement est assuré par un circuit spécial permettant de faire une lecture en inhibant les sortiesde la mémoire. On utilise des buffers ommi directionnels pour l'acheminement des données en entrée ou en sortie.

Parmi les circuits spéciaux réalisant le rafraichissement on donne le schma de la figure (suite) dont le fonctionnement est le suivant :


Buffer omnidirectionnel d'entrée
4K x 1bit
Logique de Commande
MULTIPLEXEK
HORLOGE
DECODEUR
A12
A13
A0 à A5 + A12 et A13
BUS ADRESSE
(A6 à A11)
MEMR
Buffer omnidirectionnel de Sortie.
BUS DE DONNEES



RG : Refrech Grant
(Autorisation de rafraichissement)

R̄R : Refrech. Request (Demande de rafraichement)

- le compteur de cycles de rafraichissement recevant sur son entrée CK le signal d'autorisation de rafraichissement RG, fournit l'adresse de la ligne à rafraichir. Cette adresse sera fournie à tous les boitiers (on peut s'en passer alors du circuit qui génère la sélection de lignes RAS).

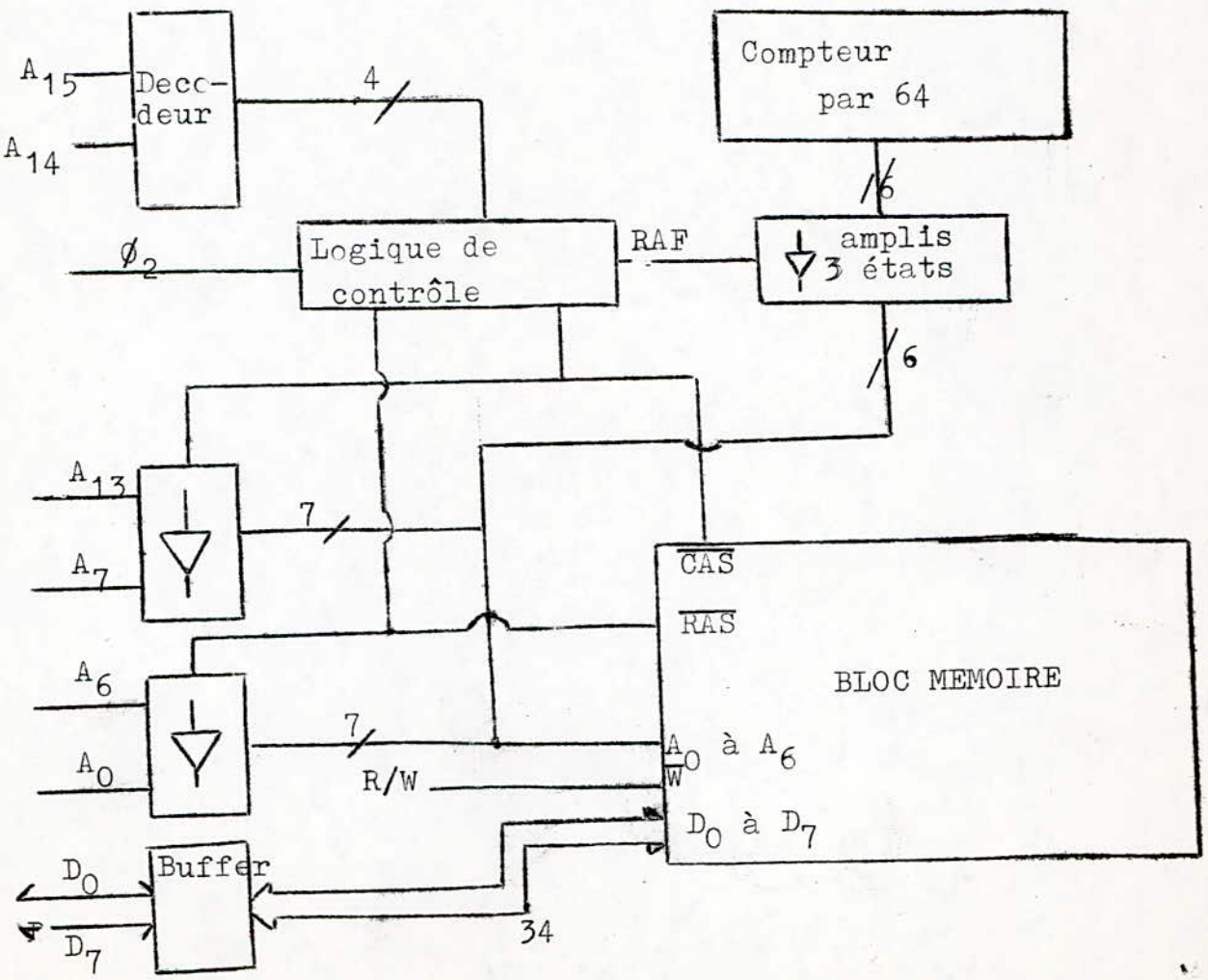
- le multiplexeur d'adresses et à 6 canaux (1 marmi 2). Il reçoit d'une part les bits d'adresse (Ao.....A5) et d'autre part les adresses générées par le compteur. Ce sont ces dernières qui adresseront les memoires pendant le cycle de rafraichissement. Le signal RG valide le multiplexeur.

- le signal R/W doit être au niveau logique 1 c'est à dire en lecture

- le signal de selection de colonne (CAS) doit être aussi au niveau logique 1.

EXEMPLE 2

Appliquons le mode "rafraichissement transparent" pour réaliser une carte RAM de 64 K mots. Les modutes RAM choisis sont des memoires de 16K X 1 Bit. La sélection d'un des bits nécessite 14 bits du bus d'adresses (Aà .....A13) Dans cet exemple, les adresses arrivent sur des amplificateurs 3 etats dont les sorties sont communes et aboutissent sur les mêmoires. Un docodage d'adresse est indispensable pour sélectionner le bloc memoire du 16K mots utilisé.

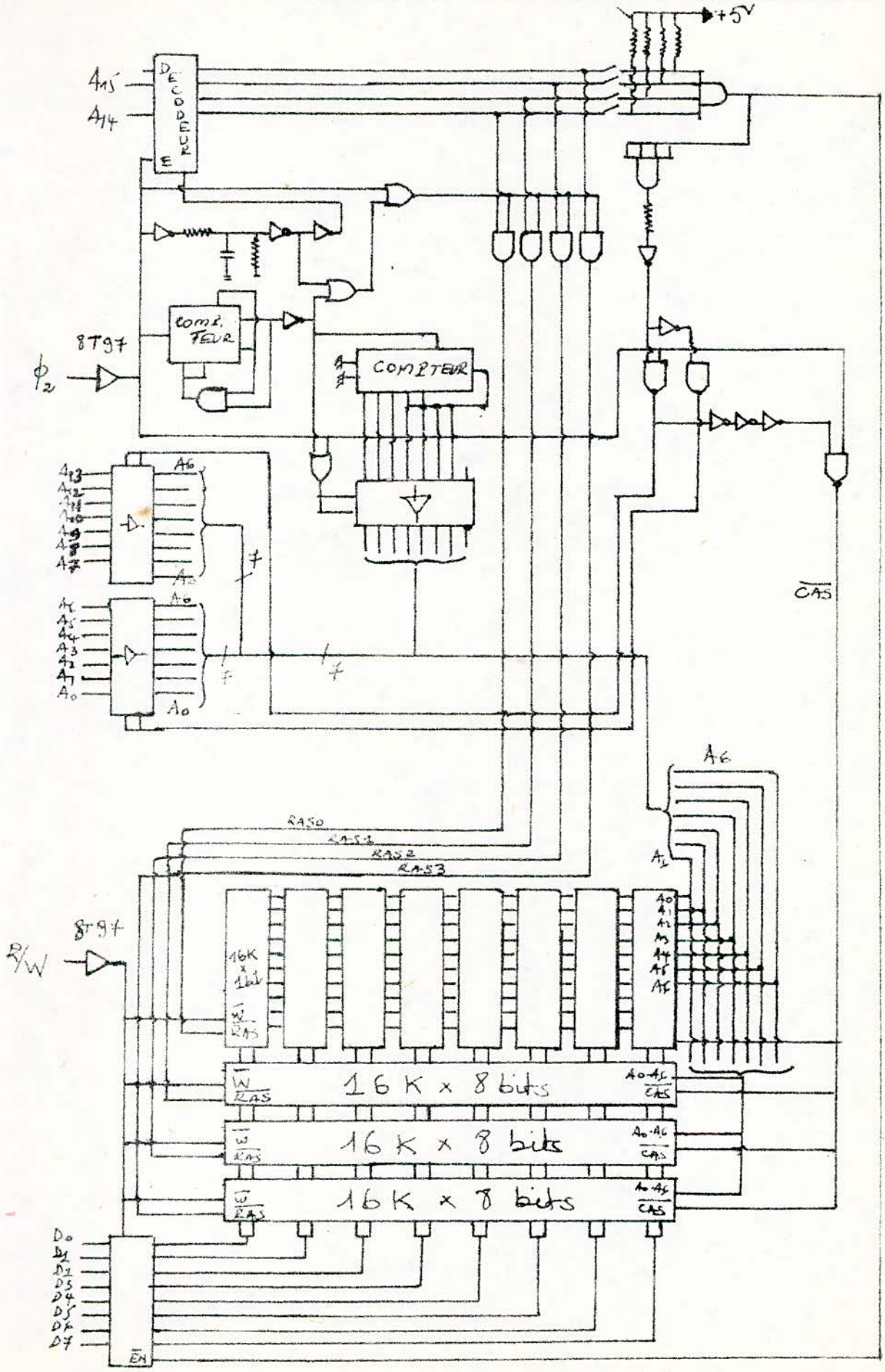


- Les échanges de données n'ont lieu que lorsque Ø2 est à l'état haut. on peut faire le rafraichissement des momoires pendant l'état bas de Ø2. pour ce faire, un compteur par 64 est incrementé à chaque descente de Ø2, ce compteur est suivi par un empli trois états qui aboutit lui aussi sur les lignes d'adresses des mémoires et qui n'est validé que lorsqu Ø2 est à l'état bas. On ajoute sur cela un autre compteur (appartient à la logique de contrôle) qui n'autorise le rafraichissement que tous les 5 cycles de Ø2.

Avec une logique supplémentaire, les sorties du decodeur permettent :
- de générer le signal CAS
- validation des amplis du bus de donnée
- génération des RAS selon le nombre de bloc équipant la carte (dans notre cas 4 blocs)

REMARQUE:

Les entrées et sorties des memoires sont reliées entre elles puisque les sorties sont en haute umpedance pendant une écriture.


+5V
A15
A14
DECODEUR
E
COMPTEUR
8T97
ϕ_2
A13
A12
A11
A10
A9
A8
A7
A6
A0
7
A6
RAS0
RAS1
RAS2
RAS3
A7
8T97
R/W
16K x 1bit
$\overline{W}$
$\overline{RAS}$
16 K x 8 bits
A0-A6
$\overline{CAS}$
D0
D1
D2
D3
D4
D5
D6
D7
$\overline{EN}$

DEUXIÈME PARTIE

CHAPITRE 3

LES ENTREES-SORTIES

NTRODUCTION:

Les capacités d'E/S d'un microprocesseur constituent sa force rincipale car leurs raisons d'être est le controle des dispositifs externes t le traitement des donnees provenant de l'extérieur, ce qui fait que e bloc dE/S est la partie la plus complexe dans la conception d'un ystème microordinateur.

Le microprocesseur doit donc être connecté à des interfaces our saisaie d'informations et par la même, disposer de macroprogrammes our le traitement de cette information et le controle des interfaces et e leurs environnements (peripheriques).

L'analyse des élements d'interfacage se fait en partant du icroprocesseur (dont les principes sont supposés connus) pour se propager ers l'environnement qu'il doit commander. (moteur, feux de carrefour, tc ...) ou qu'il doit prendre en compte.

Environements numériques :

C'est le cas des claviers, des afficheurs, des mémoires ou tres travaillant avec des données numeriques .

Environnements analogiques :

Les phéripheriques classés dans ce type d'environnement availlent avec des données analogiques, ils doivent donc communiquer avec processeur par l'intermediaire de systèmes hybrides:

-Convertisseur Numerique /Analogique ou analogique/Numerique, ltiplexeurs analogiques, echantilloneurs bloqueurs, capteurs, etc.... même la communication du microprocesseur avec l'exterieur dans les 2 ns (Entrée ou Sortie) pase le problème des modes et des moyens de ansmission des informations.

Cette transmission de données (entre UP et interface ou tre interface et peripherie) se fait caractère par caractère (mot par t) codé sur 5,6, ou 7 bits (le plus souvent) ces mots d'informations uvent être transmis soit en serie soit en parallèle. Pour la transmission rllele (le mode le plus simple) ou pourra utiliser des circuits integrés T.L ou bien des interfaces spécialisés (P.I.A) dans ce mode de transmissi r contre pour la transmission serie l'utilisation d'un circuit interface ogrammable spécialisé (dans la transmission serie) s'impose, en effet les ts d'un mot sont transmis en serie l'un après l'autre, leur transmission fera au rythme de l'horloge de l'emetteur, le recepteur doit alors oir la même frequence d'horloge. pour deco der carrectement les données i lui sont transmises.

.../...

On distingue 2 types de transmissions série :

- Transmission serie asynchrone :

Chaque caractère emis doit être accompagné d'un bit de début et de fin (pour sa reconnaissance) et eventuellement d'un ou de plusieurs bits de parité (Pour verifier la validité du caractère).

- Transmission serie synchorne.

Les caractères devant se succeder sans intervalle au rythme d'une horloge, les bits constitutifs de chaque caractère n'ont plus besoin d'être accompagnés d'un bit de debut et d'un bit de fin. Les données sont transmises en bloc de N caractères à une cadence fixe ne necessitant qu'une synchronisation au debut d'emission d'un bloc donné.

Nous verrons un peu plus loin beaucoup d'exemples d'interfaçage pour les 2 types d'environnements et les 2 modes de transmission et les différents circuits utilisés à cet effet, ces exeples sont réels et concrets d'application, nous pensons donner ainsi les ontils necessaires aux couplage d'autres systèmes différents de contexte et de complexité.

Concernant l'adressage des mémoires et interfaces ainsi que leurs connections aux différents bus du système, nous avons pu donner des methodes générales et même universelles (à toutes les familles des microprocesseurs). C'est d'ailleurs la partie la plus commune dans la conception d'un système microordinateur. Dans cette partie les choses commencent serieusement à se particulariser et ce pour les raisons suivantes :

- Chaque famille de microprocesseur offre ses propres interfaces, donc theoriquement incompatibles avec les autres familles.
- Chaque famille propose des circuits interfaces spécialisés (P I.A, ACIA.. et trés spécifiques d'applications.
- Extrême diversité des peripheriques pouvant interfacer avec un système microodinateur.
- D'une famille à une autre plusieurs ameliorations concernant les procedures et les modes d'E/S peuvent survenir.

Notre étude des E/S sera basée essentiellement sur 3 familles de microprocesseurs (9900, 8080, 6800).

.../...

Nous essayerons de tirer les points communs et differences majeures relatives aux problèmes d'E/S, seul moyen de donner des enseignements constructifs et gereraux sur les E/S d'un système microordinateur, ce qui est grandement souhaité par le concepteur, ce dernier aura beaucoup d'exemples en main pour la conception de son système approprié.

Il est important de souligner que notre but ici n'est pas de dresser des comparaisons des procedures d'E/S pour ces 3 familles Si elles ont lieu, elles ne seront faites que sur la base de la simplicité et de la souplesse dans l'application, ainsi que sur les différents signaux de commandes couplant d'une part les interfaces aux peripheriques et d'autres parts les interfaces aux microprocesseurs et leurs différents bus.

A) TECHNIQUES D'ENTREES /SORTIES

Les techniques d'E/S d'un système microodinateur sont au nombre de 3

- E/S Controlées par programme
- E/S Controlées par interruption
- E/S Controlées par DMA (Accés direct mémoire)

Ces trois techniques sont pratiquement exploitables par toutes les familles de micro processeur.

Les explications et les détails concernant les 3 procedures sont supposées connues et bien assimilées par le lecteur ous verrons ici pour les trois familles citées les deux premières techniques (E/S controlée par programme et par interruption) les E/S controlées par DMA seront vues dans un chapitre à part.

B) Les instructions d'entrées/sorties

Toutes les familles du micro processeur offrent la possibilité d'adresser et de commander un peripherique de la même façon qu'une position mémoire, les instructions d'E/S ne seront alors que des instructions de transfert, ainsi on peut manipuler les E/S avec les mêmes instructions permettant de manipuler une position mémoire c'est le mode " E/S entrélacée avec la mémoire" ou " memorymapped".

.../...

Certaines familles offrent encore la possibilité de commander les périphériques par des instructions spécifiques (valables uniquement pour les peripherique).

C'est le mode " E/S séparées" ou "ISOLATED I/O".

Dans le jeu d'instruction de la famille 8080 ou retrouve : "IN" et "OUT" dans celui de la famille 9900 on retrouve : LDCR,STCR,TB, SBZ,SBO.Nous allons voir maintenant les circuits et les dispositifs permettant de réaliser des opérations d'entrées/sorties.

Ces circuits permettront d'acheminer au processeur les informations à traiter (entrées), et aussi de communiquer ou transmettre des resultats des traitements ou des sugnaux de commande à l'usage d'application externe (sortie)

C) LES CIRCUITS D'ENTREES / SORTIES

On distingue en général deux ensembles de circuits d'E/S

1) circuits integrés TTL :

Le concepteur aura à choisir quelques uns parmi plusieurs de ces circuits tels que : multiplexeurs, décodeurs) BUFFERS, Encodeurs de priorité, etc...

Ensuite les assembler et les connecter suivant son propre genie pour realiser l'interface q'uil désire

.Ces circuits d'E/S sont compatibles donc communs à toutes les familles du microprodesseur (c'est d'ailleurs la seule chose commune) l'utilisateur devra toutefois les adresser et les commander comme des positions mémoire (memoy mapped)

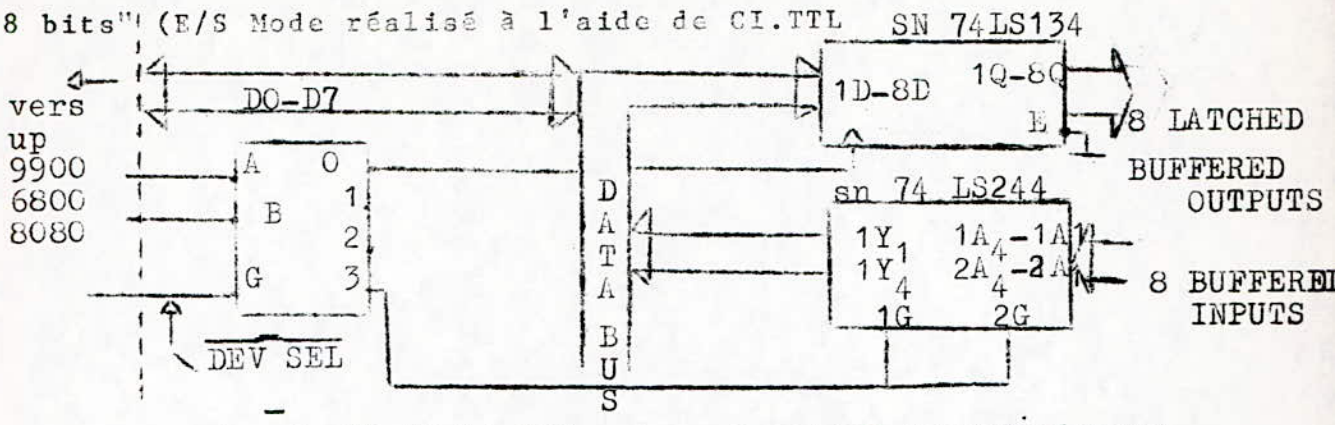
Ces circuits se connecteront comme les mémoires, leur adressage et leur commande necessitera l'utilisation de quelques bits du bus d'adresse et de commande.

Généralement dans un microordinateur on prévoit des connecteurs (comme le cas de l'APPLE II+ du KIT DS de MOTOROLA, de la carte TMS 990/189 etc...) où sont presents les 3 bus du système pour connexion eventuelle de ces circuits et une extension hors carte.

Ces circuits ne sont evidemment pas programmables ils ne peuvent donc assurer que des transmissions parallèles.

.../...

Nous donnons ci-dessous un shéma type d'interface " 8 bits" (E/S Mode réalisé à l'aide de CI.TTL



Le circuit SN 74 LS 139 est un decodeur demultiplexeur possédant "1 parmi 4" décodeurs totalement indépendants les deux entrées adressées A et B peuvent être réliées respectivement à WE et DBIN du 9900, à WR, DBIN du 8080, R/W et un bit du bus d'adresse pour le 6800 . Pour la validation d'un des BUFFERS (SN 74134 ou 244) l'entrée G doit être au niveau "0" ainsi lors d'une opération de lecteur DBIN (ou le bit du bus d'adresse utilisé) est actif, (le circuit SN 74 LS 244 est validé (le circuit d'écriture SN 74LS 184 est à l'état haute impédance).

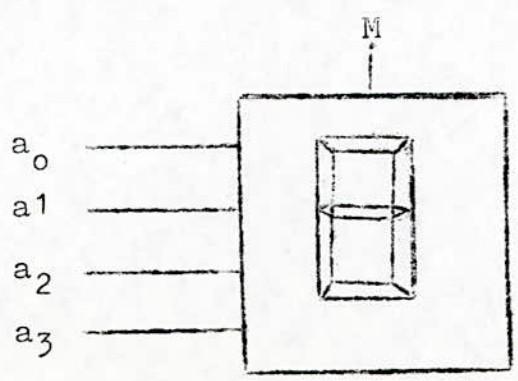
De même pour une opération d'écriture le circuit SN 74 LS 134 est validé du fait que WE est actif et DBIN inactif (entrée 00 donnant en sortie décodée 0111).

Le lecteur pourra bien assimier le fonctionnement de ce montage en se referant au DATA BOOK TTL où se trouve le brochage et le Shéma interne des circuits utilisés.

Ce circuit pourra servir dans maintes applications. Les 8 bits de sortie par exemple sont utilisés pour l'interfaçage d'officheur hexadécimaux.

Les afficheurs seront supposés disposer d'un decodeur interne 7 segments et d'un registre de memorisation

comme exemple prenons le TIL 311 de TEXAS INSTRUMENTS de la fig ci-contre



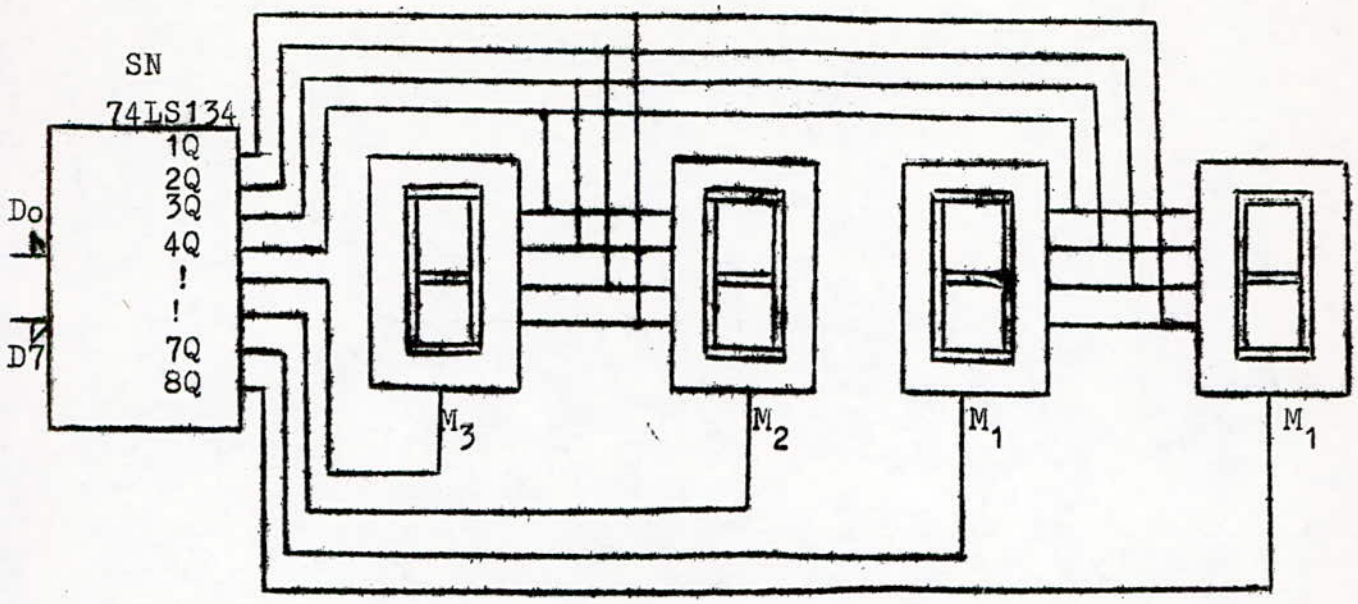
L'affichage réalisé est representé dans la table ci-dessous

si l'entrée M=0

a_3	a2	a_1	a_0	AFFICHAGE
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	A
1	0	1	1	B
1	1	0	0	C
1	1	0	1	D
1	1	1	0	E
1	1	1	1	F

Si M=1 l'afficheur memorise le dernier digit et l'affiche en permanence .

Les entrées a_0, a_1, a_2, a_3 de chaque afficheur peuvent être reliées entre elles et l'ensemble aux sorties 1Q à 4Q (du SN 74LS134). Les bits qui restent (5Q à 8Q) seront connectés aux commandes de memorisation M de chaque afficheur suivant la figure:



Les instrictions permettant l'affichage d'un nombre quelconque sur ses afficheurs ne seront au fait,quedes instructions detransfert (d'une memoire ou d'un registre vers une autre memoire qu'est l'afficheur).

Pour la famille 9900 on peut aussi utiliser des circuits integrés T T L pour realiser des interfaces en utilisant la technique d'E/S par CRU. (Une étude detaillée concernant cette technique sera exposée annexe)

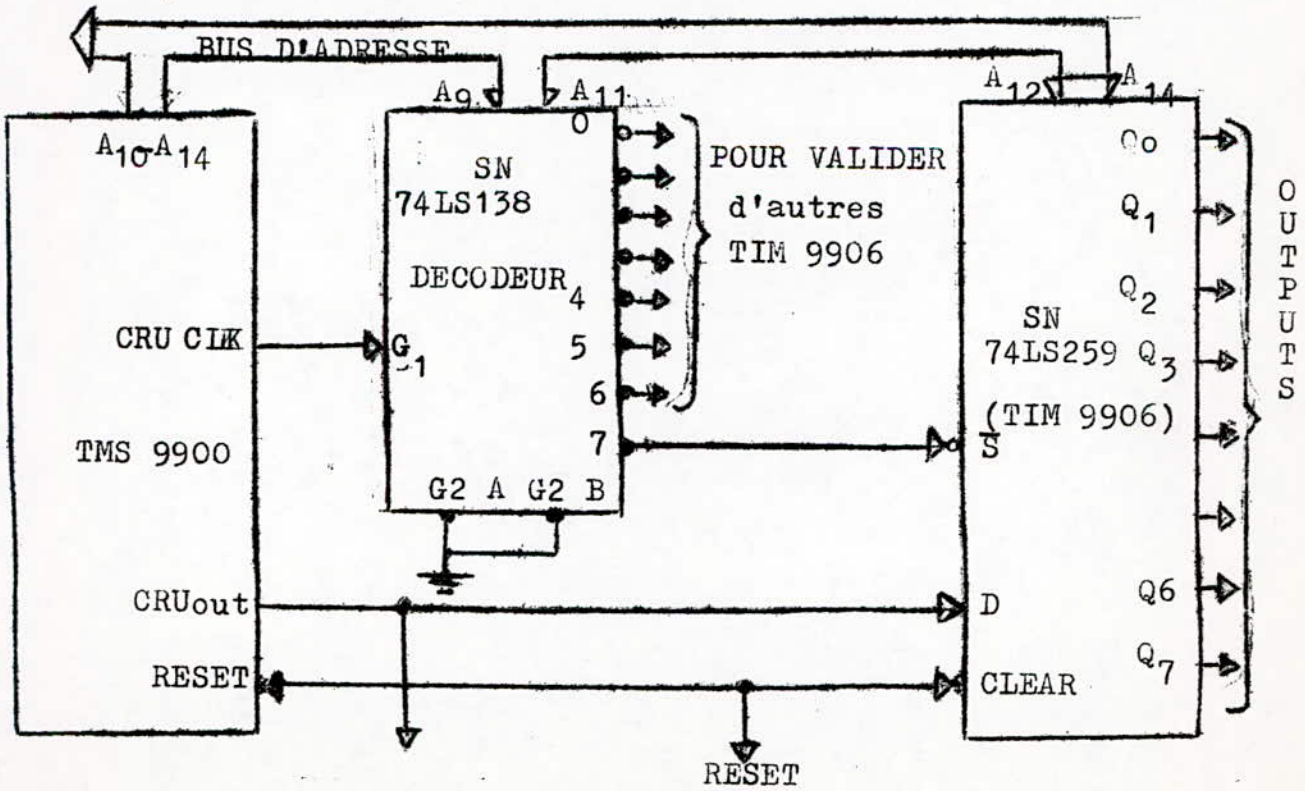
Dans ce cas plusieurs schemas peuvent être utilisés,àtitre d'exemple nous donnons ci-dessous deux montages typiques:

Ces montages peuvent servir aux mêmes applications que le montage donné plus haut ;Seulement dans ce cas,les instrutions permettant d'acheminer les données vers le microprocesseur ou bien de les presenter en sortie, seront des instructions CRU (LDCR,STCR,TB, SBZ,SBO).

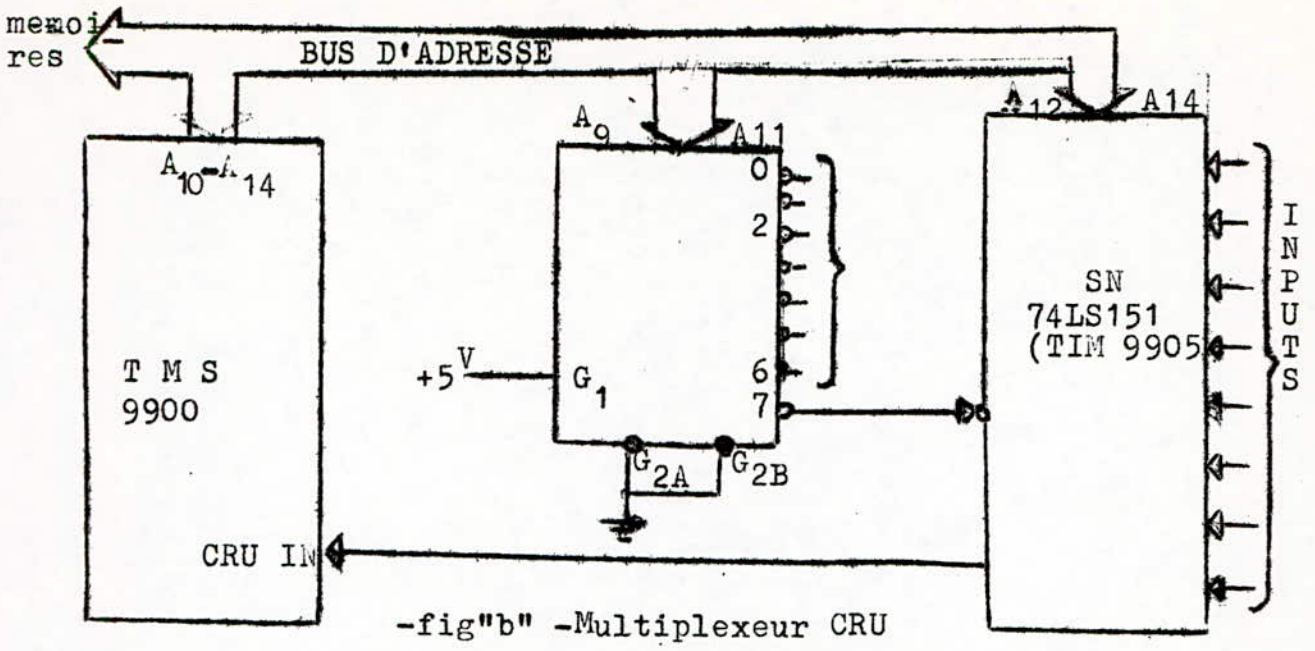
On remarque dàns la figure "A" que l'entrée G1du decodeur SN74LS138 est reliée à CRU CLK du UP,ce qui signifie que durant un transfert en sortie par nistruction CRU,les 8bits de sortie du buffer (SN 74LS259)ne peuvent être alterées que si CRU CLK pulse.

Alors que dans la figure "b" cette entrée est mise à l'etat haut (validant ainsi le decodeur)par cablage .

En effet lors d'une operation d'entrée (lecture) par CRU c'est le microprocesseur qui pilote lui même l'operation.



-fig"a" - LATCHED CRU INTERFACE



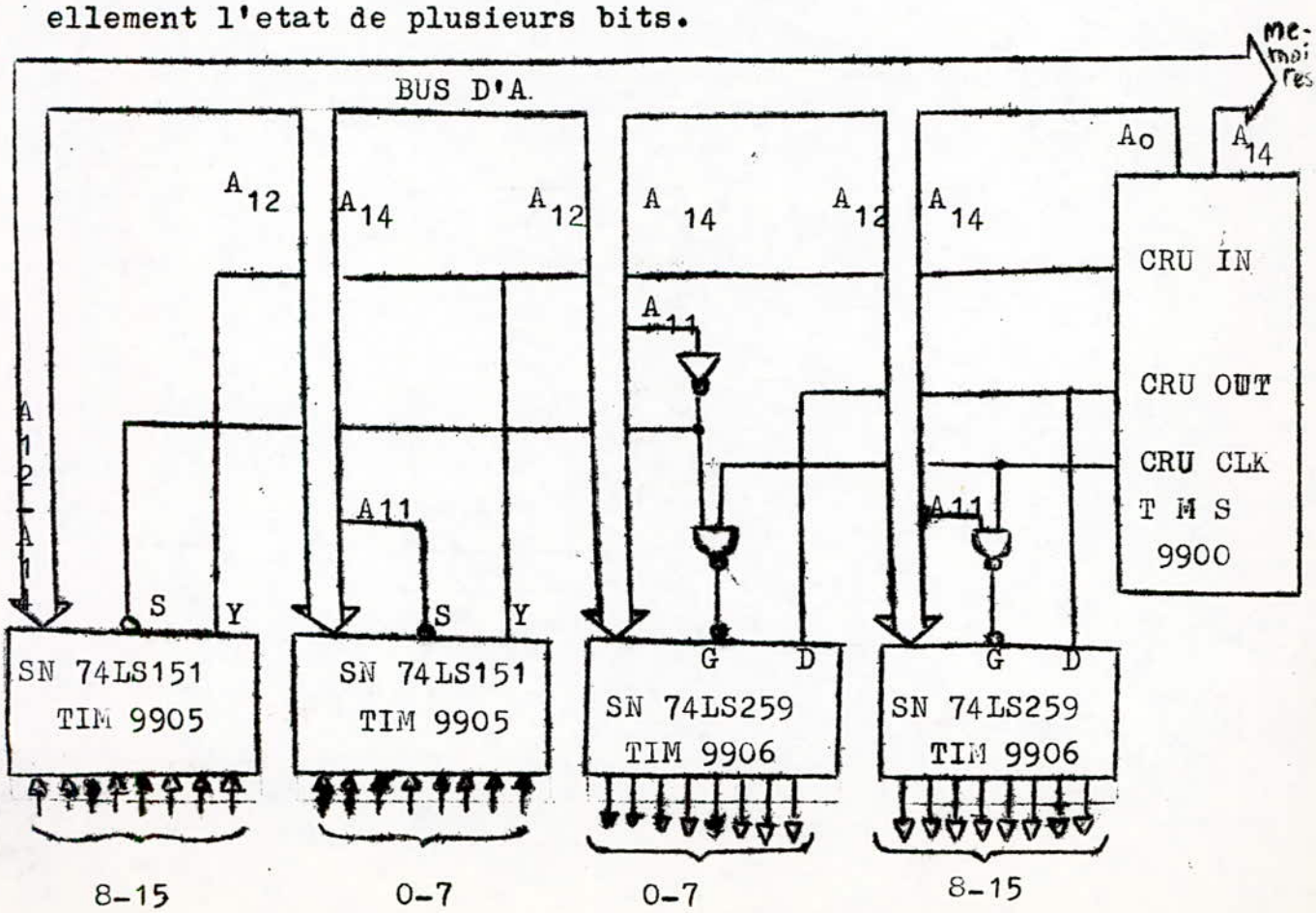
-fig"b" -Multiplexeur CRU Interface

Les bits A_9,A_{11} sont utilisés comme entrées des decodeurs 1 parmi8 (SN 74LS 138). Les sorties decodées servent à valider un TIM 9906 ou9905 parmi ceux qui existent.

Les bits A_{12},A_{14}servent à selectionner un bit parmi les 8 bits INPUTS que contient le TIM 9905 OU OUTPUTS que contient leTIM 9906.

On peut etendre l'interface à 16 bits en entree et16 bits en sortie toujours à l'aide des circuits integrés T T L comme le montre la figure de la page d'aprés.

Les bits de chaque buffer sont adressable individuellement(en entrée ou en sortie).On remarque ici la grande souplesse dans les applications surtout lorsqu'on doit commander ou tester individuellement l'etat de plusieurs bits.



B - LES INTERFACES PROGRAMMABLES:

Chaque famille propose ses propres circuits (compatibles uniquement avec le microprocesseur de cette même famille). Ces circuits permettent de réaliser des opérations d'E/S avec un minimum de logiciel et de logique supplémentaire. Ils se distinguent par le mode de transmission qu'ils peuvent assurer.

1) LES INTERFACES PROGRAMMABLES " PARALLELES":

Ce sont des circuits de transmission parallèle, ils possèdent généralement deux ou trois ports individuellement programmables en entrée ou en sortie (les bits constitufs de certains ports sont eux aussi individuellement programmables en entrée ou en sortie).

La configuration fonctionnelle de chaque port est programmée par le logiciel du système, l'initialisation de cette configuration se fait en envoyant à ces circuits des mots de commande contenant des informations sur le "mode", la remise à "zéro" ou à "un" d'un de leurs bits internes etc...

Ces circuits ont aussi l'avantage de posséder une ou plusieurs lignes d'interruptions qui acheminent directement au microprocesseur des demandes d'interruptions, ce qui élimine le recours à des circuits de logique de génération d'interruption comme dans les cas où l'on utilise des circuits intégrés T.T.L.

La famille 9900 propose le T.M.S 9901.

Le 9901 est un circuit très performant. Il possède bon nombre d'avantages le rendent très utilisable dans le domaine de la simulation et le contrôle industriel.

- Il possède un encodeur de priorité qui y est totalement intégré. Lorsqu'un périphérique (connecté à cet interface) demande une interruption et si celle-ci est valide (chaque ligne d'interruption possède une bascule de validation qu'on peut mettre par programme à "0" pour valider l'interruption ou bien à "1" pour la masquer). L'interface génère alors une demande d'interruption et son code (niveau d'interruption) associé..

  Le microprocesseur reçoit la demande et son code en même temps reconnaissant ainsi le périphérique qui l'a sollicité.

- Le concepteur n'aura pas à s'occuper du codage des interruptions en faisant appel à des encadeurs externes, celui-ci n'aura qu'à attribuer le niveau d'interruption désiré à un périphérique donné en le connectant à la ligne d'interruption adequate.

.../...

Le 9901 achemine la demande d'interruption à travers INTREQ et son code a associé à traver IC0, IC 1 , IC2, IC3, offrant ainsi 15 niveaux d'interruptions (INT 1 à INT 15).

Possibilité d'adressage bit par bit. Les bits internes de ce circuit peuvent être positionnés et contrôlés individuellement sans avoir recours au masquage des bits non concernés. Ce qui donne une grande souplesse dans le contrôle des processus industriels.

- Possession d'une horloge. Cette horloge peut servir pour la mesure d'intervalles de temps séparant 2 évenements ou encore à générer des interruptions à des intervalles de temps programmables (ce qui est d'une grande utilité pour les applications temps réel.

Application du 9901:

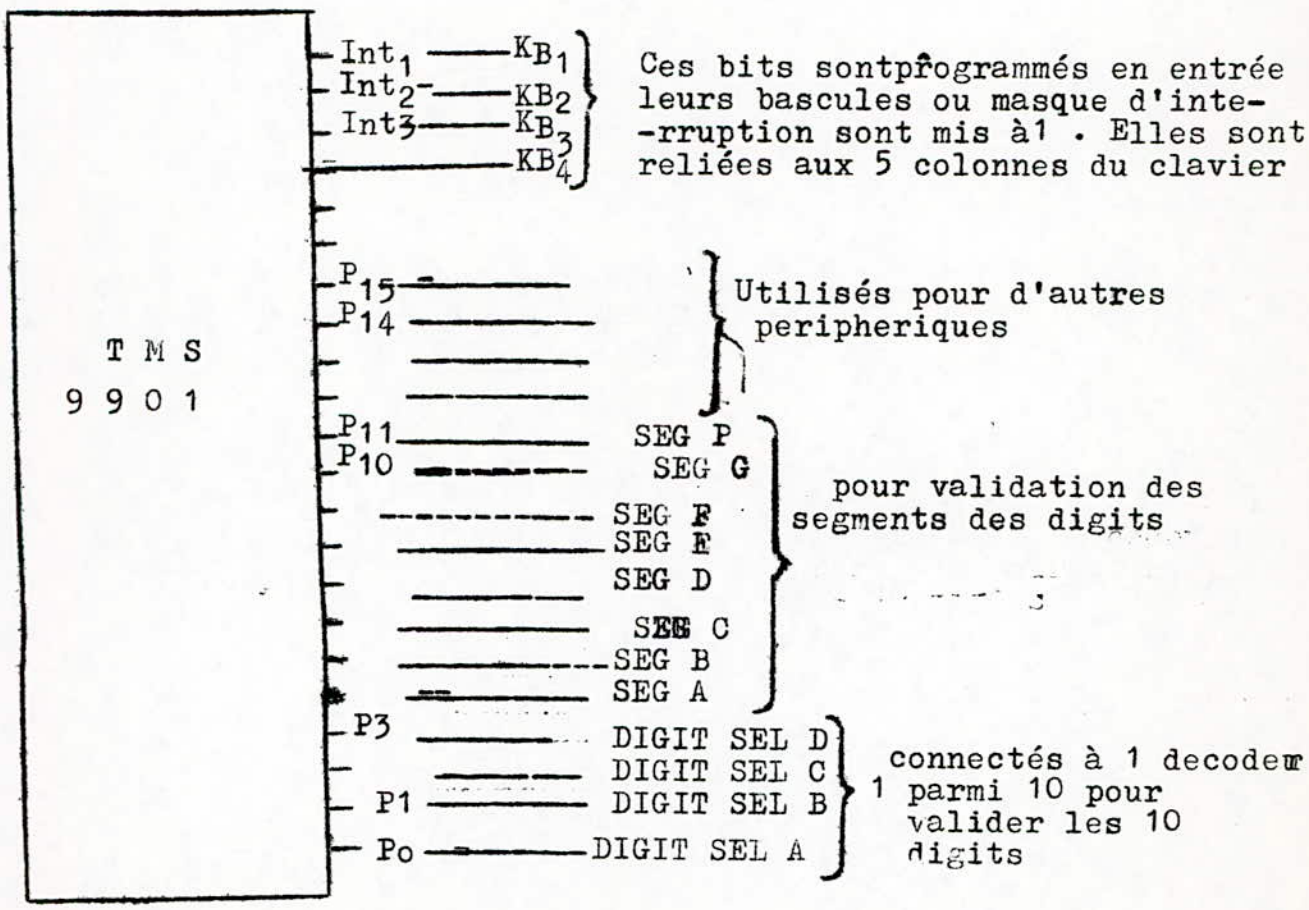
Prenons l'exemple de la carte TMS 990/189 où l'interface TMS 9901 est consacré aux périphériques sur carte comme le clavier la visualisation etc...

Nous ne verrons ici que l'interfaçage du clavier et de la visualisation.

Le clavier est à 45 touches (fonctionnant en mode principal ou secondaire) configurés en 9 Lignes de 5 Touches chacune (9 lignes X 5 colonnes).

La visualisation est à 10 digits de 8 segments chacun (le 8ème pour la virgule).

- Pour interfacer le clavier et la visualisation, le TMS 9901 de la carte est programmé comme suit:



Interfaçage de la visualisation:

- Validation des digits:

  On utilise un décodeur "1 parmi 10" SN 74LS 145, en appliquant un code de selection de 4 bits sur les entrées du décodeur (DIGIT SELA à DIGIT SEL D) on valide ainsi 1 à 10 digits de la visualisation.

- Validation des segments:
  A chaque segment des digits on attribue un fil, on aura donc 8 fils (SEG A à SEG P). Si on désire par exemple afficher "2" sur un quelconque digit on doit afficher la combinaison 11011010 reflétant respectivement le niveau logique des fils SEG A à SEG P.

Interfaçage du clavier:

L'appui sur une touche met en contact 2 Fils conducteurs (ligne et colonne) qui font partie d'un matriçage interne au clavier.

Pour détecter la touche enfoncée, le microprocesseur doit en principe explorer l'état du clavier par ligne ensuite par colonne déterminant ainsi les coordonnées de la touche activée.

Les conducteurs colonnes sont les fils KB1 à KB5 chacun possèdant une résistance de rappel en "+ 5" (niveau logique "1")

Les conducteurs colonnes sont les fils DIGIT 1 à DIGIT 9 actifs à l'état bas (rappelons que DIGIT 1 à DIGIT 10 servent aussi à valider les 10 digits de la visualisation).

Par logiciel DIGIT 1 à DIGIT 9 sont validés (mis à l'état bas) l'une après l'autre au rythme de quelques millisecondes chacun

(Le temps minimal que peut mettre un opérateur en appuyant sur une touche est $>$10 ms) et à chaque instant il y'a lecture de l'état des conducteurs colonnes.

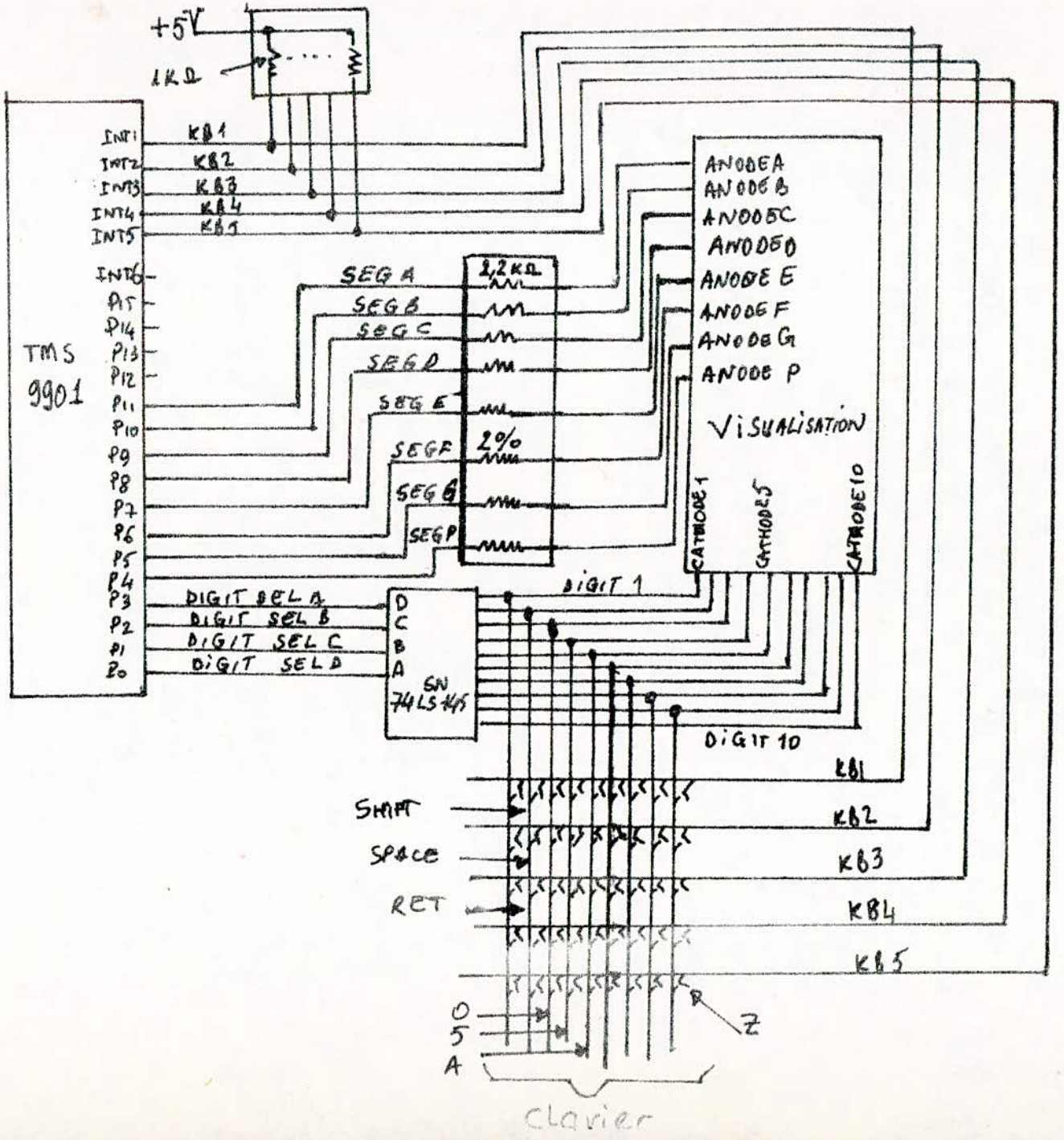
Quand on détecte un niveau logique ZERO sur une colonne, c'est ~~ex~~ que la touche correspondante à cette colonne et à la ligne validée à cet instant est enfoncée.
Il reste donc en microprocesseur (toujours sous la commande du programme de gestion du clavier) de noter les coordonnées de la touche enfoncée et d'assurer sa traduction.

Explications concernant la programmation du 9901.

Les conducteurs colonnes K B 1 à KB5 sont reliés à INT 1 à INT5 du TMS 9901 respectivement . Ces entrées interruptions peuvent être masquées (en mettant à 1 le masque d'interruption qui leur correspond.

Ces broches peuvent dans ce cas servir de broches d'entrées ordinaires, l'état des conducteur colonnes du clavier peut ainsi être lu (grâce à des instructions TB et STCR) en interrogeant l'état des bits INT 1 à INT 5 du TMS 9901.

.../...


+5V
1KΩ
TMS
9901
INT1
INT2
INT3
INT4
INT5
INT6
P15
P14
P13
P12
P11
P10
P9
P8
P7
P6
P5
P4
P3
P2
P1
P0
KB1
KB2
KB3
KB4
KB5
SEG A
SEG B
SEG C
SEG D
SEG E
SEG F
SEG G
SEG P
2,2KΩ
2%
ANODE A
ANODE B
ANODE C
ANODE D
ANODE E
ANODE F
ANODE G
ANODE P
VISUALISATION
CATHODE 1
CATHODE 5
CATHODE 10
DIGIT SEL A
DIGIT SEL B
DIGIT SEL C
DIGIT SEL D
D
C
B
A
SN
74LS145
DIGIT 1
DIGIT 10
KB1
KB2
KB3
KB4
KB5
SHIFT
SPACE
RET
0
5
A
Z
clavier

Les digits : DIGIT A à DIGIT D sont reliés aux broches Po à P3 du TMS 9901 qui peuvent être mises à ZERO ou à UN à l'aide des instructions SBO, SBZ ou LDCR de même pour les lignes validation segment (SEG A à SEG P) qui sont reliées aux broches P4 à P11 du TMS 9901.

REMARQUES/

- Quelques bits du TMS 9901 n'ont pas été utilisés mais rien n'empêche de connecter un clavier à plus de 45 Touches ou une visualisation à plus de 10 Digits.

- On peut utiliser un decodeur "1 parmi 8" pour valider les segments des digits ou encore valider chaque digit par un fil connecté directement en TMS 9901 (donc sans utiliser le décodeur 1 parmi 10). On peut aussi valider les digits séparement des conducteurs lignes ce qui nous aurait amené probablement à utiliser 2 TMS 9901 ça aurait aussi simplifié énormément le logiciel du clavier et de la visualisation.

  En conclusion nous reviendrons toujours à dire que les procédures d'interfaçage sont très spécifiques et qu'ils ne pourrait être question de donner des méthodes générales. Pour réaliser un interfaçage il y'a toujours beaucoup de configurations possibles suivant les circuits interfaces utilisés.

- La gestion du clavier et de la visualisation est assurée par 3 programmes dépendants au sein de l'UNIBNG.

1er Programme:

- Identification de la touche enfoncée par comparaison du numéro de la touche (en mode principal ou secondaire) avec la table des codes hexadécimaux 7 Segments de tous les caractères ASCII.

- Rangement du code HEXADECIMAL de la touche identifiée dans un registre.

2ème Programme:

- Analyse du caractère rangé dans ce registre (est ce un caractère ASCII ou une routine de contrôle?)
- Transferer le code 7 segments dans le tampon d'affichage.
- Gerer la routine de tabulation horizontale.

3ème programme:

- Affichage des caractères contenus dans le TAMPON d'affichage sur la visualisation.
  La famille 6800 : Elle propose les interfaces programmables parallèles suivants:

  les MC 6820, MC 6821 , MC 6822.

.../...

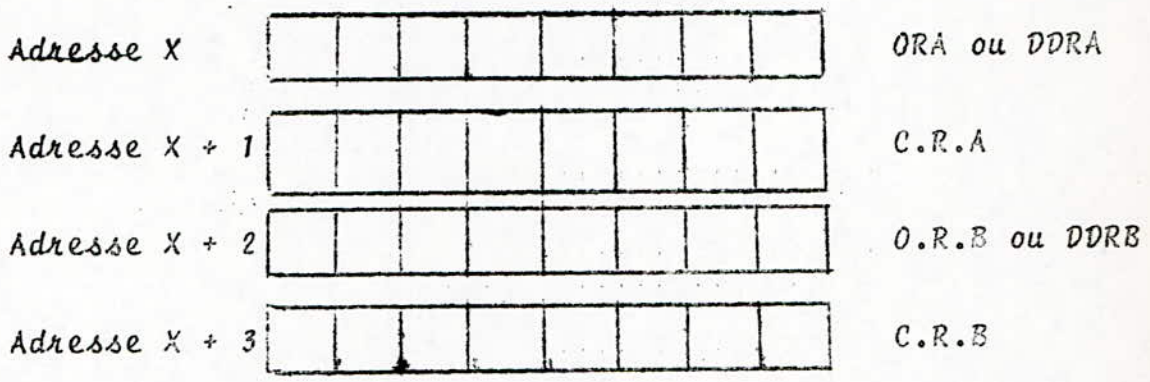
Le MC 6820:

C'est un circuit qui réalise l'interfaçage avec des périphériques (à transmission parallèle) à l'aide des deux bus de données bidirectionnels et 4 lignes de commande.

Le MC 6820 dispose de 16 Lignes d'E/S programmables individuellement et indépendemment les uns des autres en entrée ou en sortie. Il possède 4 lignes d'interruption.

Contrairement au 9901, le 6820 ne possède pas d'encodeur de priorité intégré. Les ligues d'interruption (IRQA et IRQB) sont placés à l'entrée d'un contrôleur de priorité extérieur (on cite le MC 6828 ou le mc 6830).

* La programmation du PIA permet de gagner des positions memoire (ce qui réduit à 4 le nbre d'adresses occupées en lieu de 6)



* La sélection des registres internes (en nombre de 6) permet de gagner une broche au niveau du boitier puisqu'il suffit de deux lignes RS0 et RS1

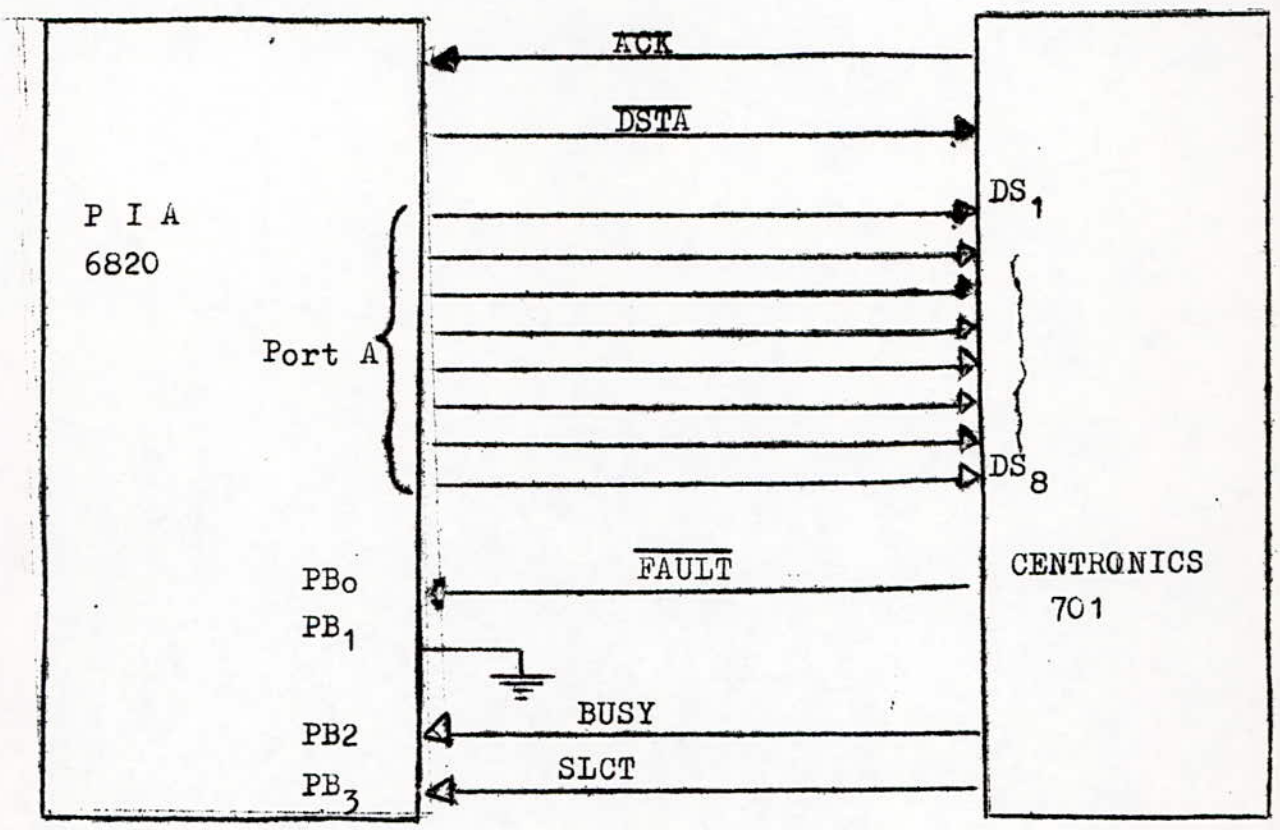
RS0	RS1	REGISTRE
0	0	ORA ou DDRA
1	0	CRA
0	1	ORB ou DDRB
1	1	CRB

EXEMPLE: 1 **Interface d'une imprimante rapide**

Une imprimante sert à récupérer des données et les stocker dans une mémoire vive interne (RAM). Elle possède un signal de reconnaissance ACK qui informe le microprocesseur qu'elle peut recevoir d'autres données. Quand cette RAM est complètement chargée, l'impimante génère un signal "BUSY" (occupé pour arrêter la transmission des données (une imprimante est capable de stocker une trace écrite du programme et des résultats sous forme de listing).

Voyons, à présent la connection du PIA avec une impimante CENTRONICS 701 (la 701 est une impimante bidirectionnelle dont la tête d'impression peut se mouvoir à droite et à gauche, cherchant toujours le prochain caractère de la nouvelle ligne avant de commencer à écrire. Les 7 aiguilles de la tête d'impression sont activées par sélection pour former, avec des points le caractère spécifié).

Connection PIA - CENTRONICS 701 :



Lignes d'entrée de l'imprimante:

DS1 - DS8 : Sont connectés aux lignes du port A dont les 7 premières représentent les 7 bits du code ASCII (le 8ème Spécifie si l'écriture doit être en caractère gras).

DSTA (DATA - STROBE).

Ce signal permet de synchroniser la transmission de données du microprocesseur vers l'imprimante.

Lignes de sortie de l'imprimante:

Busy: Signal généré si la RAM interne de l'imprimante est entièrement chargée ACK: Signal de reconnaissance pour la réception de données.

.../...

SLCT: A l'état haut, cette ligne indique que l'imprimante est sélectionnée (La 701 est munie d'un bouton de selection).

FAULT: A l'état bas, ce signal indique:

- imprimante non sélectionné
- ou bien pas de papier.

REMARQUES:

- l'Utilisateur peut utiliser le port B pour émettre les données à l'imprimante et le port A pour recevoir les commandes comme ACK, BUSY et FAULT.

- La sélection de l'imprimante peut se faire en envoyant un code octal 021 (11 en hexadécimal) par le bus de données.

- Pour éviter quelques erreurs (comme imprimante non sélectionné pas de papier) on programme l'imprimante en forçant PB0 à "1" PB1 à "0".

EXEMPLE: 2

2°) Interface d'un perforateur rapide:

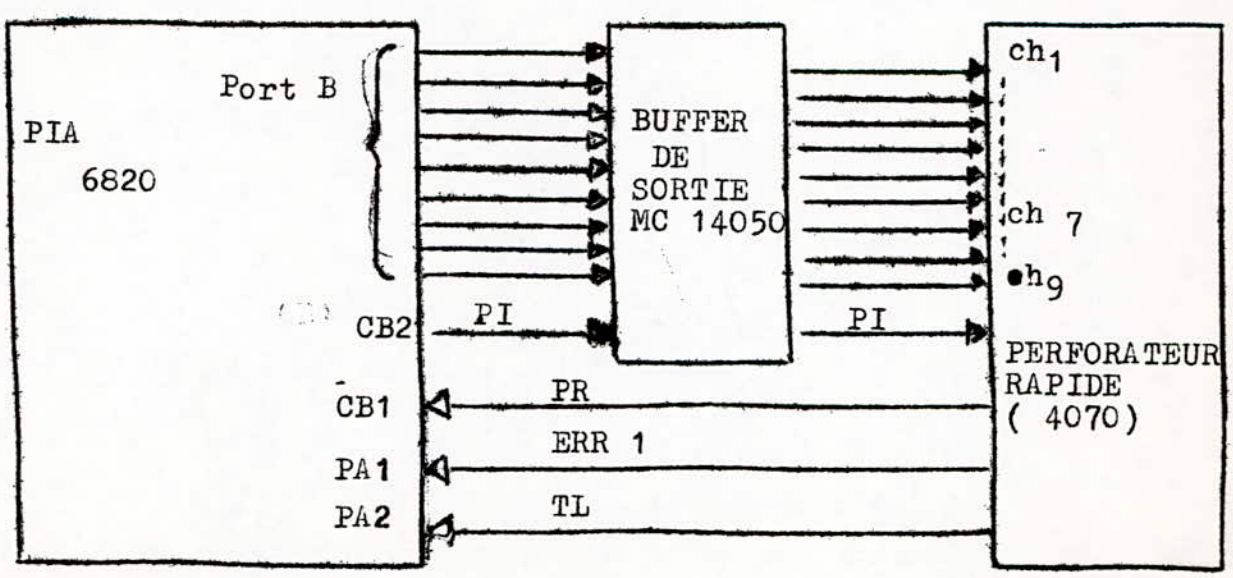
Généralement, quand il s'agit d'un nombre important de données, on commence par les stocker et les manipuler en mémoire vive, puis les sauvegarder en moyen du perforateur sur une bande au lieu de les réintroduire dans la mémoire du processeur.

On choisit le perforateur de bande modèle 4070 pour le connecter au PIA.
Le 4070 possède 8 lignes pour recevoir les données, on doit donc programmer un des ports dans le sens sortant (port B dans notre cas).

- Une ligne du port A est programmée en sortie pour perforer le trou d'entrainement, les autres lignes programmées en entrée pour servir de commandes.

REMARQUE:

La logique interne du perforateur ne contient pas d'amplificateurs d'entrée, pour cela on utilise des amplis de sortie du type MC 14050.



Lignes d'entrées du perforateur:

Ch1 - Ch8 : Lignes de données connectées au port B (Programmé en sortie).

Ch9 : Connectée à PA0, cette ligne sert à perforer le trou d'entrainement.

PI: Signal de perforation (lors de la transmission des données caractère par caractère, le signal PI actionne la commande d'avance ruban d'un pas)

Lignes de sortie du perforateur:

PR : Connecté à CB1, signal généré pour demandes des données.

ERR1 : Signal d'arrêt du perforateur si la bande est trop tendue ou cassée.

TL : Quand la fin de bande approche, le signal TL passe à l'état haut.

EXEMPLE: 3

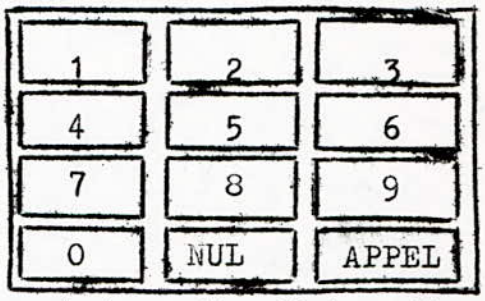
Nous allons examiner maintenant une application où l'on utilise un clavier numérique qu'on gère par microprocesseur. Ceci dans le but de réaliser une clef électrocnique déclenchant l'ouverture d'une porte (d'un immeuble par exemple) selon un code secret et de signaler toute tentative d'accès interdite.

Le clavier utilisé est à 12 Touches comme l'indique la figure.

Touches de 0 à 9 : pour taper la combinaison.

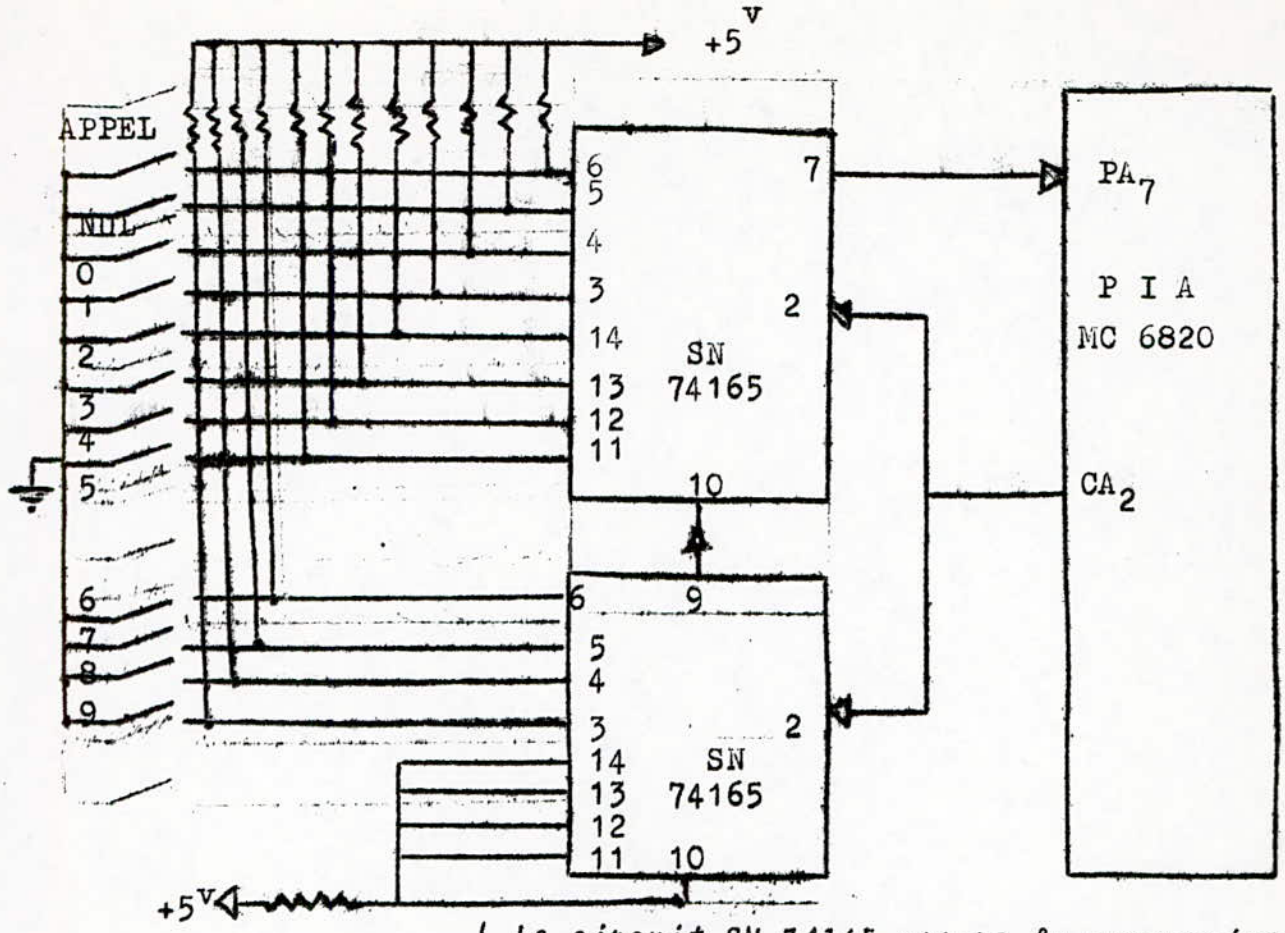
Touche APPEL : pour un appel

Touche NUL: Pour annuler la combinaison.



Si le microprocesseur s'occupe uniquement de cette tâche, il passerait tout son temps à attendre des codes du clavier. Donc entre deuxq appels, il est préférable de l'utiliser pour d'entres applications. Ainsi on propose de traiter ce problème par interruption du microprocesseur.

Chaque touche enfoncée génère un code qu'on doit alors acheminer au microprocesseur via un PIA (MC 6820). Le microprocesseur peut être très éloigné du clavier, il serait alors plus commode de transmettre ce code en série sur un seul fil. On utilise pour ce fait deux circuits intégrés SN. 74LS 165 couplés au clavier comme suit:

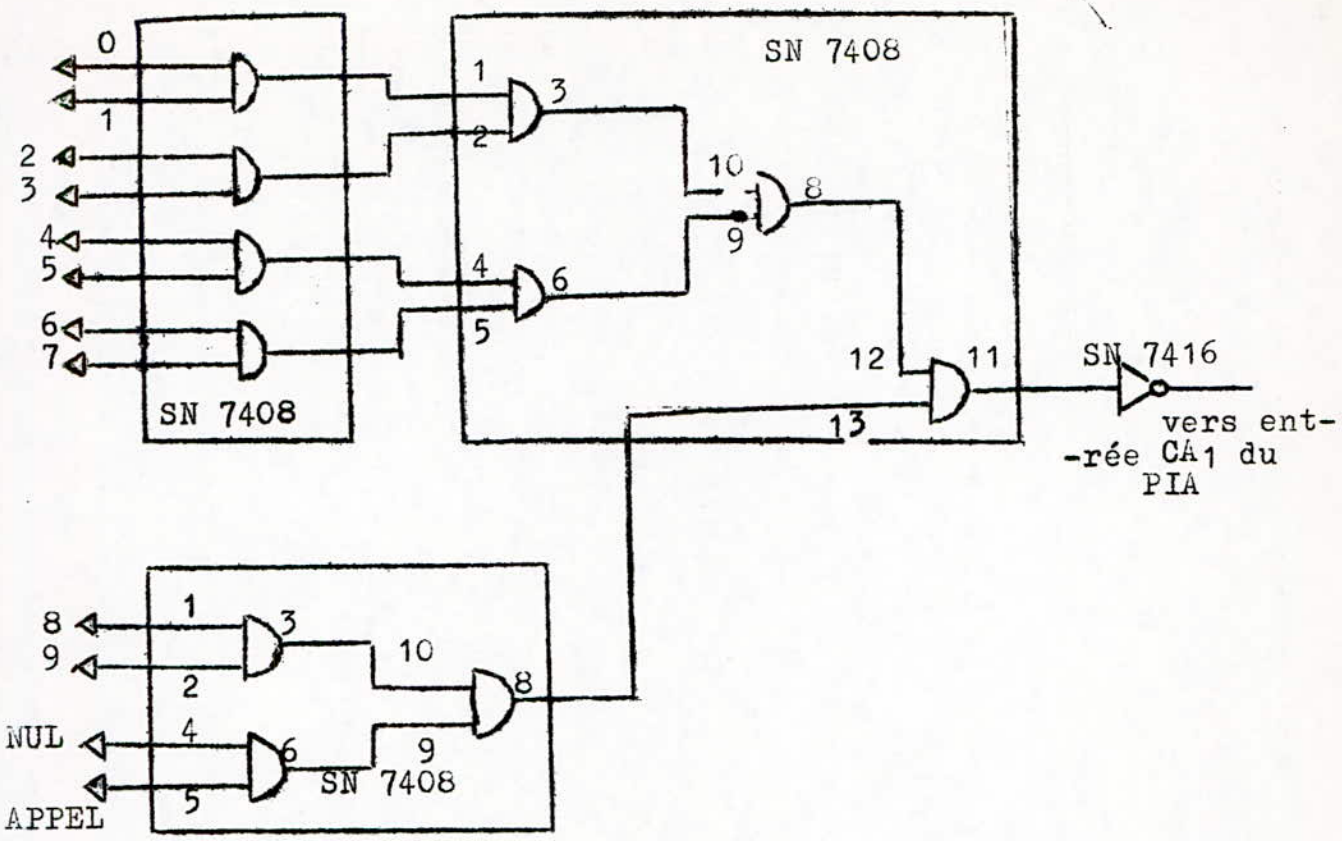


(Le circuit SN 74165 assure la conversion / série à l'aide des registres à décalage qu'il comprend).

NB: Pour plus de détails sur ce circuit (SN 74165) nous prions le lecteur de consulter le DATABOOK TTL.

- Les entrées du circuit 74165 sont au niveau logique 1 par des résistances de rappel. Le fait de fermer un interrupteur place la ligne correspondante au niveau logique ZERO (le logiciel permet la lecture des touches enfoncées). Les touches du clavier sont aussi reliées à une logique de génération d'interruption formée par des CI SN.7408 et SN.7416.

  Ainsi quand une touche est enfoncée, cette logique envoie une impulsion au PIA.



L'enfoncement d'une touche force la ligne correspondante à ZERO. Ceci se traduit par un niveau logique ZERO à la sortie de ce circuit. La demande de l'interruption se fait par une transition de 0 à 1 du signal d'où nécessité du CI SN 7416.

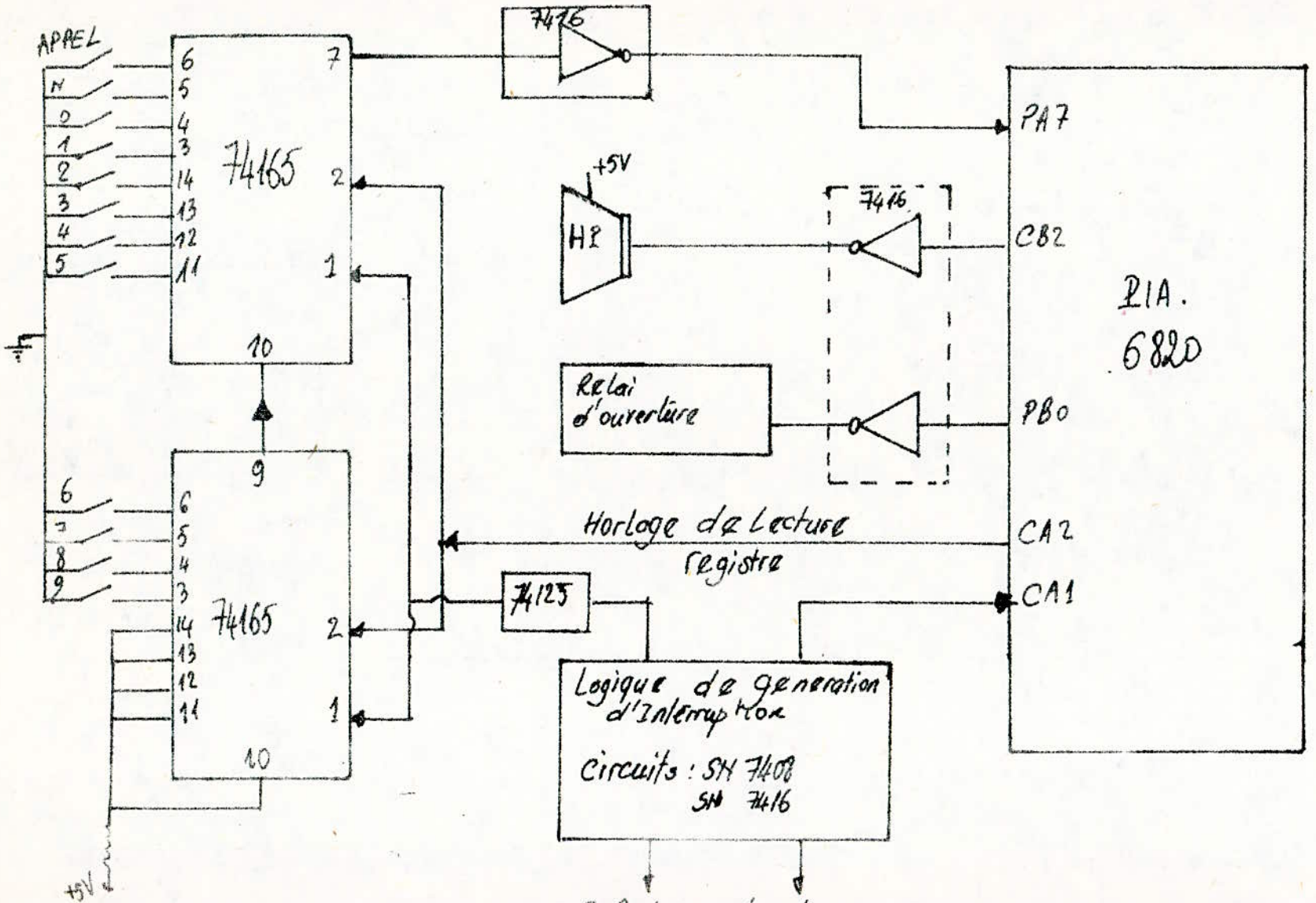
Le chargement des données est validé par un signal émis sur la broche 1 (SHIFT LOAD) par l'intermédiaire du CI SN 74123 connecté directement à la logique d'interruption.

Pour commander la sonnerie , l'alarme et l'ouverture de la porte on utilise des montages inverseurs ou portes à collecteurs ouverts .

Comment programmer le PIA ?

- Une ligne du port A programmée en entrée pour la lecture du code série (ligne P A7).
- CA1 : signal d'interruption du PIA.
- CA2 : Commande de lecture des registres à décalages (CI 74165)
- CB2 : Utilisée pour commander la sonnerie et l'alarme.
- Une ligne du port B (PB0) programmée en sortie pour déclencher l'ouverture de la porte.

.../...


APPEL
N
0
1
2
3
4
5
74165
7416
+5V
HP
7416
Relai d'ouverture
PA7
CB2
PB0
CA2
CA1
RIA. 6820
Horloge de Lecture registre
74125
Logique de generation d'Interruption
Circuits : SN 7408
SN 7416
+5V
Reliés aux touches du clavier.

Logiciel correspondant : (Initialisation du PIA)

```
CLR  CRA   - Programmation lignes
CLR DDRA   - PIA A en entrée.
LDAA =3FH  - CA1 ACTIF BAS-HAUT (+ IRQ)
STAA CRA   - CA2 Sortie niveau programme.
CLR  CRB
LDAA =01   - Un fil PB0 en sortie.
STAA ADRB
LDAB =344  - Mise à ZERO du niveau CB2.
STAB CRB
CLR  ORB   - Mise à ZERO du niveau PB0
RTS        - Retour du sous-programme.
```

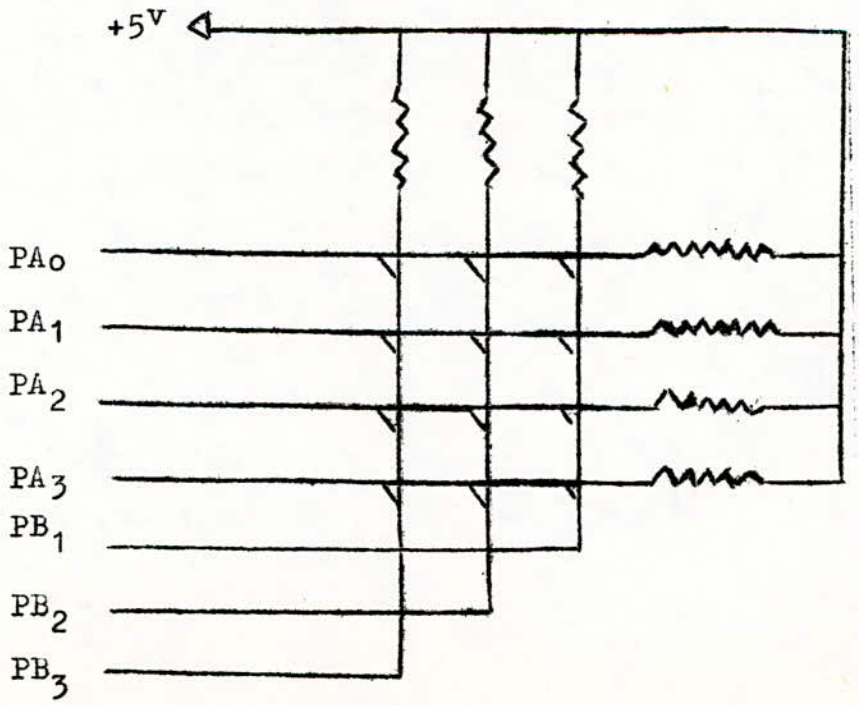
REMARQUE:

Une autre méthode de décodage du clavier peut-être employée. Elle consiste à programmer les lignes du PIA en entrée ou en sortie.

- On relie les lignes (PA0, PA1, PA2, PA3, PB1, PB2, PB3) à des résistances de rappel (5K) pour qu'elles sont au niveau logique "1"

1)- 4 Lignes du port A (PA0, PA1, PA2, PA3) sont programmés en sorti et 3 lignes du port B (PB1, PB2, PB3) en entrée.
  --On applique 0000 sur PA0, PA1, PA2, PA3.
  - En enfonçant une touche (ou vient lire PB1 à PB3 et le bit trouv "0" lors de cette lecture indique dans quelle COLONNE se trouve la touche actionnée.

2)- On programme alors PA0,PA2, PA2, PA3 en entrée et PB1, PB2, PB3 en sorties.
  - On applique 000 sur PB0,PB1, PB2.
  - On vient lire PA0, PA1:,PA2, PA3 et le trouvé à "0" correspond alors à la LIGNE où se trouve la touche enfoncée.
  * Possédant la ligne et la colonne, le programme reconnait qu'elle touche a été actionnée.



LES INTERFACES PROGRAMMABLES SERIE

Nous commencerons d'abord par sensibiliser le lecteur aux problèmes relatifs aux E/S par communication série, celui-ci est censé connaître parfaitement le mode de communication série (synchrone, asynchrone; vitesse de transmission, bit stop, parité, etc...).

Le mode de liaison parallèle convient bien lorsque les distances de transmission sont courtes, mais si celles-ci devaient être longues on préfère souvent passer par une liaison série dans laquelle le mot de donnée parallèle sera converti en mot série, transmis, puis éventuellement reconverti en parallèle évidement on y perd en vitesse mais la voie de liaison est plus simple et plus économique.

La transmission des informations numériques en série sur des distances assez longues peut se faire sur des voies téléphoniques, mais celles-ci, conçûes pour transmettre la voix humaine sur un spectre de fréquence bien déterminé ne sont pas directement compatibles avec la transmission numérique, il faut alors passer par un adapteur assurant une modulation à l'émission et une démodulation à la réception d'où l'utilité et le rôle des MODEMS (contraction de Modulateur-Démodulateur).

Dès lors (vu la diversité de ces circuits) les conditions d'interconnexions entre le micro-ordinateur ou les terminaux, et les Modems ont été appelées à être standardisées et normalisées.

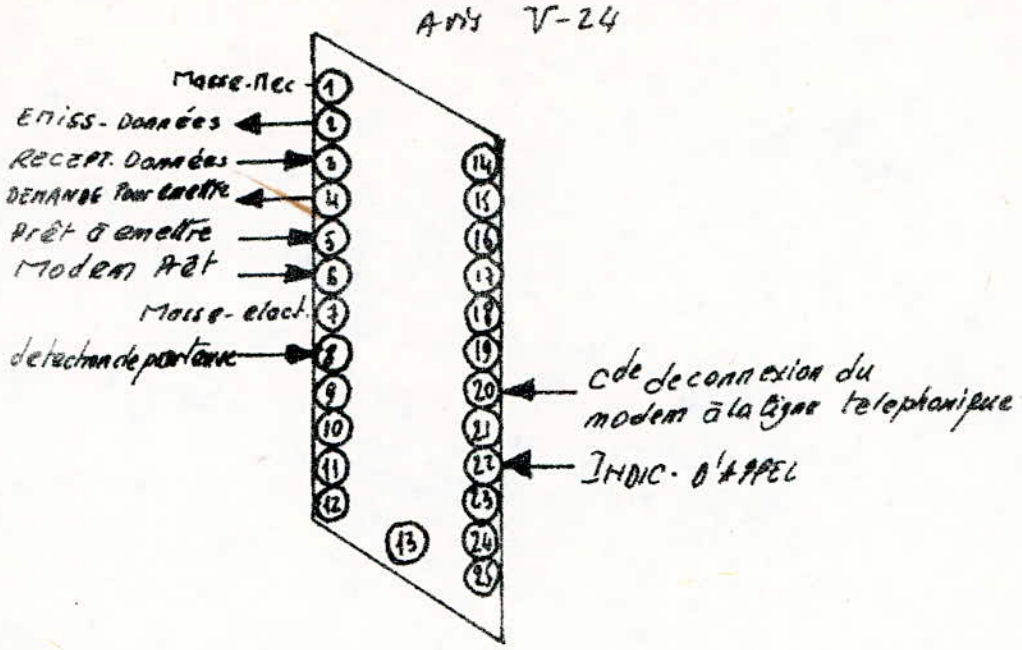
Parmi les normes les plus exploités actuellement
On cite :

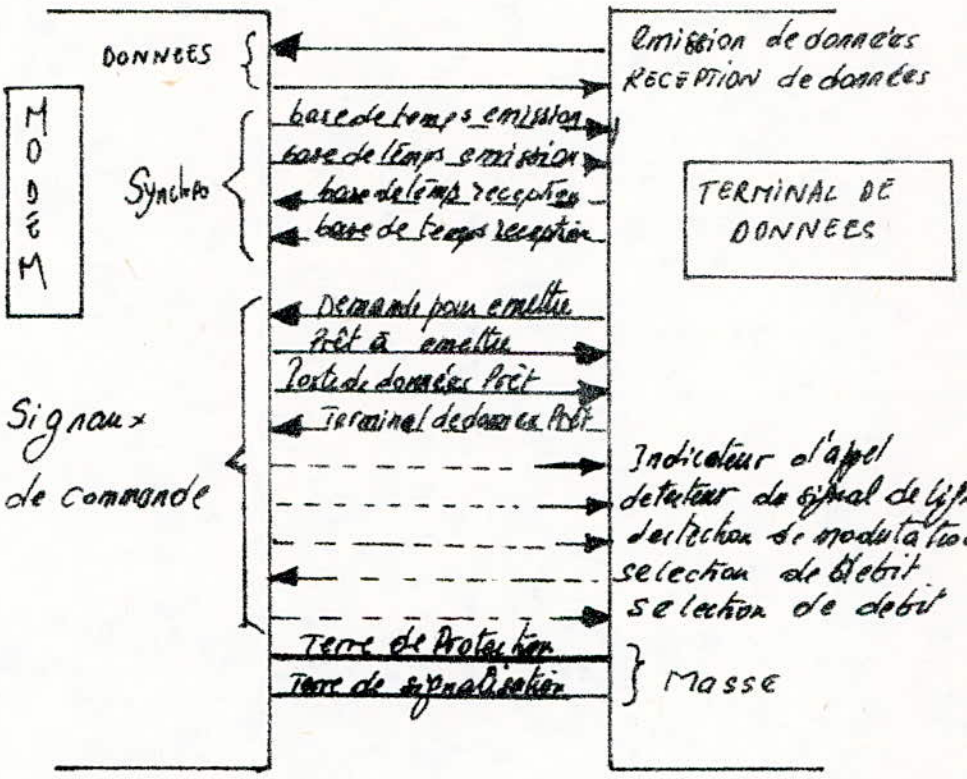
- La RS 232 C de l' EIA (Electrical Industrie Association).
- La V 24 du CCITT (comité consultatif International des téléphones et télégraphes).
- La boucle de courant 20 MA.
-

On donne dans le tableau de la page suivante les normalisations de la V 24 et de la RS 232 C (comparées).

- Voyons maintenant quelques caractéristiques de la norme EIA : RS 232 C :
  Mode : 1 Fil;
  Longueur maximale du câble : 17 mètres;
  Débit maximum : 20 K Bauds;

Connecteur Normalisé
Avis V-24





Quelques Signaux d'interfaçage pour MODEM selon la Norme V-24.

V 24		RS232C	
ref	CIRCUIT	ref	CIRCUIT
101	Terre de protection	AA	protection Ground.
102	Terre de signalisations	AB	Signal Ground.
103	Emission des données	BA	Transmitted DATA.
104	RECEPTION des donnees	BB	received DATA
105	Demande pour emettre	CB	REQUEST to Send
107	poste de donnees Prêt	CC	DATA SET. READY
108/1	connectez le poste de donnees sur la ligne	D.CD	DATA Terminal Ready
108/2	Equipement terminal de donnees prêt.		
125	Indicateur d'Appel	CE	RING. INDICATOR
109	Detecteur de signal de ligne reçu sur la voie de donnees	CF	DATA CARRIER DETECTOR
113	Base de temps Emission	DA	Transmit Signal element TIMING
114	Base de temps Emission	OB	Transmit Signal element TIMING
115	Base de temps Reception	OD	Receive Signal element TIMING.

Tension minimale de sortie en charge : ± 0,5^{V} à ± 15^{V}
Tension maximale en circuit ouvert : ± 25^{V}
Résistance sortie minimale : Ro = 300 SL.
Courant maximal de court-circuit : ± 0,5 A.
Seuil maximal du récepteur : + 3^{V} à - 3^{V}
Tension maximale à l'inter-récepteur : - 25^{V} à 25^{V}

On préfère arrêter à ce stade de sensibilisation de peur de rentre dans des détails subtils. Le lecteur pourra consulter d'autres ouvrages plus spécialisés pour recueillir de plus amples informations (on s'excuse pour ne pas avoir été un peu plus exhaustif).

Revenons maintenant aux interfaces, séries qui sont en général de 3 types
- Asynchrone: pour assurer des echanges asynchrones.
- Synchrone : ce type d'interface série assure des échanges synchrones.
- synchrone-Asynchrone : ce type d'interface assure les 2 types de transmission synchrone et asynchrone.

Ces circuits comprennent généralement des registres internes (état, contrôle, réception horloge, émission horloge, etc...). Certains sont à lecture seulement d'autres à écriture seulement. Ces circuits possèdent des entrées adresse pour la sélection d'un de leurs registres internes et éventuellement une commande pour lire leur contenu ou écrire dedans.

Le Baud-rate (vitesse de transmission), le nombre de bito stop, la parité ou l'imparité, nombre de bits par caractères d'information etc... sont programmables par des mots de commande qu'on envoie au registre de contrôle. Le registre d'état possède plusieurs bits les uns témoignent de l'état vide ou plein de certains registres les autres sont indicateurs des erreurs diverses pouvant subvenir lors de la transmission. Voyons maintenant comment se déroule généralement un transfert E/S par communication série asynchrone.

a) - Emission :

Quand le processeur veut émettre des données (en série asynchrone) à destination d'un périphérique. Il doit s'assurer d'abord que le registre de transmission de l'interface (s'occupant du périphérique) est vide (le processeur peut le faire par programme en testant un bit du registre d'état de l'interface ou bien c'est l'interface lui même qui le lui signale en lui

envoyant une demande d'interruption si son registre de transmission est vide). Si celui-ci est vide le micro-processeur envoie un caractère à l'interface qui désactive, aussitôt l'indicateur "registre transmission vide, effectue le traitement nécessaire et envoie le caractère au périphérique au rythme de l'horloge d'émission.

Une fois le caractère envoyé, l'interface active son signal demande d'interruption ainsi que l'indicateur registre de transmission vide. Le processeur peut à nouveau envoyer un autre caractère et ainsi de suite.

Dans le cas d'un modem l'interface (une fois le caractère reçu et traité) envoit d'abord le signal "RTS" au modem lui signifiant qu'une donnée est prête à lui être transmise, à son tour le modem lui envoie un signal "CTS" (clear to send) lui signifiant "modem prêt", aussitôt l'interface envoie la donnée au modem.

b) Réception :

Analogue à l'émission (seulement dans ce cas c'est le registre de réception qui entre en jeu).

Notons aussi qu'en général un interface série possède :

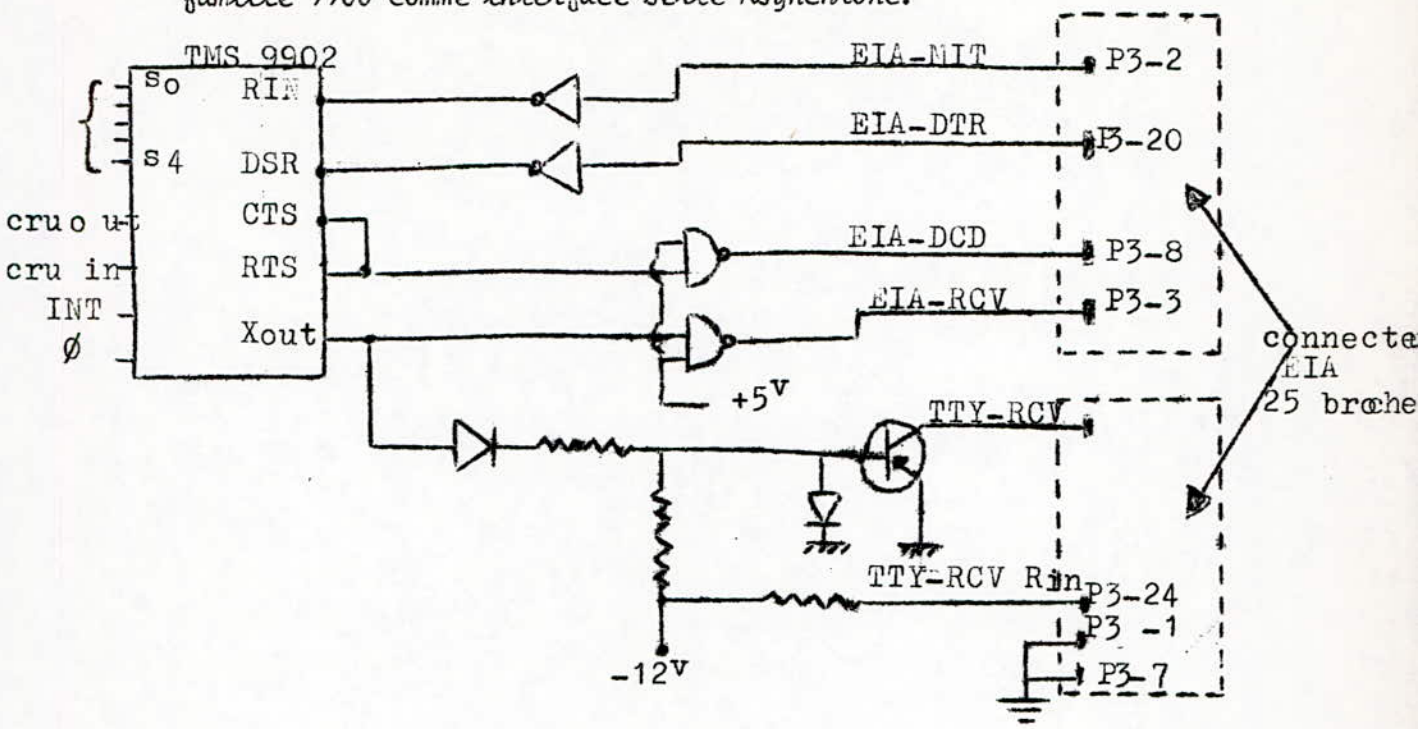
- Des entrées horloges : Une pour l'émission, une pour la réception une pour le seguencement des opérations à l'intérieur de l'interface.

- Une sortie de demande d'interruption (une seule car les interfaces série ne peuvent être connectés qu'a un seul périphérique).

- Et éventuellement une entrée "RESET" de remise à zéro des registres internes.

Les signaux de dialogue avec modem (CTS , RTS , DCD , DSR , DTR et autres qu'on a définit dans les tableaux de normalisation V 24 et RS 232 C) peuvent être utilisés comme signaux de commande ou de dialogue avec un périphérique (autre qu'un modem) par exemple : pour une TTY le signal "RTS" peut servir à la commande du lecteur de bande perforée.

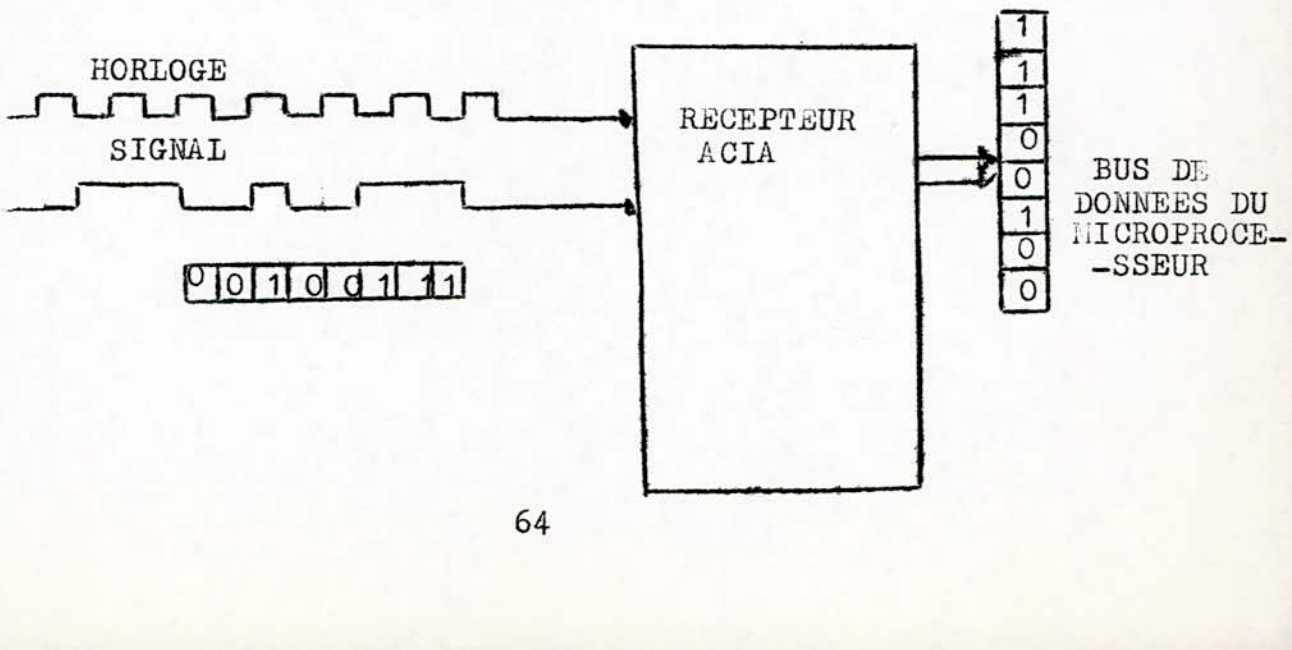
Voyons maintenant cette application du TMS 9902 proposé par la famille 9900 comme interface série Asynchrone.



Le shema ci-dessus est une voie de transmission série établissant la liaison vers le TMS 9902 (ACC) et les circuits interfaces pour ramener les signaux à des niveaux compatibles avec les normes RS.232.C et boucle de courant 20 MA. Le TMS 9902 génère les signaux de cadencement de contrôle, et d'échanges, nécessaires pour émettre des données à destination d'un périphérique (par exp. CRT, imprimante). Il reçoit également des données provenant des terminaux et les met en forme avant de les présenter sur les lignes "CRU".

L' ACIA MC 6850 : (Proposé par la famille 6800)

C'est un récepteur émetteur universel asynchrone, son principe d'utilisation apparait sur la figure suivante :



Ce circuit s'insère dans un système à minoprocesseur permettant ainsi le contrôle d'un périphérique, d'un modem ou d'un télétype.

- L'organisation interne du MC 6850 est ~~donnée dans la page~~ en annexe ~~suivante~~ :

- Le MC 6850 possède un recepteur qui convertit les données séries en un mot parallèle à 8 bits.

- Il gère les bits stops et vérifie la bonne transmission des données grâce à son registre d'état.

- Il possède également un transmetteur qui convertit les données parallèles en une forme série et va ajouter le bit départ pour séparer les mots.

Pour définir les contenus des registres internes de l'ACIA ou peut dresse un tableau comme celui de la page suivante :

Nombre de lignes du bus de données	ADRESSES BUFFERS			
	$\overline{RS}$ R/$\overline{W}$ TDR	RS R/$\overline{W}$ RDR	$\overline{RS}$ R/$\overline{W}$ Registre de contrôle	$\overline{RS}$ R/W Registre d'état
	Ecriture seule	Lecture seule	Ecriture seule	Lecture seule
0	bit de donnée 0	bit de donnée 0	CR0	RDRF B0
1	bit de donnée 1	bit de donnée 1	CR1	TDRF B1
2	bit de donnée 2	bit de donnée 2	CR2	$\overline{DCD}$ B2
3	bit de donnée 3	bit de donnée 3	CR3	$\overline{CTS}$ B3
4	bit de donnée 4	bit de donnée 4	CR4	FE B4
5	bit de donnée 5	bit de donnée 5	CR5	CNRN B5
7	bit de donnée 6	bit de donnée 6	CR6	PE B6
7	bit de donnée 7	bit de donnée 7	CR7	IRQ B7

Interface à un télétype :

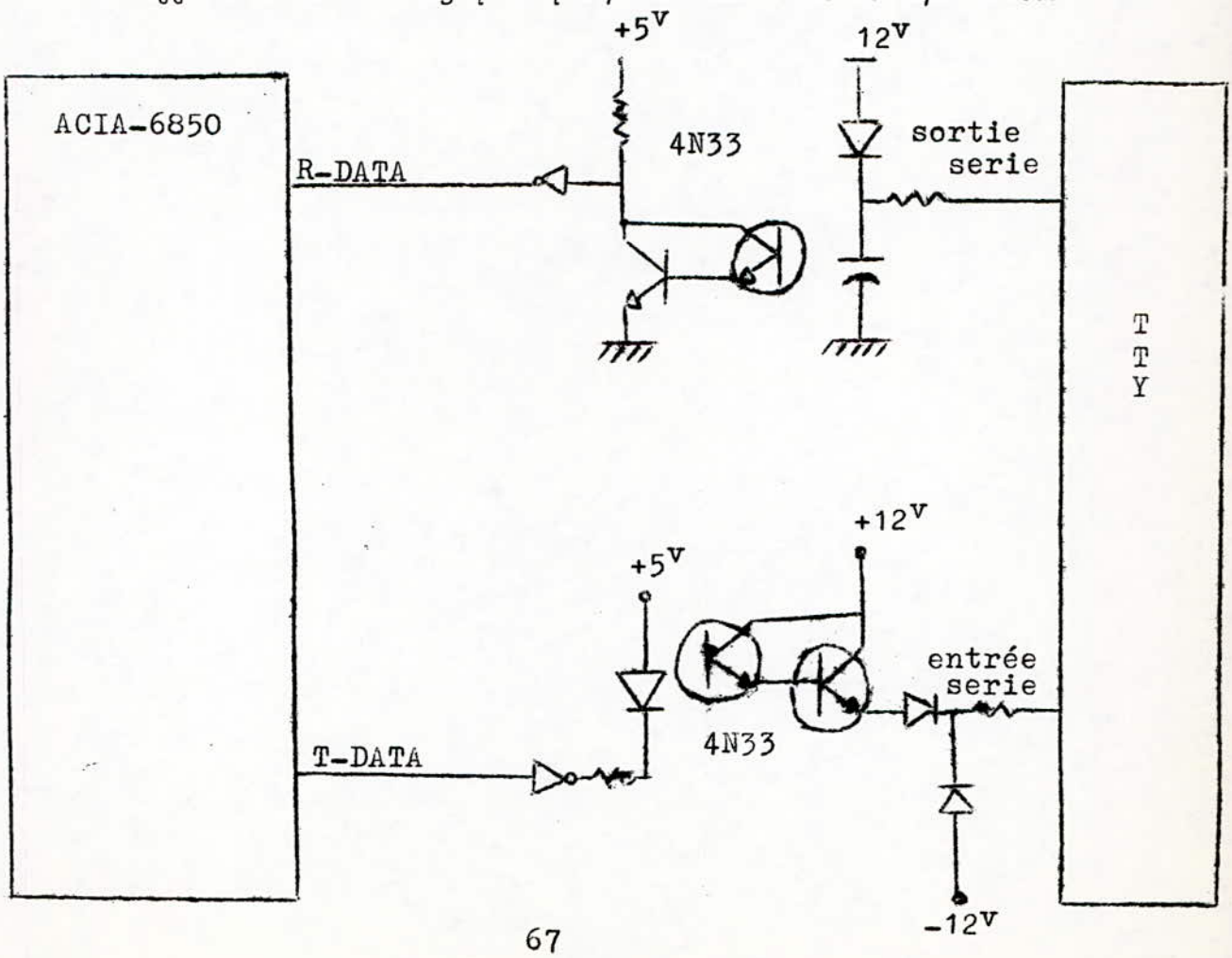
Le télétype (TTY) est un terminal qui sert à la fois à introduire les données et à les éditer grâce à son imprimante. Il dispose aussi d'un lecteur perforateur de bandes pour introduire des programmes longs. La connection au minoprocesseur nécessite un convertisseur série parallèle, ce qui revient à l'utilisation d'un ACIA.

L'interface à un télétype peut se faire suivant deux méthodes :

1) - A C I A - T T Y 33 :

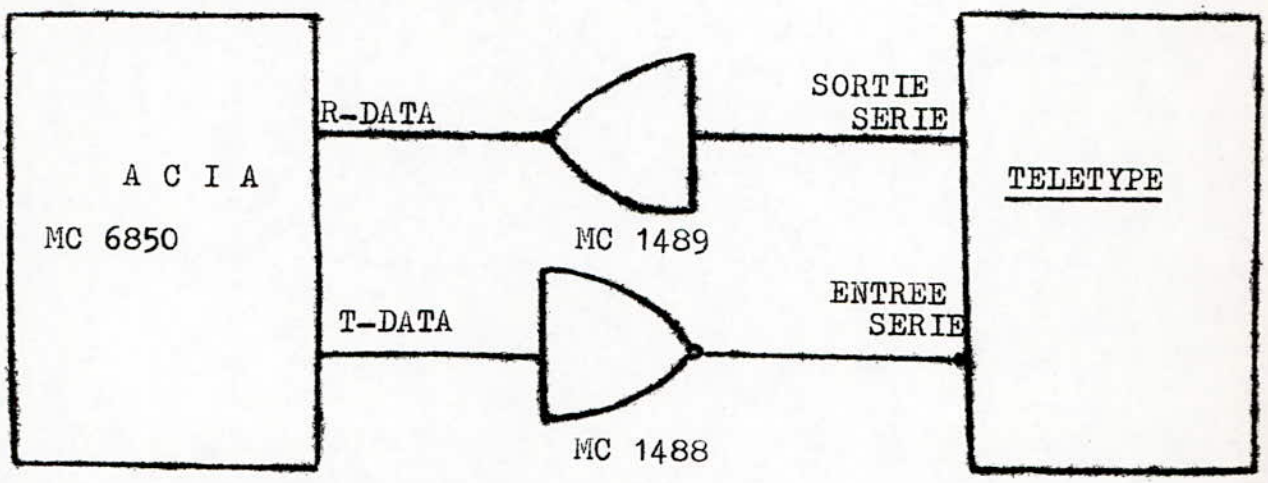
Les lignes entrée série et sortie série du télétype sont normalisées en boucle de courant 20 MA. Par conséquent elles ne peuvent pas être connectées directement aux broches T x DATA et R x DATA de l' ACIA. Pour cela la liaison ACIA - TTY est assurée par un coupleur optique 4 N 33.

Le circuit 4 N 33 permet d'isoler électriquement le télétype du système microprocesseur. Il est utilisé souvent dans l'interfaçage des systèmes de différents niveaux logiques qui pourraient être incompatibles.



3) A C I A - T T Y RS 232-C:

Un autre type de teletype normalisé suivant la RS 232-C PEU être relié à l'ACIA par l'intermediaire des dispositifs d'interfac de la RS 232-C tel que MC 1488 et MC 1489. CES circuits servent à l'adaptation des niveaux logiques.



Comment se deroulent les operations?

Pendant la reception

1) - Le clavier du teletype est actif, le lecteur de ruban est bloqué.
2) - Lorsque le clavier est inactif, l'ACIA peut recevoir les données .
3) - Le teletype doit transmettre un bit de depart pour que l'entr l'ACIA passe de "1" à "0".
4) - L'ACIA memorise la donnée de 8 bits et envoie un signal "données prêtes" au microprocesseur.

Pendant la transmission:

1) - Le microprocesseur charge le buffer de sortie de l'ACIA si celui-ci est libre.
2) - Le buffer de sortie envoie le mot d'information au regitre de transmission de données ou il sera converti en serie pour être transmis au teletype.

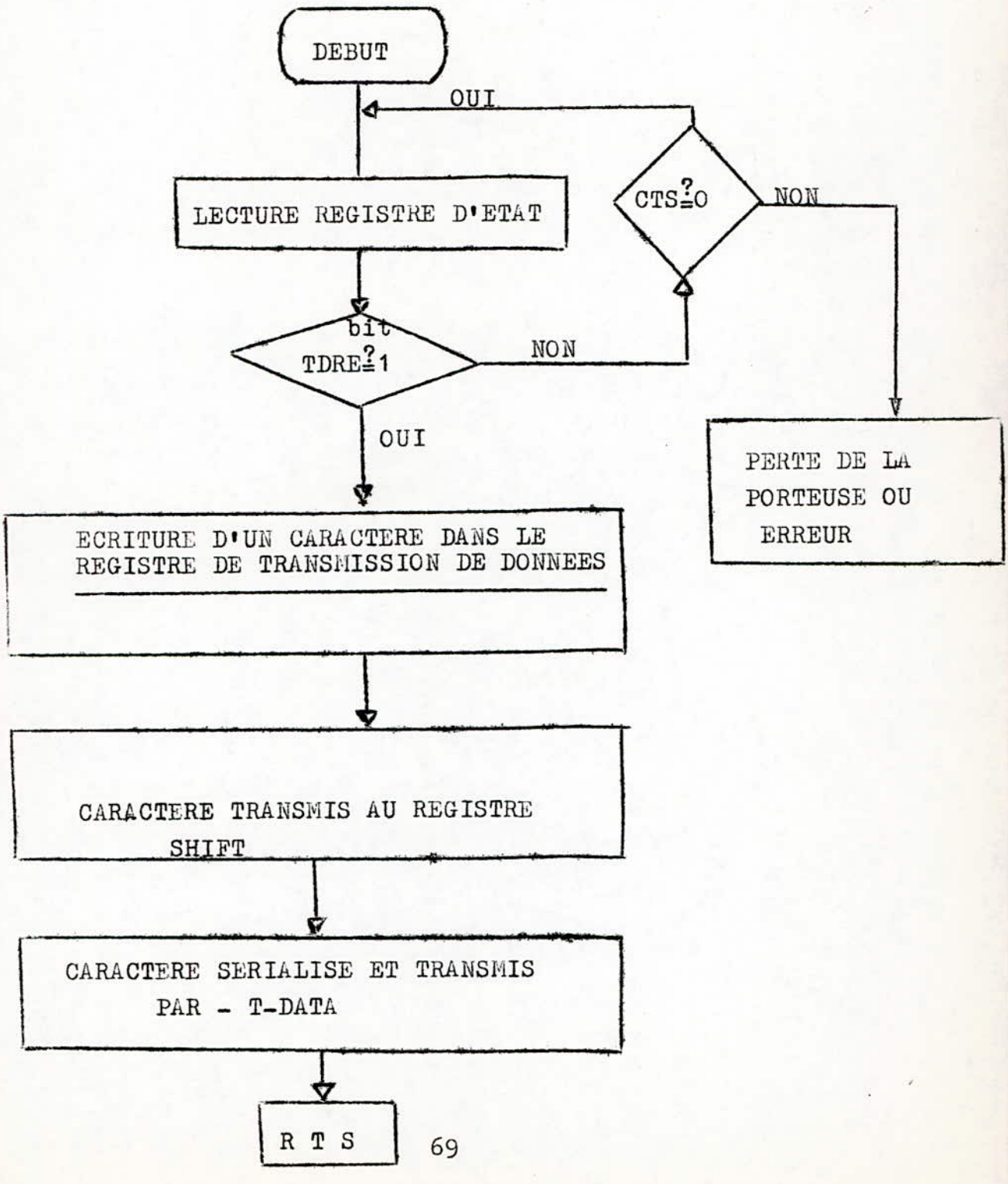
REMARQUE: On peut utiliser un PIA pour la transmission serie lors d'un interfacage d'un TELETYPE.

MOTOROLA propose un exemple d'un interface PIA avec TTY RS 232 C Par l'intermédiaire des circuits 4 N 33, MC 1488 et MC 1489.

Donnons maintenant un exemple de programmation de l'ACIA pendant la transmission et la réception.

Sousprogramme de transmission :

1) - Organigramme :



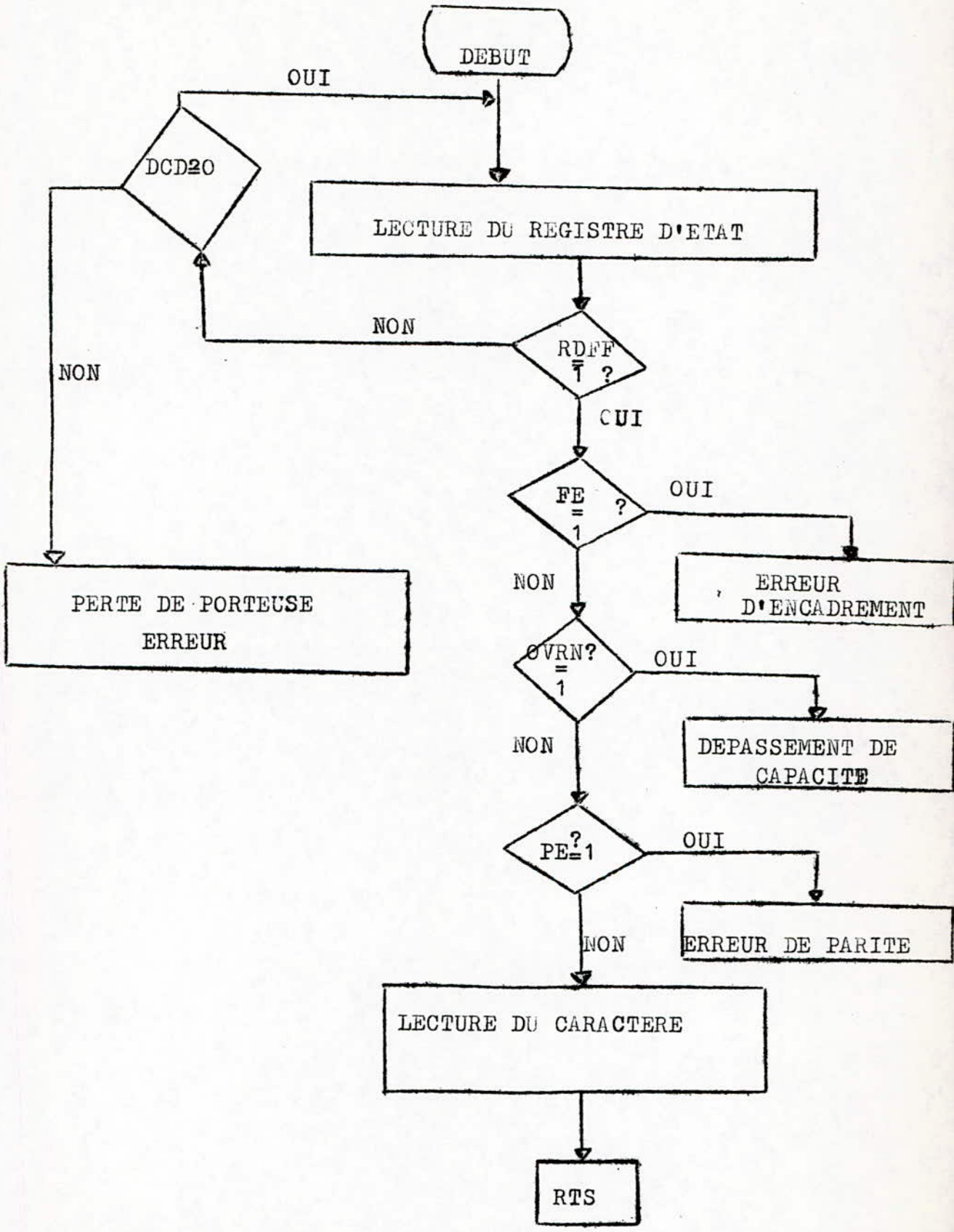
La lecture du registre d'état est nécessaire parce qu'elle permet de déterminer quel périphérique a provoqué une interruption ; de savoir si la donnée est conforme à un format programmé dans le registre de contrôle ; si la parite est bonne.

Programme :

```
NEXT  LDAA  STACON    *  Charger l'accumulateur A par le contenu du
                        registre d'état.
      ASRA            *  Décoler le bit TDRE dans la position du bit
                        carny tester le bit TDRE.
      BCC   NERT      *  Décoler le bit CTS dans la position du bit corn
                        tester le bit CTS.
      BR    ERROR 1   *  Perte de la porteuse on erreur.
TxDATA STAB TxRx      *  Stocker le caractère dans l'ICACIA.
  RTS                 *  Retour du som. programme.
```

Sous programme de réception :

Organigramme :



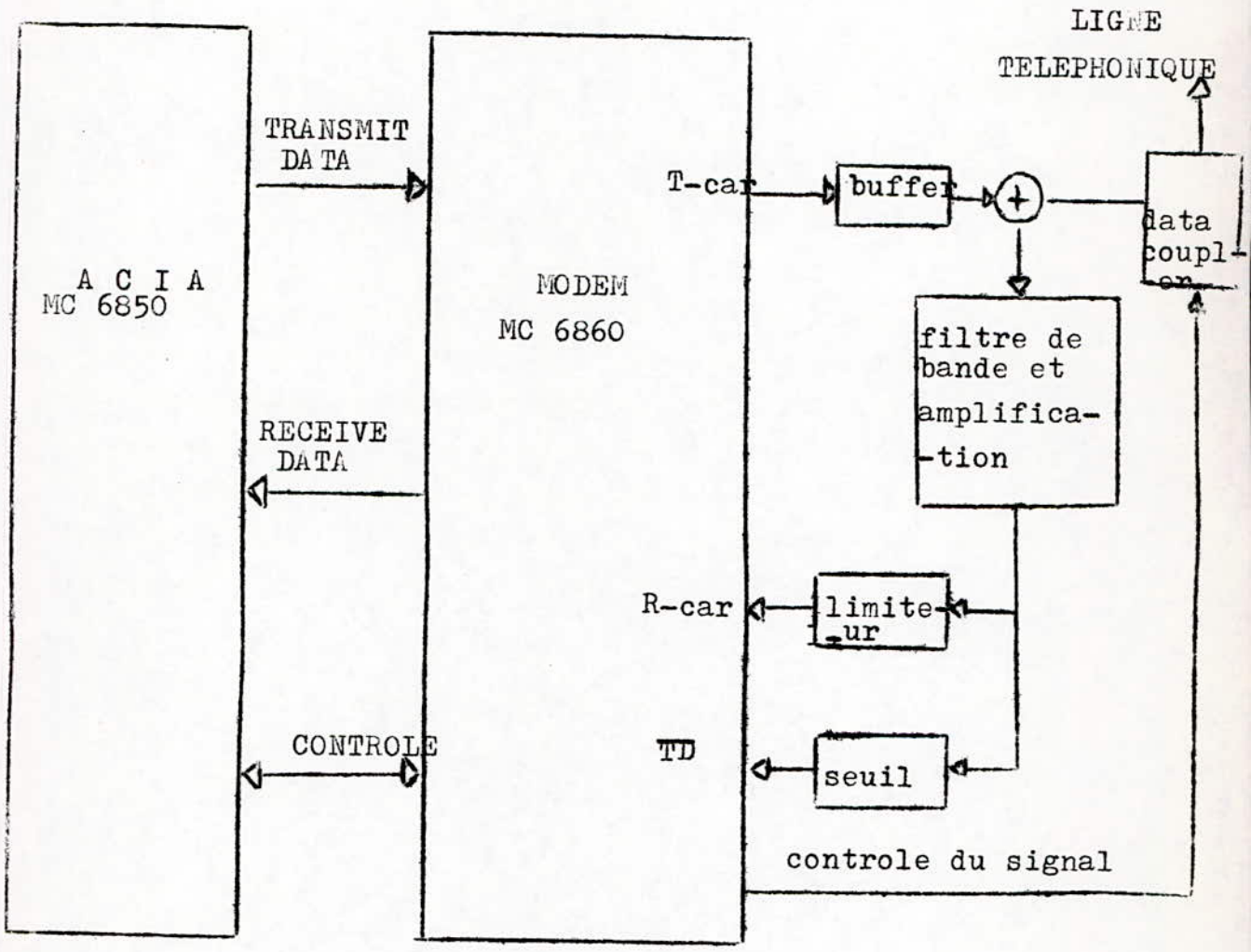
Programme :

```
NEXT 1    LDAA STACON    Charger l'accumulateur A avec les contenus du
                         registre d'état.
          ASR A          Décaler le bit RDRF dans la position du bit corny.
          BCS FRAM       Tester le bit RDRF.
          ASRA           Décaler le bit DCD sous la position du bit cony.
          BCC NAXT 1     Tester le bit DCD.
          BR ERROR2      Perte de porteuse ou erreur.

FRAM      ASRA
          ASRA           Décaler le bit FE dans la position du bit cony.
          BCC OVRN       Tester le bit FE.
          BR ERROR 3     Erreur d'encadrement.

OVRN      ASRA           Décaler le bit OVRN dans la position du bit CONY
          BCC PAR        Tester ce bit OVRN.
          BR ERROR 4     Dépassement de capacité.
PAR       ASRA           Décaler le bit PE dans la position du bit CONY.
          BCC RDATA      TESTER le bit PE.
          BR ERROR5      erreur de parité.
          LDAB Tx Rx     Charger l'accumulateur B avec le contenu du regist
          RTS            de reception de l'ACIA
```

Le MC 6800 système peut communiquer à distance par l'intermédiaire des lignes téléphoniques en utilisant l'ACIA MC 6850 et le modem MC 6860.

Le MC 6860 fonctionne en FM jusqu'à 600 bits/S. Le signal FM (émis par T x DATA) est filtré par un filtre passe bas (buffer de sortie) pour passer à la ligne téléphonique via la coupleur de données.

A la réception, le signal FM sera filtré et amplifié (filtre passe bande PLUS amplification) pour éliminer les signaux parasites.



CHAPITRE IV

LES TRANSFERTS D'E/S PAR ACCES DIRECT A LA MEMOIRE

L'accès direct à la mémoire est une technique d'E/S utilisée pour les échanges très rapides d'informations entre les mémoires et certains peripheriques (par exemple un lecteur de Floppy disk).

Dans les 2 techniques précitées (E/S par programme et E/S par interruption) le microprocesseur centralisait et surveillait lui-même tout transfert E/S limitant ainsi leséchanges de données. Dans cette technique le microprocesseur n'intervient guère et s'arrête dès qu'il reçoit une demande de transfert par DMA, des dispositifs spéciaux sont alors utilisés pour superviser ces échanges. Ces dispositifs se substituent au microprocesseur en prennant le contrôle des 3 bus du système, ils doivent alors générer les signaux de commande, de contrôle et d'horloge nécessaires pour la réalisation de ce genre de transfert.

Les E/S par DMA se font généralement suivant 3 procedures :

1) DMA par arrêt du microprocesseur :

Le circuit désirant effectuer un transfert d'E/S par cette technique envoie une "demande de DMA" au microprecesseur, qui sitôt après avoir reconnu la demande termine l'execution de l'instruction en cours et s'arrête (cessant toute activité interne) en envoyant alors un signal de reconnaissance (DMA ACKNOW LEDGE) au système demandeur. Ce dernier peut donc prendre en main le contrôle des bus pour effectuer ses transfertsaussi longtemps qu'il le voudra. Cette technique est généralement la plus utilisée.

2) DMA par vol de cycle.

Dans cette procédure, le microprocesseur ne s'arrête que pour des périodes relativement courtes puis il reprend automatiquement son fonctionnement normal. Dès qu'une demande de DMA lui parvient (signal asynchrone) le système attend que ses horloges soient en position adequate pour ordonner un blocage (signal synchrone) pendant une courte durée (4,5 µS au maximum par besoin d'aller rafraichir certains registres dynamiques) après quoi les horloges sont deploquées et le microprocesseur poursuit alors son travail normalement. Ce n'est au fait qu'un rallongement des cycles machines du processeur. Cette procédure est evidemment un peu plus complexe que la précedente et n'est employée que pour des transfertsde courte durée.

3) DMA par multiplexage.

Cette procédure est offerte par certains microprocesseurs seulement. On conçoit que si le microprocesseur travaille en cycle mémoire sur unephase de l'horloge, sur la phase suivante, il doit necessairement travailler en interne ce qui donne alors la possibilité de prendre le controle des bus du système pendant ces brefs instants.

En procedant par multiplexage ou peut donc effectuer des transferts par DMA. Cette procédure est de loin la plus compliquée parcequ'elle neccessit une synchronisation très rigoureuse et des mémoires très rapides.

Revenons maintenant aux circuits et aux dispositifs permettant de contrôler et de superviser des transferts par DMA.

L'exemple ci-dessous illustre d'une manière typique un controleur de DMA réalisé à l'aide de circuits integrés TTL (voir figure).

Un système microordinateur envoie les résultats de traitement de programmes dans une RAM (MCM 6810A) à des adresses bien determinées. SI on désire afficher ces résultats (à l'aide de 8 afficheurs 7 segments) sans faire appel au microprocesseur on procedera alors par DMA.

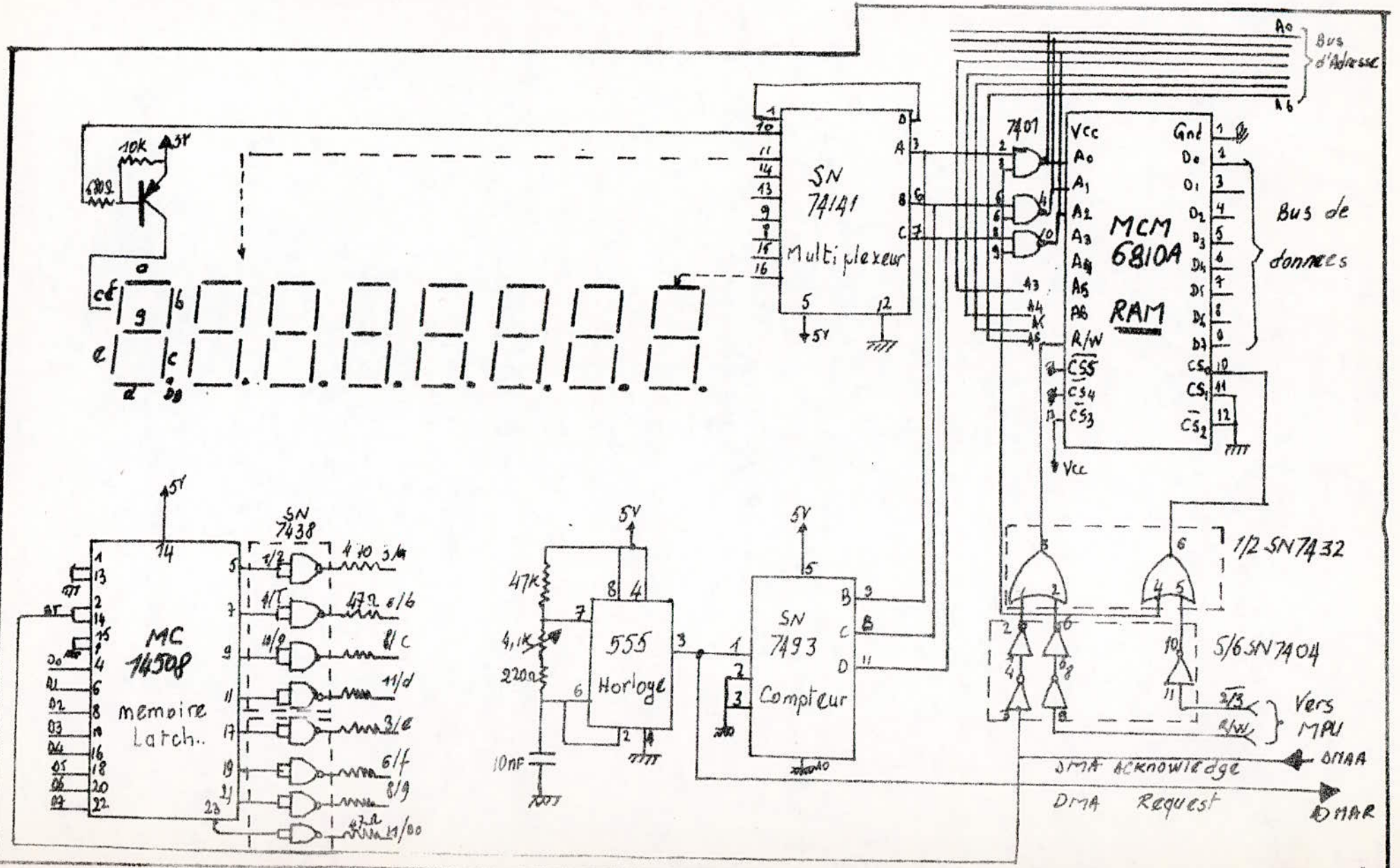
DESCRIPTION DE CE CONTROLEUR DE DMA. (voir figure de la page suivante)

Le circuite 555 est une horloge activant le compteur SN 7493 et la demande de DMA (DMA REQUEST).

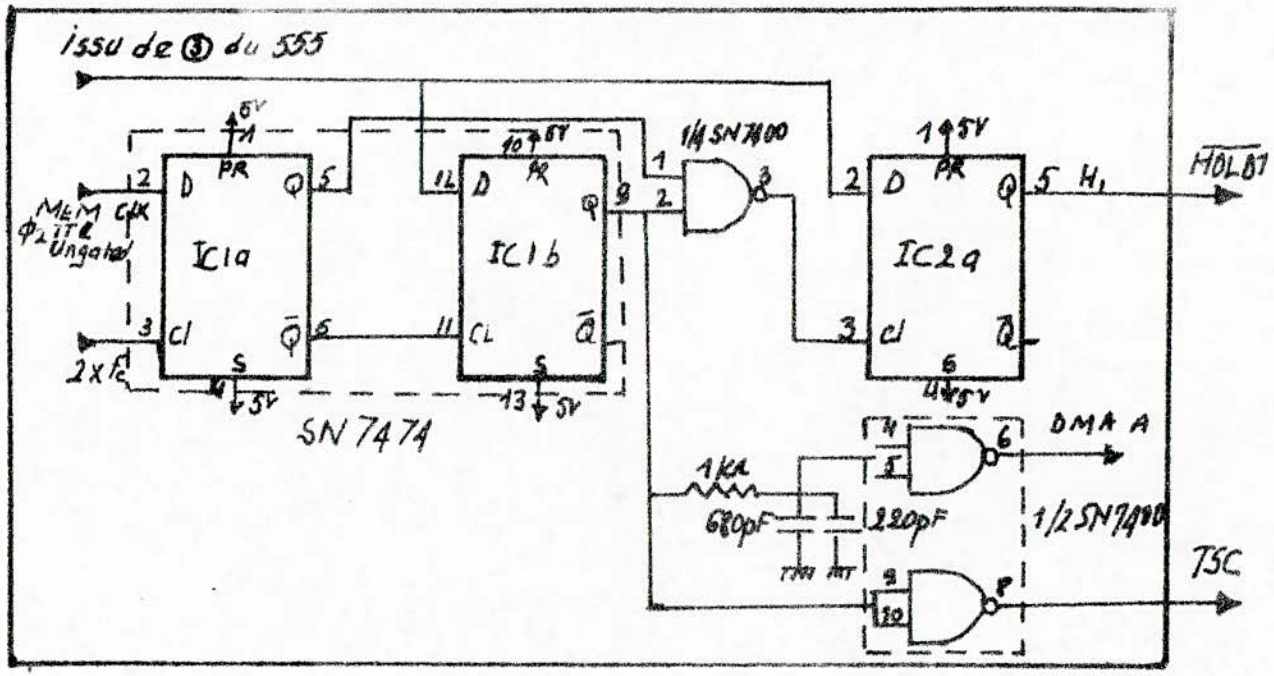
Le DMA AKCNOWLEDGE à son état actif ordonne la lecture de la RAM.

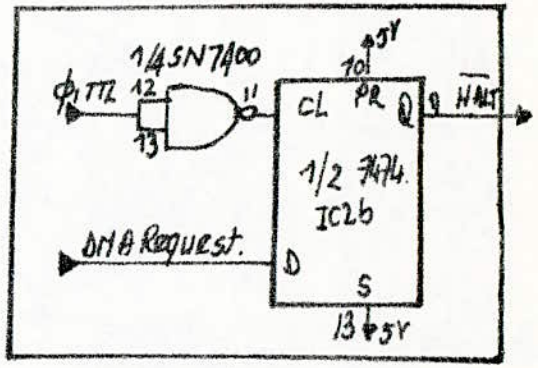
Le compteur SN 7493 s'incremente chaque fois que l'horloge lui envoie une impulsion surson entrée en délivrant sequentiellement sur ses sorties BCD une adresse qui sert à selecter un afficheur parmi les 8 dont on dispose (on utilise un multiplexeur SN 74 141 qui selecte un seul digit à la fois sachant que les segments des 8 digits sont communs).

DMA PARTIE COMMUNE

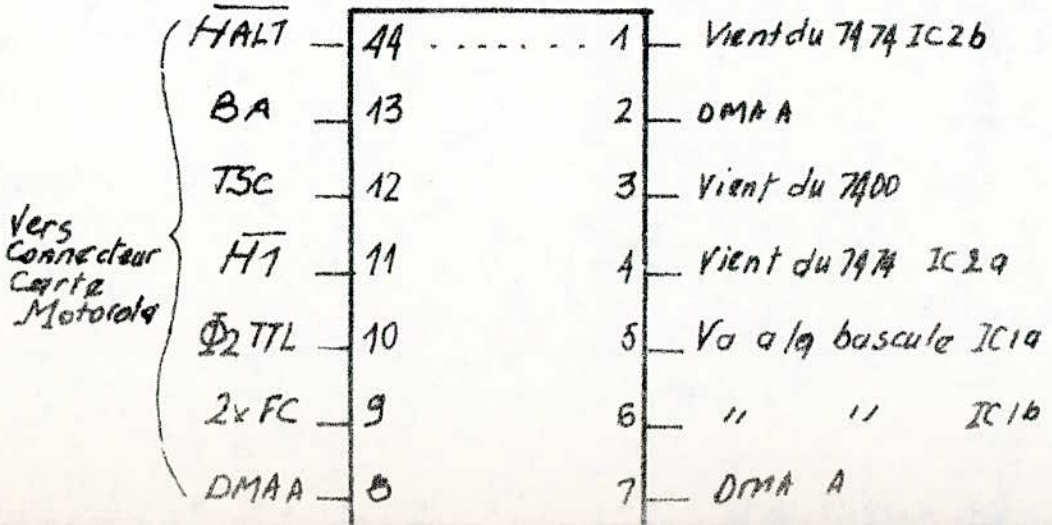


Génération de TSC.





Génération de HALT



Connections

1- Connection pour l'accès direct asynchrone : ($\overline{HALT}$)

Relier : 1 à 14 et 2 à 13

2- connection pour l'accès par vol de cycle :

Relier : 3 à 12 ; 4 à 11

5 à 10 ; 6 à 9

7 à 8

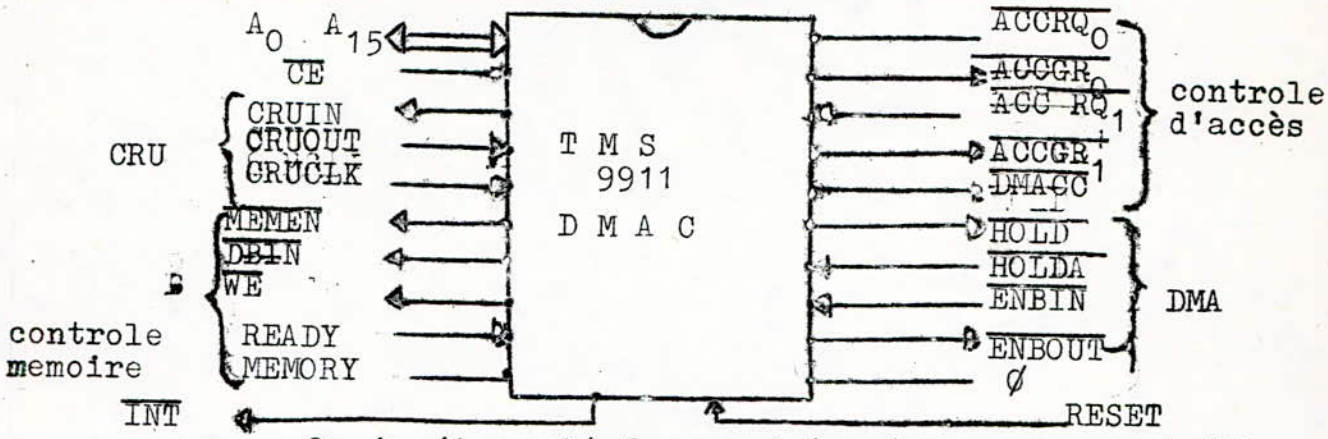
A l'aide des portes 74 141 le signal de commande d'écriture, les adresses memoires parviennent en même temps, le contenu de la position memoire sollicitée à cet instant se trouve present en sortie de la RAM (MC 68101A) qui est alors chargé dans la memoire LATCH (MC 14508) pour etre affiché sur le segment aiguillé à cet instant.

Maintenant si on désire procéder par arrêt du microprocessuer on doit génèrerun signal $\overline{HALT}$ (HOLD ou bien un signal T.S.C si on veut effectuer une DMA par vol de cycle, car le microordinateur utilisé est assemblé autour du microprocesseur MC 6800 de Motorola.

[Le MC 6800 reconnait une demande de DMA par arrêt lorsqu'il reçoit une demande á travers son entrée HALT et par vol de cycle lorsqu'il reçoit une d emande à travers son entrée TSC (Three State controle).]

Les constructeurs proposent aussi un seul circuit intégré (controleur de DMA) qui comprend plusieurs canaux (chaque canal pourra être reservé à 1 periphérique communiquant avec le système par DMA) Ce circuit comprend tous les signaux de contôle memoire, controle d'accès, de demande et de validation de DMA, et evidement les bus d'adresse et de données. Il comprend aussi des entrées horloges, des registres internes (pour le controle de la priorité des canaux, le contrôle des interruptions, l'inscription de l'adresse memoire de depart, le comptage, etc...), et d'autres dispositifs spéciaux que nous allons voir tout de suitedans les 3 circuits "controleur de DMA" que proposent les familles (8080. 6800 9900).

LE TMS 9911 DMA controller proposé par la famille 9900.



Ce circuit possède 2 canaux independants avec une priorité automatique canal 0 > canal 1 chaque canal possède 2 registres le MAR et le LAR (repect memc ıdresse register et last adress register) de 16 bits chacun le MAR contient l'adresse de la position memoire à laquelle va acceder le canal durant le prochain cycle memoire, il s'increnante automatiquement après chaque accès memoire et son contenu ext comparé à celui du LAR si la comparaison est vraie (MAR=LAR) W.bit d'état est positionné et une interruption est activée indiquant ainsi au processeur que le bloc de données a été transferé.

ACCRQo (ACCRQ1) est une demande d'accès par le dispositif de DMA connecté au canal 0. (1), ACCGRo (ACCGR1) est la réponse au dispositif lui indiquant que sa demande d'accès est autorisée. INT est actif quand chaque canal a transferé le nombre spécifié de mots (ou bytes) de données.

DMACC est actif quand les bus du système sont sous controle DMA.

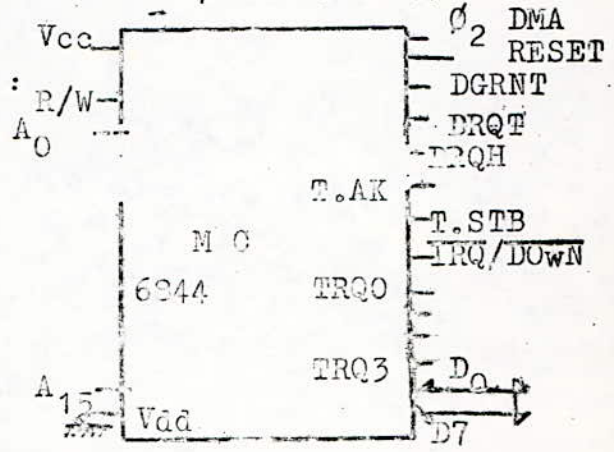
Avant chaque transfert par DMA le TMS 9911 doit être initialisé en selectant le canal qu'on desire utiliser (Quand plus de 2 canaux sont necessaires dans le système an pourra utiliser plusieurs TMS 9911 avec une priorité établie en utilisant les signaux ENBIN et ENBOUT) eten établissant le mode et la procédure opérationnelle de ce canal (chaque canal peut travailler suivant les 3 procédures de DMA et suivant 2 modes (octet ou mot).

Dans la figure ci-dessous le lecteur pourra voir la connection du TMS 9911 avec certains dispositifs DMA notamment le TMS 9909 Floppy disk controller dans un système assemblé autour du TMS 9900.

On note l'utilisation de 2 controleurs de DMA (DMAC1, DMAC2) DMAC1 est le prioritaire son entrée ENBIN est reliée à l'état actif par cablage le DMAC2 ne pourra être autorisé à effecteur un transfert que si le DMAC1 est au repos (car l'entrée ENBIN du DMAC2 est reliée à la sortie ENBOUT du DMAC1).

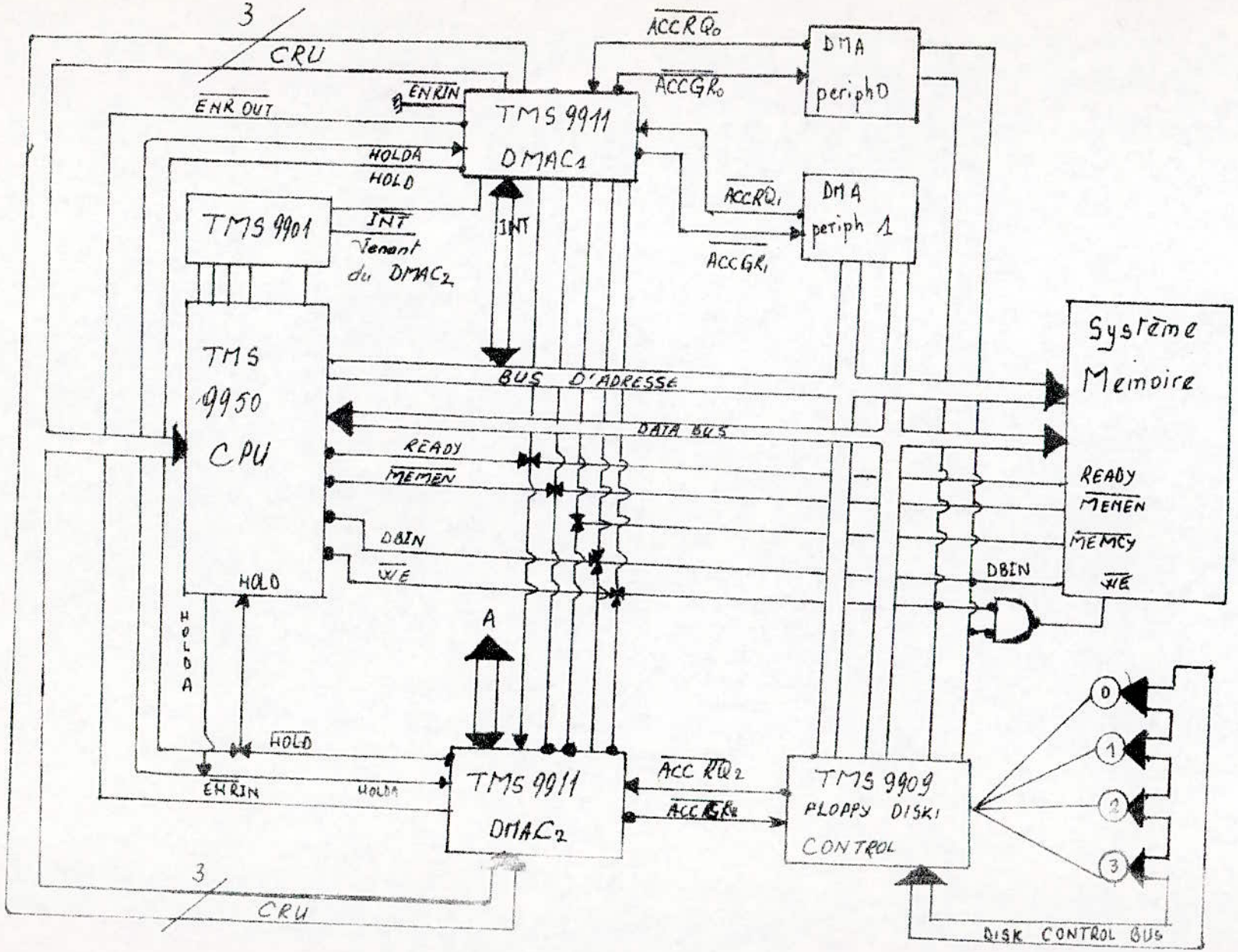
Remarquons aussi que les sorties interruptions des 2 DMAC sont reliées aux entrées interruption du TMS 9901 (Interface //) ont nous avons précedemment parlé) ceci nous éoitera d'utiliser des circuits supplémentaires pour gerer, vectoriser et hierachiser les interruption des DMAC.

LA FAMILLE M6800 propose le MC 6844 :



Le DMAC 6844 de motorola permet la gestion de 4 périphériques différents à partir de 4 canaux chaque canal pouvant travailler suivant les 2 modes (HALT, Cysteal, Multiplex), il possède également 15 registres adressables chaque canal possède un registre (16 bits) d'adresse et 1 registre (16 bits) compteur d'octet. 3 registres de controle [illegible] interruption, de chainage de données). Pour initialiser le canal pour 1 transfert on doit charger 1 registre d'adresse par l'adresse de départ (du champ Memai) et son registre octets par le nombre d'octets à transferer.

Le registre de controle doit aussi être programmé pour établir la direction du transfert, le mode, l'incrementation on la decrementation après chaque cycle.


3
CRU
ENR OUT
ENRIN
HOLDA
HOLD
TMS 9901
INT
Venant
du DMAC2
TMS 9911
DMAC1
ACCRQ0
ACCGR0
DMA
periphD
ACCRQ1
ACCGR1
DMA
periph 1
INT
TMS
9950
CPU
HOLD
HOLDA
BUS D'ADRESSE
DATA BUS
READY
MEMEN
DBIN
WE
Système
Meinoire
READY
MEMEN
MEMCY
WE
DBIN
A
HOLD
ENRIN
HOLDA
TMS 9911
DMAC2
ACC RQ2
ACC GR2
TMS 9909
FLOPPY DISK
CONTROL
0
1
2
3
DISK CONTROL BUS
3
CRU

établit

Le registre de controle de priorité une priorité fixe ou rotative il autorise la demande de transfert emise par un dispositif (periph) DMA ainsi, un peripherique desirant effectuer un transfert DMA envoi la demande au DMAC (6844) à travers le Txrequest si cette demande est autorisée et se trouve prioritaire le DMAC génère au µp demande de transfert celui ci envoi l'autorisation au DMAC qui sitôt l'avoir reçu, avise le periph demandeur et commence à transferer les données. Quand le compteur d'Octets se trouve à Zéro le DMAC emet le signal END au periph et le signal END au periph et le signal DRQH au l'avisant que le bloc de données à été transmi .

La famille 8080 propose le 8257 (INTEL).

Ce circuit possède 4 canaux CCH0, CH1, CH2, CH3) l'autorisation d'accès à un canal donné s'achemine à travers DRQ et l'autorisation d'accès à travers DACK.

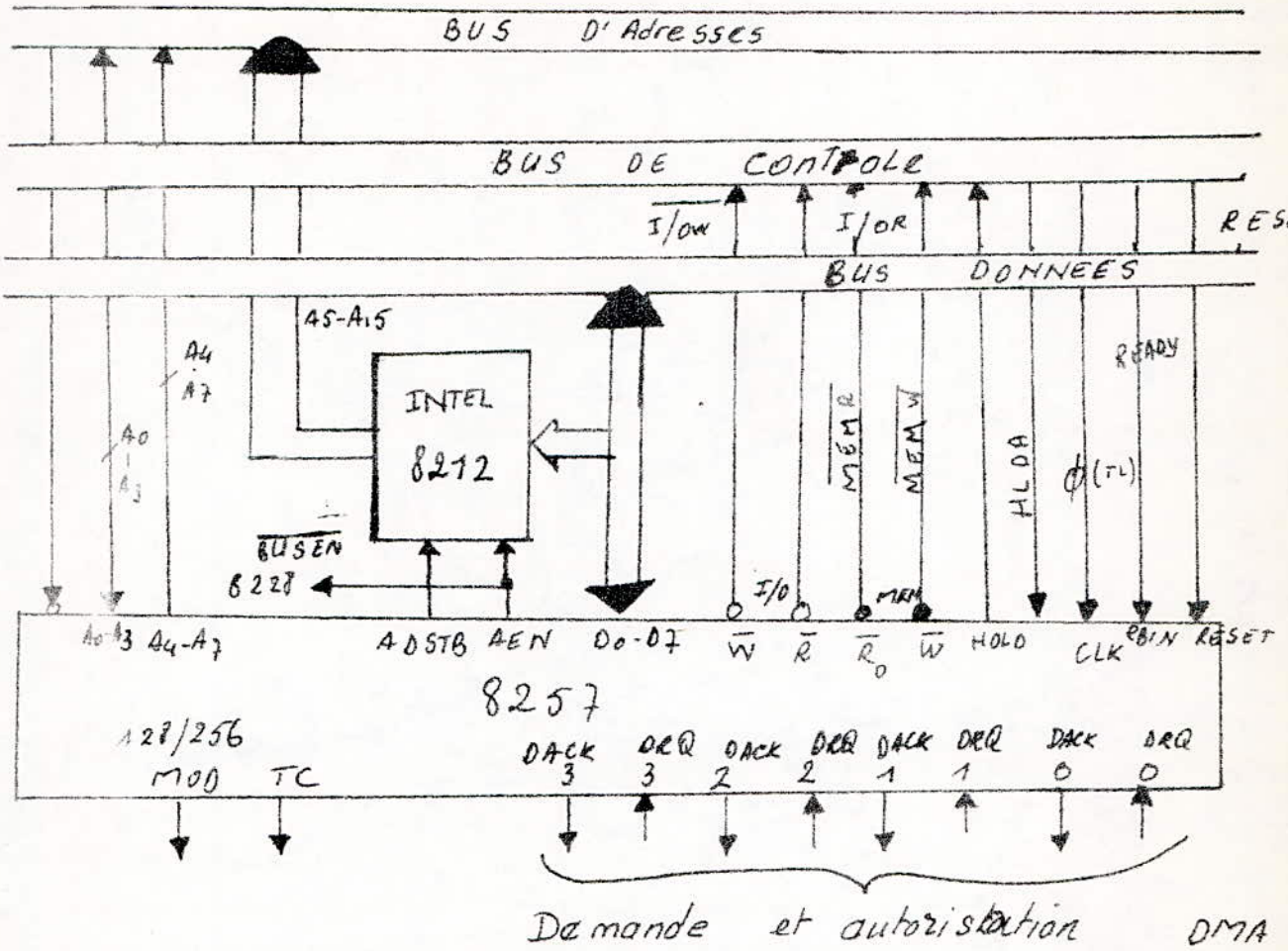
Le diagramme bloc de ce circuit est donné ~~dans la page suivante~~ en annexe on y retrouve essentiellement :

-1 buffer du bus de données

-1 logique de lecture:ecriture regroupant les signaux : I/OR, I/OW respectivement pourun transfert des memoires vers les dispositifs DMA et de ces dispositifs vers les memoires, une entrée horloge CLK pour sequencer les opérations internes du circuit, une entrée RESET pour l'initialisation et la remise à zéro des registres du circuit , des entrées adresses A0, A1, A4 A3 pour la sélection d'un registre interne et une entrée $\overline{CS}$ par la validation du circuit.

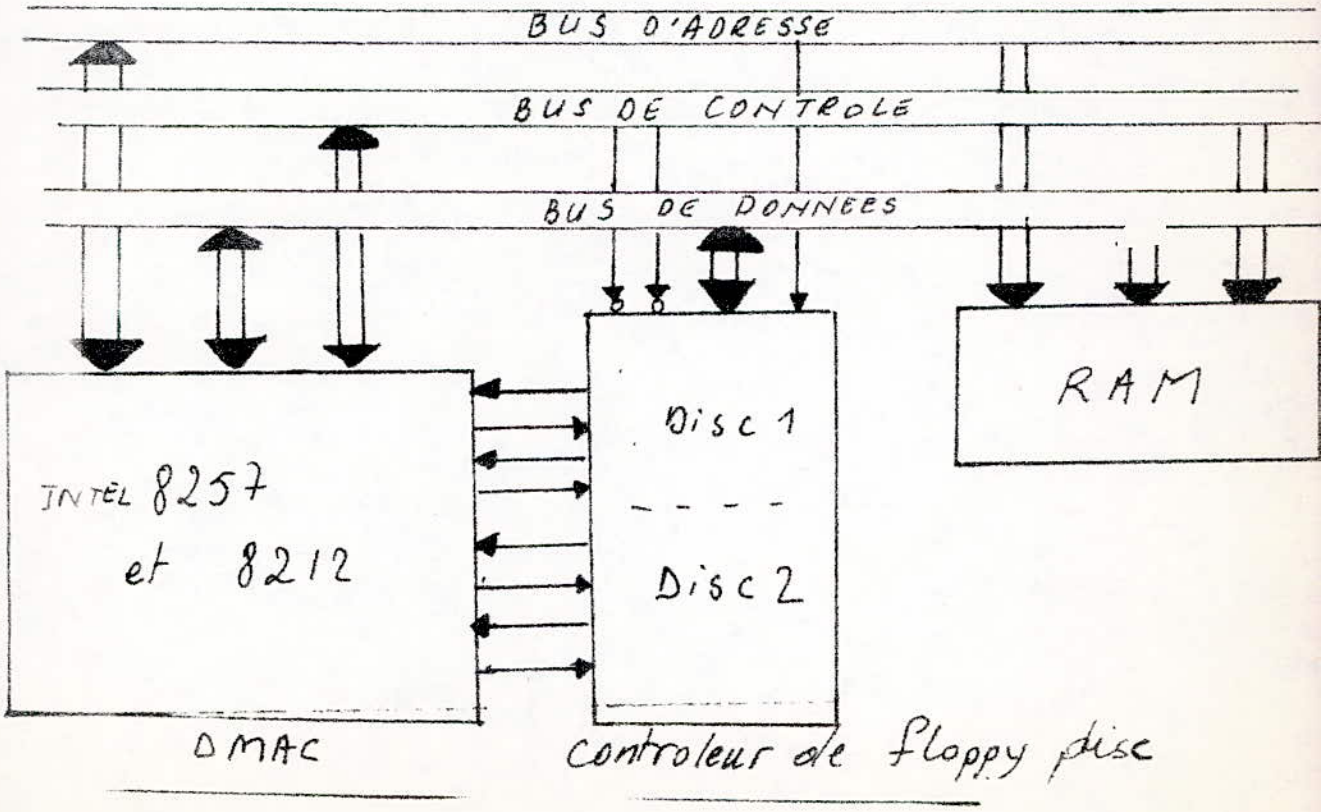
1 logique de controle regroupant les bits A4, A5, A6, A7 du bus d'adresse les signaux de controle memoire READY, $\overline{MEMR}$, $\overline{MEMW}$, les signaux HOLD et HOLD AKNOWLEDGE et d'autres signaux de controle que le lecteur pourra identifier en consultant la documentation fournie par le constructeur.

Les shemas de la page d'après montrent une façon de connecter le DMAC 8257 aux bus d'un système (Fig 1) et à un controleur de Floppy disk (Fig2).



Demande et autorisation DMA

- fig 1 -



- fig 2 -

CHAPITRE V

INTERFACAGE D'ENVIRONNEMENTS ANALOGIQUES

Avant d'aborder cette partie il est préferable de présenter et definir tous les circuits necessaires à l'interfaçage de ce type d'environnement comme les echantillonneurs/bloqueurs, les CN/A, les CA/N, les multiplexeurs analogiques, les capteurs etc....
On donnera brièvement leurs caracteristiques et les critères de leur choix necessaire au concepteur pour la comprehension des notices techniques.
On exposera enfin leur mode de fonctionnement et leur utilité à travers des exemples.
Le traitement des données analogiques d'un phénomène physique quelconque par un microcalculteur necessitela mise en place d'une chaine d'acquisition de données, celle-ci se compose généralement de :

- Capteurs;
- Echantillonneur/bloqueurs
- Filtres
- Amplificateurs
- Multiplexeurs
- Convertisseurs A/N et parfois N/A.

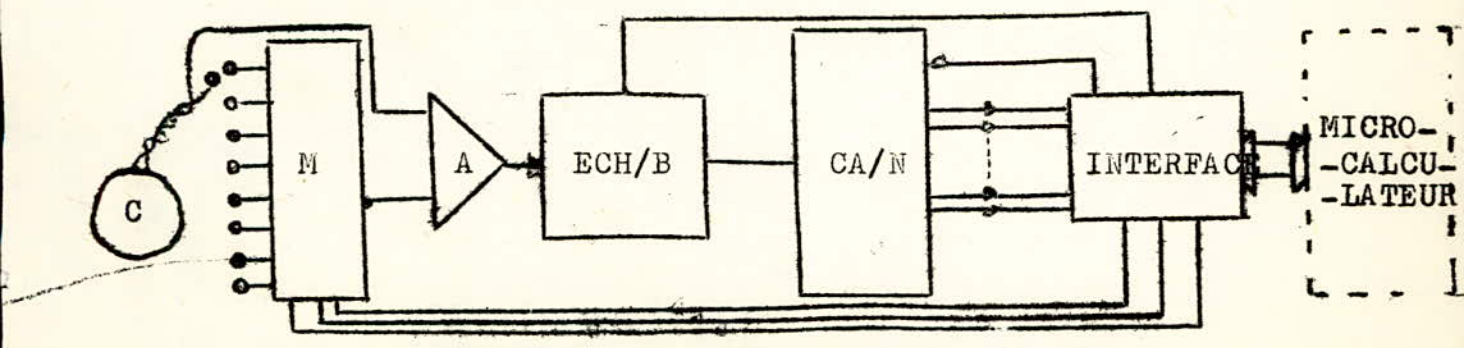
ces circuits peuvent être tous ou partiellement présents dans cette dite chaine.

Capteurs :

On les retrouve en début de chaine ces circuits permettent la transformation d'une grandeur physique de nature quelconque en une grandeur électrique, vu la diversité des phénomènes physiques, il existe plusieurs types:

- capteurs à sortie numérique : Ils délivrent en sortie des signaux numériques.
- Capteurs à sortie analogique; ensemble extremement vaste, les capteurs qui y sont classés délivrent en sortie un signal analogique

Ampli, filtres : le signal issu du capteur (à sortie analogiques) est généralement de très faible amplitude et est superposé à des signaux indesirables (bruit). Un étage amplificateur est alors necessaire de plus il est parfois indispensable dedisposer d'un ou de plusieurs filtres afin de sélectionner la bande de frequence qui représente effectivement le phénomène physique étudié.



C=Capteur; A=Amplificateur; ECH/B= Echantillonneur bloqueur
CA/N= Convertisseur analogique numerique

L'Ech/bloqueur.

Le signal analogique issu du capteur (eventuellement via l'étage ampli et les filtres) présente un caractère continu et une forme quelconque d'évolution dans le temps il doit donc être échantillonné pour être exploitable par le microcalculateur, de plus la tension à l'entrée du CA/N doit être constante pendant la conversion/donc bloquée ou maintenu) ; d'où la necessité des Ech/B.

L'échantillonage consistera en un prélevement d'une manière périodique de la valeur instatannée du signal analogique l'intervalle de temps separant deux échantillon dependera d'une part du temps de conversion des convertisseurs utilisés et d'autre part du spectre du signal analogique à echantillonner

multiplexeurs (analogique)

Dans le cas où un bon nombre de signaux analogiques sont à convertir par un seul C/AN on utilise un multiplexeur celui-ci se place entre les voies analogiques et les concertisseur, sa fonction se resume donc en un aiguillage d'une de ses entrées analogique (selectionnée parmi m entrées) sur sa sortie.

Convertisseurs :

a) CN/A : permettent de transformer une donnée binaire numérique en donnée (information) électrique analogique dont l'amplitude est determinée par la donnée numérique correspondante

b) CA/N : réalisent l'opération inverse de CN/A

Caractéristiques d'un CN/A	:	caractéristiques d'un CA/N
resolution	:	les mêmes que celles d'un CN/A
precision (Accuracy)	:	seulement on retrouve en plus
non linearite differentielle	:	— temps de conversion
gain TC (coefficient de température	:	(conversion t re)
temps d'établissement	:	

Voici achevé un tour d'horizon sur les differents circuits qu'on retrouve généralement dans l'interfaçage des environnements analogiques. On passe maintenant aux exemples où l'on verra l'utilité, l'application et la connection dechacun de ces circuits.

EXEMPLE

- utilisation du MC 6821 pour interfacer une table traçaute à entrées analogiques :

Commencons tout d'abord par définir cette table :

Elle permet la mise au point de tracés et dessins graphiques elle s'utilise de la même manière qu'un oscilloscope, elle est dotée de deux entrées X et Y. Une base de temps (~~de temps~~) pour l'enregistrement de pracessus lents* de l'éliminer pour le tracé d'une courbe Y= 7(X), cette table possède aussi une entrée pour la commande de levée ou d'abaissement de la plume tracante ; et enfin elle est à entrées analogiques.

En quoi consisterait son interfaçage ?

Les informations concernant la courbe à tracer Y=7(X) sont calculées et envoyées par le microprocesseur aux interfaces sous forme bénaire (ou bien numérique), alors que la table ne peut ~~être capable~~ **, l'interface doit être capable

- d'intercepter les données binaires concernant la courbe à tracer calculées par le microprocesseur.

De convertir ensuite ces données qui sont numériques en données analogiques.

Réalisation de l'interface :

Le PIA MC 6821 utilisé est programmé comme suit :

Le port A est utilisé pour le transfert des données numériques à convertir

Le port B est utilisé pour le transfert des commandes necessaires à la gestion des circuits interfaces venant après.

Les donnés et les commandes etant envoyées du microprocesseur vers la table traçaute, ces deux ports (A et B) seront donc programmés en sortie.

Le ordinateur calcule les coordonnéesde chaque point puis les envoie successivement vers le periphérique (table traçante).

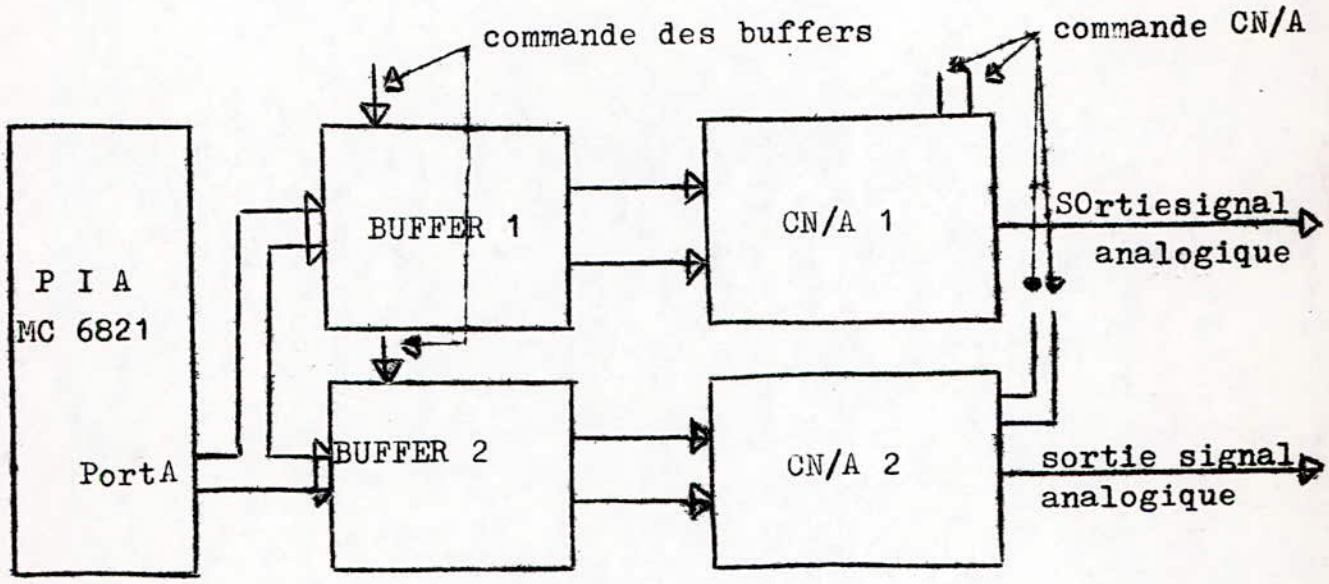
Soient X,Y les coordonnées du point qu'on désire tracer, le micro calcule et envoie la donnée (coordonnée) X et ensuite calcule et envoie la donnée (coordonnée) Y on doit donc disposer de deux concertisseurs, Analogiques-numériques l'un pour convertir la coordonnée X l'autre la coordonnée Y.

* se trouve à l'entrée X, une Commande manuelle →

* traiter que des données analogiques.

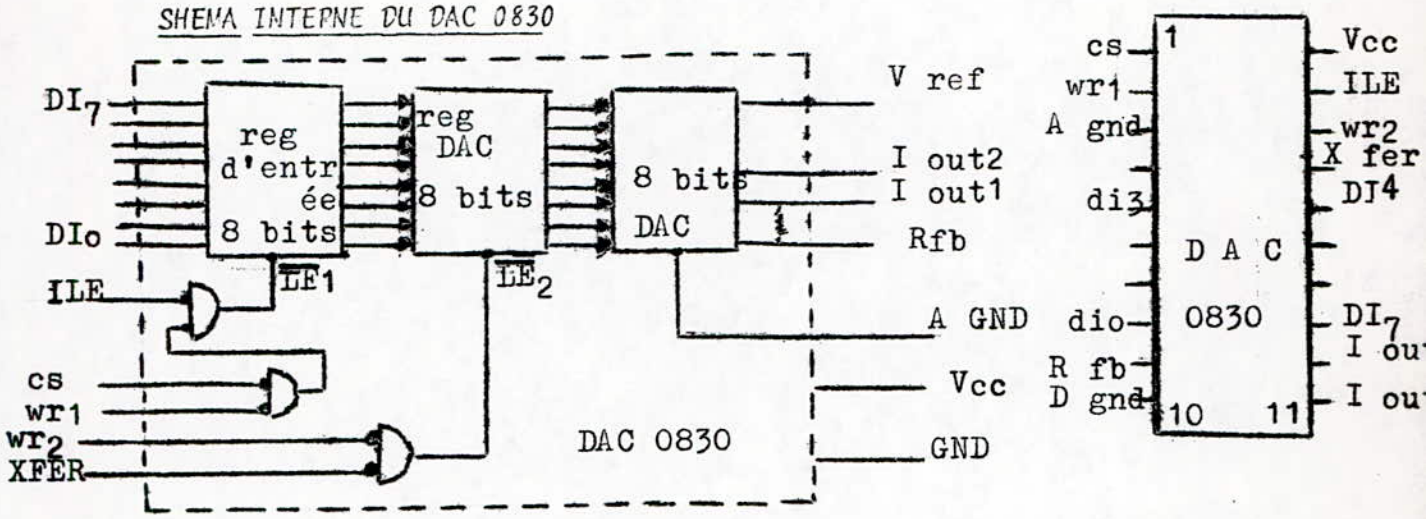
Les deux données (X et Y) doivent parvenir eu même temps aux entrées X et Y de la table tracaute ce qui impose de memoriser la donnée X (si elle est envoyée la 1ere) attendre que la donnée Y arrive ensuite ordonner la conversion des 2 données en même temps (la donnée X (ou Y) sera memorisée dans le convertisseur même qui s'occupe de sa conversion . Nous verrons plus loin le shema interne des convertisseurs N-A utilisés ainsi que leur principe de fonctionnement).

Mais au fait on n'utilise qu'un seul port (Port A) pour le transfert aussi bien de la donnée X que de la donnée Y il est donc necessaire de les diriger séparement chacune vers son convertisseur correspondant. Pour ce faire on utilise des buffers commandés par les lignes de controle venant du port B (voir Fig ci-dessous).



Ainsi lorsqu'on envoie la donnée X (abscisse du point calculé) vers le CN/A 1 le buffer 1 doit être passant et le buffer 2 en état haute impedance il enest de même pour l'envoi de la donnée Y (ordonnée du Pt) (~~ordonnée du Pt~~) avec bien sur inversion des commandes des buffers.
Les convertisseurs employés DAC 0830 sont du type numérique-Analogique 8 bits compatibles avec les microprocesseurs de la famille 6800.

SHEMA INTERNE DU DAC 0830



ce convertisseur à la possibilité de maintenir les données numériques memorisées pendant un certain temps dans un des ces registres internes commandés par les signaux $\overline{LE_1}$ et $\overline{LE2}$
$\overline{LE_1} = \overline{ILE}\ (\overline{CS}.\overline{WR})$; $\overline{LE2} = \overline{WR_1} \cdot \overline{XFER}$.

Pour revenir à notre problème de tout à l'heure (celui de memorisée la donnée X en attendant que la donnée Y arrive ensuite ordonner leur conversion simultanée) quand par exemple la donnée X parvient à l'entrée du 1ère CNA. Elle sera suivie quelques µSaprès par la donnée Y qui parviendra à l'entrée du 2ème CNA
Pour convertir les 2 données en même temps il suffira alors de mettre les 2 lignes $\overline{CS}$ de chaque convertisseur à l'état 1 (état non actif).

~~l'Etat non actif~~ puis le remettre à 0 dès-que la donnée Y arrivera [les autres lignes de commande $\overline{WR1}$, $\overline{WR2}$, $\overline{XFER}$ étant à l'état bas, **ILE** sera maintenue à l'état haut].
Ainsi les données converties seront présentés en sortie au même moment.
Finalement le port A du TMS 9901 servira pour le transfert des données sur 8 bits (PA0 à PA7 ÷ DI0-DI7) de l'interface aux buffers.
Le port B :

PB0, PB1 serviront respectivement à commander les buffers 1 et 2 .

PB2, PB3 serviront respectivement comme entrée ILE des convertisseurs 1 et 2

PB4, PB5 comme entrées $\overline{XFER}$ des 2 convertisseurs

PB6 servira comme entrée commune à la broche $\overline{WR2}$ de chaque convertisseur

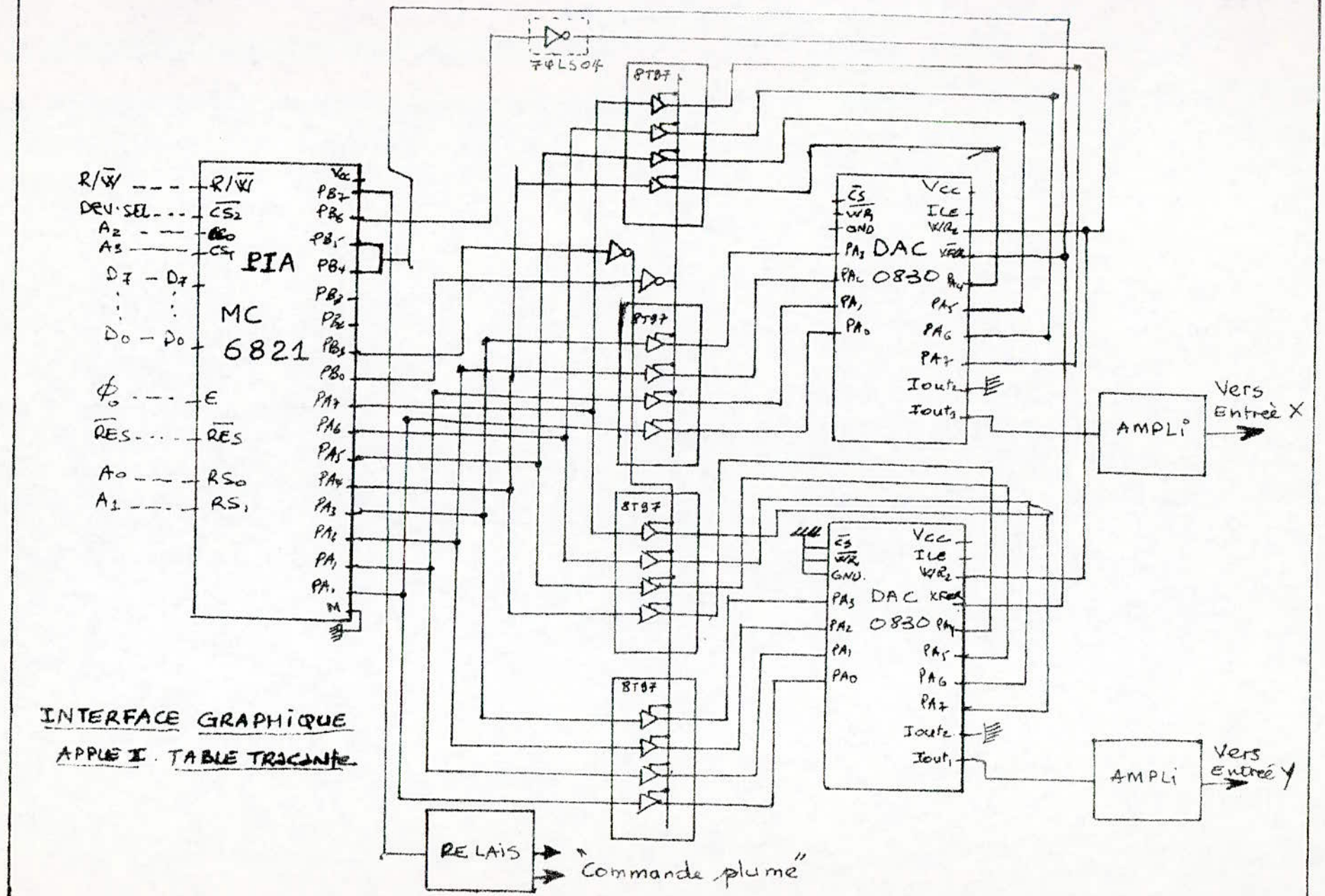
PB7 commandera la levée et l'abaissement de la plume.

Recapitulons :

PB0, PB1 commandront respectivement l'état (passant ou haute impedance) des buffers 1 et 2
PB 7 sera connectée à un circuit auxilliaire (relais) commandant la levée ou l'abaissement de la plume tracaute.

Expliquons maintenant le pourquoi des connexions PB2 à PB6.
$\overline{CS}$ et $\overline{WR1}$ seront mises à leur état actif (0) par cablage il ne nousreste donc que l'entrée ILE (voir skema interne du DAC 0830) pour commander le registre d'entrée du DAC. PB2 sera reliée à l'entrée ILE du CN/A 1
PB3 sera reliée à l'entrée ILE du CN/A 2.

Pour commander le registre DAC on utilise PB4 et PB6 qui seront reliés respectivement aux entrées $\overline{XFER}$ et $\overline{WR2}$ des 2 CN/A.



INTERFACE GRAPHIQUE

APPLE II . TABLE TRACANTE

EXEMPLE

On se propose dans cette application de procéder à l'examen d'un ERG (electroretinogramme) d'un patient par microordinateur, pour ce faire on lui placera convenablement des electrodes (capteurs). On recueillera un signal physiologique (Analogique) qu'on convertira en signal numérique (binaire) traitable par le microprecesseur, celui ci effectue les traitement appropriés pour restituer les résultats (toujours sous forme numérique) qu'on doit alors convertir en : signaux analogiques afin de les visualiser.

Le traitement des données est assuré par le KIT:D5 de Motorola(système assemplée autour du microprocesseur MC 6802) l'interfaçage est assuré par 1 PIA 6821.

Le PIA doit donc être capable :

- de faire entrée ou bien de saisir les données en provenance du CA/N (les mettant à la disposition du KIT pour le traitement).
- de faire sorttr ou bien de mettre à la disposition du CN/A les résultats du traitement qui lui sont communiqués par le up.

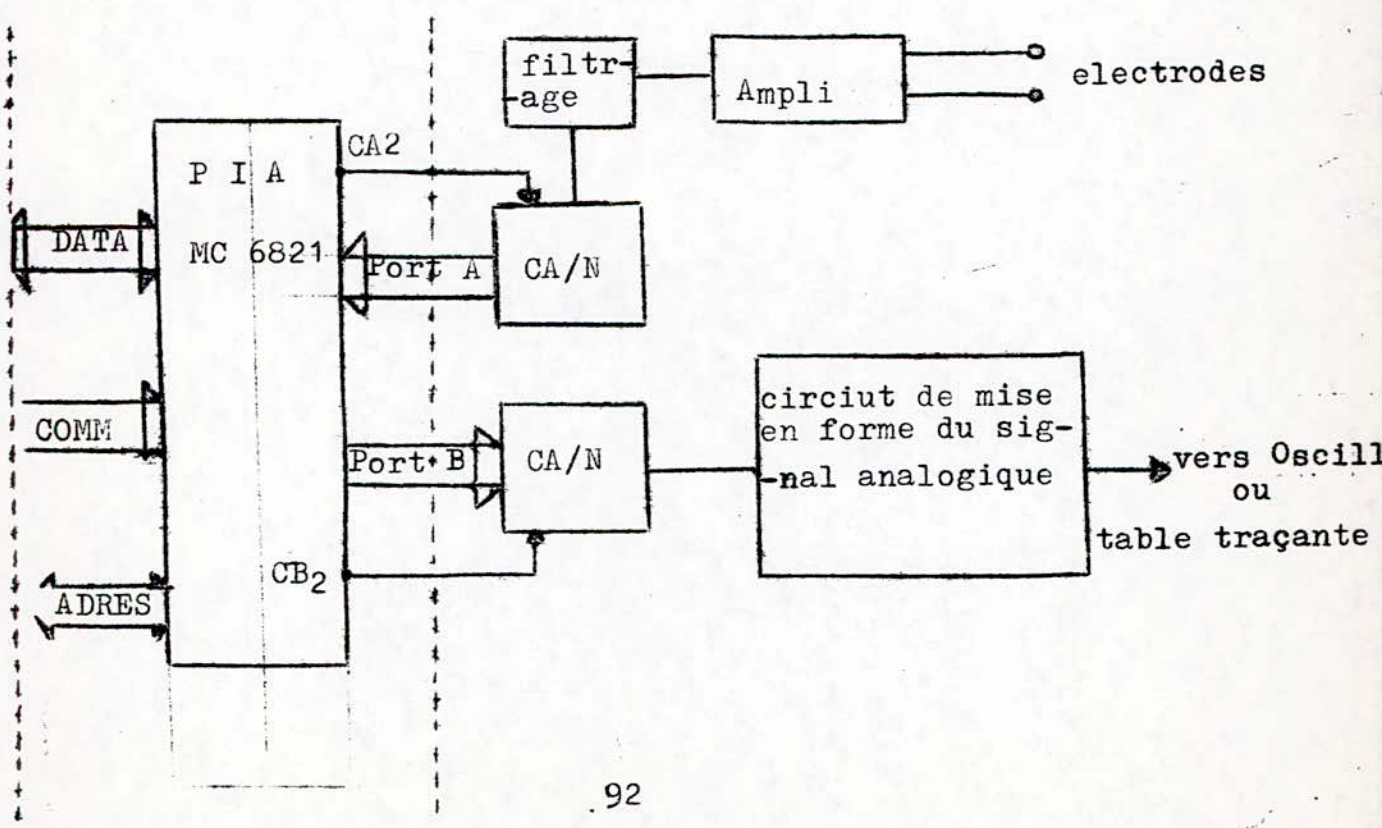
On peut alors programmer le PIA comme suit :

le PORT : A en entrée

le Port : B en sortie

La commande de conversion A/N est assurée par CA2

La commande de concersion N/A est assurée par CB2.



Notons aussi dans ce cas qu'il est inutile d'utiliser un Ech/bloqueur, le signal physiologique qu'on a à traiter est très lent (B.F) et de faible niveau ce qui rend la tâche d'échantillonnage secondaire vu que le convertisseur A/N peut lui même effectuer cette opération.

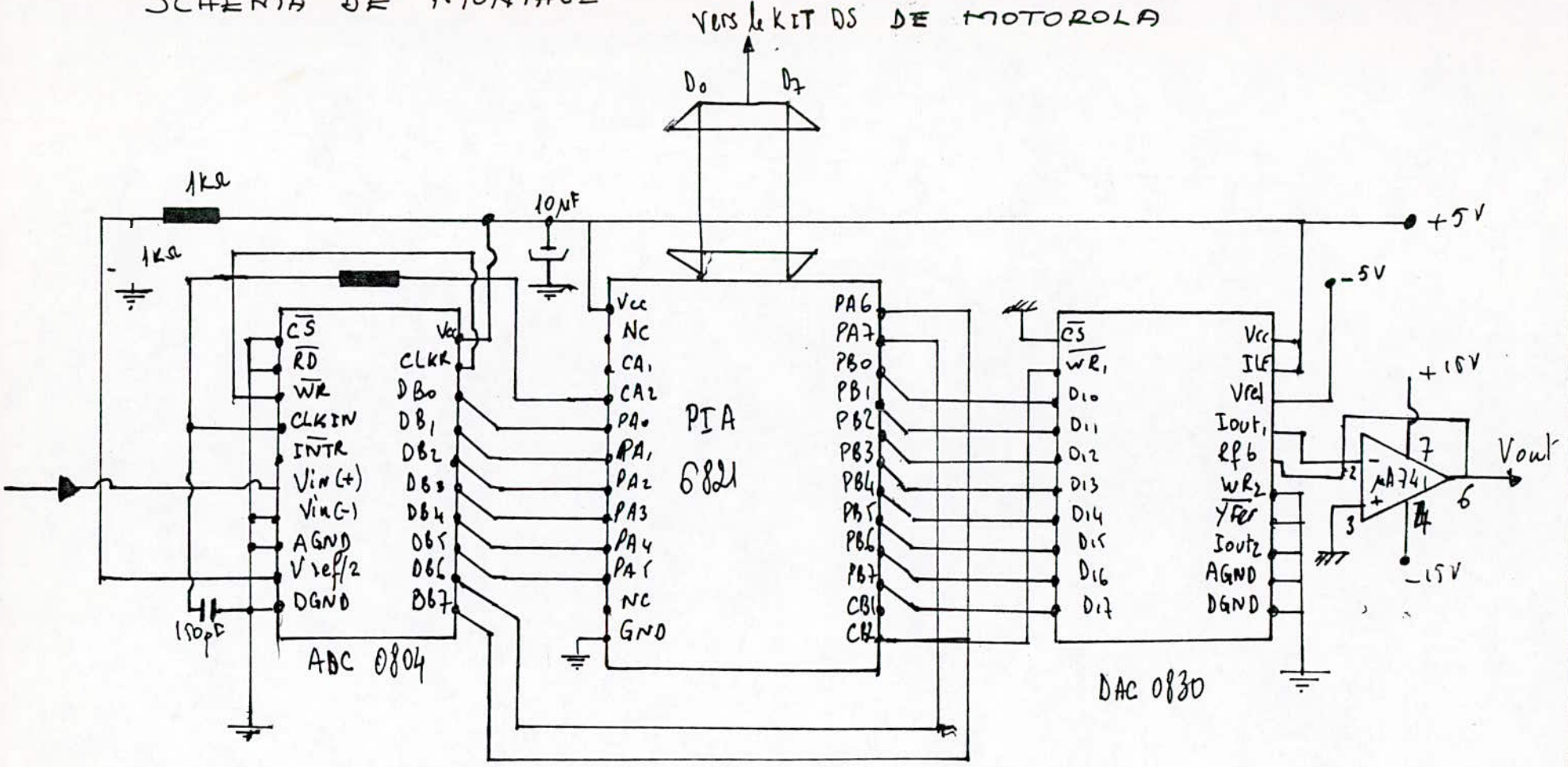
Remarquons aussi dans ce cas la possibilité d'utiliser des circuits integrés T.T.L

par exemple : 1 circuit SN 74 LS 134 remplacera le port B du PIA

1 circuit SN 74 LS 244 remplacera le port A du PIA

Les commandes de conversion N/A et AN peuvent être réalisées de plusieurs façons, on peut par exemple les generer à partir d'un autre buffer ou bien les connecter directement au bus de commande ou d'adresse du KIT. Donc sans passer par le PIA ou connecte les buffers et decodeurs sur le connecteur d'extension hors carte KIT qui présente les 3 bus du système.

SCHEMA DE MONTAGE



REMARQUE: Le schema est loin d'être complet, le signal physiologique recueilli par les électrodes est un signal électrique bruité de faible niveau necessitant donc une amplification et un filtrage avant d'attaquer le convertisseur AN (sur son entrée Vin). On utilisera pour ce faire un ampli de difference (différentiel) ainsi à sa sortie, le signal utile sera amplifié et tous les signaux indésirables qui se presentent en phases ou qui sont communs à ces deux bornes d'entrée sont en grande partie rejetés

Exemple :

Voyons maintenant une application où on réalise une chaîne d'acquisition de données analogiques comportant 3 voies (voir schéma de la page suivante) :

$\varphi(t)$ = Signal physiologique
$P(t)$ = Composante pneumatique
To) = Température

Dans ce cas on utilise des circuits intégrés TTL pour acheminer les donné et un interface programmable pour générer les commandes.

- Les signaux $P(t)$ et $\varphi(t)$ devront être convertis en même temps, chacun nécessite alors un A D C. POur la conversion de la température T° on procéde par multiglexag avec $P(t)$.

Sur le schéma on retrouve le circuit SN 7430 qui est une porte NAND servant à valider le décodeur SN 74 LS 138 qui à son tour valide l'un des 2 convertisseurs (ADC 802 utilisé).

Remarque : Les problèmes concernant la connection de ces circuits (PorteNAND et décodeur) sont des problèmes d'adressage sur lesquels on ne reviendra pas.

Branchement et commande du multiplexeur et des convertisseurs :

Le multipliplexeur MC 14053 B sert à aiguiller 2 données seulement $P(t)$ et T°, on utilise alors une seule broche interface parallèle

soit le TMS 9901 pour la famille 9900
soit le SAB 8255 pour la famille 8080
soit le MC 6820 pour la famille 6800

Cette broche commande le multiplexeur. Les deux autres entrées B et C du multiplexeur seront mises à la masse.

.../...

La commande de conversion des ADC sera reliée soit :

- à W E du microprocesseur 9 900
- à W R du microprocesseur 8 080
- à R/W du microprocesseur 6 800

(L'instruction de la commande de conversion sera alors qu'une instruction d'écriture mémoire avec bien sûr spécification de l'adresse choisie pour valider le convertisseur)

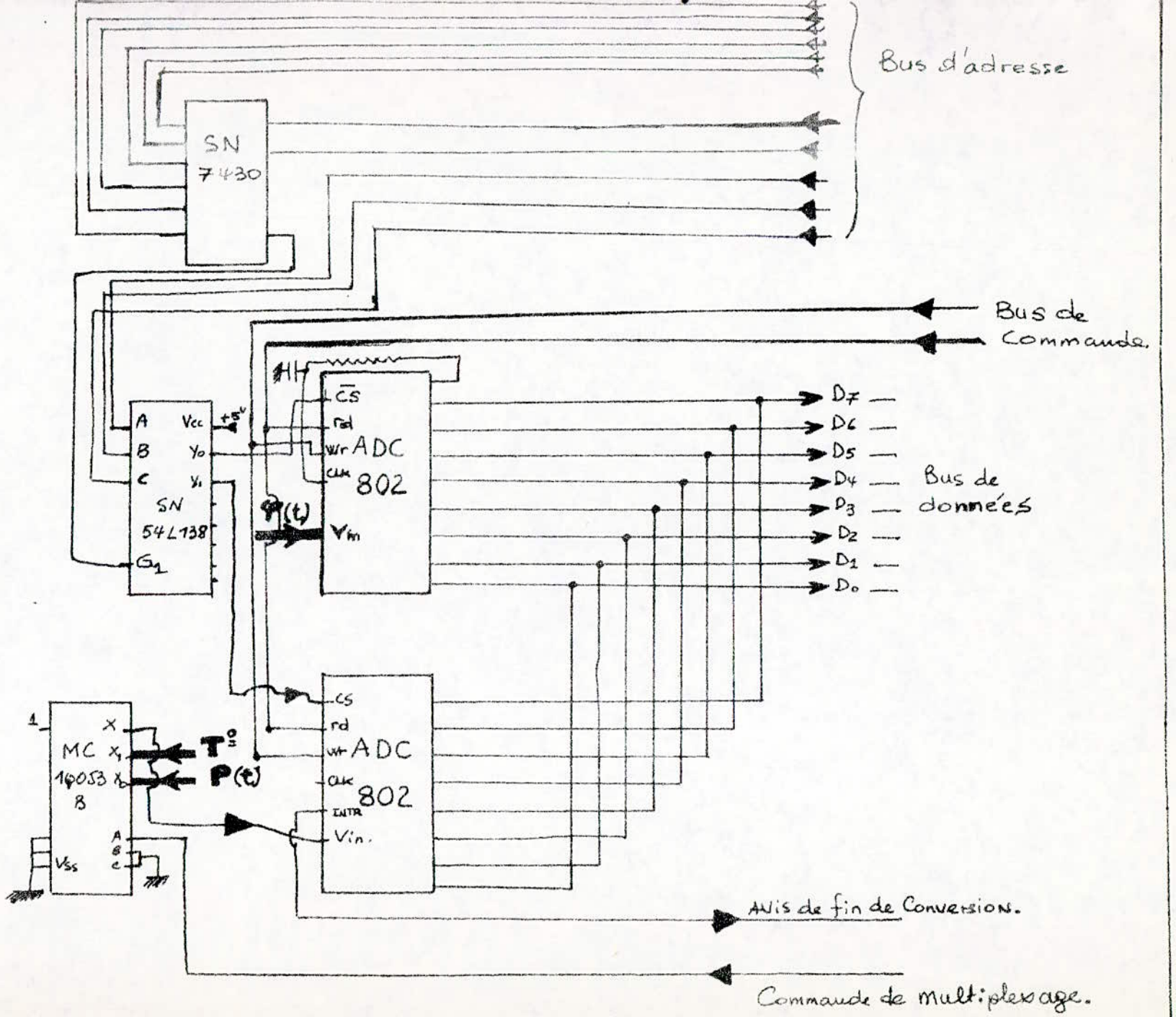
* De même pour la commande de lecture de la donnée convertie rd qui sera reliée à :

- DBIN du microprocesseur 9 900
- DBIN du microprocesseur 8 080
- Un bit du bus d'adresses pour le microprocesseur 6 800

(L'instruction de lecture de donnée convertie sera alors une instruction de lecture mémoire).

on utilise une broche de l'interface que le microprocesseur lit en permanence (par logiciel) pour renseigner sur la fin de conversion afin d'entreprendre un autre cycle de lecture des données numériques

Remarque : On connecte ce montage en système microordinateur à l'aide des connecteurs.


Bus d'adresse
SN
7430
Bus de
Commande.
A
B
C
Vcc
Yo
Y1
SN
54L138
G1
+5V
CS
rd
wr
clk
ADC
802
Vin
D7
D6
D5
D4
D3
D2
D1
D0
Bus de
données
1
MC
10053
B
X
X1
X2
Vss
A
B
C
T°
P(t)
cs
rd
wr
clk
ADC
802
INTR
Vin.
Avis de fin de Conversion.
Commande de multiplexage.

" C O N C L U S I O N "

Pour le controle de tous processus, grandeur, ou phénomène physique par microprocesseur, pour tout traitement, saisie ou transfert d'informations par microprocesseur, l'electronic en se trouvera confronté aux problèmes de conception de microordinateur. Celle-ci se fait généralement en partant de plusieurs choix :

-- Un choix de famille : le concepteur ayant en main les qualificatifs et les performances de chacune d'elles, choisira celle qui lui convient le mieux. (ou tout simplement celle avec laquelle il est le plus familiarisé).

-- Un choix de réalisation :

a) le concepteur peut utilier des cartes standards (ces cartes peuvent contenir des matrices memoires des unités centrales, et même des interfaces avec leur logique complementaire). Sans ce cas son problème de conception se reduirait à un simple problème d'adaptation et d'interfaçage.

b) Ou bien mettre au point lui même son système minimum. (quand il ne dispose pas de cartes ou que leurs qualificatifs ne lui conviennent pas); Il doit alors élaborer le programme de gestion de son système ce qui lui rendrait la tâche encore plus difficile.

Concernant les E/S, une demarche logique consisterait à choisir une procedure d' E/S (E/S controlées par programme, E/S controlées par interruption, E/S par DMA) qui determinera alors le materiel adapté. Ce materiel necessitera l'étude d'un logiciel souvent très complexe. D'autre part, les circuits eux même offrant un certain nombre de procedures d'E/S , il sera

possible d'adjoindre à un materiel; plusieurs logiciels de gestion illustrant des procedures très differentes.

Pour la conception de logiciel adapté aux microordinateurs (la partie absente dans notre étude), il serait souhaitable de lancer une ou plusieurs études qui permettraient de poser des bases et des methodes nouvelles de programmation.

-ANNEXE-

REGISTRE DE COMMUNICATION SERIE (CRU)

CRU (Communication Register Unit)

Le CRU est une voie particulière d'E/S propre à la famille 9 900 permettant une communication entre le processeur et des dispositifs externes.

L'originalité du CRU réside dans le fait que ce mode n'utilise pas le bus des données Il utilise 3 lignes CRU (CRUIN, CRUOUT, CRUCIK) couplant le microprocesseur (TMS 9 980) et l'interface s'occupant des E/S (TMS 9 901)

Les données s'acheminent en série du micro-processeur vers l'interface sur CRU OUT et l'interface vers le microprocesseur sur CRU IN. Le bit CRU CIK quand à lui emet des signaux de validation et d'échantillonnage pour piloter l'échange de ces données (voir figure 1).

La vitesse d'échange de ces données en mode CRU est bien sûr relativement vu que les données s'acheminent en série sur un seul bit.

En exécutant un programme lorsque le microprocesseur rencontre une instruction CRU, son bit MEMEN est mis à"1" (Quand MEMEN est à ZERO, l'adresse présentée sur le bus d'adresse sollicite une position mémoire et non un bit CRU d'E/S)

Une fois le bit MEMEN est à "1" les bits Ao et A1 du bus d'adresses sont forcés à "0", les bites A2 à A 12 seront chargés respectivement par les bits 4 à 14 du registre de travail R 12 de l'espace de travail du programme en exécution. Le bit A 13 servira quand à lui à l'acheminement en série des données du mricro-processeur vers l'interface d'E/S, ce n'est autre que le bit CRU OUT dont on à parlé plus haut.

Les bits A 2 à A 12 du bus d'adresses constitueront l'ADRESSE DE BASE CRU.

.../...

Qu'est ce que c'est l'adresse de base CRU ?

Avant de définir ce qu'est une adresse de base CRU il est indispensable de connaître le brochage du TMS 9 901, son architecture et son couplage avec le microprocesseur TMS 9 980.

Le TMS 9 901 est donc un circuit d'interface parallèle programmable s'occupant de la gestion d'E/S par voie "CRU". 32 bits internes de cet interface sont adressables à l'aide des instructions CRU (on entend par adressable un bit dont on peut connaître l'état 0 ou 1; le forcer à être dans un état 0 ou 1 ; ou contrôler son état qui peut être 0 ou 1).

L'adresse de base CRU n'est autre que l'adresse logicielle d'un des 32 bits adressables du TMS 9901.

Comme on l'a signalé auparavant, cette adresse doit être chargée dans le registre R 12 de l'espace travail du programme en execution

Le tableau N° 1 donne l'adresse de base CRU correspondante à chaque bit adressable du TMS 9 901.

Nous allons nous appuyer sur un exemple pour rendre aisé de compréhension le problème de l'adressage CRU.

On veut par exemple mettre à 1 la broche 38 ou le bit Po du TMS 9 901. Po correspond à la LED CR1 de la carte TMS 990/189, sa mise à "1" correspond donc à l'allumage de cette LED. Pour celà on doit spécifier notre espace travail et charger le registre R12 par l'adresse de base CRU de la LED CR1 et ensuite ordonne sa mise à "1" à l'aide de l'instruction CRU "SBO".

Le programme correspondant est donné par

```
A 200
L W P I > 300
L I   R 12, > 20  ⟶ ( Adresse de base CRU de la LED CR 1)
```

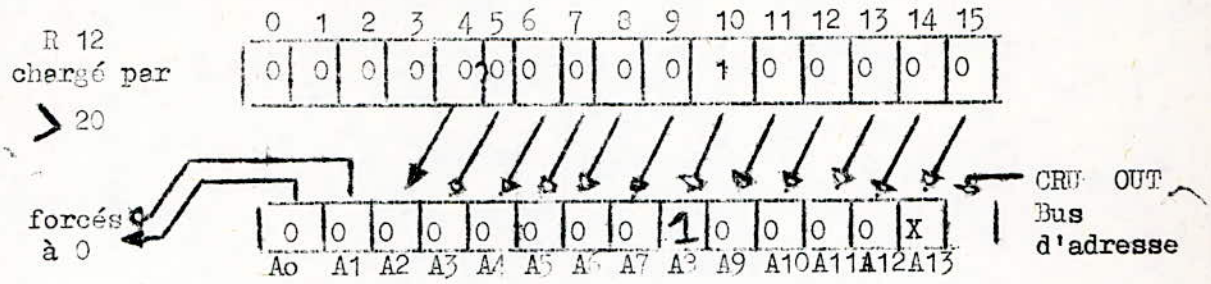
.../...

ORGANISATION des voies d'E/S

ADRESSE BASE CRU			Repérage des bits du TMS 9901			
			Entrée		Sortie	
			instruction CRU STCR TB		instruction CRU SBZ SBO LDCR	
U10	U11		Bit de contrôle		Bit de contrôle	
000	400	0	0	1	0	1
002	402	1	INT 1	CLK 1A	MASK 1	CLK 1B
004	404	2	INT 2	CLK 2A	MASK 2	CLK 2B
006	406	3	INT 3	CLK 3A	MASK 3	CLK 3B
008	408	4	INT 4	CLK 4A	MASK 4	CLK 4B
00A	40A	5	INT 5	CLK 5A	MASK 5	CLK 5B
00C	40C	6	INT 6	CLK 6A	MASK 6	CLK 6B
00E	40E	7	INT 7	CLK 7A	MASK 7	CLK 7B
010	410	8	INT 8	CLK 8A	MASK 8	CLK 8B
012	412	9	INT 9	CLK 9A	MASK 9	CLK 9B
014	414	10	INT 10	CLK 10A	MASK 10	CLK 10B
016	416	11	INT 11	CLK 11A	MASK 11	CLK 11B
018	418	12	INT 12	CLK 12A	MASK 12	CLK 12B
01A	41A	13	INT 13	CLK 13A	MASK 13	CLK 13B
01C	41C	14	INT 14	CLK 14A	MASK 14	CLK 14B
01E	41E	15	INT 15	CLK 15B	MASK 15	CLK 15B
020	420	16	P0 INPUT		P0 OUTPUT	
022	422	17	P1 INPUT		P1 OUTPUT	
024	424	18	P2 INPUT		P2 OUTPUT	
026	426	19	P3 INPUT		P3 OUTPUT	
028	428	20	P4 INPUT		P4 OUTPUT	
02A	42A	21	P5 INPUT		P5 OUTPUT	
02C	42C	22	P6 INPUT		P6 OUTPUT	
02E	42E	23	P7 INPUT		P7 OUTPUT	
030	430	24	P8 INPUT		P8 OUTPUT	
032	432	25	P9 INPUT		P9 OUTPUT	
034	434	26	P10 INPUT		P10 OUTPUT	
036	436	27	P11 INPUT		P11 OUTPUT	
038	438	28	P12 INPUT		P12 OUTPUT	
03A	43A	29	P13 INPUT		P13 OUTPUT	
03C	43C	30	P14 INPUT		P14 OUTPUT	
03E	43E	31	P15 INPUT		P15 OUTPUT	

SBO, 0 (Mettre à "1" le bit d'E/S dont l'adresse de base CRU est (contenue dans R 12)

Comment se déroule l'opération ?



Les bits A à 14 chargeront respectivement le bit A2 à A 12 . Cette adresse ainsi faite sera dénommée "adresse de base CRU"

Les bits A2 à A7 sont branchés à une logique de validation d'interface, ils nous permettent de valider un interface (l'interface utilisateur U 10, le TMS 9 901 dans notre cas) parmi d'autres connectés.

Sur la carte TMS 990/189, il existe 2 circuits d'interface TMS 9 901.

Le TMS 9 901 (U 10) est une voie d'E/S CRU réservée à l'utilisateur occupant des adresses de base CRU allant de (0000) 16 à (003E)16.

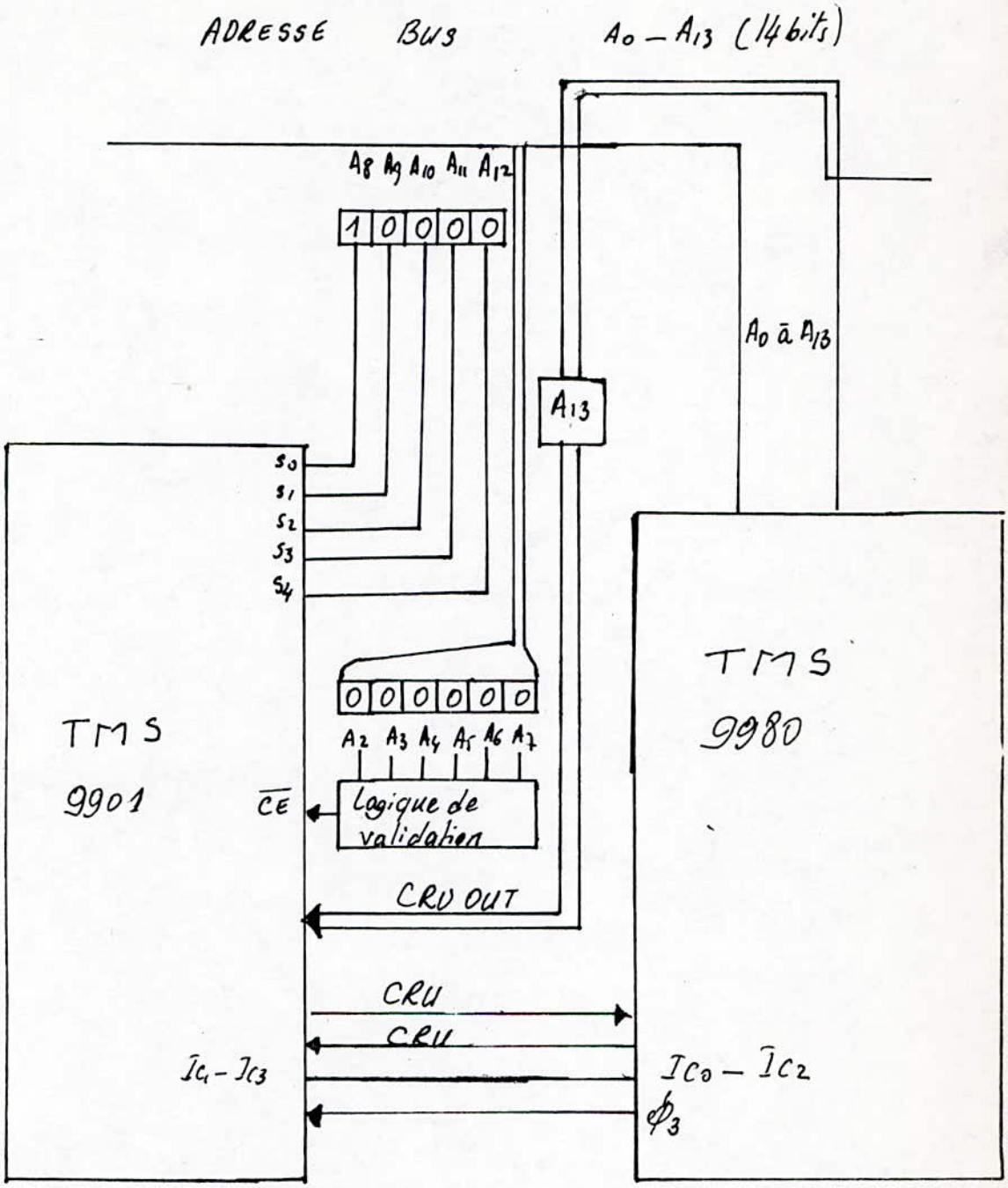
Il est donc clair que pour valider le TMS utilisateur, les bits A2 - A7 branchés à la logique de validation doivent être à"ZERO".

Le 2ème TMS 9 901 "U 11" est une voie d'E/S système, consacré aux périphériques comme le clavier, la visualisation, le disque sonore.

Cette voie d'E/S système occupe des adresses de base CRU allant de (0400)16 à (043E)16.

Les bits A8 à A12 seront connectés aux broches So...S4 du TMS 9 901 donnant 32 combinaisons.

.../...


ADRESSE BUS
A0 – A13 (14 bits)
A8 A9 A10 A11 A12
1 0 0 0 0
A0 à A13
A13
S0
S1
S2
S3
S4
TMS
9901
TMS
9980
0 0 0 0 0 0
A2 A3 A4 A5 A6 A7
CE
Logique de validation
CRU OUT
CRU
CRU
IC1 – IC3
IC0 – IC2
Φ3

On peut définir d'une façon plus claire ces instructions CRU qui sont en en nombre de 5.

Ces instructions effectuent les opérations d'E/S sur un ou plusieurs bits.

STCR : permet de nous renseigner sur l'état d'un ou de plusieurs bits adressables de l'interface d'E/S (d'adresses de base CRU contigües) en les chargeant dans un registre ou une position mémoire.

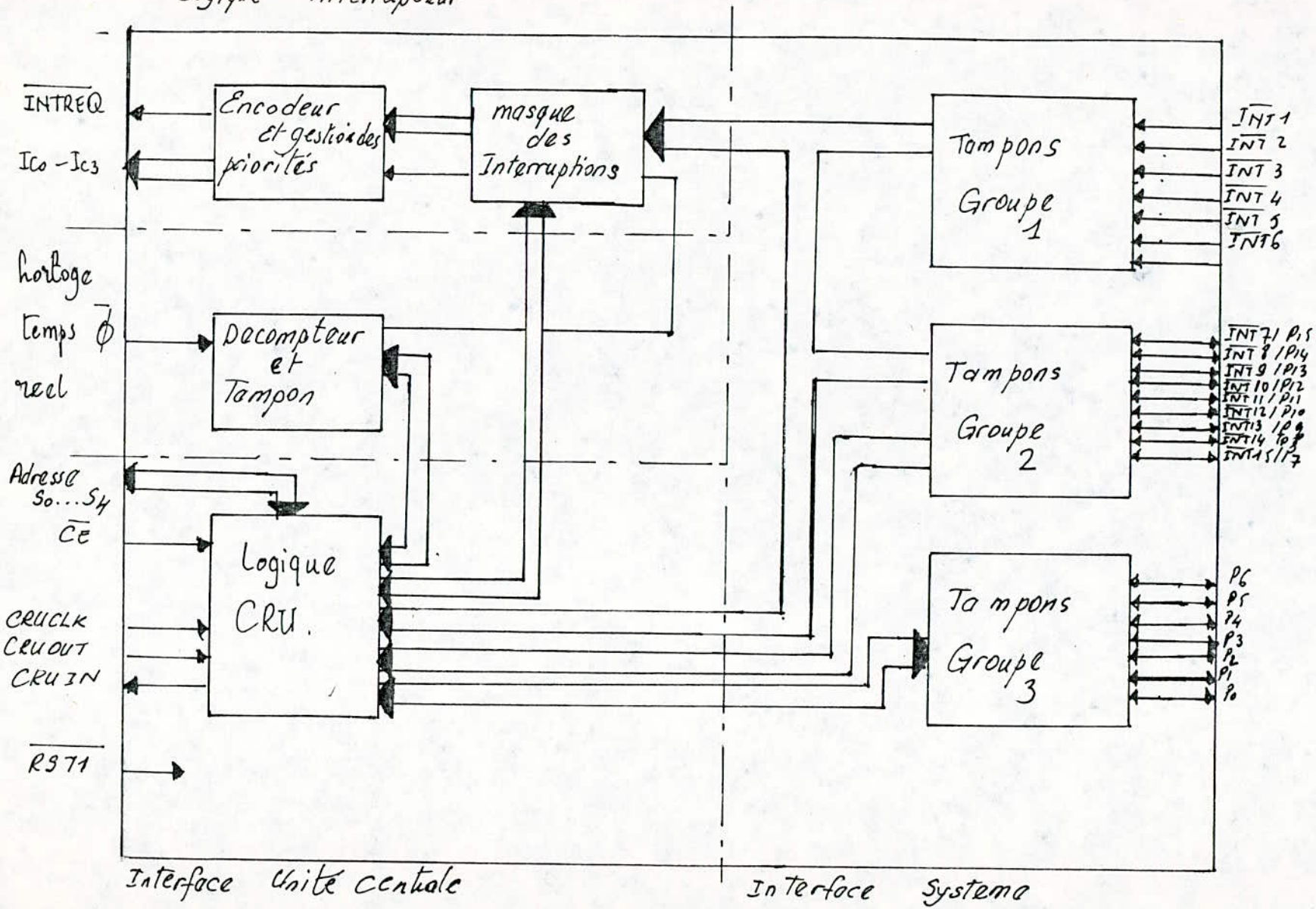
LDCR : est l'opération inverse de STCR, elle permet de charger un ou plusieurs bits adressables de l'interface d'E/S respectivement par les bits 0 à 15 d'un registre ou d'une position mémoire.

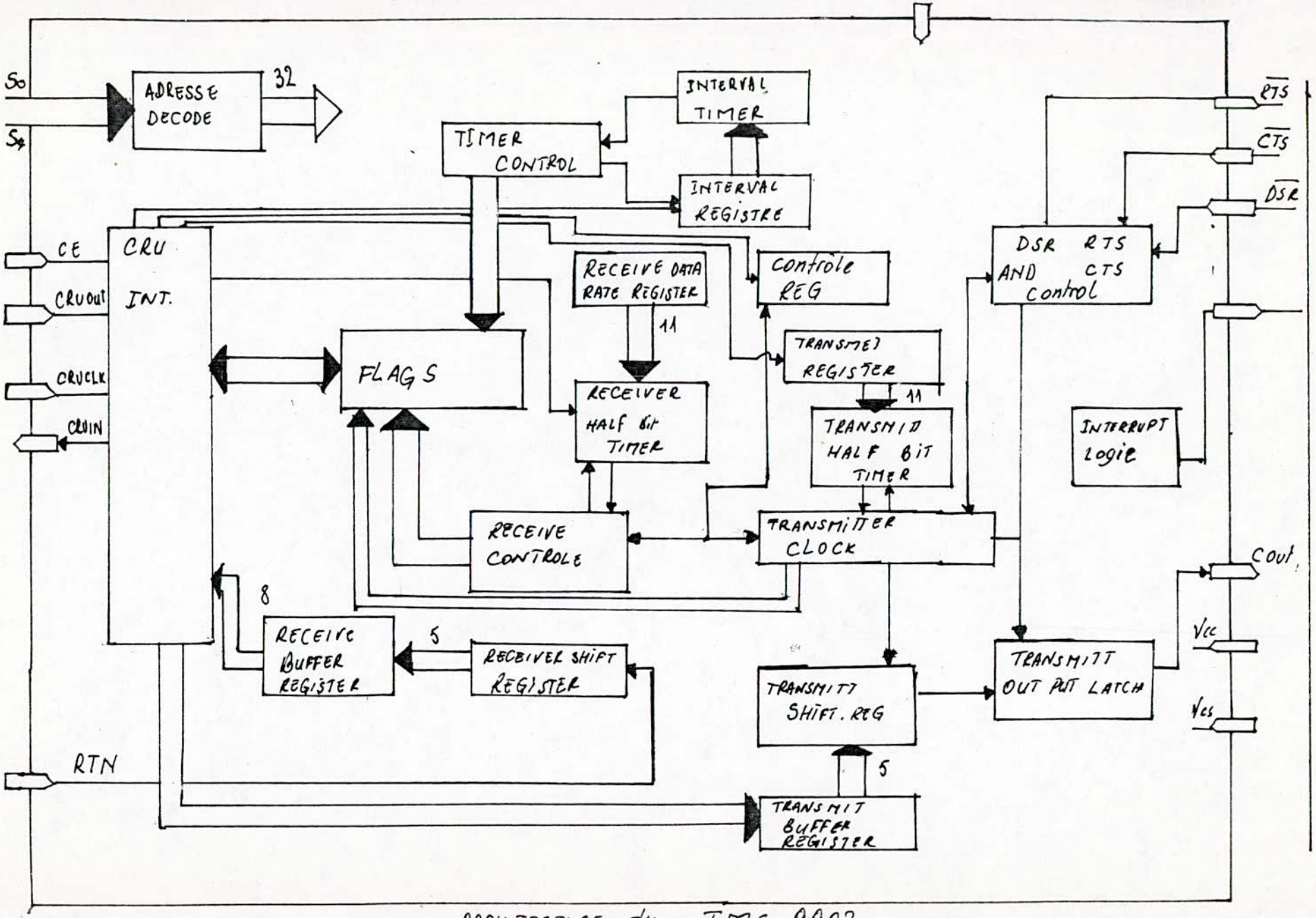
TB : permet de contrôler l'état d'un bit de l'interface des E/S CTMS 9901) en le recopiant dans le bit E Q du registre d'Etat.

SBO, SBZ : permettent respectivement de mettre à 1 (ou à 0) un des bits adressables de l'interface d'E/S.

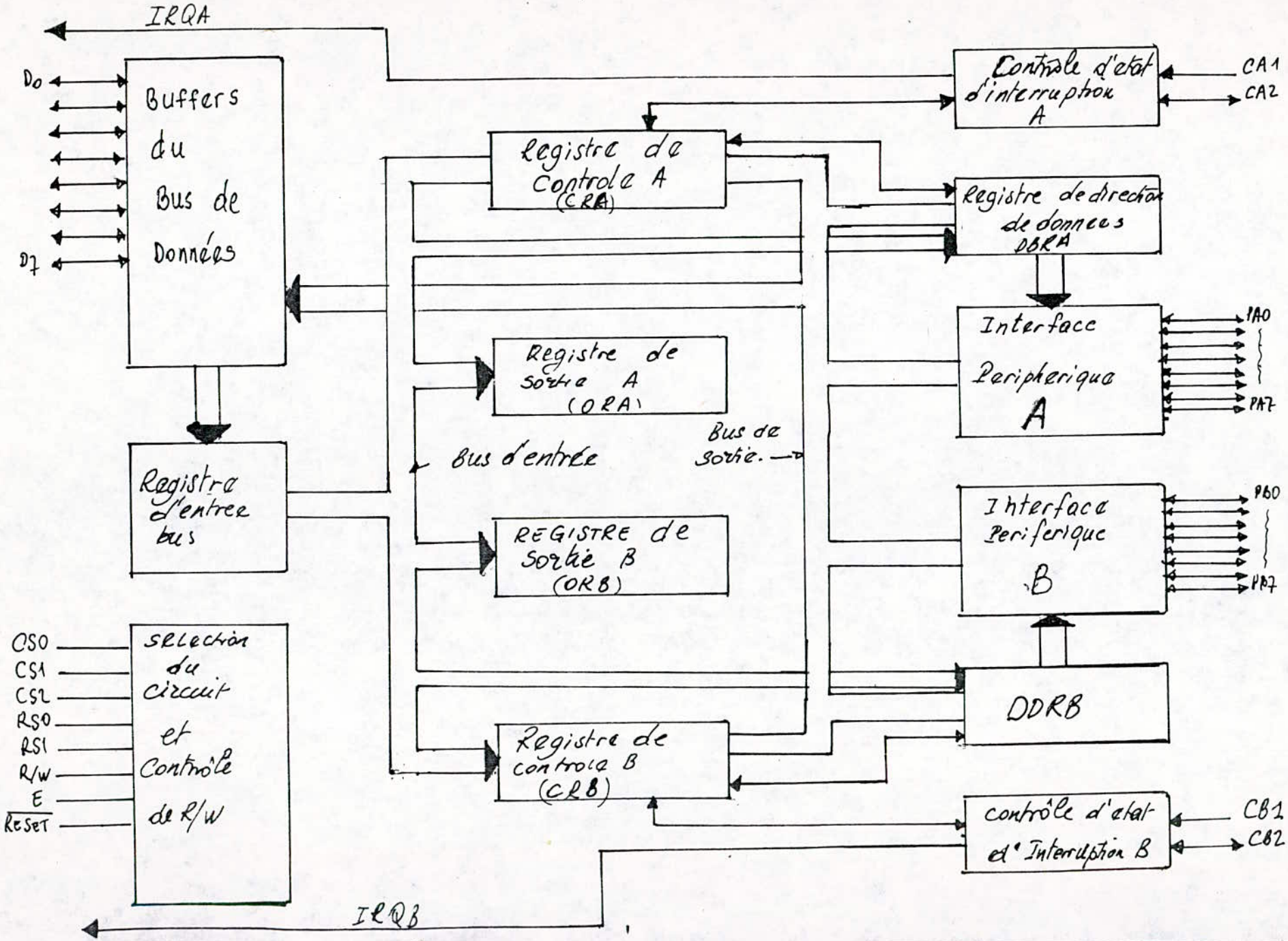
	Opération sur un seul bit	Opération sur plusieurs bits
Entrée	T B	S T C R
Sortie	S B O S B Z	L D C R

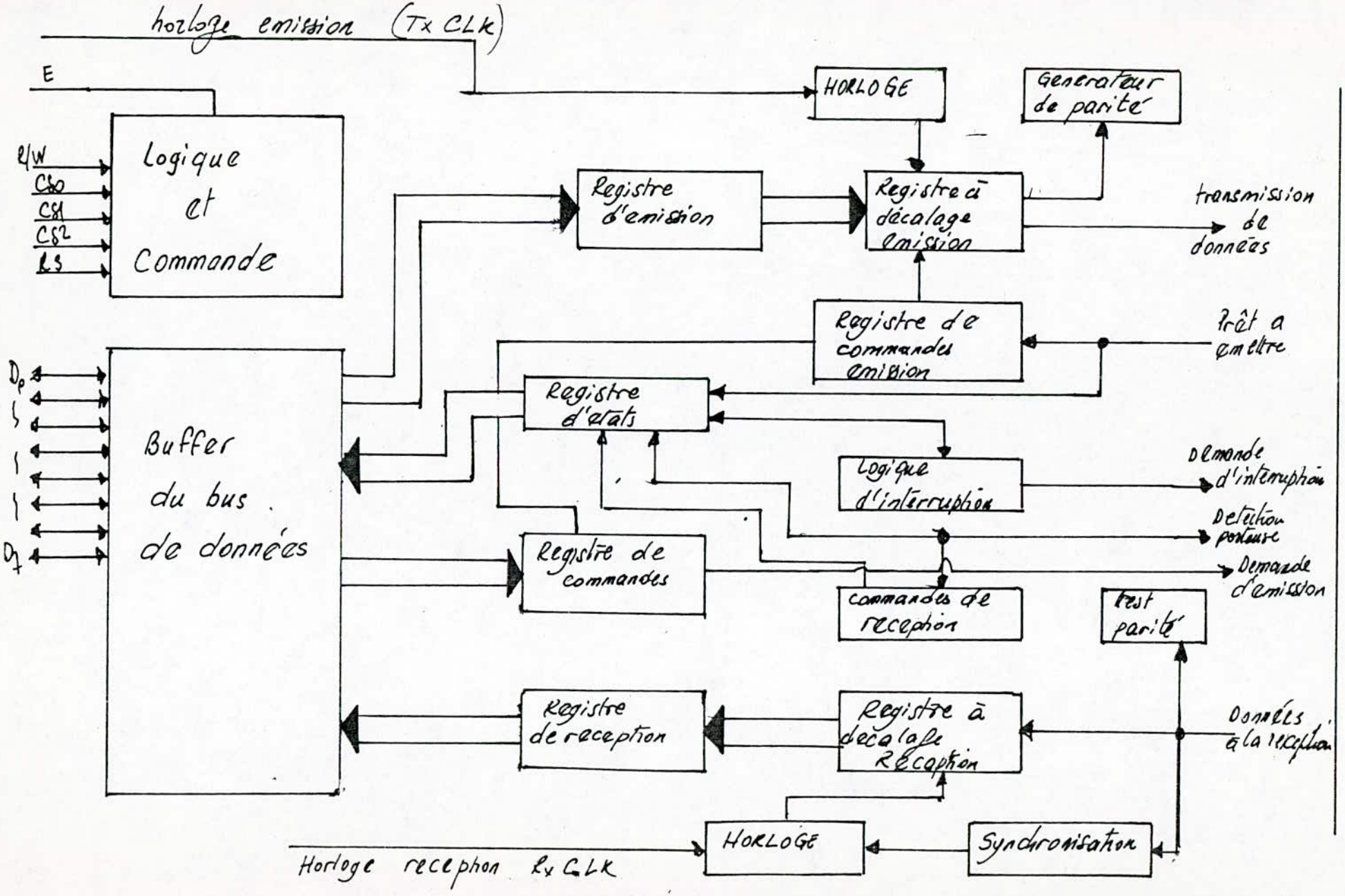
Architecture du TMS 9901
Logique interrupteur



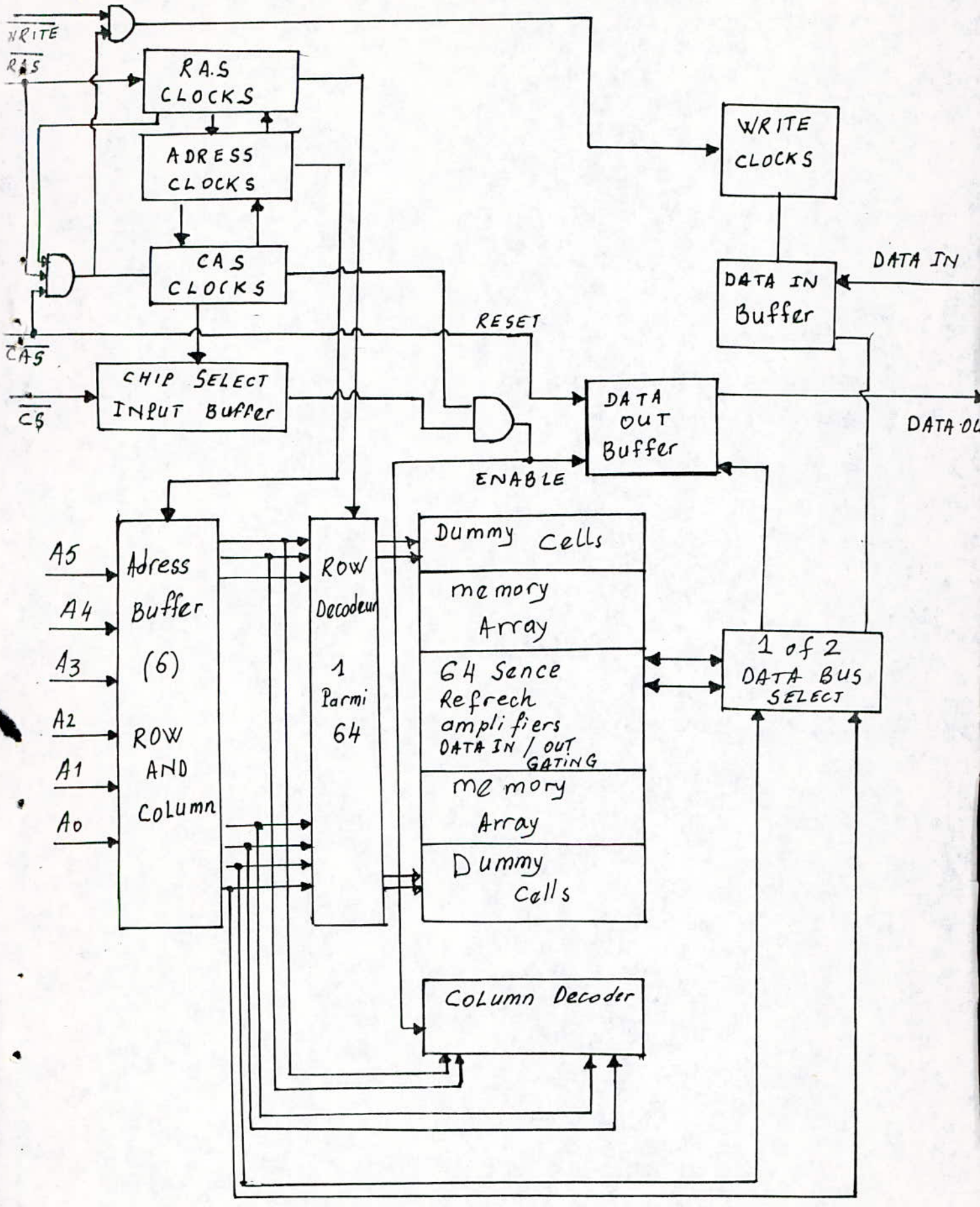


ARCHITECTURE du TMS 9902

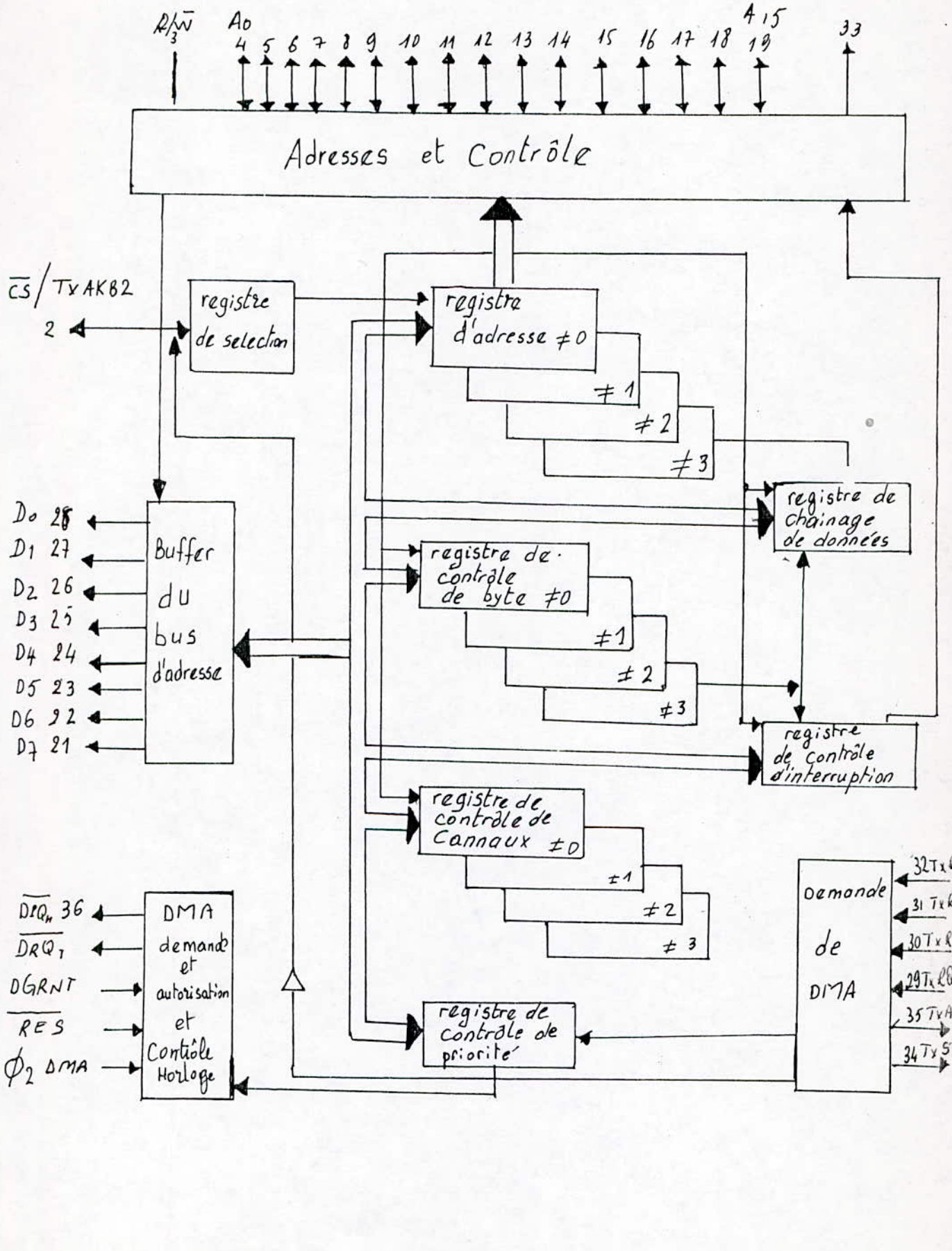

IRQA
D0
D7
Buffers du Bus de Données
Registre d'entrée bus
Contrôle d'état d'interruption A
CA1
CA2
Registre de Controle A (CRA)
Registre de direction de données DDRA
Registre de sortie A (ORA)
Bus d'entrée
Bus de sortie.
Interface Peripherique A
PA0
PA7
Interface Peripherique B
PB0
PB7
REGISTRE de Sortie B (ORB)
DDRB
Registre de controle B (CRB)
contrôle d'etat d'Interruption B
CB1
CB2
CS0
CS1
CS2
RS0
RS1
R/W
E
RESET
selection du circuit et Contrôle de R/W
IRQB


horloge emission (Tx CLK)
E
R/W
CS0
CS1
CS2
RS
Logique et Commande
HORLOGE
Generateur de parité
Registre d'emission
Registre à décalage emission
transmission de données
Registre de commandes emission
Prêt a emettre
Registre d'etats
Buffer du bus de données
Logique d'interruption
Demande d'interruption
Detection porteuse
Registre de commandes
commandes de reception
Demande d'emission
test parité
Registre de reception
Registre à decalage Reception
Données à la reception
HORLOGE
Syndronisation
Horloge reception Rx CLK

Organisation interne de La R.A.M. 4096 bits

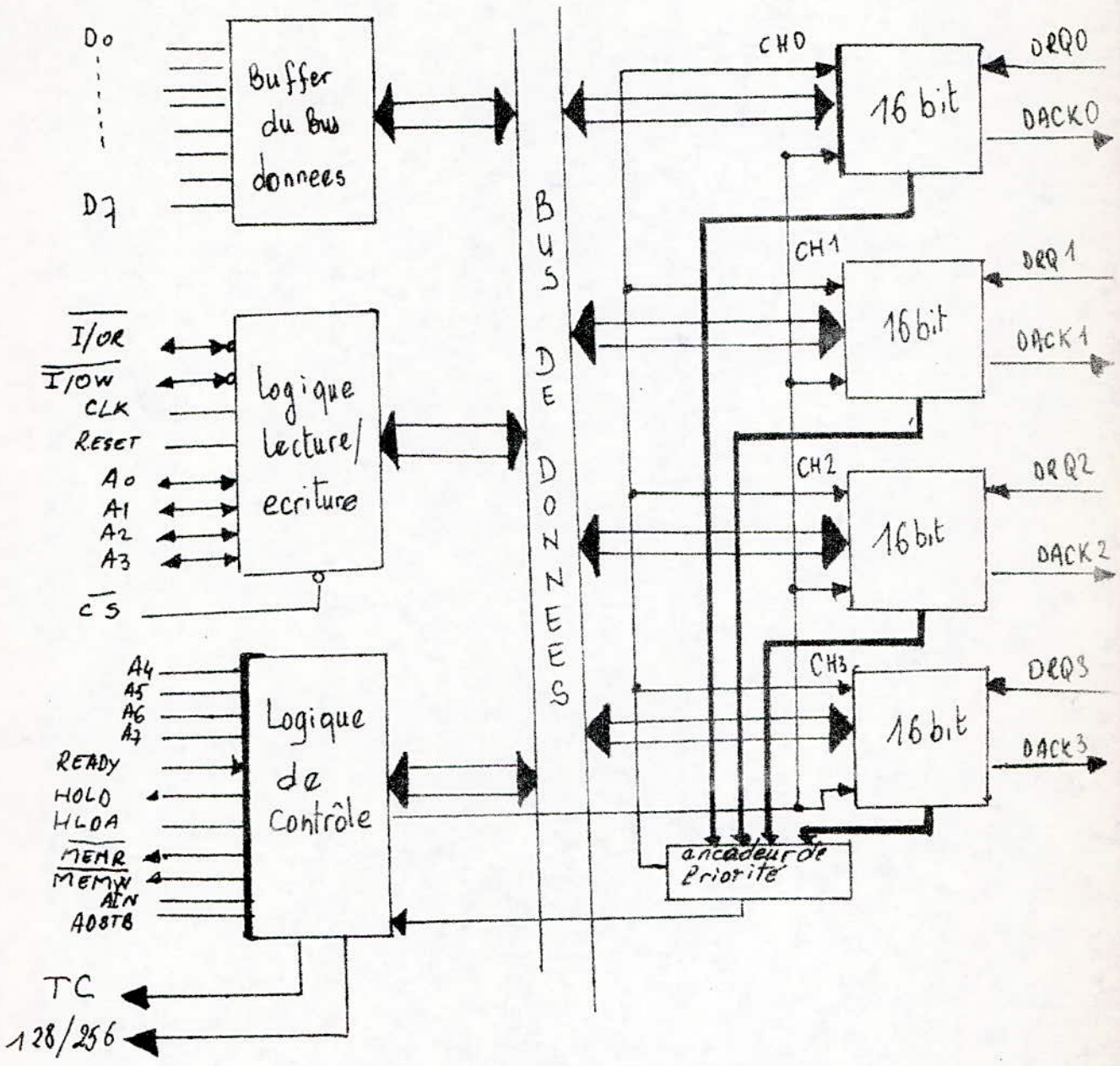


Organisation interne du MC. 6844



Organisation interne du DMAC

INTEL 8257



-ooOoo-ooOoo-
-ooOoo-ooOoo-BIBLIOGRAPHIE-ooOoo-ooOoo-
-ooOoo-ooOoo-

-EMPLOI DES MICROPROCESSEURS : M.AUMIAUX Masson 1977.

-MICROPROCESSEURS ET MICROORDINATEURS : R.L.CAEN , J.M.CROZET
Masson .1977

-INTERFACAGE DES MICROPROCESSEURS : M.ROBIN; T.MAURIN

-RENDEZ-VOUS AVEC le MICROPROCESSEUR :

-INTERFACES POUR MICROPROCESSEURS ET MICROORDINATEURS: H.LILEN
Edition Radio. 1981

-INITIATION AUX MICROPROCESSEURS : Texas Instruments

-GUIDE D'UTILISATION DE LA CARTE TM 990/189

-REVUE MICRO-SYSTEMES JANV/FEV 79 , MARS/AVRIL 79

-DATA BOOK T T L for design engineers

-M6800 MICROPROCESSOR APPLICATIONS MANUAL (Motorola) 1975.

-DATA BOOK 9900 SYSTEMS DESIGNS (Texas Instruments)

-MICROPROCESSOR DEVICES :Siemens Data Book (76-77)

-projets de fin d'etudes:

-L'electroretinogramme par microprocesseur Janv/83 (ENPA)

-Extension du micro:APPLE II ;Etude etRealisation d'une
Interface graphique Janv /83 (ENPA)

- Les E/S d'un microordinateur Janv/80 (ENPA)

- Etude et realisation d'un interface entre des signaux
physiologiques pré-traités et la carte TM 990/189
Janv :83 (ENPA)

-Techniques D'interfaces Application à une liaison:
TTY- M6800 Janv :79 (ENPA).

-MICROPROCESSEUR :DU 6800 AU 6809 MODES D'INTERFACAGE
G.REVELIN DUNOD 1981.

-REVUE LE HAUT PARLEUR N°1682 Juillet 82.

-TP SUR KIT D2 DE MOTOROLA (enita).