

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

28/82

UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE

Houari BOUMEDIENE

ECOLE NATIONALE POLYTECHNIQUE

Département d'Electronique

Projet de Fin d'Etudes

Ingéniorat d'Etat en Electronique

ETUDE ET REALISATION
D'UNE INTERFACE DE VISUALISATION
MICRO - ORDINATEUR --- CRT 606 A

المدرسة الوطنية للعلوم الهندسية
المكننة
ECOLE NATIONALE POLYTECHNIQUE
BIBLIOTHEQUE

Proposé par :

A. DERBAL

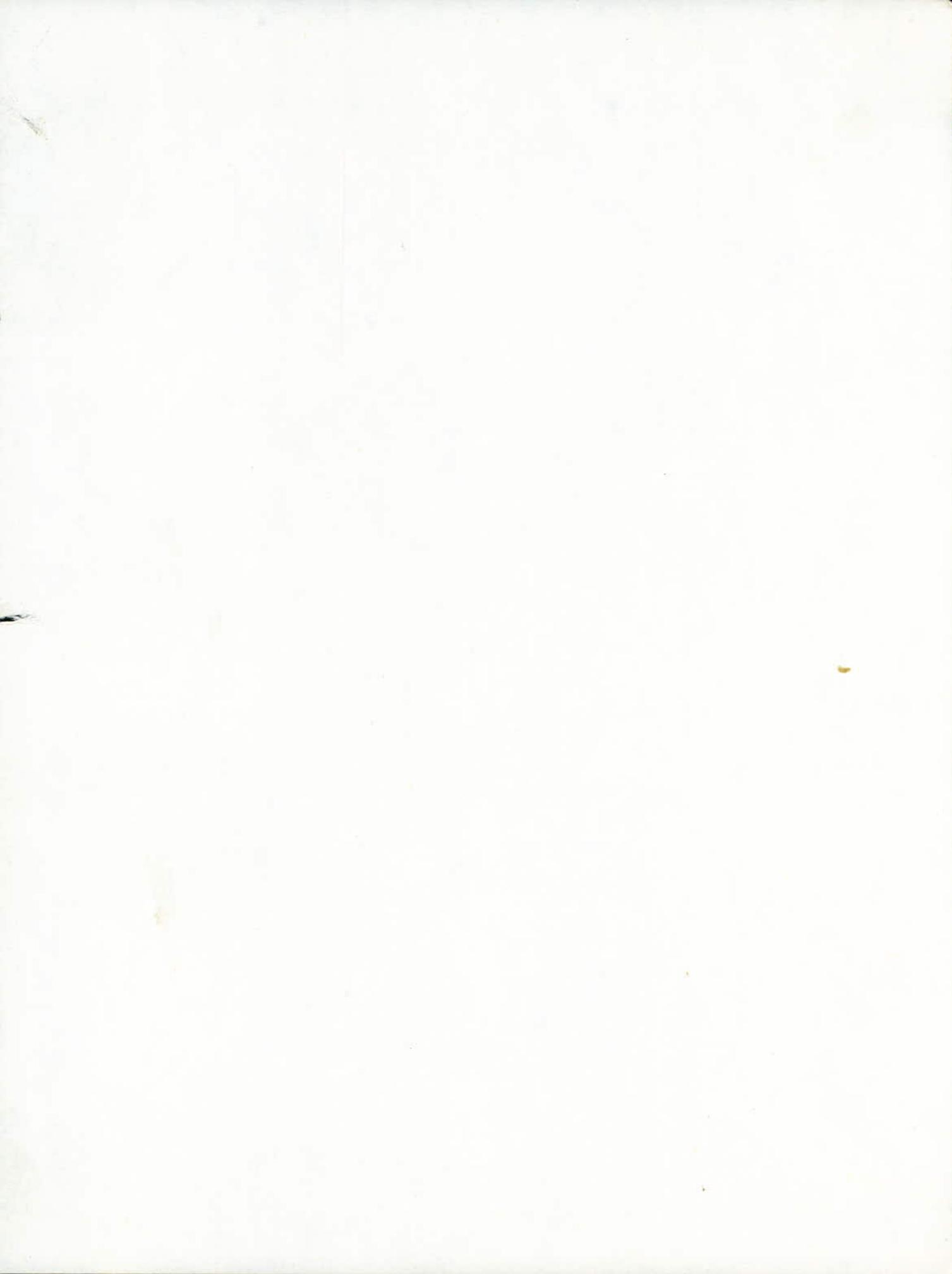
I. BOUDOUANE

Etudié et réalisé par :

S. GAOUA

M. DEBYECHE

JUIN 1982



Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE

Houari BOUMEDIENE

ECOLE NATIONALE POLYTECHNIQUE

Département d'Electronique

Projet de Fin d'Etudes

Ingéniorat d'Etat en Electronique

ETUDE ET REALISATION
D'UNE INTERFACE DE VISUALISATION
MICRO - ORDINATEUR --- CRT 606 A

Proposé par :

A. DERBAL

I. BOUDOUANE

Etudié et réalisé par :

S. GAOUA

M. DEBYECHE

JUIN 1982

REMERCIEMENTS

Au moment ou notre travail tire à sa fin, nous ressentons une joie indicible de traduire notre gratitude à tous ceux qui ont pris part à l'élaboration de ce projet.

Ainsi, nous remercions M^r B. SANSAL et M^r H. TEDJINI pour la confiance qu'ils ont placé en nous et l'agréable cadre de travail qu'ils nous ont offert.

Nous remercions M^{elles} A. DERBAL et I. BOUDOUANE pour tous les conseils et les encouragements qu'elles n'ont cessé de nous prodiguer tout au long de cette étude.

Nous remercions également M^{rs} SMARA, GRIB, HALIMI pour leur aide morale et materielle, M^{elle} Z. BENLOULOU, M^{rs} LAZIB et AEDI qui se sont occupés de la frappe et du tirage de ce projet.

DEDICACES

A La memoire de mon grand-père
A ma famille
A tous ceux qui me sont chers

Mohamed -

A la memoire de ma grand-mère
A mes parents
A mes Amis (es)

Saïd

-SOMMAIRE-

INTRODUCTION

I - PRESENTATION DU SUJET : Schéma synoptique.

II - APERCU SUR LE MICRO-ORDINATEUR.

II-1 : Le microprocesseur MC 6800.

II-2 : Les mémoires RAM et ROM.

II-3 : PIA : Etude détaillée.

III - ETUDE ET REALISATION DE L'INTERFACE.

III-1 : Liaison micro-ordinateur — PIA.

- Schéma synoptique.

- Fonctionnement.

III-2 : Liaison PIA — Carte Memoire Image.

- Schéma synoptique.

- Etude et réalisation de la Carte Memoire 64 K mots de 1 bit.

- Fonctionnement.

III-3 : Liaison Carte Memoire — Moniteur de visualisation.

- Schéma synoptique.

- Généralités sur les compteurs.

- La conversion Numérique/Analogique.

- Fonctionnement de la liaison.

- Atténuation.

IV - SCHEMA DE CABLAGE GENERAL.

V - APERCU SUR L'OSCILLOSCOPE " TEKTRONIX ".

VI - LOGICIEL.

-Organigrammes.

-Programmes de gestion.

CONCLUSION.



I N T R O D U C T I O N

Notre projet de fin d'études se situe dans le cadre d'un axe de recherche "Traitement Numerique d'image " développé actuellement au sein de la division : Electronique et Simulation du centre des sciences et de la technologie nucleaire

Il se limitera à une application précise :

La visualisation graphique sur ecran cathodique

Elle permet de restituer , sur l'ecran , sous forme d'image graphique , une information binaire rangée en memoire , pouvant constituer par exemple des caracteres Alpha numeriques ou une image quelconque .

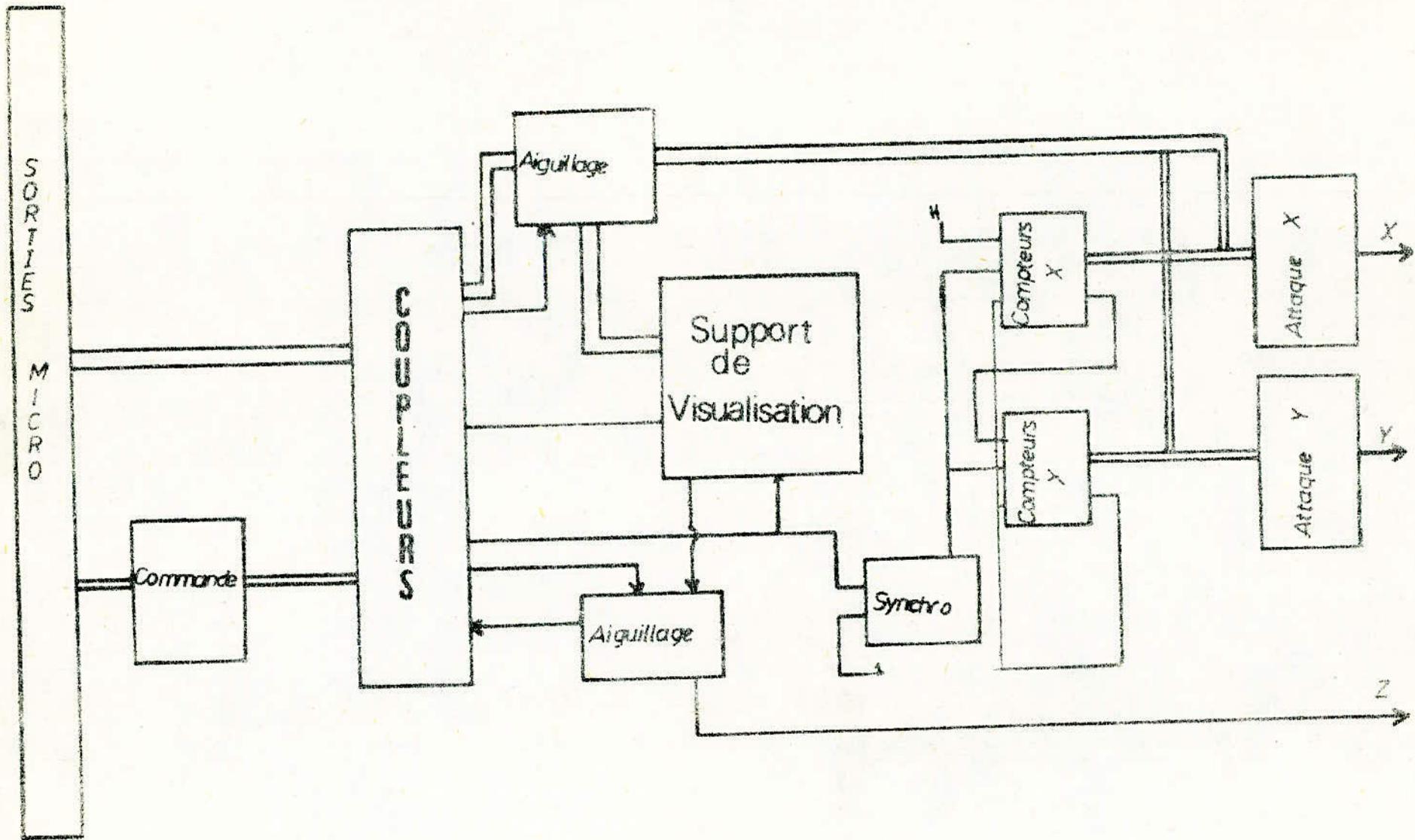
I - Presentation du sujet

Le travail réalisé en guise de projet de fin d'études consiste en l'étude et la réalisation d'une interface entre un micro-ordinateur et un moniteur de visualisation .

Cette interface comprendra essentiellement deux cartes

- Une carte Memoire Image de 64k mots de 1bit servant de support d'informations à visualiser .
- Une deuxieme carte permettant de generer des bus d'adresses ; des signaux de commande et l'echange des informations .

Le schema de cablage general est donne a la page suivante



II-- APPERCU SUR LE MICRO-ORDINATEUR UTILISE

Le micro-ordinateur de la division simulation et contrôle est à base de microprocesseurs MC 6800 de motorola. Pour cela nous jugeons utile de donner un bref rappel sur ce que nous estimons être indispensable à la compréhension de ce travail

II-- 1-- Rappels sur le microprocesseur MC 6800

1-- Généralités : Le MC 6800 est un circuit intégré traitant des mots de huit bits et constitue une unité centrale complète en un seul boîtier DIL 40 broches. Il est réalisé en technologie Mos à canal N et porte au Silicium, il est compatible TTL.....

2-- Caractéristiques essentielles du MC 6800

- Tension d'alimentation unique : +5v.
- Bus de données bidirectionnel.
- Bus d'adresse de 16 bits, espace d'adressage de 64 K octets.
- Six registres internes.
- Temps d'exécution de l'addition : 2us.
- Sept modes d'adressage.
- Pile ext^rne de longueur variable.
- Vecteur d'interruption masquable.
- Interruption non masquable.
- Possibilité d'accès direct memoire (DMA) et de configuration multiprocesseur.
- Possibilité d'arrêt et d'exécution pas à pas.
- 72 instructions de longueur variable.

3- Constitution

L'organisation générale du MC 6800 est donnée fig 1. Nous remarquons :

- L'unité de contrôle associée au registre d'instruction, analyse et decode les instructions présentes sur le bus données.
- La memoire interne à l'unité centrale représentée par six registres mis à la disposition de l'utilisateur et les amplificateurs des sorties et de données aussi appelés "buffers".

La figure 2 montre les différents registres et les lignes d'entrées/sorties accessibles.

Ces lignes d'entrées/sorties se composent des bus et des signaux suivants.

- Bus de données 8 bits bi-directionnel.
- Bus d'adresses 16 bits.
- Bus de contrôle représenté par : R/W ; $\overline{\text{IRQ}}$; $\phi 1$; $\phi 2$; et $\overline{\text{RESET}}$.
- Bus de commande : DBE ; TSC ; BA ; $\overline{\text{HALT}}$; NMI ; et $\overline{\text{RESET}}$.

Le détail des six registres internes du MC 6800 est donné fig 3.

4- Définition des signaux

La définition succincte de chacun des signaux d'entrées/sorties ainsi que le mode de brochage du MC 6800 sont donnés fig 4.

5- Le logiciel

Le micro-processeur MC 6800 est doté de 72 instructions exécutables et assemblées, suivants le mode d'adressage choisi, en 1, 2 ou 3 octets.

Modes d'adressage du MC 6800

Adressage immédiat : L'operande est contenu dans le 2ème ou 3ème octet de l'instruction selon qu'on s'adresse aux accumulateurs ou aux registres.

Les instructions correspondantes servent généralement au chargement
addition, comparaisons.....

Adressage direct : (2 octets) C'est le mode le plus simple et le plus
utilisé qui consiste à utiliser les adresses fournies sur le bus adresse pour
accéder directement à des données dans la ~~la~~ position correspondante de
la mémoire.

Adressage indexé : (2 octets) L'adresse contenue dans le 2ème octet de
l'instruction est ajoutée à l'octet de poids faibles du registre d'index.
Le résultat est utilisé pour adresser l'opérande à rechercher ou à stocker
en mémoire.

Adressage étendu : (3 octets) L'adresse en mémoire de la donnée est contenue
dans le 2ème octet (~~po~~**ids fort**) et dans le 3ème octet (poids faible) de
l'instruction.

Adressage implicite : (1 octet) Dans ~~ce~~ mode, l'opérande est indiqué par
le code opération de l'instruction.

Adressage relatif : (2 octets) L'adresse contenue dans le 2ème octet
de l'instruction est ajoutée à l'octet de poids faible du compteur de
programme plus deux. Les limites d'adressage sont donc de -126 à +126
par rapport à l'instruction courante.

Adressage d'accumulateur : (1 octet) La donnée est contenue soit dans
l'accumulateur A soit dans l'accumulateur B.

6- Instructions

Ces instructions permettent d'effectuer les opérations suivantes :

-Arithmétiques (binaire et décimale) .

-Logique.

-Décalage à droite ou à gauche.

-Chargement des registres.

-Stockage.

-Branchement conditionnel ou inconditionnel de traitement des interruptions et des instructions de manipulations de pile.

7 -La famille du MC 6800

La famille MC 6800 comprend le MPU MC 6800, des memoires dynamiques et statiques RAM MC 6810, ROM MC 6830. la necessité d'utiliser des fonctions spéciales pour le transfert des données à conduit à la conception des circuits d'interface. Ils sont de plusieurs types suivant les applications.

-L'ACIA MC 6850 (Asynchronous Communication interface Adqpter) réalise la mise en format des données et la commande pour la communication série asynchrone.

-Le PIA MC 6821 (Peripheral I_nterface Adqpter) realise l'interface des peripheriques à l'aide de deux bus de données bidirectionnels et quatre lignes de commande.

Notons que ces circuits sont entièrement programmables à travers le bus.

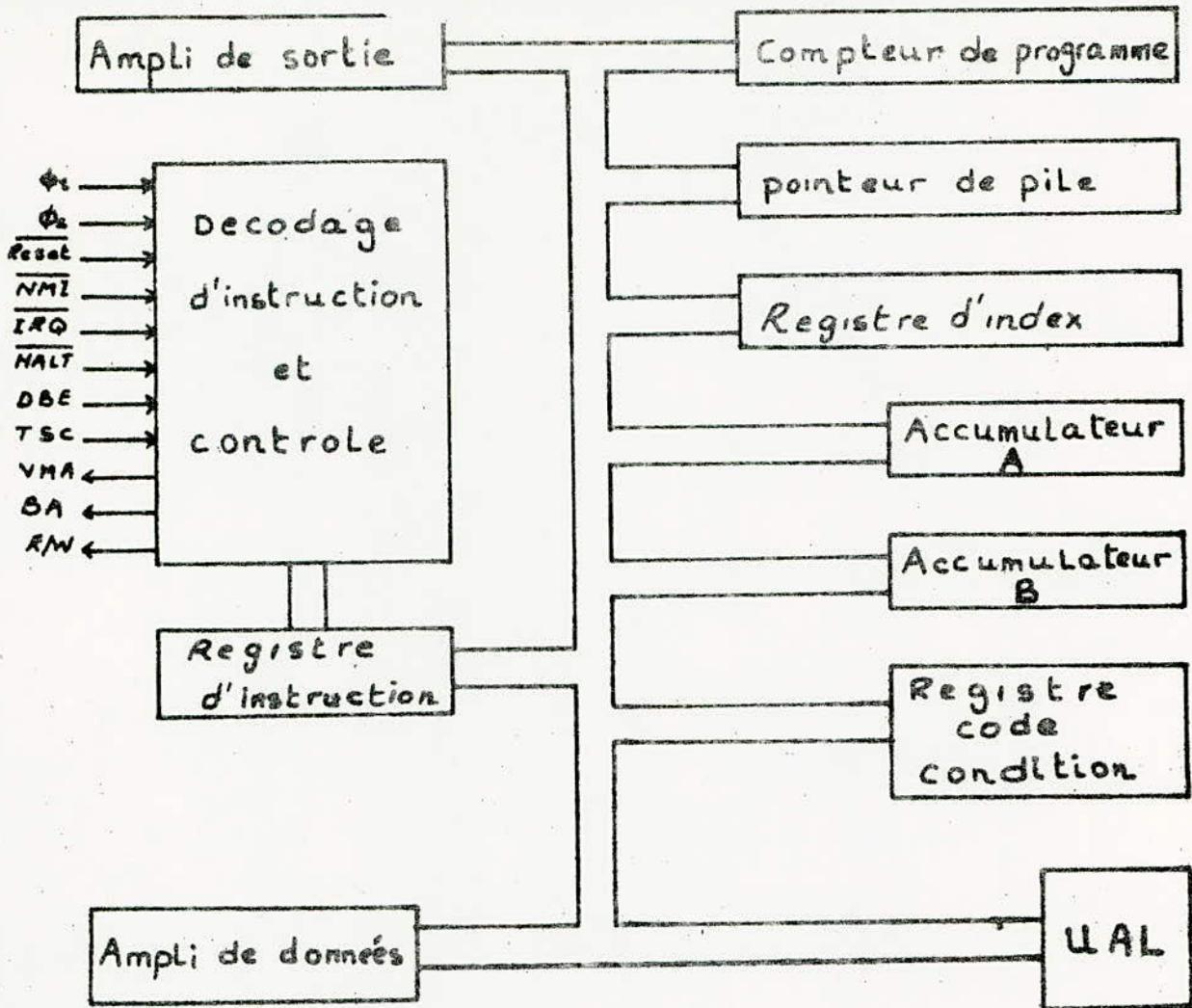


fig - 1 schema fonctionnel

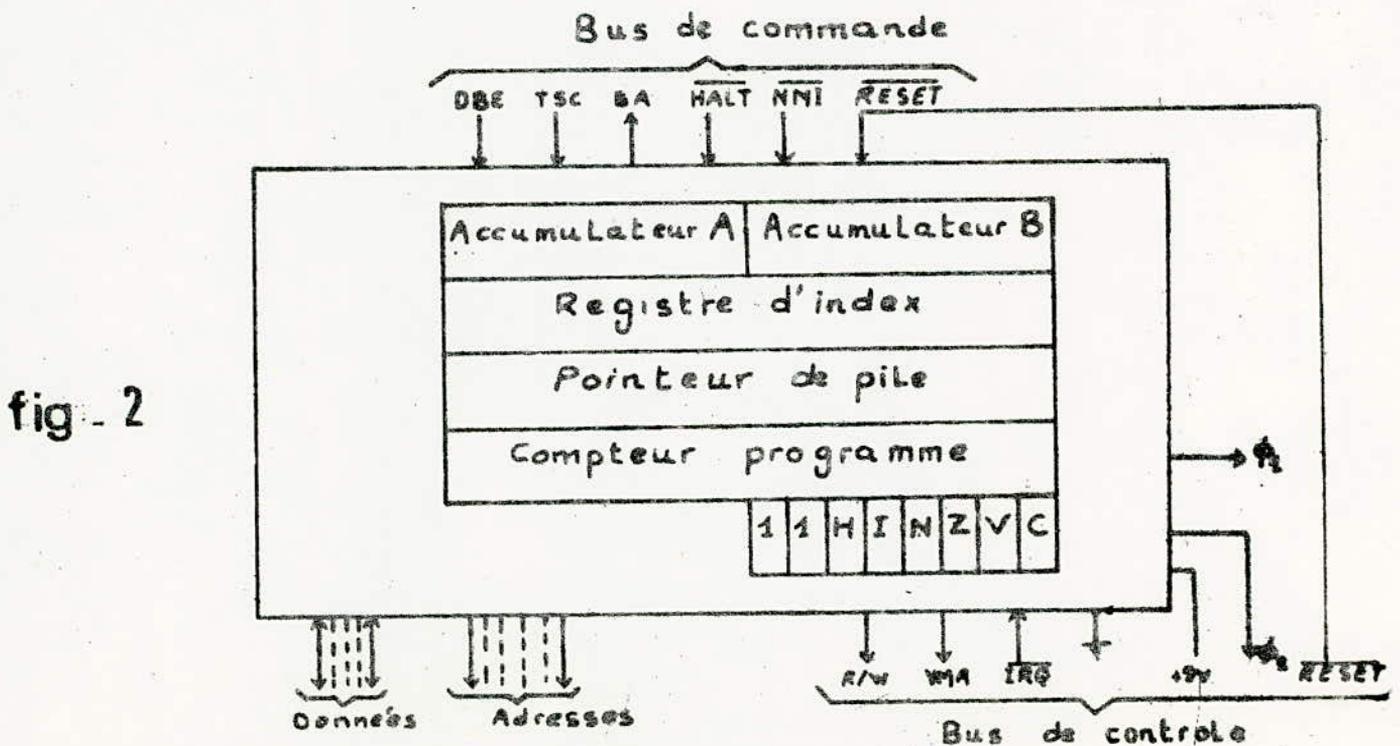
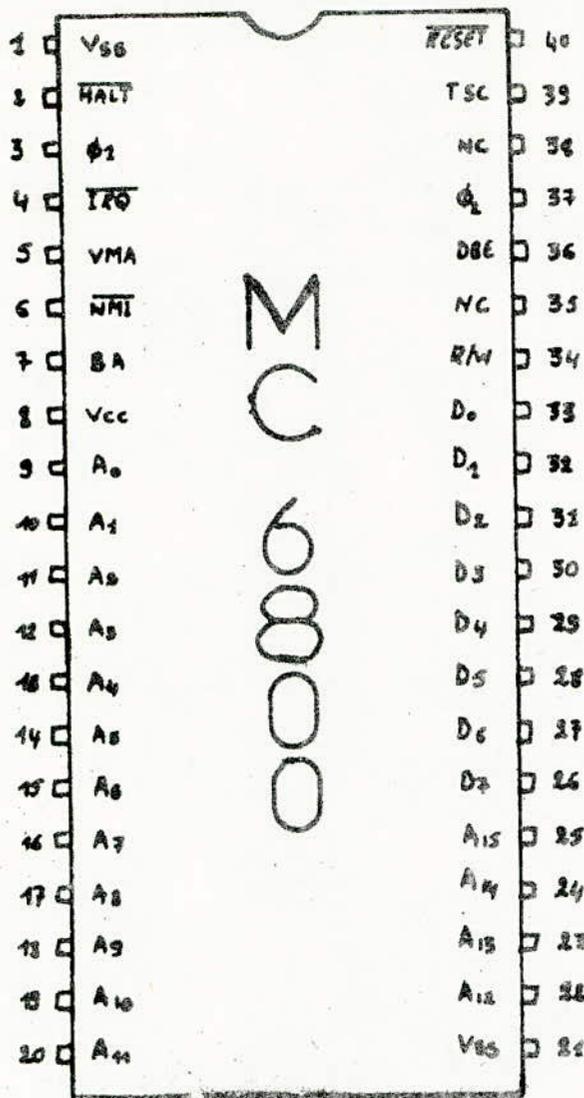


fig - 2



Commentaires

Lignes de commande

- RESET : Remise à zero
- HALT : Arrêt
- NMI : Interrup. non masquable
- IRQ : Interrup. masquable
- R/W : Lecture - écriture
- DBE : Activation bus données
- VMA : Adresse memoire valide
- TSC : Commande trois etats
- BA : Bus disponible

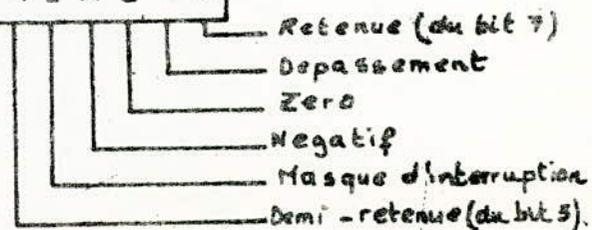
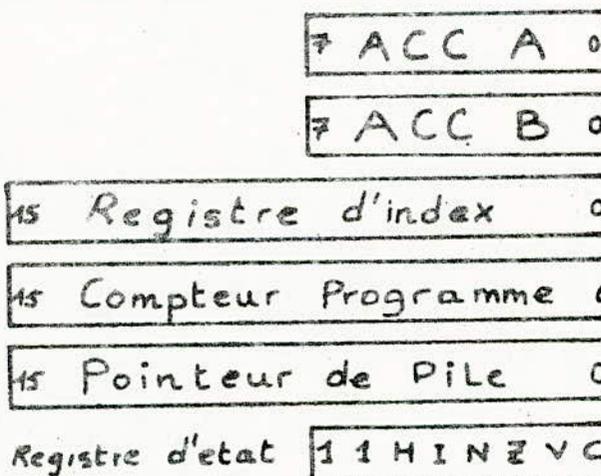
Autres Lignes

- A₀-A₁₅ : Lignes d'adresses
- D₀-D₇ : Lignes de données
- ϕ_1 et ϕ_2 : Phases d'horloge non chevauchante

fig - 4 Brochage du MC 6800

fig - 3

Régistres programmables du MPU



II-2 Les memoires ROM et RAM.

1- Les RAM MC 6810

La RAM MC 6810 est une memoire de 128 octets utilisable dans des systemes organises autour d'un bus. Elle est fabriquée dans la technologie MOS canal N, grille Silicium. Ce circuit n'a besoin que d'une seule tension d'alimentation (+5v).

De plus, il est compatible TTL, DTL et etant de fonctionnement statique, n'a besoin d'aucune horloge ou de signal de rafraichissement.

Le circuit est compatible avec la famille du MC 6800. L'extension de la memoire est possible grâce à plusieurs entrées de selection.

Schéma fig 1.

2- Les ROM MC 2708

Les ROM MC 2708 sont en réalité des EPROM. C'est à dire des memoires programmables, effaçables. Elles sont utilisables pour la mise au point de systemes et pour des applications similaires demandant une memoire non volatile qui doit être programmée périodiquement.

Une fenêtré transparente sur le boitier permet d'effacer aux rayons ultra-violetes le contenu de la memoire.

Caractéristiques :

- Organisée en 1024 octets (mots de 8 bits).
- Fonctionnement statique.
- Tension d'alimentation standards : +12v ; +5v ; -5v.
- Faible dissipation.
- Entrée de selection du boitier pour l'extension de la memoire.

-Sorties trois etats.

-Compatible TTL.

Schéma fig. 2.

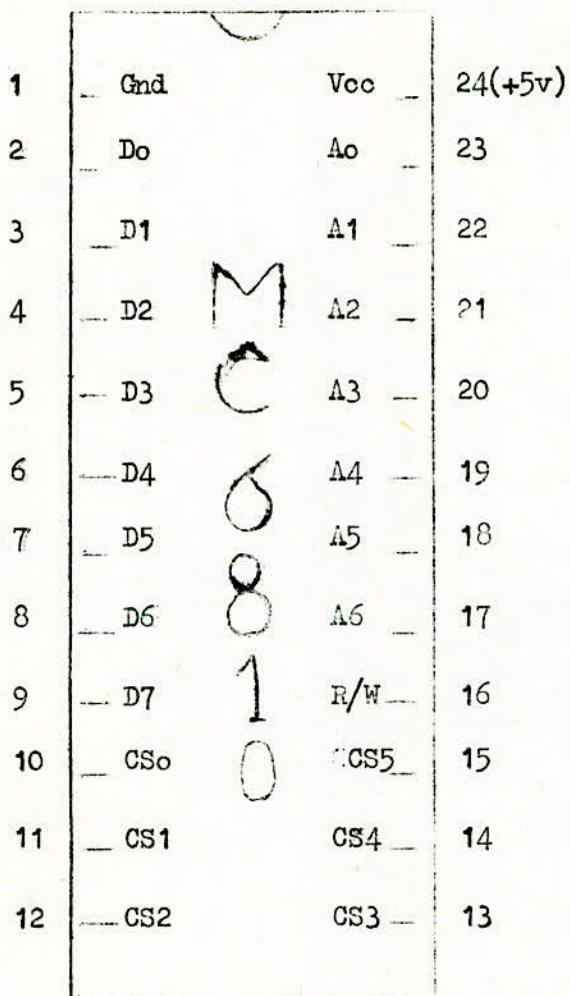


Fig 1 RAM

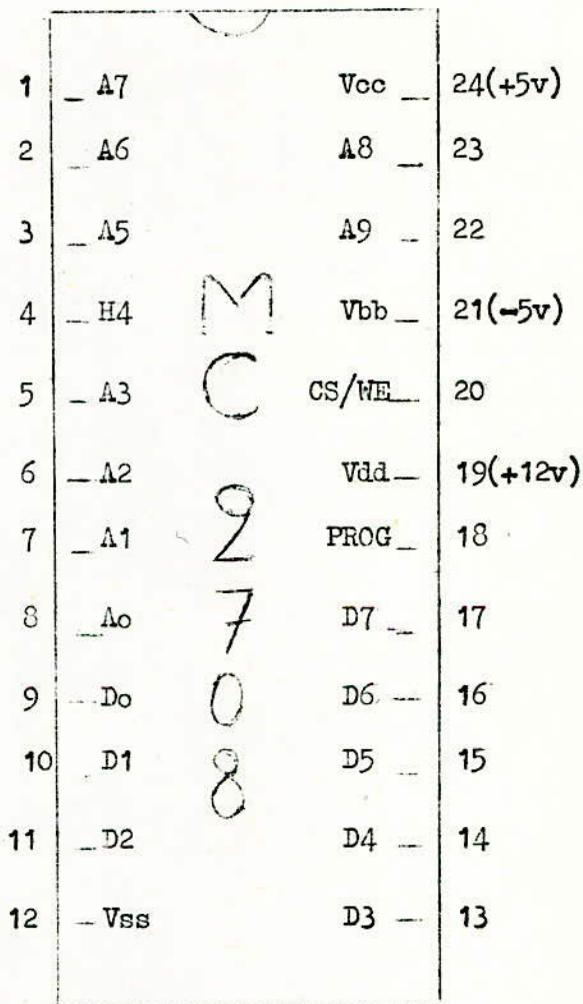


Fig 2 ROM

II-3 PIA : ETUDE DETAILLEE

L'importance particulière de ce composant dans notre réalisation nous a poussé à étudier de façon détaillée d'un des modèles les plus connus et utilisés le MC 6821. L'unité d'interface MC 6821 est un produit N-MOS logé dans un boîtier 40 broches et utilisé comme un moyen d'interface d'équipement périphérique et de signaux externes avec le MPU.

Le PIA communique avec le MPU à travers le même bus de données bidirectionnel de huit bits utilisés par les RAM et les ROM.

Il a deux bus de données périphériques bidirectionnels de huit bits séparés, pour communiquer avec le monde extérieur.

Les 16 lignes entrées/sorties peuvent être programmées pour jouer le rôle soit d'entrées, soit de sorties. En plus de ces 24 broches, le PIA possède :

- 3 broches de sélection boîtier (CS0 , CS1 , CS2)
- Une broche d'initialisation (RESET)
- Deux broches d'interruption (\overline{IRQA} , \overline{IRQB}).
- Une broche lecture/écriture (R/W).
- Quatre broches de lignes de commande (CA1 ; CA2 ; CB1 ; CB2).
- Une broche de validation (E).
- Deux broches de sélection de registre (RSO ; RS1).
- Deux broches d'alimentation (+5v et la masse).

Le schéma de brochage est donné fig 1.

Le MC 6821 a deux voies de communication, une partie A et une partie B. Chaque partie a un registre de données périphérique, un registre de direction données et un registre de commande.

1-Signaux échangés avec le système

-Entrées de sélection de boîtier CS0 , CS1 , $\overline{\text{CS2}}$ ("Chip Select")

Ces trois signaux d'entrées sont utilisés pour accéder à un PIA. Un état haut (état logique "1") doit être appliqué aux lignes CS0 et CS1 pendant qu'un état bas (état logique "0") est appliqué à la ligne $\overline{\text{CS2}}$. Les transferts de données sont alors effectués sous le contrôle de l'impulsion d'activation (E) et la ligne R/W.

-Entrées de sélection des registres : RSO et RS1.

Ces deux lignes sont utilisées en conjonction avec les registres de contrôle pour sélectionner un des registres internes du PIA. En conséquence, le PIA occupe 4 adresses mémoires.

-Ligne d'activation E : ("Enable")

C'est le signal d'activation des échanges, cette entrée est reliée à la phase ϕ_2 du MPU.

-Entrée R/W : "Read/Write".

Ce signal est produit par le microprocesseur pour spécifier le sens de transfert de données.

R/W = 0 la donnée est transférée du MPU au PIA.

R/W = 1 la donnée est transférée du PIA au MPU.

-BUS de données : D0-D7.

Les huit lignes de données bidirectionnelles permettent le transfert de données entre le MPU et le PIA. Le MPU envoie et reçoit à la fois des données vers et du monde extérieur à travers le PIA via ces huit lignes de données. Les amplificateurs de sortie de ces lignes sont des produits Tri-State (3 états) qui restent à l'état haute impédance, excepté lorsque le MPU exécute une opération de lecture du PIA.

-Ligne d'initialisation : RESET.

Cette ligne, initialise tous les registres du PIA, elle est utilisée principalement durant une operation de remise à zéro ou de mise sous tension. Elle est active à l'état bas.

-Ligne de commande d'interruption : IRQA et IRQB.

Ces 2 lignes de demande d'interruption sont destinées à interrompre l'exécution d'un programme par le MPU. Elles sont actives à l'état bas. Chaque ligne est associée à 2 indicateurs (bit 6 et 7 du registre de contrôle).

2-Signaux échangés avec le peripherique

-Le bornier A : PA0-PA7.

Chacune de ces huit lignes de données est programmable pour être soit comme une entrée, soit comme une sortie. Ceci est fait par l'intermédiaire du registre sens de transfert des données DDRA ("Data Direction Register A") A chaque bit de ce registre correspond une ligne de données vers le peripherique (Au bit zero correspond la ligne PA0 ;, bit 7 correspond la ligne PA7.)

Pour les lignes devant être en sortie, on mettra à un "1" les bits correspondants du DDRA. Pour les lignes devant être en entrée, on mettra à zero "0" les bits correspondants.

Le contenu du registre de sortie A ("OUT PUT Register A") apparait sur les lignes programmées comme sorties. A un "1" logique dans un bit du registre de sortie correspond un état haut sur la ligne correspondante et à un "0" logique correspond un état bas.

-Le bornier B : PB0-PB7.

Chacune des lignes PB0-PB7 joue le même rôle que les lignes PA0-PA7 du

du bornier A. Il y a seulement une différence de structure, les lignes PB pouvant être mises à l'état haute impédance (Tri-State) lorsqu'elles sont programmées en entrée.

3-Organisation interne

Le schéma de la fig 2 représente le synoptique du MC 6821. Nous remarquons que le microprocesseur peut adresser six registres en lecture ou en écriture.

Ces registres sont réparties en 2 groupes de 3 registres relatifs à chacun des borniers.

-Registre de commande : CRA et CRB ("Control Registers").

Ces 2 registres contiennent les paramètres de fonctionnement et permettent au MPU de contrôler le fonctionnement des lignes de commande d'interruption CA1 , CA2 , CB1 , CB2. L'état des huit bits peut être lu par le MPU, bien qu'il puisse écrire seulement dans les bits 0 à 5. La table 1 montre le format de ces registres.

-Registre sens de transfert des données : DDRA et DDRB.

Ces 2 registres permettent de définir le sens de transfert sur les lignes de données vers le périphérique.

Un "1" (respectivement "0") dans un bit de ces registres définit la ligne correspondante comme une sortie (respectivement entrée).

-Registre de données : ORA et ORB.

Ces 2 registres mémorisent les données en sortie lors d'une écriture.

4- Selection et adressage des registres

La sélection d'un de ces registres se fait par les entrées RSO et RS1 associées aux bits 2 des registres de commande.

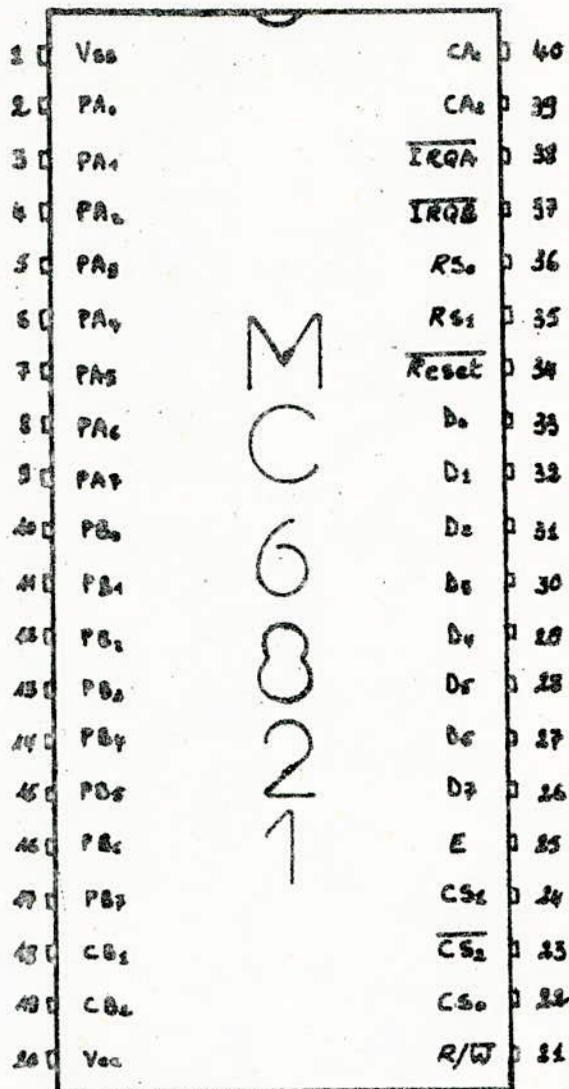


fig.1 Brochage du MC 6821

7	6	5	4	3	2	1	0
IRQA1	IRQB2	Commande de CA2			Accès à DDRA	Commande de CA1	

CRA

7	6	5	4	3	2	1	0
IRQB1	IRQB2	Commande de CB2			Accès à DDRB	Commande de CB1	

CRB

Table. Format des mots de commande

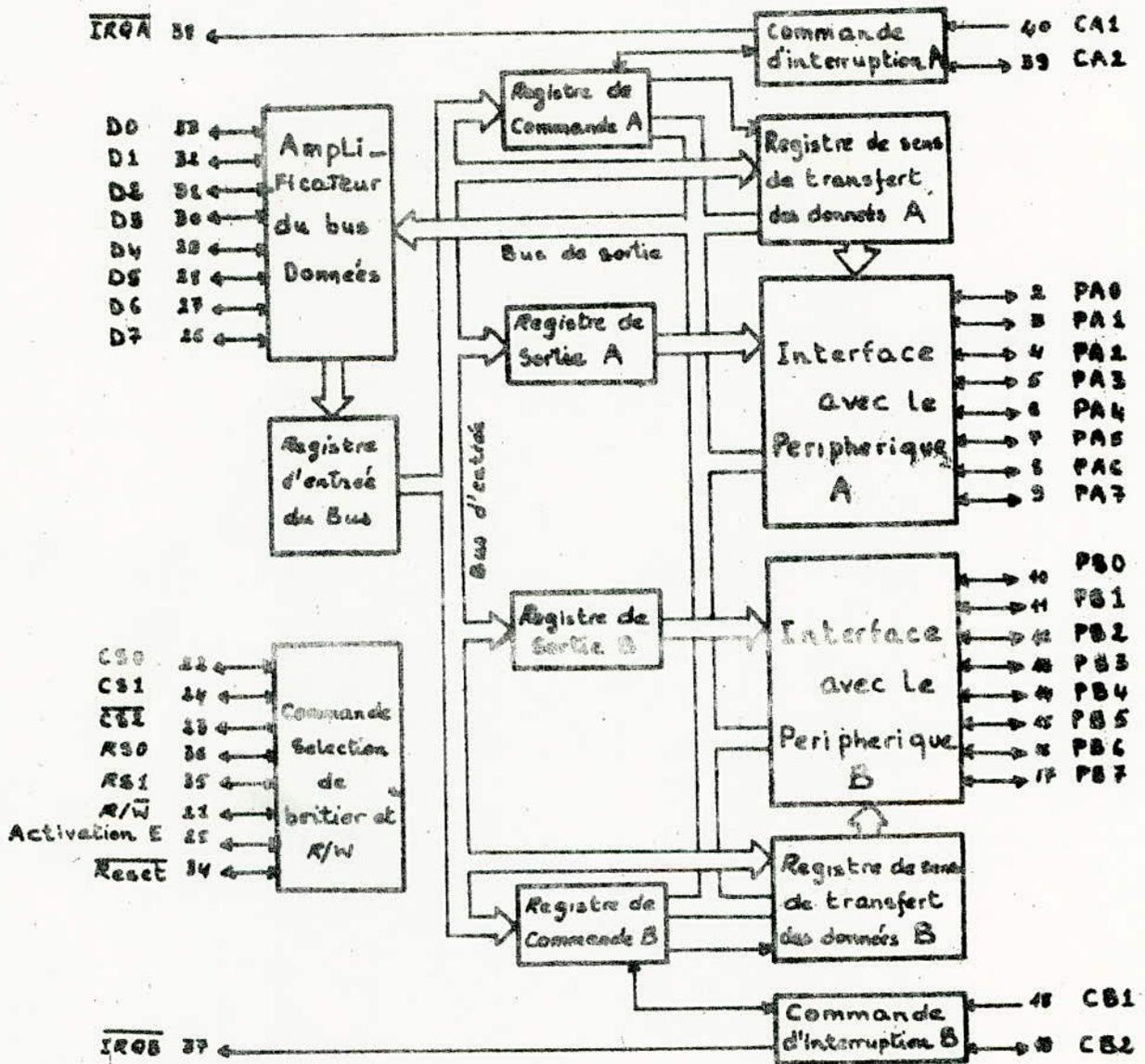


fig. 2. Schema fonctionnel

Ces 6 registres sont toutefois considérés par le microprocesseur comme 4 ~~adresses~~ memoires. Ainsi, 2 registres sont adressés directement ^{DDRB} CRA et CRB, les quatre autres ORA , ORB et DDRA , etant adressés indirectement. Le choix entre ORA , ORB et DDRA , DDRB etant fonction du bit 2 écrit au préalable dans CRA , CRB. Cet adressage est resumé sur la table 2.

RS1	RS0	CRA_2	CRB_2	Registre Sélectionné
0	0	1	X	ORA
0	0	0	X	DDRA
0	1	X	X	CRA
1	0	X	1	ORB
1	0	X	0	DDRB
1	1	X	X	CRB

Table 2 - Adressage Interne

CHAPITRE III

ETUDE ET REALISATION DE L'INTERFAÇE

L'interface realisee peut être decomposée en trois parties essentielles

Liaison Micro ordinateur ——— PIA

Liaison PIA ——— Carte Memoire Image

Liaison Carte Memoire Image ——— Moniteur de Visualisation

Ceci nous permettra de donner une explication par etape de son mode de fonctionnement.

III -1- Liaison Micro-ordinateur — PIA

Le transfert des données du micro-ordinateur vers le monde extérieur nécessite un circuit d'interface, ces circuits sont de plusieurs types suivant les applications.

Le choix de la transmission parallèle des données, la communication avec le support de visualisation nécessite un certain nombre de lignes ne pouvant être fourni par un seul PIA, nous ont conduit à utiliser deux PIA comme circuits d'interface des périphériques.

1- Realisation de la liaison Micro-ordinateur — PIA

-Schema synoptique

Le schéma synoptique de la liaison est donné à la page suivante:

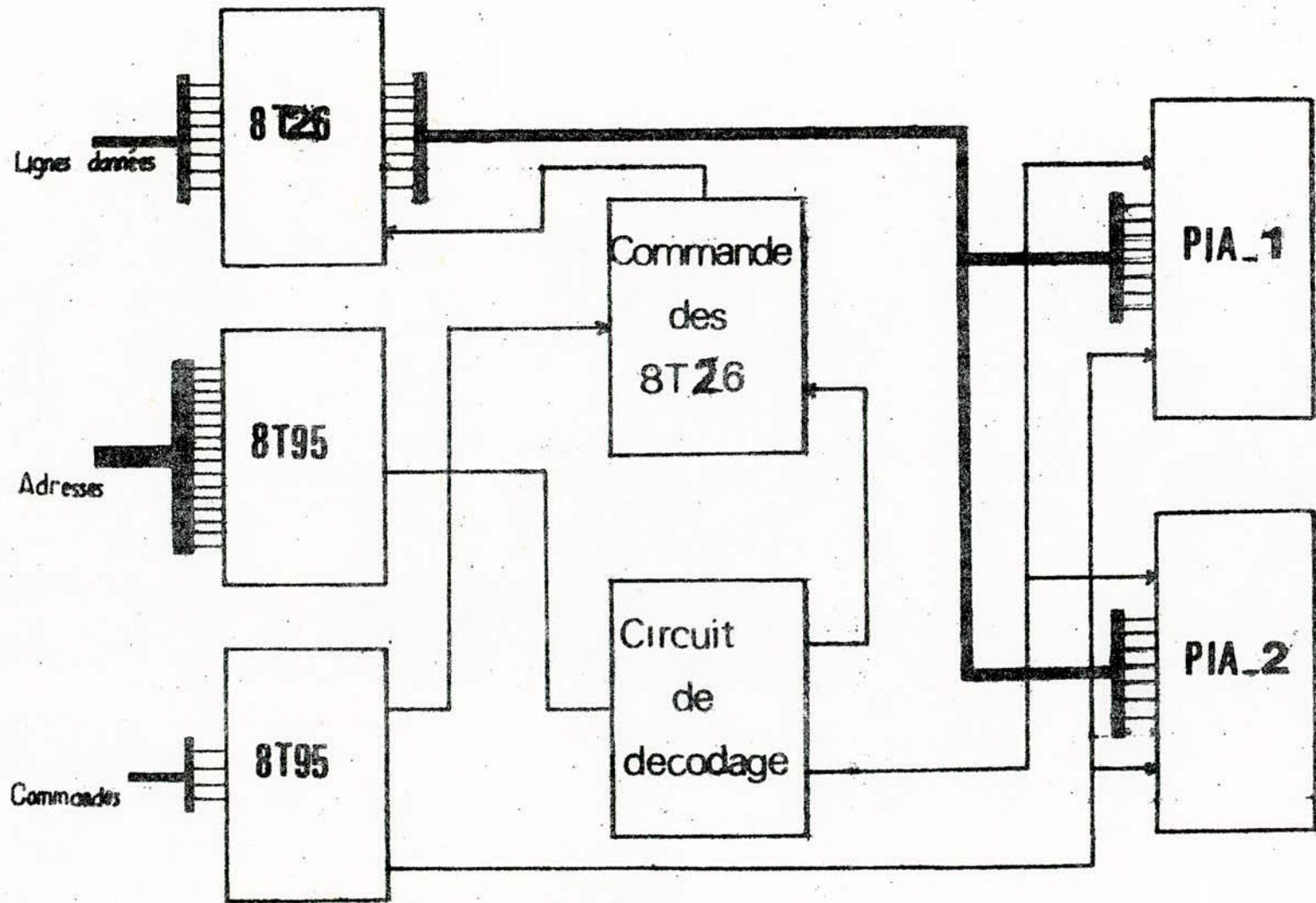
-Adressage des PIA

La sélection d'un PIA est donnée par les 3 lignes (select chip) CS0, CS1, CS2 ayant pour états respectifs 1, 1, 0.

Les adresses des registres des deux PIA sont les suivantes.

PIA (I)	8000	DDRA ou ORA.
	8001	CRA.
	8002	DDRB ou ORB.
	8003	CRB.

PIA (II)	8004	DDRA ou ORA
	8005	CRA.
	8006	DDRB ou ORB.
	8007	CRB.



Synoptique de la liaison MPU - PIA

Le choix des adresses successives des deux PIA nous permet de simplifier le circuit de decodage.

Le tableau suivant donne les lignes d'adresses des registres des 2 PIA.

	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
PIA(I)	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
PIA(II)	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Les lignes A0 et A1 sont reliées respectivement à RSO et RS1 de chaque PIA.

Les lignes A2 et A3 permettent de selectionner les 2 PIA.

Les 12 lignes du bus d'adresses constituent les entrées du circuit de decodage.

Circuit de decodage

Ce circuit est constitué de trois comparateurs de bits, ayant chacun huit entrées et trois sorties.

Quatre entrées (A) fixées par l'utilisateur, les quatre autres (B) à comparer sont reliées à quatre lignes d'adresses. Il existe trois cas de comparaisons (sortie) $A=B$; $A < B$; $A > B$; et suivant le cas désiré,

on choisit l'une de ces sorties. Dans notre cas c'est la ligne $A=B$ qui nous intéresse. Donc à l'égalité des états (ceux fixés par l'utilisateur et ceux présentés par le bus d'adresse) la sortie "6" ($A=B$) du comparateur passe à l'état "1".

-le choix de ce circuit de décodage (comparateurs) nous permet d'avoir un adressage translatable ce qui donne une large utilisation de l'interface.

Selection des PIA

D'après le tableau d'adressage ci-dessus la sélection des PIA est faite comme suit:

PIA(I) $\overline{CS1=A3}$
 $\overline{CS0=A2}$

PIA(II) $\overrightarrow{CS1=A3}$
 $\overrightarrow{CS0=A2}$

Pour avoir $\overline{CS2=0}$ pour chaque PIA, on l'a relié à la sortie du comparateur et du VMA via une porte NAND.

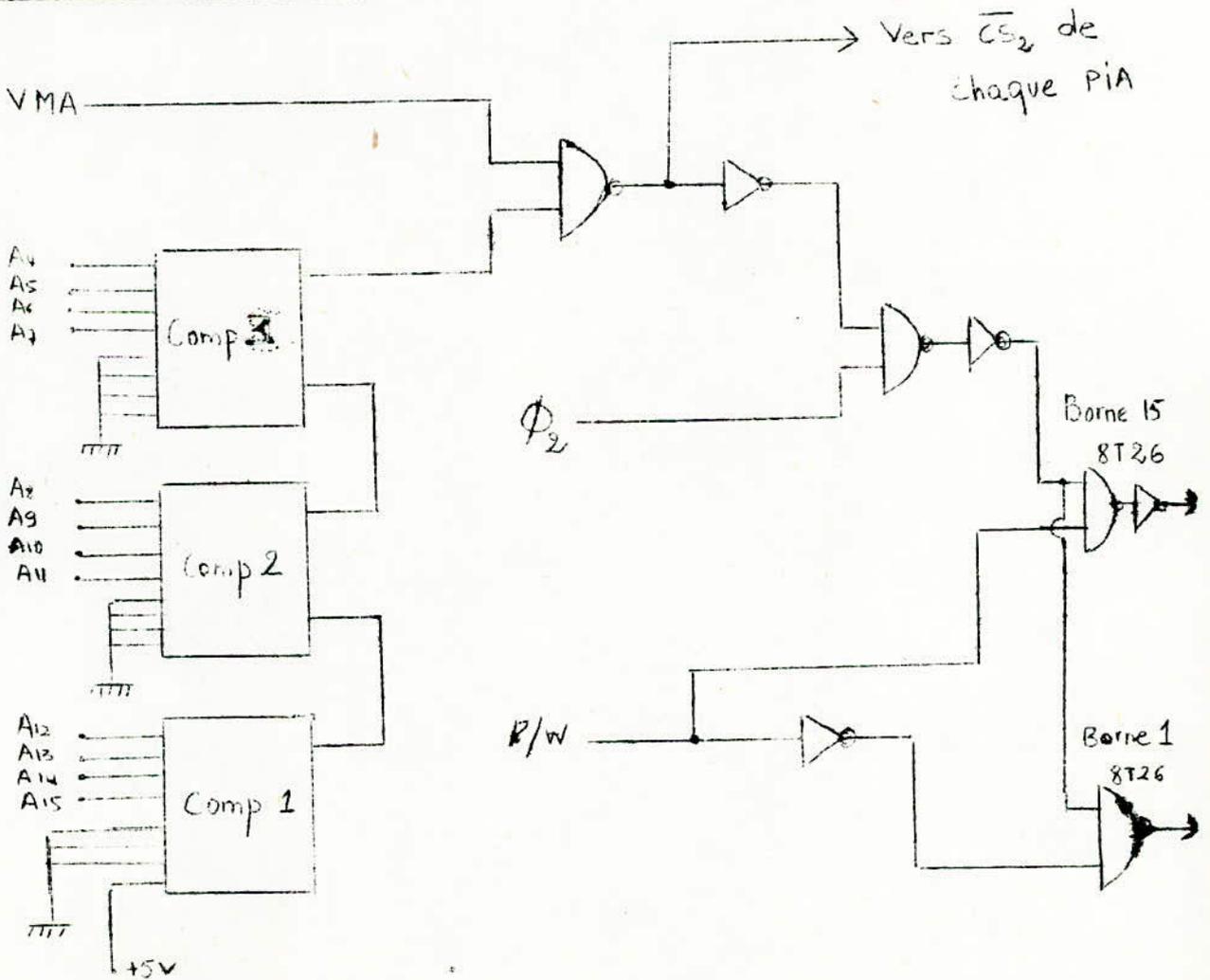
Commande des buffers à 3 états: 8T26

Les 8T26 sont des circuits amplificateurs, inverseurs, bidirectionnels. Ils sont connectés sur le bus de données. La commande de ces buffers définit le sens de transfert (lecture ou écriture). Ainsi elle synchronise le transfert des données et la sélection des PIA adressés.

Table de vérité de la logique de commande 8T26.

PIA sélectionné ϕ_2	R/W	Sortie Comparateurs	A3	A2	Borne1 8T26	Borne15 8T26	Etat des 8T26
I	1	1	0	0	1	1	Lecture
I	1	0	0	0	0	0	Ecriture
I	0	\emptyset	0	0	1	0	H.impedance
II	1	1	0	1	1	1	Lecture
II	1	0	0	1	0	0	Ecriture
II	0	\emptyset	0	1	1	0	H.impedance

Circuit de commande des 8T26:



Buffers adresses 8T95

Ce sont des circuits amplificateurs, unidirectionnels mais contrairement aux 8T26 ils sont non inverseurs.

Table de verité du 8T95

Borne 15 du 8T95	Borne 1 du 8T95	Entrée	Sortie
0	0	0	0
0	0	1	1
0	1	∅	H. impedance
1	0	∅	"
1	1	∅	"

III-2 Liaison PIA — Carte memoire image

-Le schéma synoptique est donné à la page suivante:

III-2-1 Etude et realisation de la carte memoire image 64 K mots de 1 bit.

A-Introduction:

Entièrement wrappée sur une carte MOTOROLA, cette carte memoire servira de support de visualisation. Avant de passer à sa réalisation, nous avons constaté la nécessité de rappeler quelques généralités relatives au fonctionnement des memoires à accès aleatoires (RAM).

B-Généralités pratiques sur les memoires RAM

Grâce à ces circuits integrés, il est possible de mettre des informations logiques (bit: etat 0 ou 1) en memoire. On appelle:

-Mode ecriture: l'action d'enregistrer des bits.

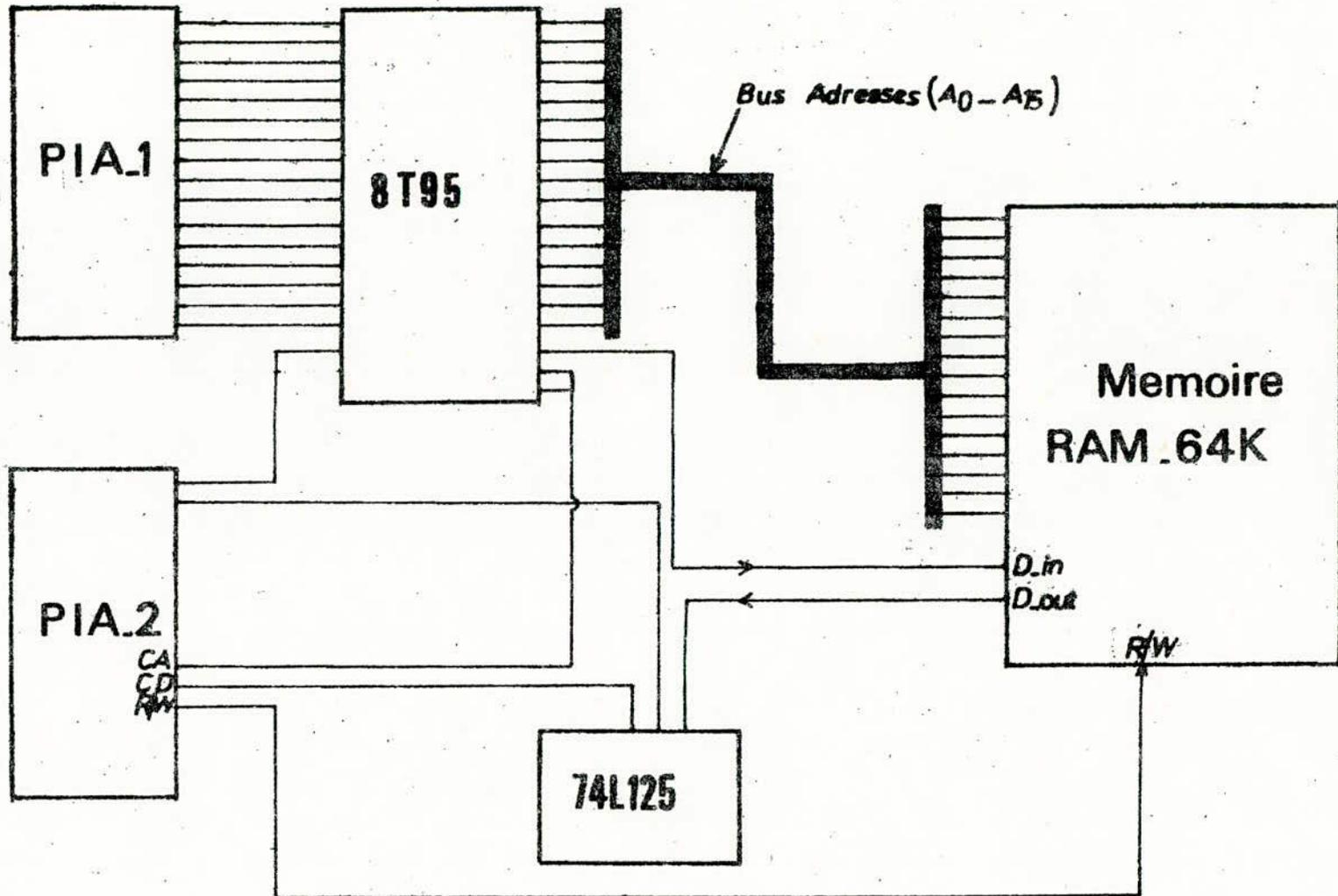
-Mode lecture : l'action de lire ces bits.

Il existe deux types de RAM.

1-Les RAM DYNAMIQUES :

Existent seulement en technologie C-MOS. L'élément de memoire est constitué par le condensateur formé de la grille et de la source d'un MOS-FET. Le "1" correspond au condensateur chargé et le "0" au condensateur dechargé. Malheureusement, ces condensateurs ont une faible valeur (moins de 30 pF) et un courant de fuite non negligable.

Si on ne recharge pas ces condensateurs, l'information mémorisée va disparaitre. Il faut donc procéder à un rafraichissement périodique.



Synoptique de la liaison PIA - Carte RAM

2-RAM STATIQUES :

Se rencontrent en technologie TTL dite bipolaire, la tension d'alimentation est +5v.

L'element de memoire est constitué par une bascule bistable et selon le "0" ou le "1" que l'on veut mémoriser, la sortie de la bascule sera mise à l'état correspondant.

3-Inconvénient des RAM

Ces memoires perdent le programme prealablement chargé dès que l'alimentation est coupée même brièvement. Pour cette raison, elles sont dites volatiles. Si on les remet sous tensions, elles se chargent par des valeurs aleatoires.

C-Description des fonctions des RAM statiques

1-Borne Read/Write (R/W) :

- Un "0" logique sur cette borne correspond au mode ecriture.
- Un "1" logique correspond au mode lecture.

2-Bornes DATA INPUT / OUTPUT

Ces bornes peuvent être communes ou séparées. Les données sont des informations electriques (bits "0" ou "1") que l'on désire écrire ou lire dans la memoire. Il est bien évident que pour faire apparaitre des données en sortie, il faut d'abord les enregistrer.

Avant cette opération, il faut preparer le mot (le programme), c'est à dire présenter sur chaque ligne d'entrée de données (DATA INPUT) les informations desirées ("0" ou "1").

Quand ces informations d'entrée sont prêtes, on peut les enregistrer

en appliquant une impulsion "0" logique sur la borne R/W. On aura donc écrit un mot sur une ligne.

Dès la fin de l'impulsion, le code écrit (programme) apparaît sur la sortie, on peut contrôler ainsi le mot que l'on vient d'écrire.

3-Bornes adresses :

Le programme à enregistrer est constitué de mots. On écrira ces mots (qui auront tous la même longueur) les uns après les autres.

En résumé, on trouvera un mot par adresse. Ainsi avec une mémoire qui possède 4 bornes adresses, on aura $2^4 = 16$ positions adresses, soit 16 mots à écrire.

Par ailleurs, pour passer d'une ligne à une autre, il faut piloter la mémoire par un programme ou par un circuit intégré compteur qui à chaque instruction ou impulsion de changement de ligne avancera d'un pas.

Il est à noter que le compteur doit avoir le même nombre de sorties que de nombre de lignes d'adressage de la RAM.

Suivant la RAM utilisée, la longueur du mot à écrire sur une ligne (adresse) peut être de 1 bit, 2 bits, 4 bits,

On remarque aussi que la plupart des RAM, les lignes de données d'entrées et de sorties sont distinctes.

4-Borne chip Enable (CE) :

-Si un "1" logique est présent sur cette borne, les sorties se trouvent en haute impédance, de ce fait la RAM ne peut être ni lue, ni écrite.

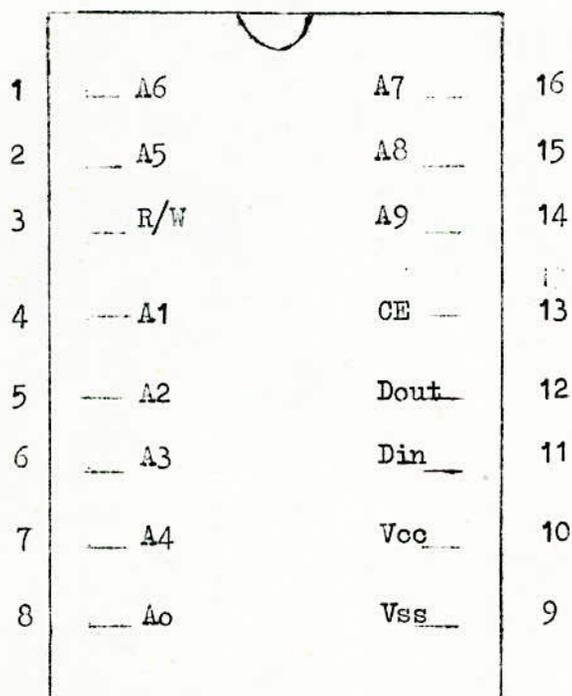
(on utilise cette borne pour grouper plusieurs RAM).

-Si un "0" est present, la RAM est en ordre de fonctionnement, on peut ecrire ou lire.

C- Les memoires utilisees : RAM MCM 2102

Ce sont des memoires statiques ayant une capacite de 1024 mots de 1 bit et necessitant une alimentation de +5v. Elles ont un temps d'accès de 250ns et disposent de 10 lignes adresses (A₀ , A₉), une ligne de selection de **boitier** (CE) et deux lignes de donnees (entree et sortie).

Comme le montre la figure suivante:



Comme nous l'avons dit auparavant, cette carte memoire à une capacité de 64 K mots de 1 bit. Elle necessite par conséquent 64 boitiers RAM RAM MCM 2102.

E-Adressage de la carte

1- Les signaux utilisés sont :

-Bus adresse :

Il comporte 16 lignes amplifiées par des buffers 3 etats 8 T95; une partie de ce bus (A_{10} , A_{15}) est utilisée pour l'adressage externe (les 64 boitiers), l'autre pour l'adressage interne.

-Bus de données :

La DATA INPUT est amplifiée par un 8T55 et la DATA OUTPUT par un buffer 74 LS 125.

-Signal de lecture-écriture (R/W) :

Il sert à determiner le sens de transfert des données (lecture ou ecriture)
Il est relié directement à chaque boitier RAM.

2-Circuit de decodage :

Les 10 lignes (A_0 , A_9) sont reliées à chaque boitier. La selection d'un des boitiers parmi les 64 est faite grâce à un circuit de decodage composé de 4 decodeurs 74LS154 (voir schéma de brochage du decodeur). Chacun de ces decodeurs monolithiques a 4 lignes d'entrées et 16 sorties compatibles aux montages. Il est utilisé pour decoder 4 entrées (codées binaire) en une "1" parmi 16 à la sortie, quand les deux entrées G_1 et G_2 sont à l'etat bas, comme le montre la table de verité.

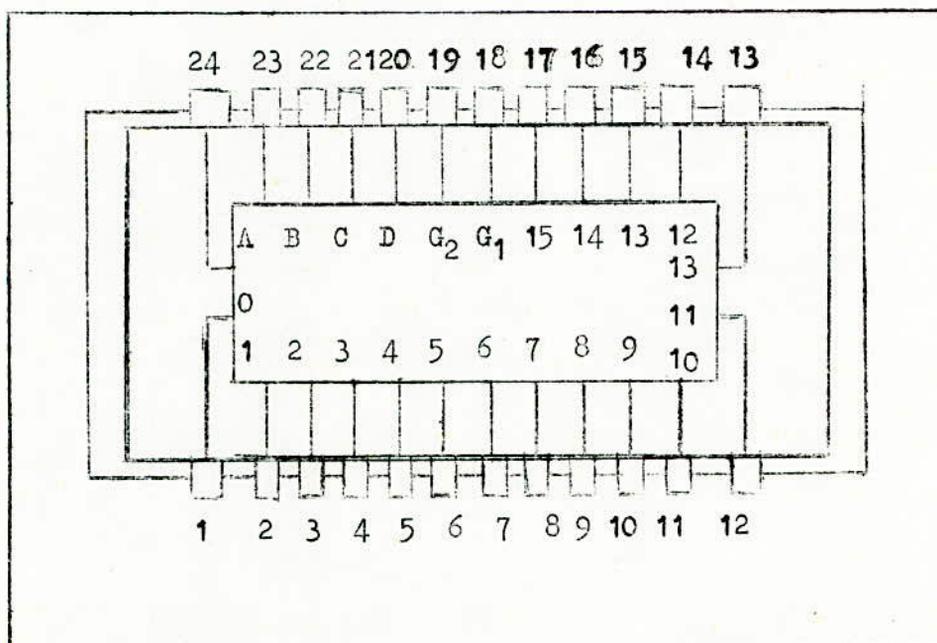
Entrées						Sorties															
G	B ₂	D	C	B ₁	A	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
L	L	L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	H	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	L	L	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	L	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H
L	L	H	L	L	L	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H
L	L	H	L	L	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H
L	L	H	L	H	L	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H
L	L	H	L	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H
L	L	H	H	L	L	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H
L	L	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H
L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H
L	H	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
H	L	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
H	H	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H

Table de Vérité

H: état haut

L: état bas

X: état indifférent

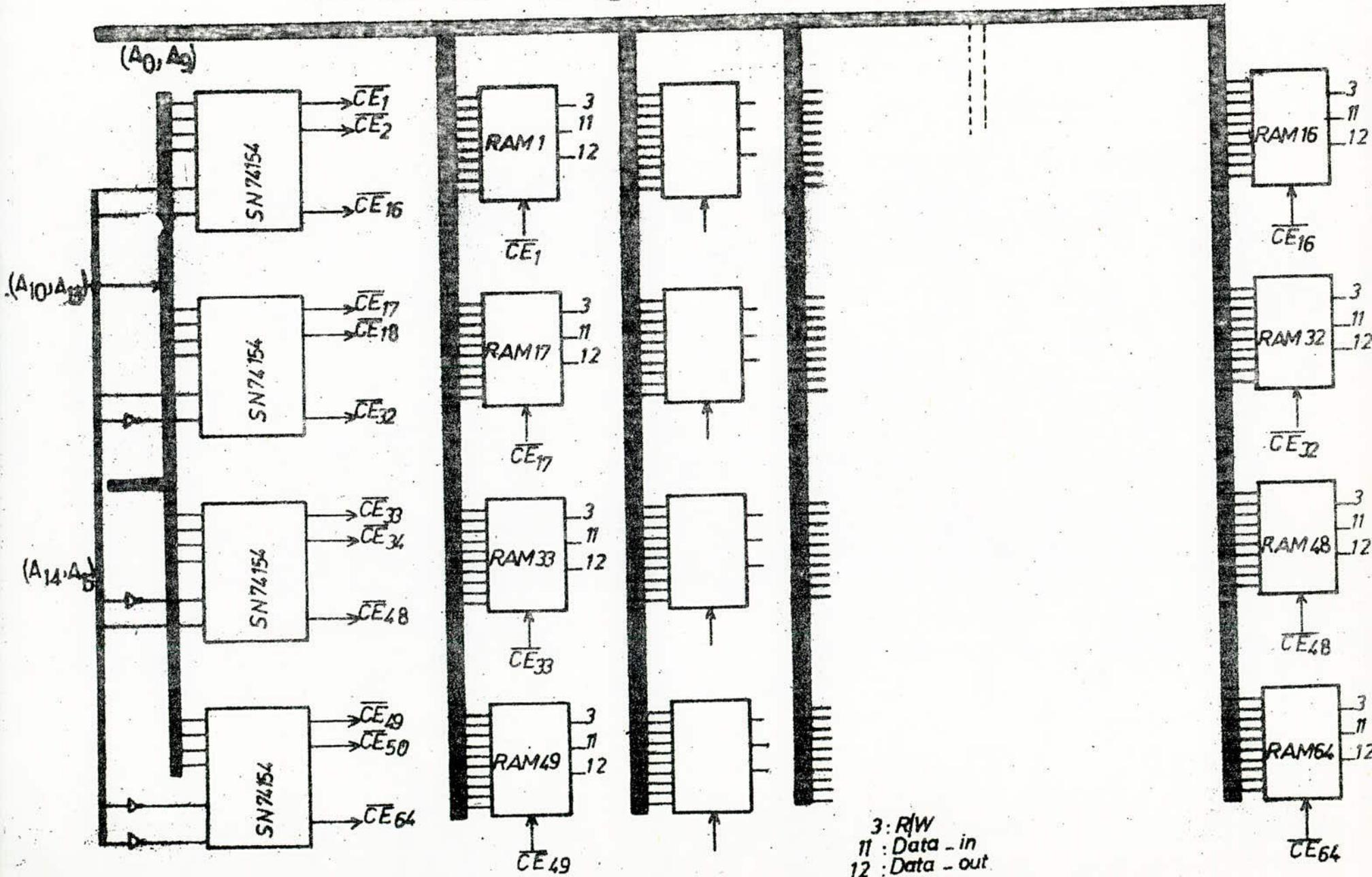


BROCHAGE DU DECODEUR 74LS154

Les 4 lignes (A_{10} , A_{13}) sont reliées directement aux 4 entrées A, B, C, D de chaque decodeur. La combinaison de (A_{14} , A_{15}) selectionne 1 des 4 decodeurs.

A_{14}	A_{15}	Decodeur selectionné
0	0	1
0	1	2
1	0	3
1	1	4

Schema de Cablage de la Carte RAM



Pour cette liaison la carte memoire image constituera le peripherique

La communication entre le MPU et la carte memoire image (peripherique) à travers les PIA necessite 21 lignes, qui seront choisies parmi les 32, dont nous disposons à la sortie des deux PIA utilisés.

1- Choix des lignes d'adressage et de commande

La capacité de la memoire image etant de 64 K de un bit, son adressage necessite 16 lignes.

Les deux ports A et B du 1er PIA ont été pris pour cet usage.

La ligne PA0 du deuxième PIA est reliée au bit Data INPUT (Donnée en entrée) de la carte memoire. Elle permet le transfert de la donnée, du MPU vers la carte (c'est le mode ecriture).

Ces lignes (les deux ports A et B du 1er PIA et la ligne PA0) passent à travers des buffers trois etats (8T95) commandés par la ligne PBO, dans le but de faire un aiguillage indispensable aux deux modes qui sont

-Mode ecriture memoire par le MPU.

-Mode visualisation de son contenu sur le moniteur.

Si PBO = 1 les buffers sont ouverts (en haute impedance)

Si PBO = 0 les buffers sont passants, l'adressage de la carte memoire peut avoir lieu.

-La ligne PA1 est reliée au bit Data OUTPUT (Donnée en sortie) de la carte memoire. Cette ligne passe à travers un buffer (74125) commandé par la ligne PB1 (commande donnée).

Si PB1 = 1 la Data OUTPUT est transférée vers le moniteur

Si PB1 = 0 elle est transférée vers le MPU.

La ligne PB2 est reliée à R/W de la carte memoire.

Si PB2 = 1 on a le mode lecture de la carte memoire.

Si PB2 = 0 mode ecriture de la carte memoire.

Ayant defini le rôle de chacune des lignes necessaires à la communication et à l'aiguillage, nous pouvons passer à la séquence de fonctionnement de la liaison après avoir au préalable predisposé le système c'est-à-dire initialiser les PIA selon le mode de fonctionnement voulu.

2 - Programme d'initialisation du PIA I

Les deux ports A et B sont programmés en sortie pour permettre

l'adressage de la carte memoire par le MPU. Pour cela

--on remet à zero les registres de commande des deux ports par une instruction CLEAR.

--on determine le mot de direction qui sera envoyé dans le registre de direction. A cet effet on ecrira ce mot (FF) dans L' ACC A par une instruction LDA A puis on transfèrera ce mot dans le registre de direction par l'instruction STA A.

Enfin pour avoir accès aux registres de sorties (ORA et ORB) on met les bits 2 des registres de commande à 1.

3 - Programme d'initialisation du PIA (II)

Les lignes PA0 et PA1 sont programmées respectivement comme sortie, entrée.

Les lignes PB0 , PB1 , PB2 , sont programmées en sorties.

De la même manière, pour avoir accès aux registres de sorties (ORA et ORB), on met les bits 2 des registres de commande des deux ports à 1.

Voir programmes d'initialisation des PIA

INIT I et INIT II.

Après avoir initialiser les deux PIA, on passe au mode écriture mémoire.

A cet effet les bits de commande seront programmés comme suit:

-PB0 est mis à zero, les buffers 8T95 sont passants l'adressage peut avoir lieu.

-PB2 est mis à zero, permet l'écriture de la mémoire.

Voir programme d'écriture mémoire par le MPU (PROG 1).

Nous faisons remarquer que ce programme permet le transfert de données du MPU vers la carte par zones de 8 K.

Une fois la carte mémoire écrite, on passe à la phase lecture par le MPU qui n'est en fait qu'une manière simple, nous permettant de vérifier ce que l'on a écrit dans cette carte.

Pour cette phase les bits de commande sont programmés de la manière suivante

-PB0 est mis à zero

-PB1 est mis à zero, dans ce cas le buffer 74125 ne laisse passer la donnée qu'en direction du MPU.

-PB2 (R/W) est mis à 1, mode lecture de la carte.

Voir programme de lecture mémoire par le MPU (PROG 2).

De même ce programme permet la lecture de la carte par zones de 8 K.

III- 3 Liaison Carte Memoire ——— Moniteur de visualisation.

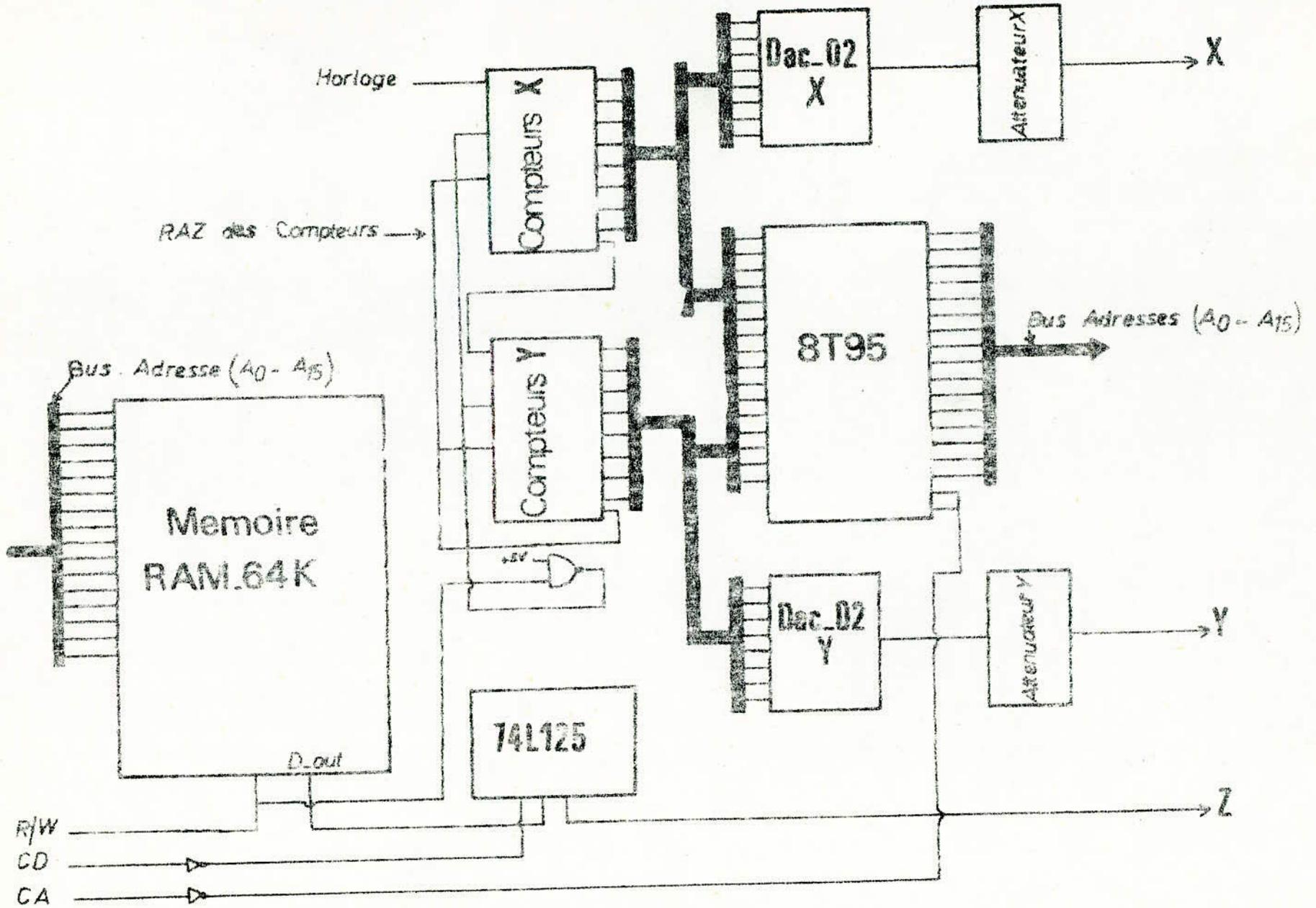
-Le schéma synoptique est donné à la page suivante.

Cette liaison permet de visualiser sur le moniteur le contenu de la carte memoire préalablement enregistré.

Elle comporte essentiellement

- un système de comptage
- une conversion numérique/analogique suivie d'une atténuation.

Avant de passer à son mode de fonctionnement nous donnons quelques généralités relatives aux systèmes de comptage (compteurs) et à la conversion N/A ainsi que nous présenterons les types de circuits utilisés.



Synoptique de la liaison Carte RAM . VISU

Compteurs

Généralités:

On emploie le terme compteur pour définir un dispositif séquentiel dont les états successifs représentent un nombre N qui évolue chaque fois qu'une **impulsion** se présente sur l'entrée de comptage.

On classe les compteurs suivant plusieurs modes:

-Mode de fonctionnement-

- Compteurs synchrones
- Compteurs asynchrones

-Sens de comptage

- Sens direct ($0 \rightarrow q$)
- Sens inverse ($q \rightarrow 0$)
- Compteurs réversibles ou compteur-décompteur.

-Code utilisé

C'est généralement un code standard, les plus utilisés étant le code binaire et le code B.C.D (binary coded decimal). On trouve aussi:

- les compteurs gray (utilisant le code gray).
- les compteurs spéciaux: en anneaux, etc.....

Compteurs binaires

Les compteurs binaires sont des circuits logiques séquentiels composés d'une suite de bistables; ceux-ci peuvent être de différents types: JK ; RS ; D ; Puisque les bistables ne fonctionnent qu'en tout ou rien, ces compteurs ne peuvent compter qu'en binaire.

Compteurs binaires synchrones

Toutes les bistables reçoivent en même temps le même signal d'horloge H.

Compteur utilisé : SN 74 193

C'est un compteur-decompteur binaire 4 bits. Il possède :

-Une entrée "clear" dont le passage à "1" positionne les sorties au niveau bas "0".

Le compteur change d'état sur les transitions positives du signal d'horloge. La remise à zero (entrée "clear") est prioritaire sur tous les autres signaux.

-Une entrée "Load" dont le passage à zero permet le chargement en parallèle des sorties par les datas présentes à l'entrée du compteur.

Dès que le "load" est au niveau "1", le comptage ou le décomptage peut avoir lieu selon qu'une impulsion est envoyée sur l'entrée "count up" ou sur l'entrée "count down". En mode de comptage : l'entrée "count down" doit être nécessairement à "1", l'impulsion est envoyée sur l'entrée "count up". En mode de décomptage : l'entrée "count up" est à "1", l'impulsion est envoyée sur l'entrée "count down".

-Deux sorties :

Carry out put pour le compteur

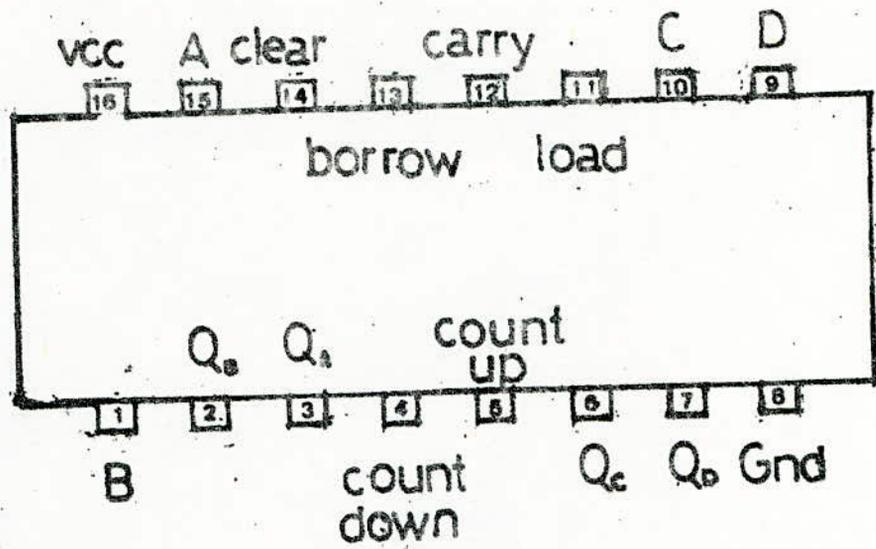
Borrow out put pour le décompteur

Elles sont commandées par les entrées du comptage (count up) ou de décomptage (count down).

Ces deux sorties donnent une impulsion négative dont la largeur est celle de l'impulsion d'horloge.

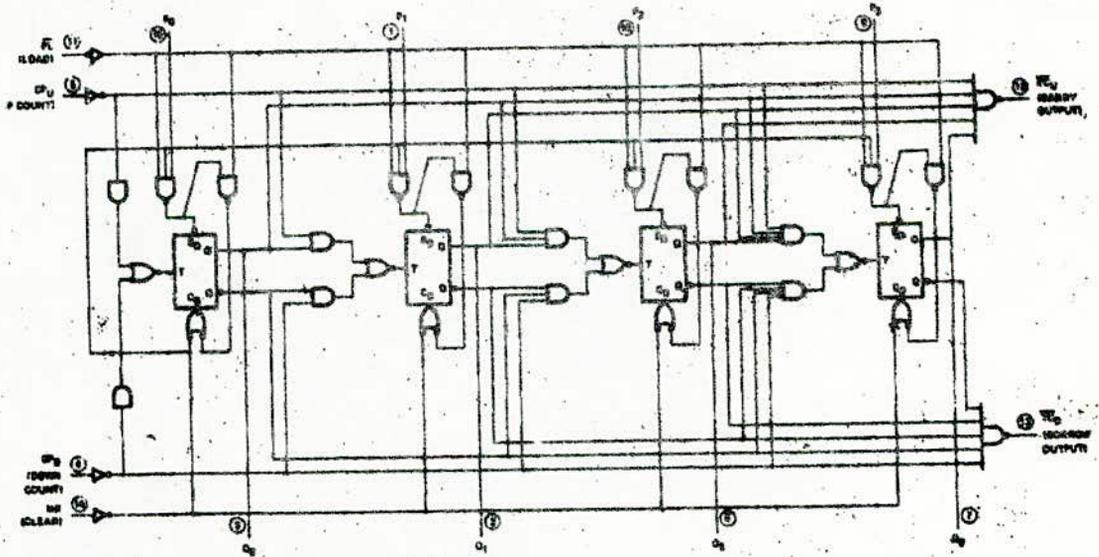
Elles sont destinées à la mise en cascade de plusieurs compteurs de 4 bits.
En conséquence, elles jouent le rôle d'horloge pour le compteur ou de compteur
suivant.

fig-1



brochage du SN74193

fig 2



schema interne du SN74LS193

VCC = Pin 16
GND = Pin 8
○ = Pin Number

La conversion numérique-analogique

Généralités :

Les systèmes de traitement de l'information peuvent se décomposer en systèmes analogiques et systèmes numériques. Les paramètres sur lesquels travaillent les systèmes analogiques varient d'une manière continue. Par opposition, les systèmes numériques travaillent sur des variables discrètes appelées nombres.

Tout traitement numérique d'un signal analogique nécessite une opération préliminaire de conversion analogique-numérique. De même tout traitement analogique d'une suite de nombres nécessite donc une opération de conversion numérique-analogique.

Lors du traitement, on préfère généralement travailler avec des signaux numériques (Nombres), vu qu'ils sont beaucoup moins sensibles aux imperfections des systèmes de transmission (distorsion, bruit).

Pour ces signaux, il suffit de détecter des impulsions pour posséder l'information et la mémoriser pour la conserver.

Convertisseurs numériques/analogiques

Définition:

un convertisseur numérique/analogique (CNA) est un dispositif qui reçoit une information numérique sous forme d'un mot de n bits et qui la transforme en un signal analogique. C'est donc un système hybride. Il fait correspondre à l'une des 2^n combinaisons binaires possibles à l'entrée (correspond à un signal d'entrée de n bits), une parmi 2^n tensions discrètes obtenues à partir d'une tension de référence U_{ref} .

Le nombre N que l'on veut convertir s'exprime en fonction des puissances entières de 2.

$$N = d_1 2^{n-1} + d_2 2^{n-2} + \dots + d_n 2^0$$

di peut prendre les valeurs 0 ou 1.

$$U = N U_{ref} \quad (\text{tension delivrée par le CNA})$$

$$U = d_1 U_{ref} 2^{n-1} + d_2 U_{ref} 2^{n-2} + \dots + d_n U_{ref} 2^0.$$

Pour un nombre inférieur à l'unité, il s'écrit de la manière suivante

$$N' = \frac{N}{2^n}$$

donc
$$U' = N' U_{ref} = \frac{N}{2^n} U_{ref}$$

d'ou
$$U = \sum_{i=1}^n d_i \frac{U_{ref}}{2^i}$$

on définit ainsi une fonction de transfert pour le CNA. La tension de reference peut être constante, on a alors un CNA classique, ou bien variable on parle alors de convertisseur multiplicateur ou plus simplement de multiplicateur N/A.

Caractéristiques d'un CNA

-Fonction de transfert ideale (sans erreur): Elle est donnée par

l'expression:
$$U = U_{ref} \left[\frac{d_1}{2} + \frac{d_2}{2^2} + \dots + \frac{d_n}{2^n} \right]$$

-Amplitude de la tension de sortie Umax: correspond à la variation maximale de la tension de sortie lorsque tous les bits passent de l'etat "0" à l'etat "1" et definit la dynamique du convertisseur.

-Pour un convertisseur unipolaire : $U_{max} = U_{ref} \left(1 - \frac{1}{2^n}\right)$

-Pour un convertisseur bipolaire : $U_{max} = U_{ref} \left(2 - \frac{1}{2^n}\right)$.

-Temps de conversion: C'est le temps necessaire pour que le signal de sortie atteigne la valeur desiree a l'erreur specifiee. Il depend des elements utilises interrupteurs, amplificateurs.....

-Resolution d'un CNA: Elle est donnee par le nombre de bits que le convertisseur peut accepter. On peut la definir comme etant la plus petite variation de l'amplitude de la tension de sortie que l'on peut obtenir compte-tenu du nombre de bits impose a l'entree.

$$r = \frac{1}{2^n}$$

ou comme le rapport de la plus petite quantite que l'on peut obtenir a la tension maximale

$$r = \frac{U_{ref}}{2^n} \frac{1}{U_{ref} \cdot \frac{2^n - 1}{2^n}} \frac{1}{2^n - 1}$$

Principaux convertisseurs numeriques/analogiques

Parmi les CNA, il existe deux familles.

-CNA directs: passent sans intermediaire du mot binaire au signal analogique.

-CNA indirects: utilisent une variable intermediaire .

Les CNA directs se divisent en CNA paralleles et CNA serie.

Les CNA paralleles sont les plus rapides.

CNA parallèles

Le schéma de principe d'un CNA parallèles peut se deduire directement de la formule donnant sa fonction de transfert.

$$U = d_1 \frac{U_{ref}}{2} + d_2 \frac{U_{ref}}{2^2} + \dots + d_n \frac{U_{ref}}{2^n}$$

Il comprend les elements suivants:

- Une grandeur de reference
- Un système de pondération: multiplication par les coefficients de la suite: $\frac{1}{2}$; $\frac{1}{4}$; ; $\frac{1}{2^n}$
- Une commande numerique: multiplication par di qui vaut "0" ou "1".
- Une sommation de ces différents signaux (tensions ou courants).
- Une transformation eventuelle: courant en tension ou tension en courant

Habituellement, il est plus facile d'additionner des courants que les tensions

$$U = R \left(d_1 \frac{U_{ref}}{2R} + d_2 \frac{U_{ref}}{4R} + \dots + d_n \frac{U_{ref}}{2^n R} \right)$$

$$U = R(d_1 I_1 + d_2 I_2 + \dots + d_n I_n)$$

Après avoir effectué la sommation des courants ponderés, on peut operer la transformation courant-tension. On utilise , par exemple, le montage de la fig. 1
Le schéma d'un CNA parallèle est représenté fig .2. La pondération utilisée transforme la tension Uref en un courant

Les interrupteurs realisant la fonction commande sont eux-mêmes contrôlés par des bits du mot binaire en entrée. Suivant que le bit est à l'etat 0 ou 1 le courant traversant la resistance correspondante est envoyé à l'entrée de l'amplificateur pour être additionné aux autres courants ou bien au contraire dirigé vers la masse.

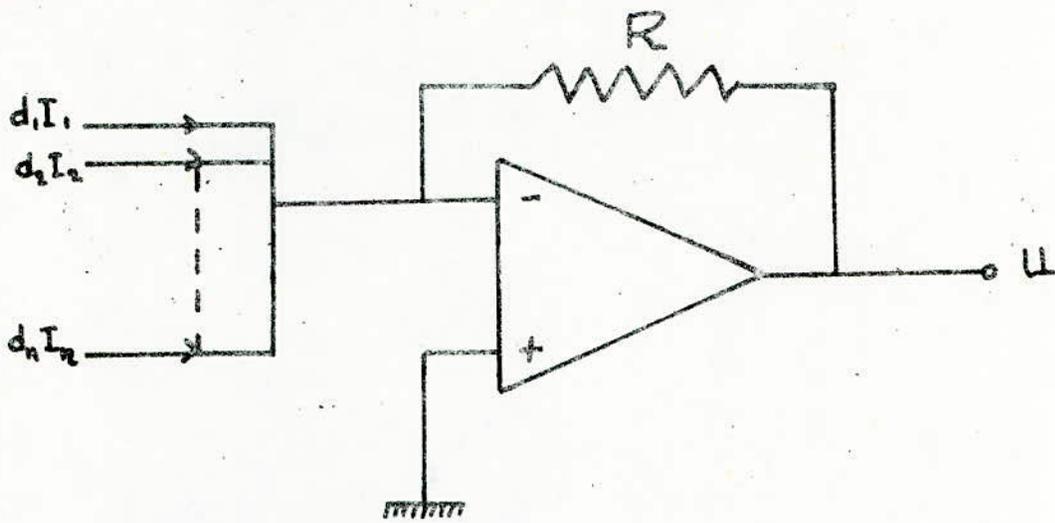


fig.1

$$I_i = \frac{U_{ref}}{2^2 R}$$

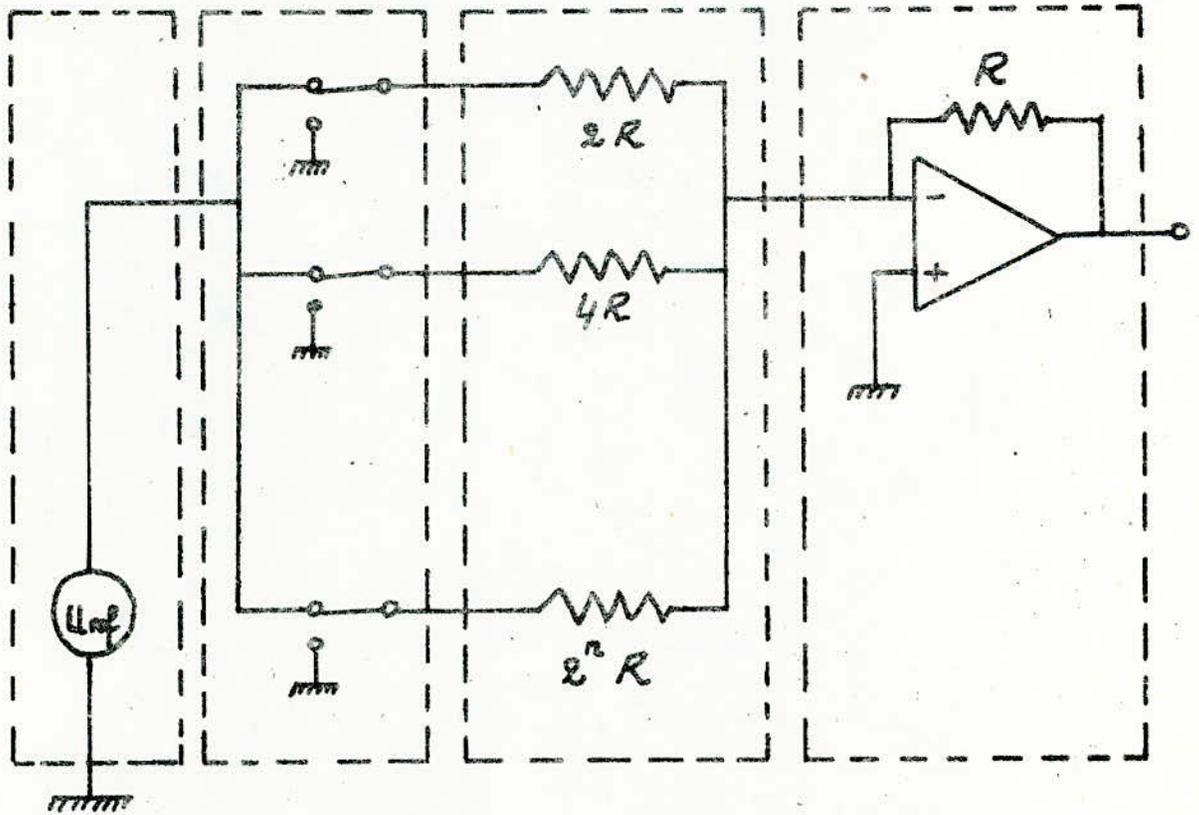


fig. 2.

Convertisseur utilisé: DAC 02

C'est un convertisseur parallèle, il est constitué d'un assemblage complet comprenant référence et amplificateurs opérationnels de sortie.

Il est du type R/2R et les courants résultants sont appliqués à 2 amplificateurs de sortie ($-V_s$ et $+V_s$ selon le bit de signe) par l'intermédiaire de 10 transistors montés en base commune.

La tension de référence est fournie par l'ensemble diode zener, résistance R_{ref} et amplificateur.

Le transistor T_1 sert à régler automatiquement la tension des transistors T_2 à T_{11} en cas de variation de la température.

Les transistors T_2 à T_{11} étant fabriqués de la même façon sur une même pastille, sont très appariés et leur tension V_{be} variera de la même manière que celle de T_1 ; ainsi leurs émetteurs resteront toujours au même potentiel et la précision sera maintenue.

Caractéristiques techniques:

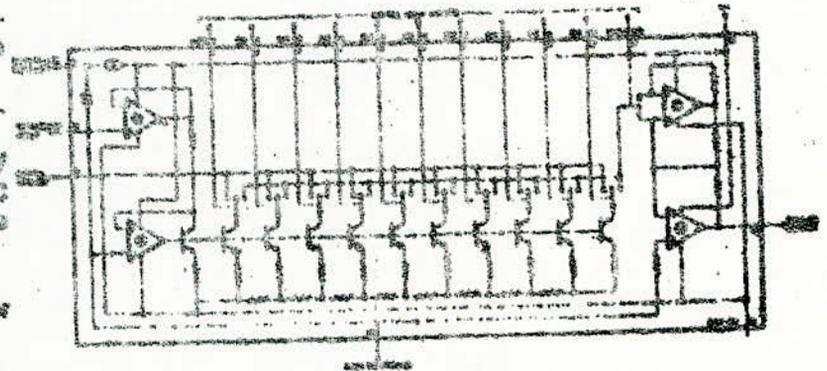
- Les entrées sont compatibles avec la TTL;DTL; et C.MOS.
- Tensions d'alimentations de : (+12v à +18v) et de (-12v à -18v).
- Le temps de conversion: 1,5 μ s.
- Résolution : 10 bits+bit de signe.
- Monotonie garantie : 0 à 70°C.
- Stabilité : dans toute la gamme de température 60ppm % c max.
- Sortie bipolaire : ± 10 v.

La constitution interne et l'ajustage de la tension de sortie sont représentées à la fig. B.

Ajustement de la tension de sortie: la tension maximale de sortie etant légèrement inférieure à celle de reference, nous l'avons ajusté à 10v à l'aide du potentiomètre (10K) en mettant toutes les entrées à 1.

DAC-02 - CONVERTISSEUR D/A MONOLITHIQUE, 10 BITS + SIGNE

- ⊗ assemblage complet comprenant références et amplificateurs opérationnels de sortie
- ⊗ rapides 1,8 µs de temps d'établissement
- ⊗ sortie bipolaire $\pm 10V, \pm 5V$
- ⊗ monotonie garantie 0° à 70°C
- ⊗ stabilité dans toute la gamme de température 50 ppm/°C max.
- ⊗ alimentations standard $\pm 12V$ à $\pm 15V$
- ⊗ compact ... un simple chip dans un boîtier DIP 18 sorties



Type	Tension de sortie OpAmp Buffer		Résolution (bits)	Monotonie (0° à 70°C) (bits)	Linéarité (0° à 70°C)	Coef. de Temp. en /°C max.
	+10V	±5V				
DAC-02AC	X1	X2	10 + Signe	10	±0.1%	50
DAC-02BC	X1	X2	10 + Signe	9	±0.1%	50
DAC-02CC	X1	X2	10 + Signe	8	±0.2%	50
DAC-02DD	X1	X2	10 + Signe	7	±0.4%	100

AJUSTEMENT DE LA TENSION DE SORTIE

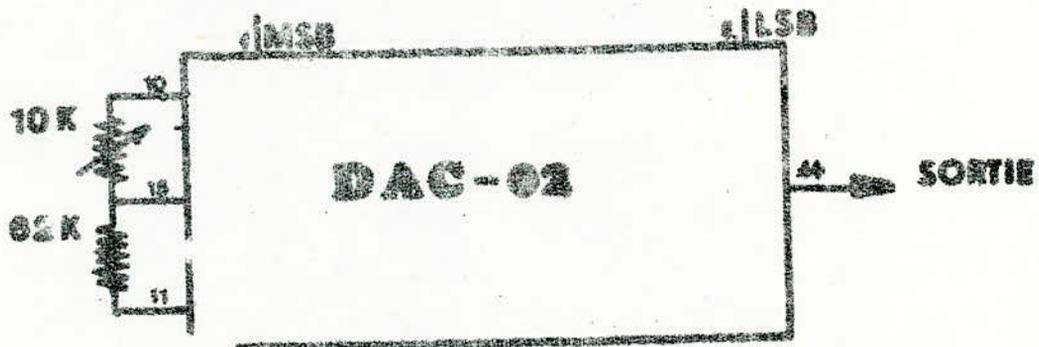


fig-3

III - 3 --3 FONCTIONNEMENT

Les 4 compteurs SN 74 193 sont montés en cascade et forment un compteur de 16 bits.

Les deux premiers donnent le balayage horizontal, et les deux derniers le balayage vertical.

Le premier compteur est piloté par une horloge de 1 MHz (le choix de cette fréquence sera justifié dans ce qui suivra) à la fin de chaque cycle de comptage il délivre une impulsion négative "Carry" qui sert d'horloge pour le compteur suivant. Chaque impulsion "Carry" du 2ème compteur indique la fin de ligne et celle du 4ème compteur la fin d'image. Les sorties de ces compteurs sont reliées d'une part aux lignes adresses de la carte image à travers des buffers 3 états commandés par la ligne PBo.

-PBo = 0 buffers en haute impédance.

-PBo = 1 buffers passants, l'adressage peut avoir lieu.

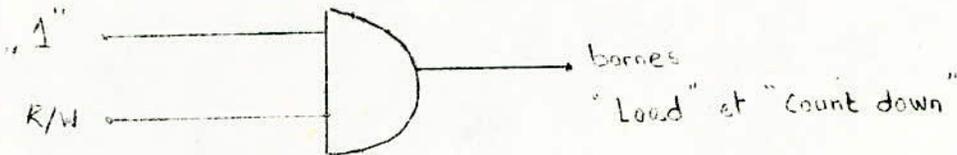
D'autre part à 2 convertisseurs N/A permettant l'attaque, après avoir au préalable effectué une atténuation, des entrées X et Y du moniteur.

SYNCHRONISATION

Le comptage ne peut avoir lieu que lorsque la borne "Load" et l'entrée "Count down" de chaque compteur sont à "1".

Dans notre système, nous voulons à ce que le début du comptage (l'adressage de la carte) s'effectue lorsque $R/W = 1$ c'est-à-dire juste au moment où l'on passe au mode lecture de la carte. Pour cela nous avons pensé à relier ces deux lignes ("Load" et "Count down") à la sortie d'une porte "ET" dont ses entrées sont composées par un "1" logique et le signal R/W.

Donc la sortie de la porte "ET" ne sera à "1" que lorsque R/W est à "1".



La remise à zero des compteurs est assurée par la borne "Carry" (fin d'image) issue du quatrième compteur.

De même que la liaison précédente, celle-ci fonctionne en mode voulu par programmation .

Programme

D'une manière générale ce programme est identique à celui de la liaison précédente avec seulement une différence au niveau de l'initialisation des PIA et plus particulièrement au niveau des bits de commande

- R/W est mis à 1, ce qui permet la lecture de la carte mémoire.
- PBo (CA) = 1 les buffers reliés aux compteurs sont passants, l'adressage de la carte est possible.
- PB1 (CD) = 1 le buffer SN 74 125 est ouvert seulement pour la donnée vers l'entrée Z du moniteur.

III- 3 - 4 CALCUL DE LA FREQUENCE D'HORLOGE.

Une image stable sur l'ecran du moniteur dure $\frac{1}{20}$ de seconde.

Elle est constituée de 256 lignes sur 256 colonnes, de ce fait elle sera formée de 256×256 points.

Donc un point sur l'ecran apparait chaque $\frac{1}{20 \times 256 \times 256}$ secondes.

d'où $f = \frac{1}{T} = 20 \times 256 \times 256 = 1,31 \text{ MHz.}$

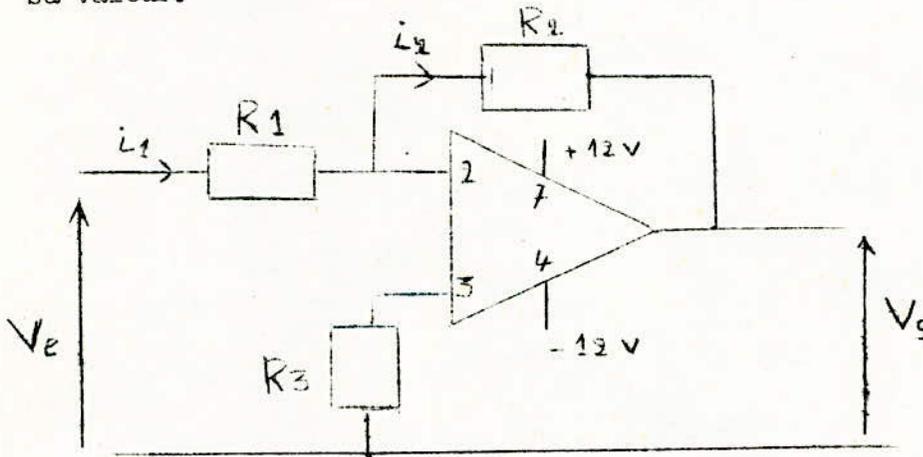
III- 4 - Attenuation

Le composant utilisé est un circuit intégré UA 741 nécessitant 2 alimentations distinctes. L'une d'elles est positive par rapport à la référence commune des entrées tandis que l'autre est négative.

Ce circuit peut avoir plusieurs fonctions suivant les applications.

- Amplification
- Attenuation
- Adaptation

Le montage suivant nous permet d'atténuer la sortie du DAC à $\frac{1}{10}$ de sa valeur.



La borne 2 présente une très grande impédance d'entrée, ainsi le courant qui traverse R1 traverse aussi R2, ce qui permet d'écrire

$$i_1 = i_2$$

$$V_s = -R_2 i_2$$

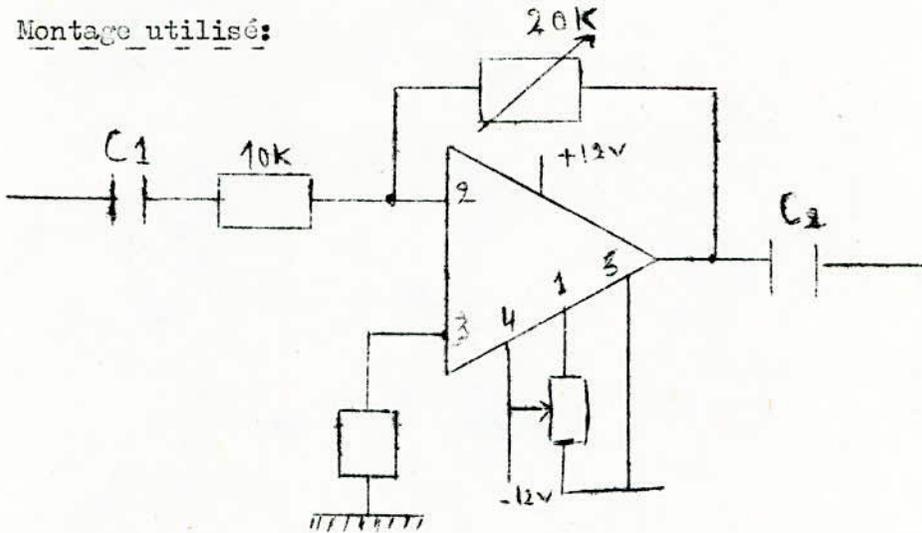
$$V_e = R_1 i_1$$

$$A = \frac{V_s}{V_e} = \frac{-R_2 i_2}{R_1 i_1} = -\frac{R_2}{R_1}$$

Pour avoir $A = \frac{1}{10}$

on prendra $R1 = 10 R2$

Montage utilisé:



Le potentiomètre de 10 K sert à annuler la tension d'offset.

Les condensateurs C1 et C2 sont des capacités de liaison de 1 μ F.

V - APERCU SUR L'OSCILLOSCOPE "TEKTRONIX".

La partie visualisation de notre système est constituée d'un oscilloscope "TEKTRONIX" du type "606" A".

Sa technologie de pointe permet d'aboutir à des résultats de hautes performances dans les différents cas d'utilisation.

Grâce à son excellente résolution ainsi qu'à son spot extrêmement réduit, il permet, outre une présentation très fidèle des informations

Alpha numérique ou graphiques, des applications dans les domaines.

- de l'exploration thermique et de radiations.
- des phénomènes vibratoires.
- des systèmes de microscopes électroniques.
- des systèmes biophysiques et médicaux etc....

Caractéristiques techniques :

1 - Dimension de l'écran : 8x10cm.

2 - Dimension du spot : 0,079cm.

3 - Ampli X et Y.

3-1 : Temps de montée : 116ns.

3-2 : Bande passante : 3 MHz.

3-3 : Facteur de reflexion.

vertical et horizontal 1v nominal pour la pleine échelle,
il existe un ajustage interne qui permet de passer de 0,5v
à 2,5v pour la pleine échelle.

3-4 : Impédance d'entrée : 1M Ω // 47pF

3-5 : Tension d'entrée max : \pm 100v.

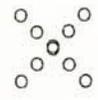
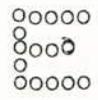
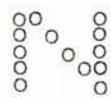
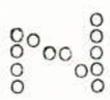
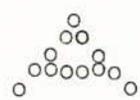
4-Ampli Z : permet la modulation d'intensité.

4-1 : Bande passante : 10 MHz.

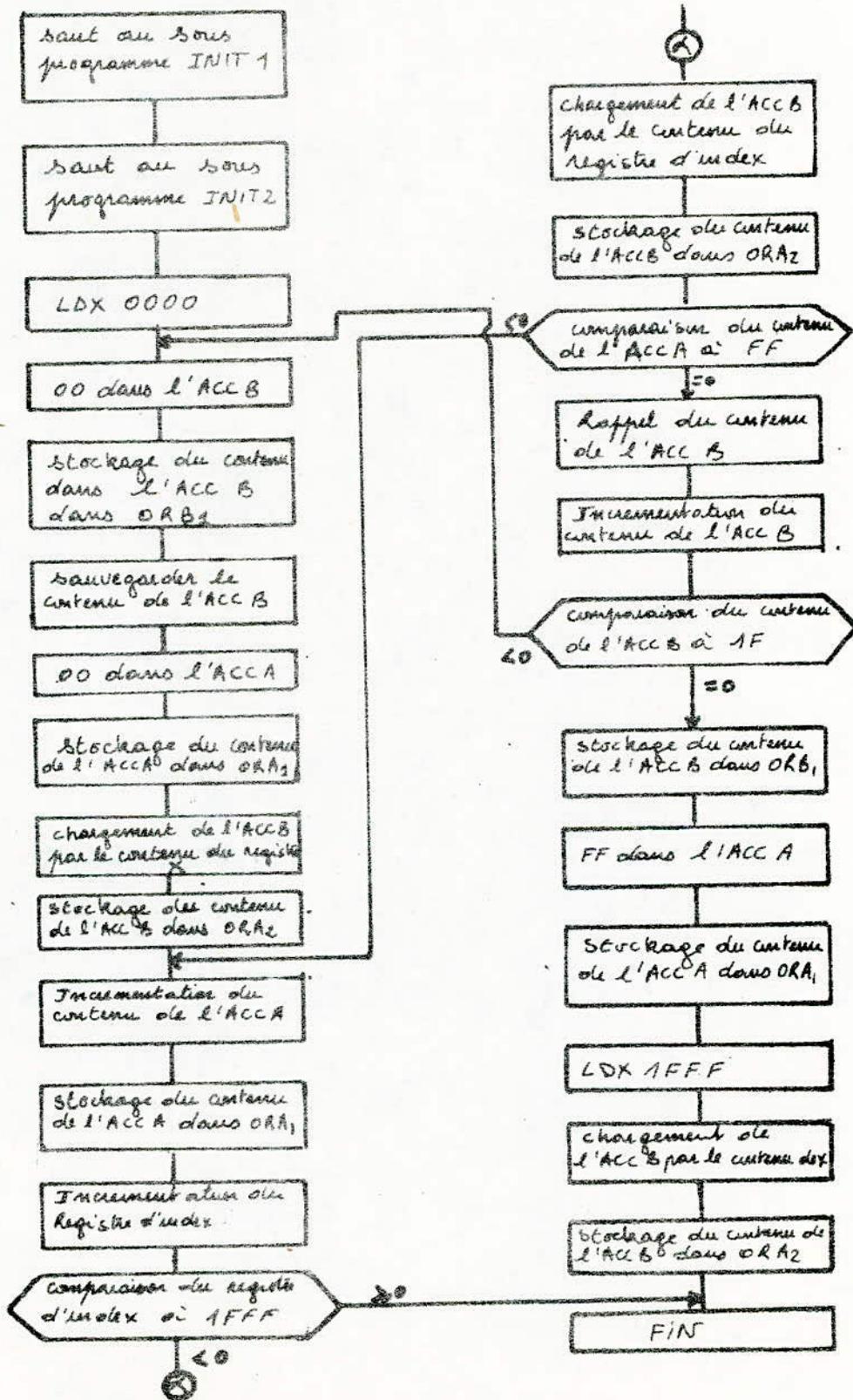
4-2 : Temps de montée : 35 nS.

4-3 : Tension d'entrée max : 100v.

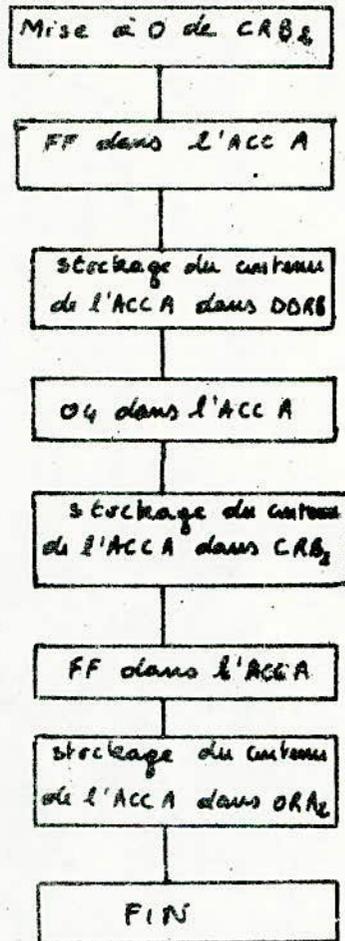
4-4 : Gamme de sensibilité : ajustable entre 0-1v et 0-5v.



Organigramme de TRANSFERT



Organigramme de Visualisation



PROGRAMME D' INITIALISATION DU PIA I : INIT (I)

CLR	§ 8001	7F / 8001
CLR	§ 8003	7F / 8003
LDA A	≠ § FF	86 / FF
STA A	§ 8000	B7 / 8000
LDA A	≠ § FF	86 / FF
STA A	§ 8002	B7 / 8002
LDA A	≠ § 04	86 / 04
STA A	§ 8001	B7 / 8001
LDA A	≠ § 04	86 / 04
STA A	§ 8003	B7 / 8003
RTS		39

PROGRAMME D' INITIALISATION DU PIA II : INIT (II)

CLR	§ 8005	7F / 8005
CLR	§ 8007	7F / 8007
LDA A	§ FF	86 / FF
STA A	§ 8004	B7 / 8004
LDA A	§ FF	86 / FF
STA A	§ 8006	B7 / 8006
LDA A	§ 04	86 / 04
STA A	§ 8005	B7 / 8005
LDA A	§ 04	86 / 04
STA A	§ 8007	B7 / 8007
LDA A	§ 00	86 / 00
STA A	§ 8006	B7 / 8006
RTS		39

PROGRAMME DE TRANSFERT : MPU vert MEMOIRE IMAGE (PROG I)

	Saut à INIT I	
	Saut à INIT II	
	LDX \$ 0000	CE / 0000
	LDA B 2 \$ 00	C6 / 00
BA	STA B \$ 8002	F7 / 8002
	PUSH B	37
	LDA A 2 \$ 00	06 / 00
	STA A \$ 8000	B7 / 8000
	LDA B \$(0,X)	E6 / 00
	STA B \$ 8004	F7 / 8004
AB	INC A	4C
	STA A \$ 8000	B7 / 8000
	INX	08
	CPX \$ IFFF	8C / IFFF
	BEQ END	27
	LDA B \$(0,X)	E6 / 00
	STA B \$ 8004	F7 / 8004
	CMPA 2 \$ FF	8I / FF
	BNE AB	26
	PULB	33
	INC B	5C
	CMPB 2 \$ IF	CI / IF
	BNE BA	26
	STA B \$ 8002	F7 / 8002

LDA A	≠ \$ FF	B6 / FF
STA A	\$ 8000	B7 / 8000
LDX	\$ IFFF	CE / IFFF
LDA B	(0,X)	E6 / 00
STA B	\$ 8004	F7 / 8004
END		3F

PROGRAMME D'INITIALISATION DU PIA (II) POUR PERMETTRE
LA VERIFICATION DU CONTENU DE LA MEMOIRE : INIT' (II)

CLR	§ 8005	7F / 8005
CLR	§ 8007	7F / 8007
LDA A	≠ § 01	86 / 01
STA A	§ 8004	B7 / 8004
LDA A	≠ § FF	86 / FF
STA A	§ 8006	B7 / 8006
LDA A	≠ § 04	86 / 04
STA A	§ 8005	B7 / 8005
LDA A	≠ § 04	86 / 04
STA A	§ 8007	B7 / 8007
LDA A	≠ § 04	86 / 04
STA A	§ 8006	B7 / 8006
RTS		39

PROGRAMME DE VERIFICATION DU CONTENU DE LA MEMOIRE (PROG II)

	Saut à INIT (I)	
	Saut à INIT' (II)	
	LDX \$ 0400	CE / 0400
	LDA B Z \$ 00	86 / 00
SAD	STA B \$ 8002	B7 / 8002
	PUSH B	37
	LDA A Z \$ 00	86 / 00
	STA A \$ 8000	B7 / 8000
	LDA B \$ 8004	86 / 8004
	STA B \$ (0, X)	E7 / 00
BYE	INC A	4C
	STA A \$ 8000	B7 / 8000
	INC X	08
	CPX \$ 28FF	8C / 07FF
	BEQ END	27
	LDA B \$ 8804	F6 / 8004
	STA B \$ (0', X)	E7 / 00
	CMP A \$ FF	81 / FF
	BNE BYE	26
	PUL BB	33
	INC B	5C
	CMP B \$ 23	CI / IF
	DNE SAD	26
	STA B \$ 8002	F7 / 8002

PROGRAMME DE VERIFICATION DU CONTENU DE LA MEMOIRE (PROG II)

	Saut à INIT (I)	
	Saut à INIT' (II)	
	LDX \$ 0400	CE / 0400
	LDA B Z \$ 00	06 / 00
SAD	STA B \$ 8002	B7 / 8002
	PUSH B	37
	LDA A Z \$ 00	06 / 00
	STA A \$ 8000	B7 / 8000
	LDA B \$ 8004	06 / 8004
	STA B \$ (0, X)	E7 / 00
BYE	INC A	4C
	STA A \$ 8000	B7 / 8000
	INC X	08
	CPX \$ 20FF	8C / 07FF
	BEQ END	27
	LDA B \$ 8004	F6 / 8004
	STA B \$ (0', X)	E7 / 00
	CMP A \$ FF	81 / FF
	BNE BYE	26
	PUL BB	33
	INC B	5C
	CMP B \$ 23	C1 / 1F
	DNE SAD	26
	STA B \$ 8002	F7 / 8002

LDA A	7 \$ FF	86 / FF
STA A	\$ 8000	86 / 8000
LDX	\$ 7FFF	CCE / 7FFF
LDA B	\$ 8004	C6 / 8000
STA B	\$ (0, X)	E7 / 00
END		3F

PROGRAMME DE VISUALISATION

Pour permettre la visualisation, il suffit seulement de changer le programme d'initialisation du PIA (II) ce qui revient à mettre :

- à la ligne R/W à 1
- La ligne commande adresse (CA) à 1
- La ligne commande donnée (CD) à 1

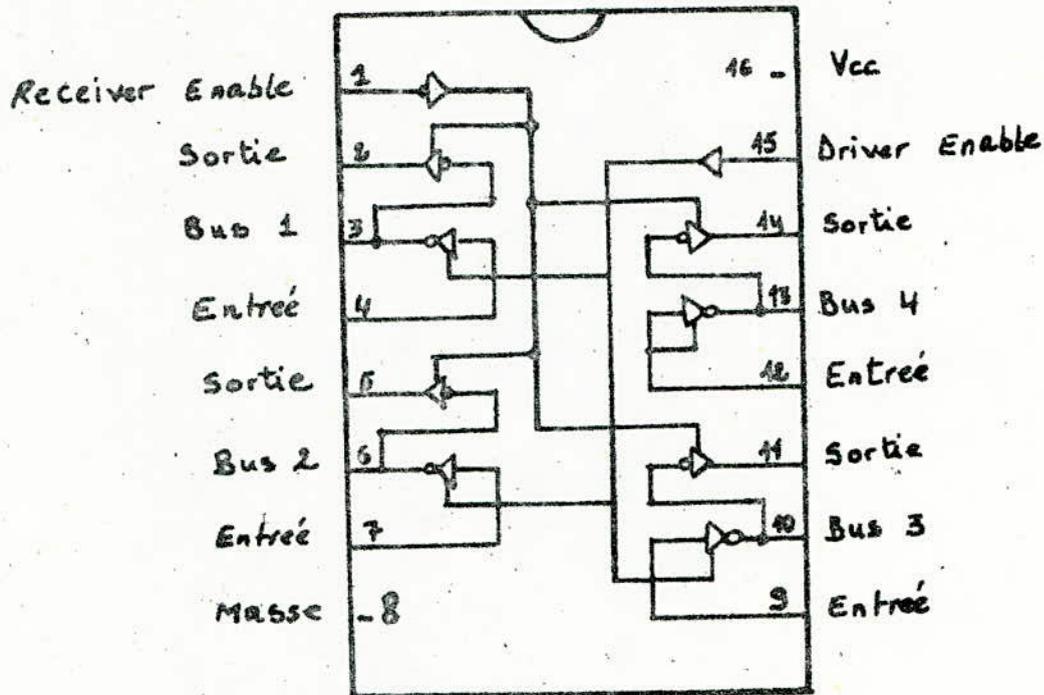
CLR	§ 8007	7F / 8007
LDA A	z § FF	86 / FF
STA A	§ 8006	B7 / 8006
LDA A	z § 04	86 / 04
STA A	§ 8007	B7 / 8007
LDA A	z § FF	86 / FF
STA A	§ 8006	B7 / 8006
END		3F

INDEX REGISTER AND STACK		IMMED	DIRECT	INDEX	EXTND	INNER	BOOLEAN/ARITHMETIC OPERATION	5	4	3	2	1	0
POINTER OPERATIONS	MNEMONIC	OP	OP	OP	OP	OP		H	I	N	Z	V	C
Compare Index Reg	CPX	0C 3 3	0C 4 2	AC 8 2	0C 5 3		$(X_H/X_L) - (M/M + 1)$	*	⊕	*	*	*	*
Decrement Index Reg	DEX					09 4 1	$X - 1 \rightarrow X$	*	*	*	*	*	*
Decrement Stack Ptr	DEB					34 4 1	$SP - 1 \rightarrow SP$	*	*	*	*	*	*
Increment Index Reg	INX					08 4 1	$X + 1 \rightarrow X$	*	*	*	*	*	*
Increment Stack Ptr	INS					31 4 1	$SP + 1 \rightarrow SP$	*	*	*	*	*	*
Load Index Reg	LDX	0E 3 3	DE 4 2	EE 8 2	FE 5 3		$M \rightarrow X_H, (M + 1) \rightarrow X_L$	*	⊕	*	*	*	*
Load Stack Ptr	LDS	0E 3 3	DE 4 2	EE 8 2	FE 5 3		$M \rightarrow SP_H, (M + 1) \rightarrow SP_L$	*	⊕	*	*	*	*
Store Index Reg	STX		DF 5 2	EF 7 2	FF 8 3		$X_H \rightarrow M, X_L \rightarrow (M + 1)$	*	⊕	*	*	*	*
Store Stack Ptr	STS		DF 5 2	EF 7 2	FF 8 3		$SP_H \rightarrow M, SP_L \rightarrow (M + 1)$	*	⊕	*	*	*	*
Index Reg → Stack Ptr	TXS					35 4 1	$X - 1 \rightarrow SP$	*	*	*	*	*	*
Stack Ptr → Index Reg	TSX					30 4 1	$SP + 1 \rightarrow X$	*	*	*	*	*	*

JUMP AND BRANCH OPERATIONS		RELATIVE	INDEX	EXTND	INNER	BRANCH TEST	5	4	3	2	1	0
MNEMONIC	OP	OP	OP	OP	OP		H	I	N	Z	V	C
Branch Always	BRA	20 4 2				None	*	*	*	*	*	*
Branch If Carry Clear	BCC	24 4 2				$C = 0$	*	*	*	*	*	*
Branch If Carry Set	BCS	25 4 2				$C = 1$	*	*	*	*	*	*
Branch If = Zero	BEQ	27 4 2				$Z = 1$	*	*	*	*	*	*
Branch If > Zero	BGT	26 4 2				$N \oplus V = 0$	*	*	*	*	*	*
Branch If >= Zero	BGE	26 4 2				$Z + (N \oplus V) = 0$	*	*	*	*	*	*
Branch If Higher	BHI	22 4 2				$C + Z = 0$	*	*	*	*	*	*
Branch If <= Zero	SLE	2F 4 2				$Z + (N \oplus V) = 1$	*	*	*	*	*	*
Branch If Lower Or Same	PLS	23 4 2				$C + Z = 1$	*	*	*	*	*	*
Branch If < Zero	SLT	2D 4 2				$N \oplus V = 1$	*	*	*	*	*	*
Branch If Minus	BMI	28 4 2				$N = 1$	*	*	*	*	*	*
Branch If Not Equal Zero	BNE	25 4 2				$Z = 0$	*	*	*	*	*	*
Branch If Overflow Clear	BVC	28 4 2				$V = 0$	*	*	*	*	*	*
Branch If Overflow Set	BVS	29 4 2				$V = 1$	*	*	*	*	*	*
Branch If Plus	BPL	2A 4 2				$N = 0$	*	*	*	*	*	*
Branch To Subroutine	BSR	0D 5 2					*	*	*	*	*	*
Jump	JMP		0E 4 2	7E 3 3		See Special Operations	*	*	*	*	*	*
Jump To Subroutine	JSR		AD 5 2	BD 0 3			*	*	*	*	*	*
No Operation	NOP					BT 2 1						
Return From Interrupt	RTI					3B 10 1						
Return From Subroutine	RTS					3S 5 1						
Software Interrupt	SWI					3F 12 1						
Wait for Interrupt	WAI					3E 9 1						

CONDITION CODE REGISTER		INNER	BOOLEAN	5	4	3	2	1	0	CONDITION CODE REGISTER NOTES
OPERATIONS	MNEMONIC	OP	OPERATION	H	I	N	Z	V	C	
Clear Carry	CLC	0C 2 1	0 → C	*	*	*	*	*	*	① (Bit V) Test Result = 10000000 ?
Clear Interrupt Mask	CLI	0E 2 1	0 → I	*	*	*	*	*	*	② (Bit C) Test Result = 00000000 ?
Clear Overflow	CLV	0A 2 1	0 → V	*	*	*	*	*	*	③ (Bit D) Test Decimal value of most significant BCD Character greater than nine ? (Not cleared if previously set)
Set Carry	SEC	0B 2 1	1 → C	*	*	*	*	*	*	④ (Bit V) Test Operand = 10000000 prior to execution ?
Set Interrupt Mask	SEI	0F 2 1	1 → I	*	*	*	*	*	*	⑤ (Bit V) Test Operand = 01111111 prior to execution ?
Set Overflow	SEV	08 2 1	1 → V	*	*	*	*	*	*	⑥ (Bit V) Test Set equal to result of M ⊕ C after shift has occurred
Accum A → CCR	TAP	0B 2 1	A → CCR	*	*	*	*	*	*	⑦ (Bit N) Test Sign bit of most significant (MS) byte of result = 1 ?
CCR → Accum A	TPA	07 2 1	CCR → A	*	*	*	*	*	*	⑧ (Bit V) Test Z's complement overflow from subtractive LS bytes ?

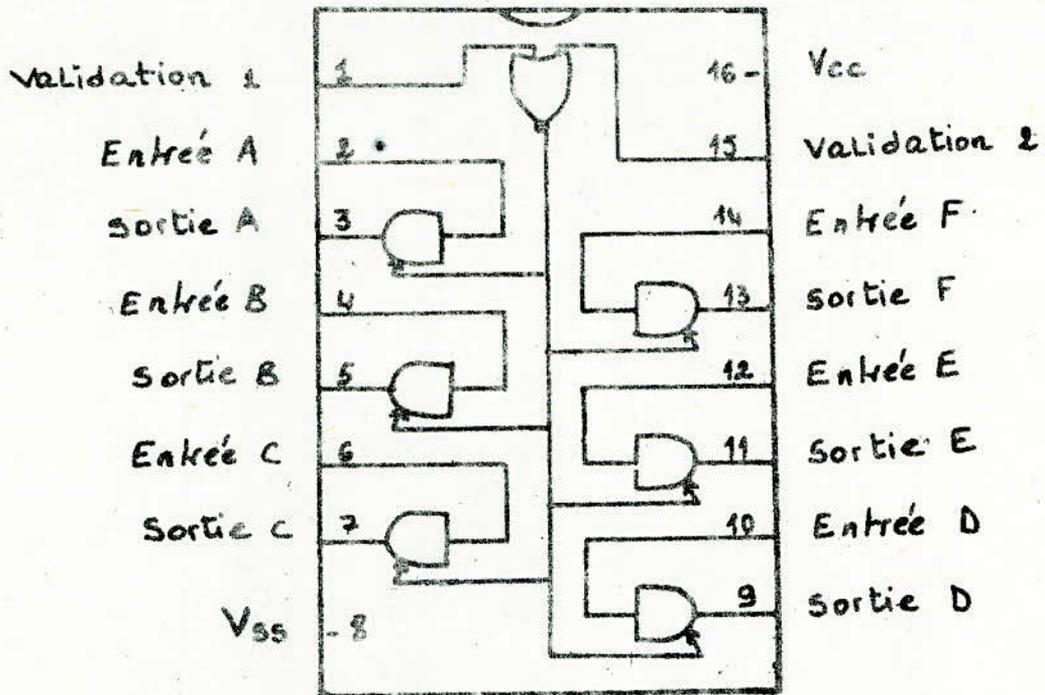
- LEGEND**
- 00 Sign = Zero
 - OP Operation Code (Hexadecimal)
 - ~ Number of MPU Cycles
 - Number of Program Bytes
 - + Arithmetic Plus
 - Arithmetic Minus
 - Boolean AND
 - M₂₀ Contents of memory location pointed to by Stack Pointer
 - + Boolean Inclusive OR
 - ⊕ Boolean Exclusive OR
 - M Complement of M
 - Transfer Into
 - 0 Bit = Zero
 - H Half carry from bit 3
 - I Interrupt mask
 - N Negative (sign bit)
 - Z Zero (byte)
 - V Overflow, 2's complement
 - C Carry from bit 7
 - R Reset Always
 - S Set Always
 - † Test and set if true/clear otherwise
 - Not Affected
 - CCR Condition Code Register
 - LS Least Significant
 - MS Most Significant
- ① (Bit V) Test Result = 10000000 ?
 ② (Bit C) Test Result = 00000000 ?
 ③ (Bit D) Test Decimal value of most significant BCD Character greater than nine ? (Not cleared if previously set)
 ④ (Bit V) Test Operand = 10000000 prior to execution ?
 ⑤ (Bit V) Test Operand = 01111111 prior to execution ?
 ⑥ (Bit V) Test Set equal to result of M ⊕ C after shift has occurred
 ⑦ (Bit N) Test Sign bit of most significant (MS) byte of result = 1 ?
 ⑧ (Bit V) Test Z's complement overflow from subtractive LS bytes ?
 ⑨ (Bit N) Test Result less than zero ? (Bit 15 = 1)
 ⑩ (ALL) Load Condition Code Register from Stack (See Special Operations)
 ⑪ (Bit I) Set when interrupt occurs. If previously set, a Non Maskable interrupt is required to exit the wait state.
 ⑫ (ALL) Set according to the contents of Accumulator A



8T26

Driver Enable	Receiver Enable	Sortie	Entrée	Bus
L	L	L	X	L
L	L	H	X	H
L	H	X	X	isolé
H	H	X	L	L
H	H	X	H	H

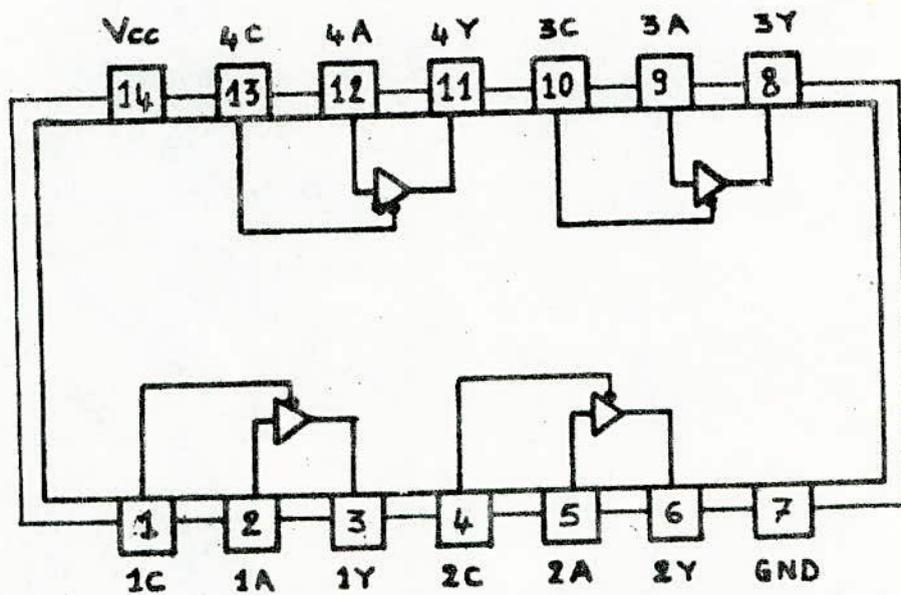
Table de verité



8T95

Valid 2	Valid 1	Entrée	Sortie
L	L	L	L
L	L	H	H
L	H	X	Z
H	L	X	Z
H	H	X	Z

Table de verité



SN 74 125

- Positive Logic

$$Y = A$$

- Output is off (disabled) when C is high.

C O N C L U S I O N

La présente étude a été très bénéfique pour nous; en effet, elle nous a permis de nous familiariser avec la micro-informatique, domaine ignoré par nous jusque là.

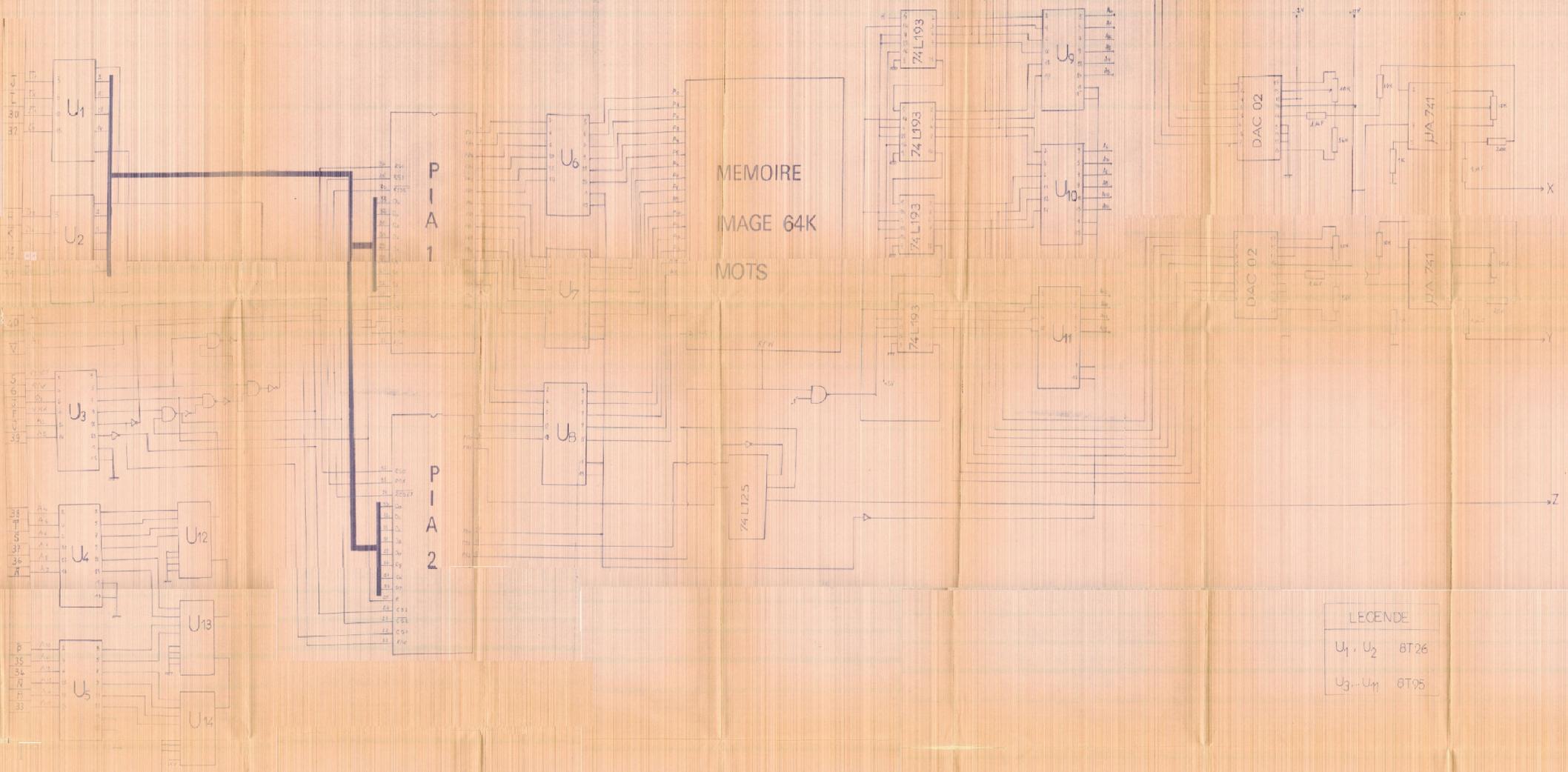
La partie hardware a été menée avec beaucoup d'attention, malheureusement l'indisponibilité d'une horloge de 1,3 MHz et des convertisseurs rapides nous a posé un problème car comme vous auriez pu le remarquer le temps de conversion des DAC est de 1,5 us alors que chaque 1 us une information à convertir se présente. Aussi l'image obtenue sur l'écran est instable chose due à une fréquence faible (1MHz).

Pour la partie software, nous nous sommes limités juste aux programmes de transfert et de visualisation. Il est bien évident que beaucoup d'applications peuvent être faites suivant le programme de chargement.

Enfin, nous espérons que d'autres groupes amélioreront notre travail dans les semestres à venir.

--oOo-- B I B L I O G R A P H I E --oOo--

- MICROPROCESSEURS ET MEMOIRES : THOMSON EFCIS
- DIALOGUE MICRO-MICRO : CATALOGUE 1980
- THESE D'INGENIORAT : RESTITUTION D'IMAGE SUR OSCILLOSCOPE
JUN 80
- THESE D'INGENIORAT : ETUDE DES ENTREES ET SORTIES D'UN MICRO-
ORDINATEUR
- REVUE MICRO SYSTEME N° 5
- TEKTRONIX - INC INSTRUCTION MANUEL
- LA CONVERSION N/A : BERNARD LORIFERNE



LEGENDE	
U ₁ , U ₂	8T26
U ₃ ...U ₁₁	8T95

Ecole Nationale Polytechnique
 Projet Réalisé Par
 M. Debyeche
 S. Gaoua

**SCHEMA DE CABLAGE
 GENERAL DE L'INTERFACE**