الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم و البحث العلمي
MINISTERE DE L'ENSEIGNEMENT ET DE LA RECHERCHE SCIENTIFIQUE

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT : G. MECANIQUE

PROJET DE FIN D'ETUDES

SUJET

ETUDE du micro-ordinateur MPZ-80/EV et de son application a la mise en œuvre du mini-robot RB-4/EV

Proposé Par : A. ZERGUERRAS

Etudié par : A. MEDJEDOUB

Dirigé par : A. ZERGUERRAS

PROMOTION : JUIN 88

E.N.P. 10, Avenue Hacen Badi – EL-HARRACH – ALGER

الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم و البحث العلمي
MINISTERE DE L'ENSEIGNEMENT ET DE LA RECHERCHE SCIENTIFIQUE

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT :

PROJET DE FIN D'ETUDES

SUJET

ETUDE du micro-ordinateur MPZ-80/EV et de son application a la mise en œuvre du mini-robot RB-4/EV

Proposé Par :

Etudié par :

Dirigé par :

PROMOTION :

E.N.P. 10, Avenue Hacen Badi – EL-HARRACH – ALGER

MINISTERE DE L'ENSEIGNEMENT
SUPERIEUR
ECOLE NATIONALE POLYTECHNIQUE

Departement: GENIE MECANIQUE
Promoteur : Mr A. ZERGUERRAS
Eleve ingenieur: A.MEDJEDOUB

الموضوع: دراسة قيادة إنسان آلي مصغر مبرمج بواسطة ميكرو كمبيوتر.

الملخص: هذه الدراسة تتناول تركيب وبرمجة الإنسان الآلي المصغر RB4/EV بواسطة ميكرو كمبيوتر MPZ-80/EV ، وإنجاز تعاقبات الحركة ، ودراسة الملحقات المختلفة.

Sujet:Etude d'un micro-ordinateur et de son application a la mise en oeuvre d'un mini-robot.

Resume:Ctte etude porte sur la commande du mini-robot RB-4/EV a l'aide du micro-ordinateur MPZ-80/EV.Et etude des differents peripheriques de ce dernier.

Subject:Study of a computer and its application to move the mini-robot.

Summary:This study revolve around the mini-robot RB-4/EV command using the MPZ-80/EV computer.And study the differents peripheriques.

Dedicaces

je dedie
ce modeste travail a

mes chers parents
mes freres et soeurs
mes amis

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

Je tient a exprimer ma forte reconnaissance et mes vifs remerciments a monsieur A. ZERGUERRAS qui a eu l'aimable gentiellesse de me fornir son aide a realiser ce travail,ainsi qu'a tous les enseignants qui ont contribue a ma formation.

Que tous ceux qui ont participe de loin ou de pres a la realisation de ce projet trouveront ici ma sincere gratitude.

********** T A B L E D E S M A T I E R E S **********

PAGE

INTRDUCTION.....1

1 DESCRIPTION DU MICRO-ORDINATEUR MPZ-80/EV.....3
1.1 Le systeme a microprocesseur.....3
1.2 Structure du systeme MPZ-80/EV.....10
1.3 Exploitation avec le moniteur MON-80.....21
1.4 Exploitation avec le systeme operationnel CP/M.....23
1.5 Introduction sur le programme d'edit,d'assembleur et de debug.....24

2 DESCRPTIONDESPERPHERIQUES.....27
2.1 Liaison du terminal video et de l'imrimante.....27
2.2 Fonctionnement des cartes CAR A/D , D/A.....32
2.3 L'alimentation stabilisee.....44
2.4 L'unite d'extension industrielle VEZ-80/EV.....49

3 DESCRIPTION DU MINI-ROBOT RB-4/EV.....53
3.1 Structure mecanique.....53
3.2 Relation entre le pas du moteur et l'accroissement angulaire.....55
3.3 Commande electronique.....56

4 COMMANDE PROGRAMME DU MINI-ROBOT RB-4/EV.....63
4.1 Le programme de gestion.....63
4.1.1 Description du programme AROBOT.....63
4.1.2 Description du programme ROBCOM.....64
4.1.3 Description du programme MROBOT.....65
4.2 Link des paquets du programme.....65
4.2.1 Interprete basic.....66
4.2.2 Compilateur basic.....66
4.3 Description et utilisation du programme.....67

5 APPLICATION A LA MISE EN OEUVRE DU MINI-ROBOT.....71

CONCLUSION.....74

BIBLIOGRAPHIE.....75

ANNEXES

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

La naissance des microprocesseurs vers les annees 70 a marquee une nouvelle etape du processus de robotisation. Ainsi en 1974 apparurent les premiers robots universels et facilement programmables pour la construcion automobile.Ils sont un bon exemple des robots industriels perfectionnes: particulierement aptes a la reconnaissance des formes , ils effectuent notament les travaux de soudure,de peinture et de manutention.

Ajourd'hui , la robotique represente un champ d'application important dans les domaines industriels pour l'automatisation des chaines de montage , l'assistance et l'intervention en milieux et environnements nocifs ou hostiles (domaine spacial, sous-marin,nucleaire,etc...).

Ceci a conduit les chercheurs de divers domaines tels que la mecanique , l'electronique , l'automatique ,l'informatique et l'intelligence artificielle,a orienter leurs recherches entre autre :

Vers le developpement de structures materielles (aussi bien sur le plan des manipulateurs eux-memes que sur celui des capteurs).

vers la resolution des problemes lies a la commande.

L'objet de notre projet est l'etude du micro-ordinateur MPZ-80/EV et de son application a la commande du mini-robot RB-4/EV.Il constitue une initiation didactique a la robotique

qui bien comprise pourrait conduire a des travaux de recherche aux perspectives non negligeable avec le meme equipement.

1 DESCRIPTION DU MICRO-ORDINATEUR MPZ-80/EV

1.1 LE MATERIEL(HARDWARE)D'UN SYSTEME A MICROPROCESSEUR

1.1.1 INTRODUCTION SUR LES MICROPROCESSEURS:

Le microprocesseur est une consequence de la miniaturisation des circuits integres:C'est un circuit LSI(Large Scale Integration),c'est a dire resultant d'une integration a tres grande echelle de plusieurs miliers de transistors sur un carre de quelques milimetres de cote et appelle PUCE .La technologie MOS qui se prete a une forte densite d'integration,a permis d'integrer sur une puce,l'unite centrale d'un ordinateur, appelee aussi Processeur,d'ou le nom de Microprocesseur. Le premier microprocesseur introduit sur le marche a ete le 4004 qui est un microprocesseur 4 bits.En 1972,la meme societe,Intel,proposa le premier microprocesseur,8 bits, le 8008.Devant son succes,de nombreuses societes etudierent un nouveau microprocesseur en se basant sur les avantages et inconvenients du 8008.C'est ainsi qu'en 1974 furent proposes le 8080 par Intel,le 6800 par motorala,le 6250 par Signetics,le PPS8 par rockwell et d'autres encores. Chaque constructeur a ameliore son microprocesseur (ce qui a donne le 8085 A d'Intel,le 6802 de Motorala) et concu un micro-systeme,c'est a dire un micro-ordinateur sur un seul circuit,par exemple le 8048 d'Intel,le 6801 de Motorala,le

3870 de Fairchild. (1)

Les plus largement utilises aussi bien dans l'industrie que dans l'enseignement sont le 8080 et 8085 qui sont tres voisins du populaire Z 80 a 8 bits de Zilog et du puissant 8086/8088 a 16 bits d'Intel.

A l'heure actuelle,le 8088 est la base du PC d'IBM.Quand au 6800,il est tres utilise dans les systemes de formation et qu'il appartient a la famille des 6800 de Motorala (qui est caracterisee par une exellente maintenance) qui comprend, aussi,des processeurs ameliores tels que le 6809 a 8 bits et le 6800 a 16 bits.Les premiers prototypes commerciaux a 32 bits existent deja au japon (Toshiba) et meme aux U.S.A ; leurs performances sont tres superieures et leur systeme dispose d'une capacite de 1 Mega byts a 5 Mega byts. (2)

1.1.2 ARCHITECTURE D'UN MICRO-ORDINATEUR:

Un systeme de micro-ordinateur est un ordinateur numerique.Il est classe dans les micro par suite de sa faible taille et de son bas prix.

En general,le microprocesseur constitue la partie unite centale du systeme,ou s'effectuent toutes les operations arithmetique / logiques fondamentales de l'ordinateur,elle constitue aussi le centre ou sont connectes les peripheriques.

Dans ce systeme electronique d'elaboration de donnees,on identifie 3 unites fondamentales qui realisent sa structure.

1-L'UC(unite centale d'elaboration-CPU:Central Processing Unit).

2-Des RAM(memoires vives) et des ROM(memoires mortes) constituent les Memoires.

3-Les dispositifs d'E/S (Input/Output) ou Interfaces.

Ces memoires et ces dispositifs d'E/S sont connectes a l'UC par un moyen physique appelle BUS.

On identifie 3 types de Bus. (fig 1.1) (M1) et (1)

BUS:

Les bus sont des connexions electriques qui unissent l'UC avec les dispositifs d'E/S et de memoire,et qui permettent l'echange du flux continu d'informations entre ceux-ci.

a-Bus De Donnees (Data Bus):

c'est l'ensemble des fils par lesquels transitent les donnees,i.e les operandes,les les resultats de calculs,les caracteres de textes a imprimer, etc....

Ce bus est generalement bidirectionnel et il est de 8 bits.

(1)

b-Bus Des Adresses:

C'est par ce bus que le CPU adressera les dispositifs d'E/S et la memoire,cad donnera l'adresse de la position memoire ou sont contenues les data recherchees.

Ce bus est generalement unidirectionnel,car c'est le CPU qui

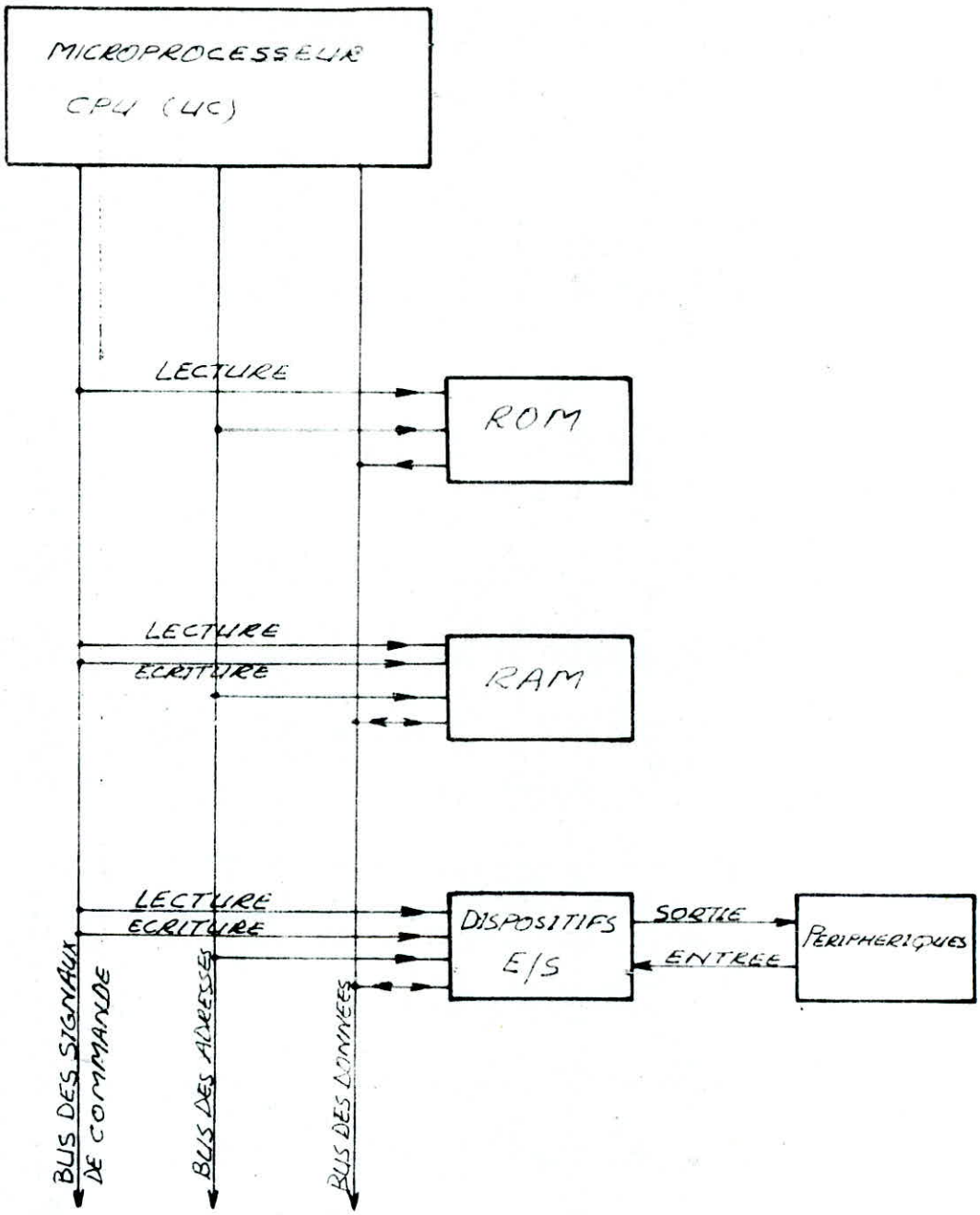


FIG. 1.1 : SCHEMA A BLOCS D'UN MICRO-ORDINATEUR

fournit l'adresse a la memoire ou aux peripheriques (sauf exception).

Ce bus est un 16 bits. (1)

c-Bus Des Signaux De Controle (Control Bus):

Il est constitue de quelques fils destines a commander des operations d'E/S telles qu'une lecture-memoire,une ecriture-memoire,une lecture a partir d'un peripherique,une ecriture a partir d'un peripherique.

La les signaux change d'un constucteur a un autre,ainsi que leur nombre. (1)

D'autre signaux particuliers se trouvant sur le bus des signaux de controle servent a signaler a l'UC des situations particulieres qui apparaissent au cours du processus d'elaboration ou de controle des unites peripheriques a la suite de celle-ci,il faut executer des parties de programme determinees et/ou varier les connexions entre les dispositifs d'E/S (interruptions,demande de bus,etc...). 0

MEMOIRE:

La memoire est formee de l'ensemble d'un grand nombre de positions de memoire(ou cellule de memoire),chaque cellule est identifiee par une adresse se trouvant sur le bus des adresses.

Une memoire contenant des mots de 8 bits(octets ou bytes)

sera une juxtaposition de cellules de 8 bits.La capacite totale de la memoire est variable mais non infinie.

Les memoires se divisent generalement en deux groupes:

-La ROM : qui est une memoire a lecture contenant le le programme.Elle est ineffacable et est progammee par le constucteur.

Ces ROM apparaissent en quatre versions.

. Les ROM standard sont programmees par le constucteur.

. Les PROM (pour ROM programmable) peuvent etre programmees de maniere permanante par l'utilisateur ou le revendeur utilisant un un equipement special.

.Les EPROM (pour PROM effacable) peuvent etre programmees et effacees par l'utilisateur.Les donnees stockees dans une Eprom peuvent etre effacees par exposition au rayons UV.

.L'EAROM est un autre type de prom effacable,son effacement necessite un equipement special.

-La RAM : C'est une memoire a lecture et ecriture contenant les donnes et les resultats,elle est effacable et programmee par l'utilisateur.

Ces RAM sont divisees en deux sous-groupes:

.Si la RAM comporte des circuits de type bascules comme cellules elementaires de memoire,elle est dite RAM Statique.

.La RAM dynamique est de conception beaucoups plus simple et fondee sur des proprietes de capacitance, mais elle implique un refroidissement des cellules de l'ordre de plusieurs centaines de fois par seconde.

Dans les micro,les RAM servent au rangement temporaire des programmes et des donnees de l'utilisateur.Les ROM servent le plus souvent,a stocker les instructions en langage machine qui constitue le programme moniteur.Le moniteur comporte des programme inalterables d'intialisation,d'E/S et des algorithmes arithmetiques. (2) et (3)

-DISPOSITIFS D'E/S (INTERFACES) :

Les dispositifs d'E/S sont les circuits relies directement a l'UC et qui permettent sa connexion aux equipements peripheriques. Les dispositifs d'E/S sont necessaire vu qu'il n'y a aucune standardisation dans les peripheriques de sorte qu'un microprocesseur ne peut pas commander directement les peripheriques.
Le role des dispositifs d'E/S est d'etablir une compatibilite entre les E/S du processeur et celles du peripherique. (3)

1.2 DESCRIPTION GLOBALE DU SYSTEME MPZ-80/EV

1.2.1 DESCRIPTION DU SYSTEME:

Le mini-ordinateur Trainer MPZ-80/EV est un systeme base sur un microprocesseur Z80 concu et realise pour les applications didactiques dans le domaine d'etude des systeme de traitement de donnees et de leurs nombreuses applications.

Le systeme presente une architecture materiel a modules,ayant des possibilites d'extension et de souplesse du materiel du systeme permettant ainsi a l'utilisateur de diversifier et de specialiser les fonctions executees par le micro-ordinateur, grace a l'insertion d'une ou plusieurs cartes avec lesquelles on pourra realiser l'interface de divers appareils.

La structure generale du systeme de base comprend 4 unites fondamentales:

1-Unite Centrale (UC):

Elle est essentiellement composee du rack porte-cartes et de carte que ce dernier contient et qui constitue l'ordinateur et l'alimentation.

Le tout est assemble dans une structure ayant la forme d'un boitier.

Cette UC est le centre ou sont connectees les 3 unites ci-dessous.

2-Unite Disque Souple:

Elle est composee de deux floppy disk drivers qui permettent l'utilisation de disquettes flexibles comme dispositif de memoire de masse du systeme.
Les programmes qui constituent le logiciel de base, les programmes d'application et les differentes archives de donnees se rapportant a ces programmes sont mis en memoire dans ces disquettes.

3-Terminal Video :

Il permet la communication entre l'ordinateur et l'utilisateur.
Il est constitue d'un moniteur et d'un clavier.

4-Imprimante :

Elle permet d'ecrire sur papier toutes les informations traitees par l'ordinateur.

La fig(1.2), montre la configuration complete du systeme oriente vers les applications de l'automation industrielle et le controle de processus.
Cette configuration est obtenue en ajoutant a l'UC les modules materiels approprie qui servent a piloter toutes les unites peripheriques dont on a besoin.
De cette maniere on peut commander des dispositifs comme les instruments programmables(en utilisant l'interface standart IEEE 488),les programmateur EPROM,les Convertisseurs A/D et D/A,les moteurs pas a pas ou en cc,etc...,jusqu'a ce qu'on

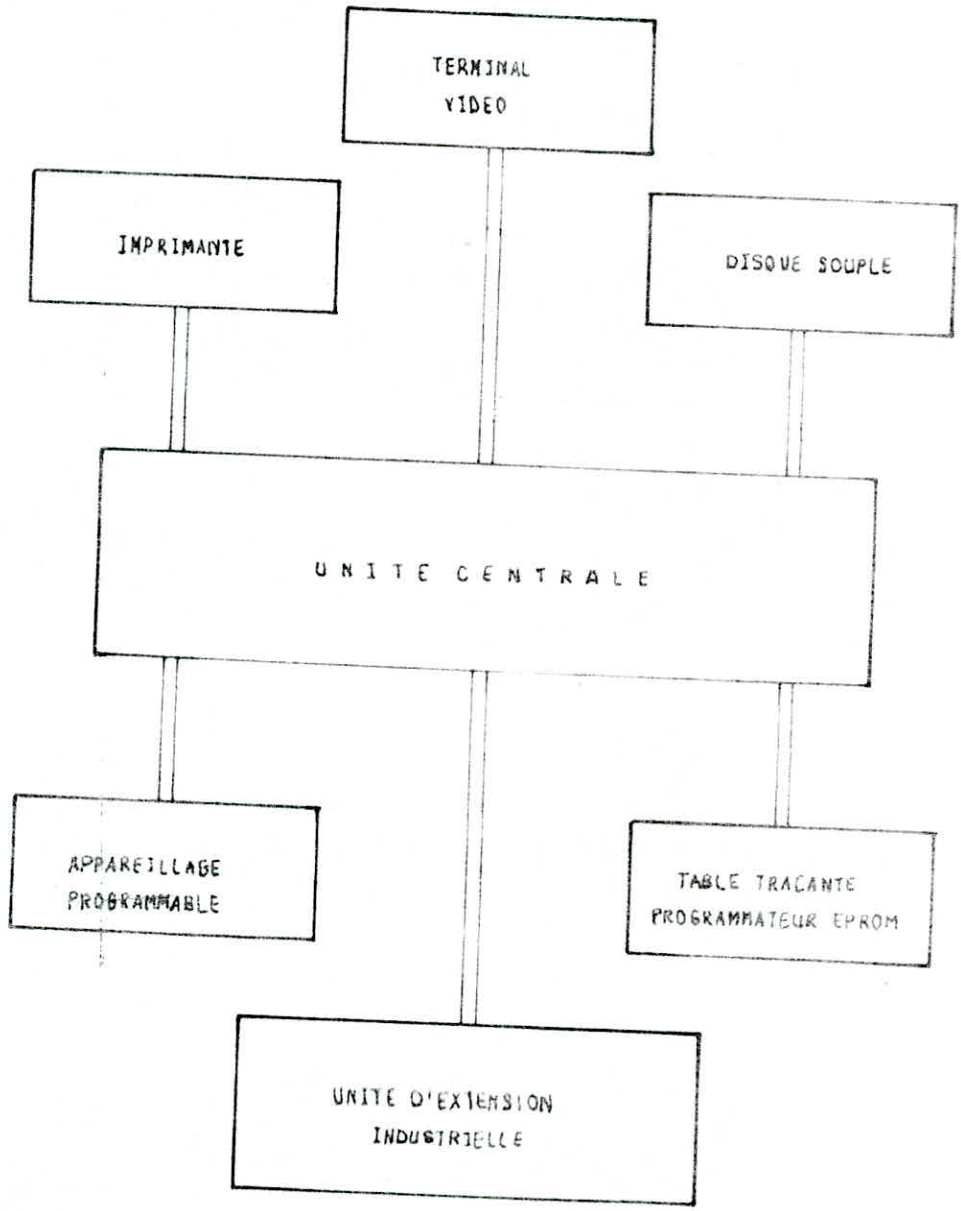


FIG. 1.2 MINICOMPUTER TRAINER NPZ-80/EV DANS LA CONFIGURATION "COMPLETE"

arrive a construire des systemes complets qui simulent l'utilisation effective d'un ordinateur dans l'automation industrielle.

(M1) et (M2)

1.2.1.1 REALISATION PHYSIQUE DES TROIS UNITES FONDAMENTALES DANS LE MPZ-80/EV:

Les configurations de base du systeme sont:

-L'unite centrale est constituee du microprocesseur Z80.

-64 Kbytes de memoire MEV dynamique.

-4 Kbytes de memoire EPROM contenant le microprogramme du systeme.

-Controle du disque souple (dispositif de memoire de masse), (voie parallele).

-Deux voies de communication serie pour la liaison du terminal video et de l'imprimante.

-Le BUS utilise est le MMS-8 qui est un standard pour les les systemes a microprocesseurs.

La configuration de base du systeme MPZ-80/EV prevoit la presence de six cartes placees a l'interieur de l'UC. Toutes les connexions qui constituent les bus des donnees, des adresses,des signaux de controle,ainsi que les connexions qui amenent sur les cartes les tensions d'alimentation necessaire au fonctionnement sont contenues sur une carte appellee CARTE MERE. voir fig(1.2.1) (M2)

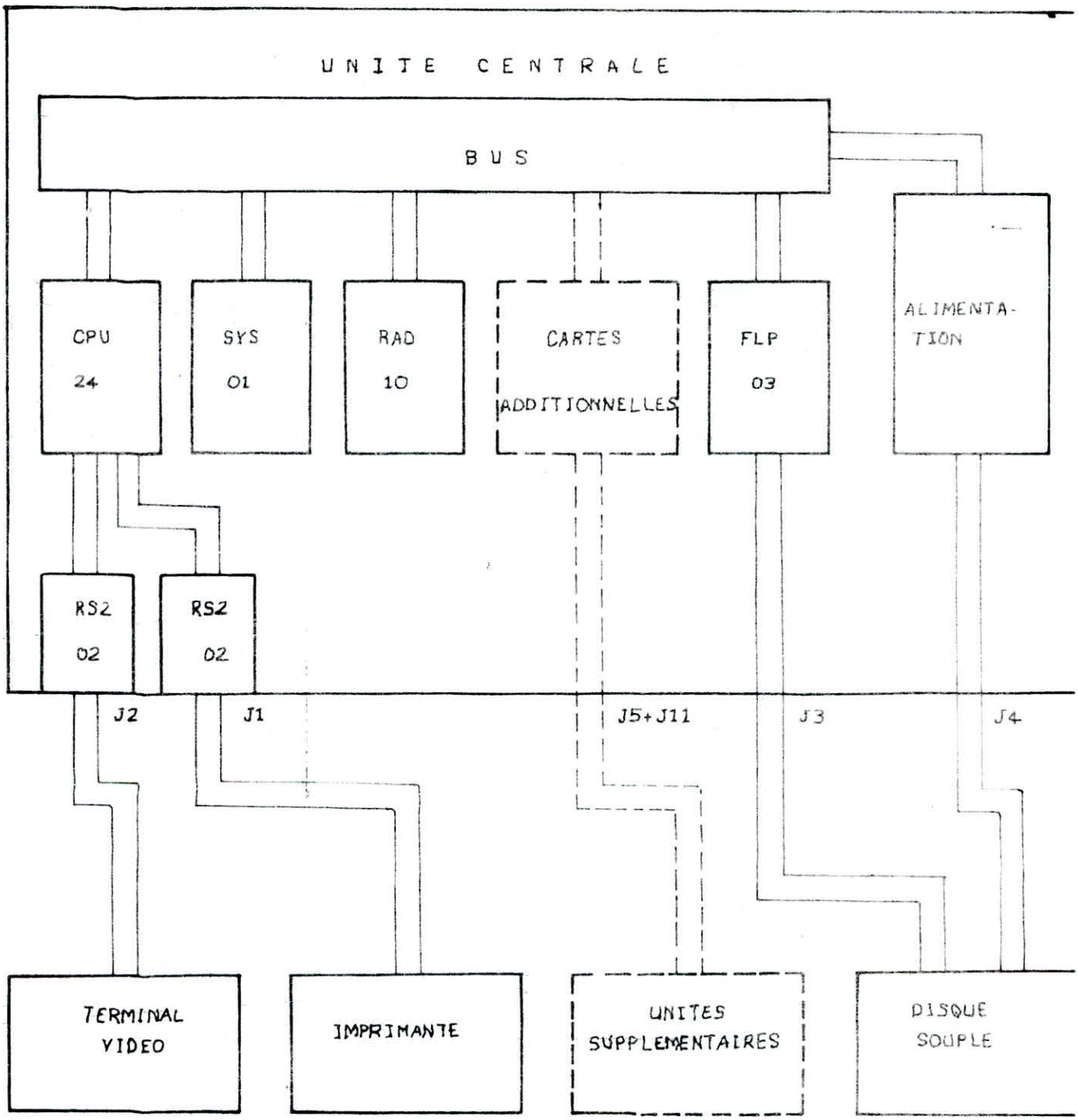


FIG. 12.1 : Organigramme du Minicomputer-Trainer mod. MPZ-80/EV

1-Carte CPU-24:

Cette carte a pour but de traiter les informations par l'intermediaire du microprocesseur Z80 se trouvant sur cette carte;un autre dispositif se trouvant sur cette meme carte(circuit integre LSI Z80-SIO(Serial Input/ Output)) au moyen duquel le systeme est dote de deux voies d'interface serielle vers les systemes peripheriques (terminal video et imprimante).
Le circuit integre LSI Z80-CTC(Counter Timer Circuit) est programme et controle par l'UC comme dispositif d'E/S.

(M1) et (M2)

2-Cartes RS2-02:

Ces cartes sont connectees aux deux portes serielles du SIO contenu dans la carte CPU-24 et permettent d'adapter les caracteristiques electriques de l'interface a celles qui sont requise par l'unite peripherique connectee a celle-ci.

(M1) et (M2)

3-Carte SYS-01:

Cette carte Contient un ensemble de memoires EPROM;dont la caracteristique est de conserver les informations qu'elles contiennent meme en l'absence de la tension d'alimentation.

(M1) et (M2)

4-Carte RAD-10:

Cette carte contient les circuits integre qui

constituent la memoire MEV du systeme et contient en outre l'ensemble des circuits necessaire a la synchronisation parfaite des operations de lecture et d'ecriture que l'UC doit realiser sur la memoire.Elle contient 64 Koctets de MEV dynamique.

C'est dans cette partie de memoire qu'on charge les programmes de developpement et d'application,quand l'utilisateur veut les mettre en execution.

(M1) et (M2)

5-Carte FLP-03:

Cette carte constitue le module d'interface de l'UC avec deux floppy disk drivers places dans l'unite disque souple.

Elle convertie les informations traitees par l'UC (1 octet de 8 bits) dans la forme requise par le floppy disk drivers pour leur enregistrement magnetique sur les disquettes et inversement (conversion inverse).

Le circuit integre LSI appelle floppy disk controller (controlleur disque souple) sert a effectuer les operations complexes sur les donnees necessaires a realiser sur ce type de fonctionnement.

Le circuit F.D.C gere tous les signaux de controle accompagnant les flux de donnees a partir et vers les floppy drivers,en signalant les erreurs et mauvais fonctionnement.

(M1) et (M2)

1.2.2 DESCRIPTION DU Z80,DU 8203 D'INTEL ET DU 8272 D'INTEL:

1.2.2.1 DESCRIPTION DU Z80:

Le microprocesseur(CPU:Central Processing Unit) est compose :

-d'une U.A.L d'execution des calculs et operations logiques sur les donnees des registres internes.

-d'un decodeur d'instuction.

-d'une unite de commande 'sequencant' le systeme.

-du bus de liaison avec l'exterieur.

La fig ci-dessous (1.2.2) montre les registres internes du CPU. A l'interieur du CPU,le programmeur n'a acces qu'aux registres internes.Ce sont des cases particulieres qui contiennent 1 ou 2 octets.

Accumulateur		Plage		Accumulateur		plage		Interruption vecteur		Memory refresh	
	A		F		A'		F'		I		R
	B		C		B'		C'	Registre index			IX
	D		E		D'		E'	Registre index			IY
	H		L		H'		L'	Pointeur de pile			SP
								compteur programme			PC

fig (1.2.2)

(M1) et (4)

Remarquer que les registres 8 bits ont un nom d'une seule lettre,et ceux de 16 bits ont un nom de deux lettres.

Examinant tous ces registres et leurs utilisations.

-Le registre A ou accumulateur (8 bits) a un role preponderant: c'est sur lui que porteront les operations arithmetiques et et logiques.

-Les registres B,C,D,E,H,L (8 bits) sont utilises pour stocker temporairement des donnees destinees a etre traite par l'accumulateur.

Le temps d'acces a ces registres est bien inferieur a celui de la memoire externe,d'ou l'interet a bien les utiliser.

-Les registres A',B',C',D',E',F',H',L',sont exactement symetriques.Ils prennent la place de A,B,C,D,E,F,H,L grace aux instuctions EX et EXX.Ils constituent la deuxieme banque de registres.On travaille avec la premiere ou la deuxieme, mais jamais les deux en meme temps.

La banque unutilisee peut conserver temporairement des donnees.Leur appel sera plus rapide que si on les stocke en memoire.

-Les registres IX et IY (16 bits) servent d'index (I index), pour acceder a une donnee.En general,ils contiennent une adresse,a laquelle on ajoute,ou on retranche,un deplacement pour trouver la donnee recherchee.

-Le registre R contient une adresse qui change sans cesse,et qui assure le rafraichissement des memoires vives externes.

-Le registre SP est le pointeur de pile (stack pointer).

La pile est une zone de memoire ou l'on pousse (PUSH) les informations sans avoir a se soucier d'adresses.Le moment venu,il suffira de les retirer (POP).

-Le registre F (Flag-drapeau) n'est pas un registre de travail,chacun des bits qui le composent est un indicateur qui,suivant son etat (0 ou 1),renseigne sur le resultat de certaines operations.

-Le registre PC (Compteur Programme) contient a chaque instant l'adresse de la prochaine instruction a executer.

Le Z80 possede pres de 158 instructions dont 78 sont celle qu'utilise le 8080 A.

(M1) et (4)

1.2.2.2 DESCRIPTION DU 8203 D'INTEL:

Le 8203 d'Intel est un circuit integre LSI se trouvant sur la carte RAD-10 et il est appele CONTROLEUR DE MEV dynamique.

Les taches fondamentales effectuees par ce bloc controleur de de MEV dynamique sont:Les operations de renouvellement des donnees et de multiplixage des adresses.

Ce controleur de MEV dynamique est pourvu d'une temporisation fournie par un quartz et s'occupe de la production et de la synchronisation des signaux au fonctionnement correct des circuits de MEV dynamique.

Quand l'UC n'effectue pas d'operations de lecture ou d'ecriture de la memoire,le controleur envoie les adresses qui provoquent le retablissement cyclique des informations (operations de renouvellement) aux MEV dynamique.

(M1)

1.2.2.3 DESCRIPTION DU 8272 D'INTEL:

Le 8272 d'Intel est un circuit integre LSI appele CONTROLEUR DE DISQUE SOUPLE (FDC) dont il constitue le circuit principal de la carte FLP-03.

Ce circuit est connecte au bus de donnees et des signaux de controle de l'UC et realise en outre,avec d'autres circuits, la connexion avec les floppy disk drives.

Les opperations effectuees par le FDC et se referant a son interface avec l'UC est:

-Le FDC recoit de l'UC les codes de commande et les parametres relatifs aux operations qu'il faut executer.

-IL recoit de l'UC les donnees qu'il faut ecrire sur le disque et envoie a l'UC les donnees lues sur le disque.

-Il effectue un diagnostique sur le resultat des operations effectuees en passant a l'UC les codes d'identification des des erreurs relevees eventuellement pendant les operations.

En ce qui concerne la commande des floppy disk drives,le controleur execute les fonctions suivantes:

-Selection du drive sur lequel il faut effectuer l'operation et si cela est necessaire,de la face du disque conserne.

-Commande pour l'abaissement de la tete ou pour la mise en fonctionnement du moteur de rotation du disque.

-Verification de la condition de READY du drive selectionne.

-Envoi du moteur pas a pas du drive des signaux de direction

et de STEP qui realisent le positionnement de la tete sur la trace requise.

-Verification et signalisation a l'UC des conditions de trace 0,de write protect et de disque à double face.

-Commande des signaux qui realisent l'ecriture des donnees.

-Realisation de la conversion parallele/serielle des donnees au cours des operations d'ecriture.

-Realisation de la conversion serielle/parallele des donnees au cours des operations de l'ecture du disque.

(M1)

1.3 UTILISATION AVEC LE MONITEUR MON-80:

Le moniteur MON-80 constitue le microprogramme du systeme situe dans la memoire EPROM,il est actif des qu'on allume le systeme; ces routines peuvent etre appellees par des programmes utilisateurs.

Ce programme moniteur peut accomplir les fonctions suivantes:

-Mise au point de depart du materiel du systeme et programmation des dispositifs d'E/S.

-Conversion avec l'operateur grace a des commandes realisant des fonctions d'utilite generale.

-Fonctions de debug sur des programmes ecrits par l'usager.

-Gestion des situations d'erreur ayant lieu lorsqu'on accede aux dispositifs du systeme.

-Memorisation du systeme operationnel et mise en execution de celle-ci.

Les fonctions realisees par le moniteur MON-80 permettent a l'usager,grace a des commandes appropriees,d'acceder a la memoire interne de l'ordinateur et de realiser des operations de lecture,d'ecriture,de copiage et de controle de la memoire. Tout ceci permettra a l'operateur d'ecrire de simples programmes d'essai en code objet Z80 dans la memoire.
Ceci afin de transferer des donnes d'un secteur a un autre secteur de memoire,d'effectuer des operations logiques et arithmetiques,d'envoyer des commandes et des sequences de caracteres au terminal video et a l'imrimante,etc...
Les commandes du programme MON-80 permettent a l'UC de mettre en execution des programmes memorises et d'en arreter le cours en certains points preetablis;elles permettent aussi d'examiner l'etat des registres internes de l'UC lors des phases statiques et d'en modifier le contenu,ou d'examiner les registres durant les phases dynamiques(i.e:quand le programme passe par certains points preetablis.
Ce programme MON-80 contient des routines qui permettent aux programmes utilisateurs d'acceder aux elements de materiel de l'ordinateur(terminal video,l'imprimante ou les disques).
Cependant quand l'usager aura une experience telle qu'il ne pourra plus se passer de ces dernieres caracteristiques du MON-80,il se rendra compte en poursuivant son travail que les instruments mis a sa disposition par le systeme operationnel sont encore d'autant plus valables.

(M2)

1.4 UTILISATION AVEC LE SYSTEME OPERATIONNEL CP/M :

Le systeme operationnel est un systeme qui met a disposition des fonctions qui se referent au controle de l'ordinateur sur l'acces aux dispositifs composants;il represente le logiciel de base de l'ordinateur.

Dans le but de developper le logiciel d'application utilisateur,l'utilisation de ces instruments de logiciel (programme de developpement) qui existent,peuvent fournir de tres hautes prestations.

Les programmes d'application utilisateurs sont ecrits dans une forme proche du langage humain,grace aux programmes d'EDITION (ED et WS).

La redaction des programme pourra se faire soit en langage Assembler,soit en langage plus evolues et de tres haut niveau telque le Basic,le Fortran,etc...

En outre ce logiciel de developpement comprend:

-Les programmes Assembler et Macro-Assembler (ASM et M80), employes quand les programmes ont ete rediges en langage assembler,qui permettent de traduire le texte des programmes d'application en langage machine,et permettent aussi a l'ordinateur de les executer.

-Des Compilateurs qui traduisent les programmes d'application ecrits en langage plus evolue.

-Les programmes de Debug (DDT et ZSID),qui permettent d'executer partie par partie les programmes d'application

developpes,et de verifier leur exactitude du point de vue logique et syntaxique.

L'emploi de ce systeme operationnel CP/M founit des moyens de tres haute capacite utiles au developpement du logiciel d'application,et permet aussi de faire tourner sur son propre ordinateur un grand nombre de programmes d'application deja rediges et capables de fournir des fonctions de traitements de donnees dans les champs d'application les plus varies.
Ce logiciel peut etre divise en plusieurs branches:

-Traitement de textes.

-Banque de donnees.

-Logiciel de gestion d'entreprises.

-Logiciel scientifique,mathematique et statistique.

-Applications industrielles.

-Transmission de donnees.

(M2)

1.5 INTRODUCTION SUR LE PROGRAMME D'EDIT,D'ASSEMBLEUR ET DE DÉBUG:

Un programme ecrit en code machine est de 10 a 100 fois plus rapide qu'un meme ecrit en basic.Cependant,le code machine est vraiment tres abstrait,illisible,et elloigne du langage humain (car il n'est compose que de nombres binaires,suite de 0 et 1) on ne peut envisager de l'ecrire directement.
On va donc employer un terme: On ecrira dans un langage

symbolique qu'un programme special se chargera de traduire en code machine.Ce programme s'appelle ASSEMBLEUR.On nomme langage machine ou par extension Assembleur le langage symbolique dans lequel on ecrit.

Outre la rapidite,on gagnera egalement en place memoire.

On utilise donc generalement une representation plus parlante: C'est le code mnemonique.Bien entendu,ces mnemoniques ne sont pas comprise par le microprocesseur et il est necessaire de les traduire en code machine pour les executer.Pour cela, il existe un programme assembleur,qui effectue la traduction mnemonique-code machine.

L'assembleur est par consequent, un programme traducteur et, par extension,il designe aussi la programmation en langage machine.

Le programme ecrit en mnemoniques constitue un texte peu litteraire ;on l'appelle programme source qui obeit a certaines regles de syntaxe.

L'assemblage,c'est a dire la traduction du programme source generera un programme objet en code machine directement executable.

L'ecriture du programme source est grandement facilitee par un EDITEUR de texte qui permet de rentrer les instuctions aisement,de retrouver tel ou tel passage,de corriger les erreurs et d'inserer des commentaires.Les erreurs de syntaxe sont detectees par l'assembleur au moment de l'assemblage,mais non les erreurs de logique,ni de l'oubli d'une ou plusieurs

instructions.

On utilise alors un moniteur pour le debugging,c'est a dire la chasse au fautes.Le moniteur admet un nombre varie de commandes,comme l'execution pas a pas,la visualisation des registres,la recherche d'une suite d'octets en memoire,leur modification,etc... (4)

Le programme d'edit,d'assembleur et de debug sont identifies dans le disque par:

ED.COM , ASM.COM , ZSID.COM

Se referer au manuel du systeme operationnel pour une etude detaillee.

2 DESCRIPTION DES PERIPHERIQUES

2.1 LIAISON A DISTANCE ENTRE LE TERMINAL VIDEO ET L'IMPRIMANTE PAR MODEMS DE TRANSMISSION DE DONNEES.

L'emploi du mini-ordinateur dans les liaisons par tansmission de donnees,se traite comme dans le laboratoire de telematique. Pour la realisation d'une liaison inherente a la transmission de donnees,il faut tenir compte de plusieurs parametres,telque le protocole de communication employe et la vitesse d'echange de l'information.

On remarque qu'il existe une nette difference entre les modes de transmission synchrone et asynchrone,en choisissant le type de liaison en fonction du protocole de communication employe.

Dans une liaison inherente a la transmission de donnees (modem-ligne-modem) il n'est pas toujours necessaire de respecter les conditions de synchronie ou d'asynchronie imposees par les appareils utilisateurs de donnees.

En effet,il est possible d'avoir une liaison synchrone dans le cas d'utilisateurs asynchrones,et il est impossible d'avoir une liaison asynchrone dans le cas d'utilisateurs synchrones.Car pour transmettre des donnees asynchrone,il est necessaire que la vitesse nominale de la liaison soit un multiple entier et egal a la vitesse nominale des utilisateurs a relier.Pratiquement,le compromis entre la qualite de la transmission et le cout de la liaison comporte l'emploi d'un multiple de 4.D'habitude les liaisons pour utilisateurs

asynchrones sont realisees avec des modems bande basse (synchrone par nature) en mettant en pratique ce qui a ete vu. En general,dans le cas de liaisons locales avec emploi de modems bande basse,la methode du multiple de la vitesse de transmission s'impose,ceci a cause du cout peu eleve de ce type de liaison. (M2)

2.1.1 LIAISON DU TERMINAL VIDEO:

La liaison du terminal video est assez simple.En effet,une fois la condition classique (multiple de la vitesse) est realisee,aucune autre disposition n'est a prendre.
On rappellera seulement que si le multiple de la vitesse est superieiur a 4,il ne cree aucun inconvenient.
Sur le plan local,le desavantage du a l'uniformite des modems a la vitesse de fonctionnemnt maximale,est lie a la distance a franchir;en fait le modem bande basse MBB-83/EV est a meme de franchir,a la vitesse de 19200 bauds et sur un cable de telephone de 6/10 mm,une distance de 10 km.
Lorsque l'on realise une ligne pour terminaux asynchrones employant des modems synchrones,la structure du reseau est en principe du type point a point,car en general les terminaux asynchrones ne possedent pas une intelligence propre permettant de les orienter dans les liaisons multipoints. Naturellement tout ceci rend la realisation des liaison plus simple.En particulier,dans ce genre de liaison il est conseille d'employer les modems predisposes sur porteuse fixe.

Grace a cette solution,on a une liaison toujours prete et facilement diagnosticable en cas d'interruption ou de panne de l'un des deux modems.La liaison devra etre en duplex;par consequent,dans le cas des modems MBB-83/EV,on pourra la realiser avec 4 fils. (M2)

2.1.2 LIAISON DE L'IMPRIMANTE:

Contrairement au terminal video, la liaison de l'imprimante est plus compliquée quand a la programmation et a la realisation,ceci a cause de la nature du fonctionnement de celle-ci.

Le fonctionnement de l'imprimante est lie a la capacite des buffers de reception;dans la pratique toutes les donnees envoyees a l'imprimante ne peuvent pas être immediatement imprimées;en effet,des pauses sont necessaires a l'imprimante pour regler la vitesse de reception des donnees sur la vitesse d'impression(celle-ci étant presque toujours limitée par des problèmes mécaniques).

Il existe de nombreux systémes pour resoudre ce probléme;ils peuvent être de nature Logiciel ou de nature Materiel.

On va analyser un systeme de nature materiel.

2.1.2.1 CONTROLE DE L'IMPRESSION A L'AIDE DE L'INTERFACE:

Il est possible de gerer le controle du buffer de facon semblable a celle du systeme 'X-ON,X-OFF',grace a l'interface de donnee normalisee(CCITT V24/V28 ou EIA RS232-C).

L'obtension du niveau d'alerte dans le buffer entraine le

changement d'etat d'un BUSY.

Il n'existe pas de norme prescrivant le juste contact a utiliser,car il existe differentes imprimantes qui peuvent mettre a disposition du BUSY des contacts differents.

On remarque que certaines imprimantes fournissent le critere d'appareil terminal de donnees pret en utilisant le fil 20 (dans ce cas le critere busy utilisera un autre contact).

Dans l'imprimante (EPSON RX-80) employee dans le laboratoire de telematique,le critere de busy se trouve au contact 20 (DTR).

Si l'imprimante(EPSON RX-80) doit etre employee sur longue distance avec modems,il sera necessaire de transferer sur la ligne le critere de busy se trouvant au contact 20(DTR).

Vu que les modems regis par les normes CITT,on n'a pas prevu le transfert de critere de busy,ni celui du DTR,il est necessaire d'effectuer certaines adaptations.

Avec l'imprimante reliee a l'aide de modems MBB-83/EV,on utilisera le sens de transmission ordinateur-imprimante pour effectuer un envoi normal,et le sens imprimante-ordinateur pour obtenir le retour du signal de gestion.

En deduit que la liaison doit etre en duplex,car l'imprimante pourrait a tout moment envoyer le signal d'alarme de buffer plein.

Dans les modems bande basse,les deux signaux de l'interface devant etre transferes sont le critere de demande de transmission et le signal correspondant aux donnees recues.

On transfert le critere de busy par l'entremise du fil de transmission des donnees,et ceci ne comporte pratiquement aucun retard de transfert.

Ainsi la disposition exacte des cables est celle qu'on voie a la fig(2).

Les envois des fils d'interface peuvent etre effectues avec les visualisateurs d'interface VIF-83.

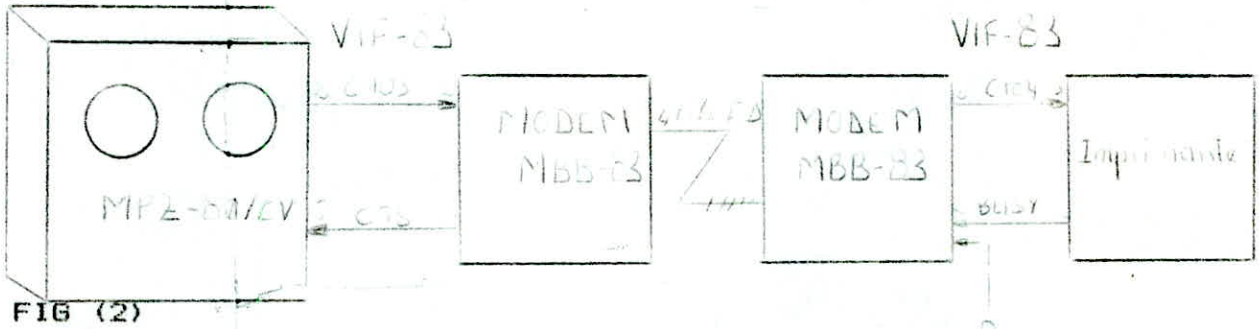


FIG (2)

NB: Dans cette figure sont montres les criteres necessaires a la gestion du buffer;en se qui conserne les autres criteres,se referer a une liaison normale point a point duplex a 4 fils.

Le busy est un fil de l'interface.

(M2)

2.2 FONCTIONNEMENT DE LA CARTE D/A A/D

2.2.1 DESCRIPTION DE LA CARTE:

La carte CAR est constituee par deux sections qui se distinguent non seulement du point de vue fonctionnelle et circuitale,mais aussi correspondant a deux circuits imprimes.

-La section correspondant a la conversion D/A avec resolution 12 bits est est un circuit de base,car il porte le connecteur pour le Bus d'extension,la plaque frontale avec les terminaux pour les signaux analogiques,le decodeur d'adresse et la tension reglable de reference pour les convertisseurs.
Elle peut fonctionner de facon autonome.

-L'autre section correspondant a la conversion A/D avec resolution 12 bits est un circuit du type'PLUG-IN'(fiche); il doit etre installe sur la plaque de contact appropriee 'DUAL-IN'.

L'emploi de cette carte implique toujours le montage du module CAR-A/D sur le module CAR-D/A.

Dans sa version complete CAR ,les differents potentiometres de reglage,les ponts de selection pour le fonctionnement unipolaire / bipolaire et les points de test sont toujours accessibles (M7)

2.2.2 STRUCTURE FONCTIONNELLE:

2.2.2.1 SECTION COMMUNE:

Cette section contient le generateur

de la tension de reference des convertisseurs,reglable de -4 a -8Vcc au moyen d'un potentiometre,ainsi que le systeme de decodage de l'adresse de la carte ou d'une partie de la carte,car la partie A/D peut etre mise en adresse independement de la partie D/A,avec production de signaux de controle (CD4-CD1) pour la section D/A et (CA4-CA1) pour la section A/D. Les adresses attribuees a chaque voie de conversion sont determinees par des micro-interrupteurs (4 bits pour A/D et 4 bits pour D/A).

On remarque seulement l'emploie du byte le moins significatif du mot d'adresse (AD8-AD1).

2.2.2.2 CARTE D/A (CAR D/A)

Cette section comprend tous les dispositifs necessaires aux trois voies de conversion D/A et distribue la donnee 12 bits (constituee apres deux lectures du bus des donnees de 8 bits) au convertisseur D/A selectionne. voir fig(2.2).

En examinant le schema a blocs (fig.2.2.1),on voit que les 8 bits provenant du bus des adresses (AD8-AD1) sont,du point de vue fonctionnel,inseres separement dans les deux 'NIBLES' (demi-bytes).Le nible d'ordre superieur (AD8-AD5) determine l'adresse de la carte ou d'une partie de la cart que l'on veut utiliser;le nible d'ordre inferieur porte l'information relative a la voie A/D ou de la voie D/A.

Ainsi,apres les 'bus buffer',ces memes lignes D'adresse

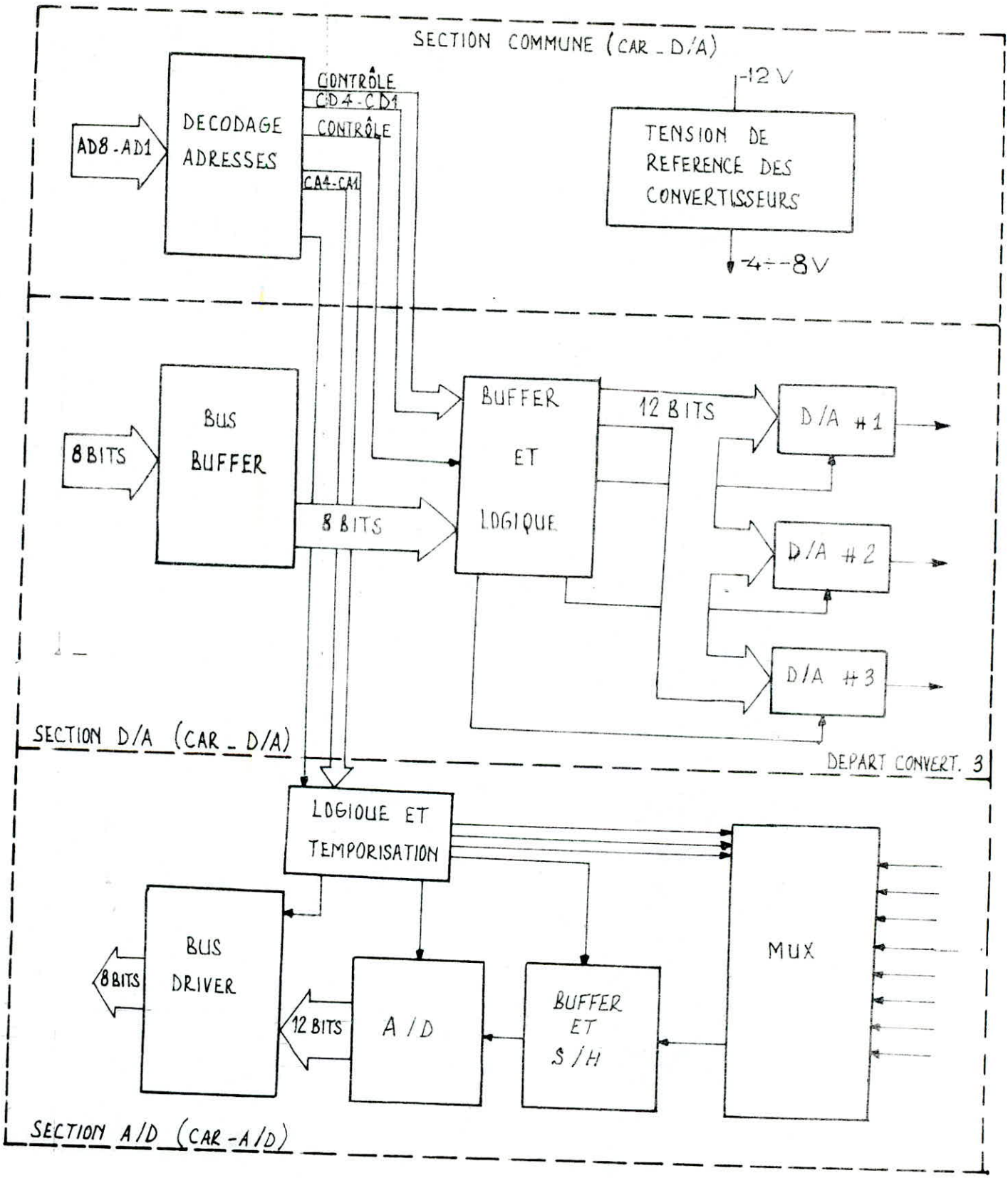

SECTION COMMUNE (CAR_D/A)
DECODAGE ADRESSES
AD8-AD1
CONTRÔLE
CD4-CD1
CONTRÔLE
CA4-CA1
-12 V
TENSION DE REFERENCE DES CONVERTISSEURS
-4÷-8V
BUS BUFFER
8 BITS
8 BITS
BUFFER ET LOGIQUE
12 BITS
D/A #1
D/A #2
D/A #3
SECTION D/A (CAR_D/A)
DEPART CONVERT. 3
LOGIQUE ET TEMPORISATION
BUS DRIVER
8 BITS
12 BITS
A/D
BUFFER ET S/H
MUX
SECTION A/D (CAR-A/D)

Fig. 2.2

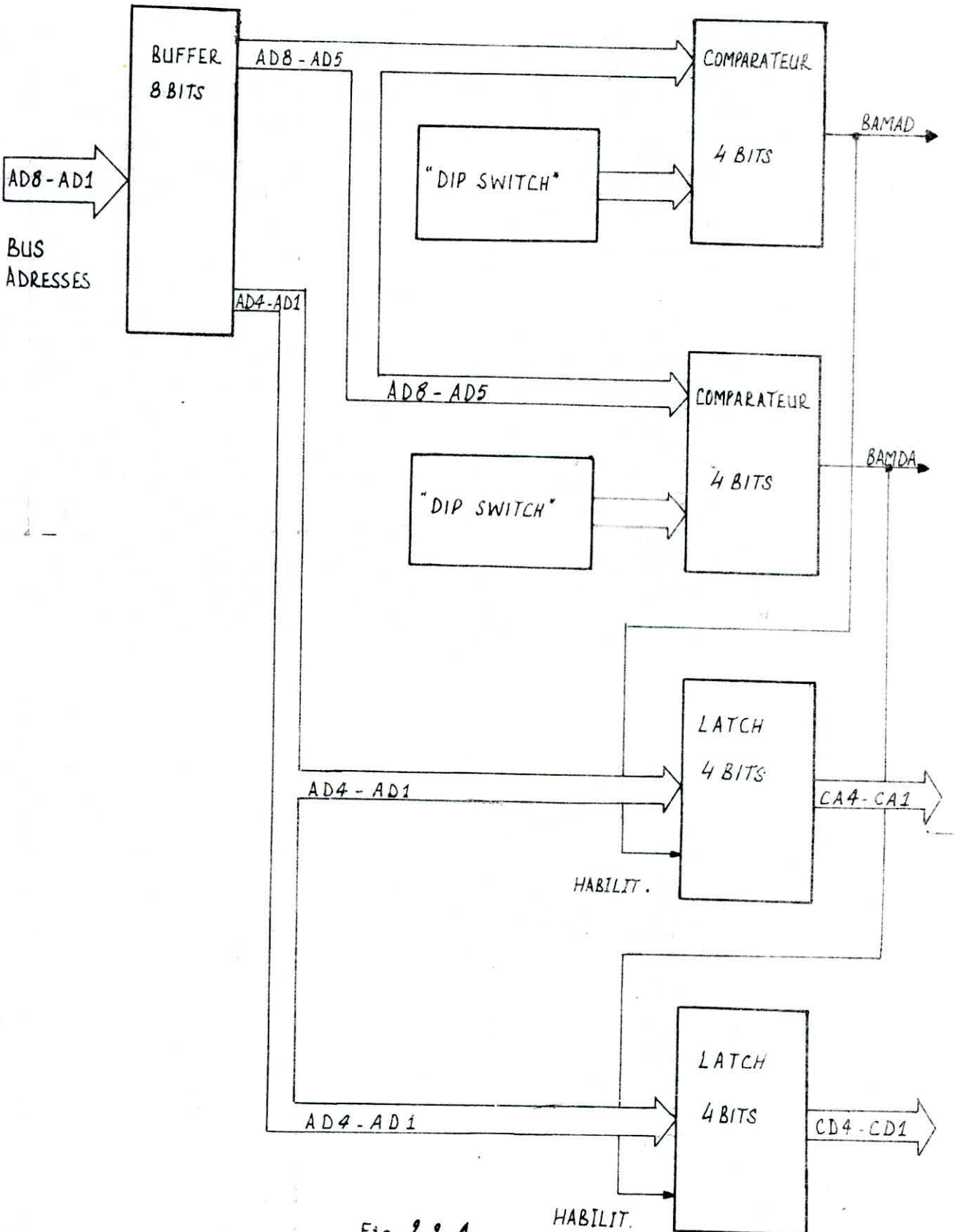


Fig. 2.2.1

DECODAGE DES ADRESSES ET SIGNAUX DE CONTRÔLE

(AD8-AD5) arrivent a deux comparateurs distincts qui les comparent a celles qui proviennent des interrupteurs de preselection (DIP suitches).
Si l'adresse,codifiee a l'aide de l'un de ces interrupteurs est egale a l'adresse provenant des 'bus',l'un des signaux BAMAD ou BAMDA aura la valeur logique 1
BAMAD=BOARD ADRESS MATCH A/D;BAMDA=BOARD ADRESS MATCH D/A).
L'activation de l'un de ces signaux donne lieu a la memorisation du nible inferieur pour controler la conversion (CD4-CD1,CA4-CA1).
Une fois que la carte et sa section ont ete selectionnees, la composition de la donnee 12 bits a lieu grace a deux emissions successives des bytes les plus ou les moins significatifs sur le bus des donnees.
Ces bytes sont memorisees dans les latchs intermediaires habilites par CD3(byte LS) et CD4(byte MS).
Ensuite grace au bus interne de 12 bits,la donnee a convertir est distribuee au latch specifique de la voie,ce dernier etant habilite par le signal correspondant CONV3-CONV1 qui est le produit du decodage CD2-CD1.
Chaque convertisseur produit sans interruption le signal analogique correspondant a la valeur digitale mamorisee dans la section precedant immediatement le convertisseur (memoire ou Latch de voie). fig(2.2.2).
Voir les deux schemas electriques de CAR-D/A

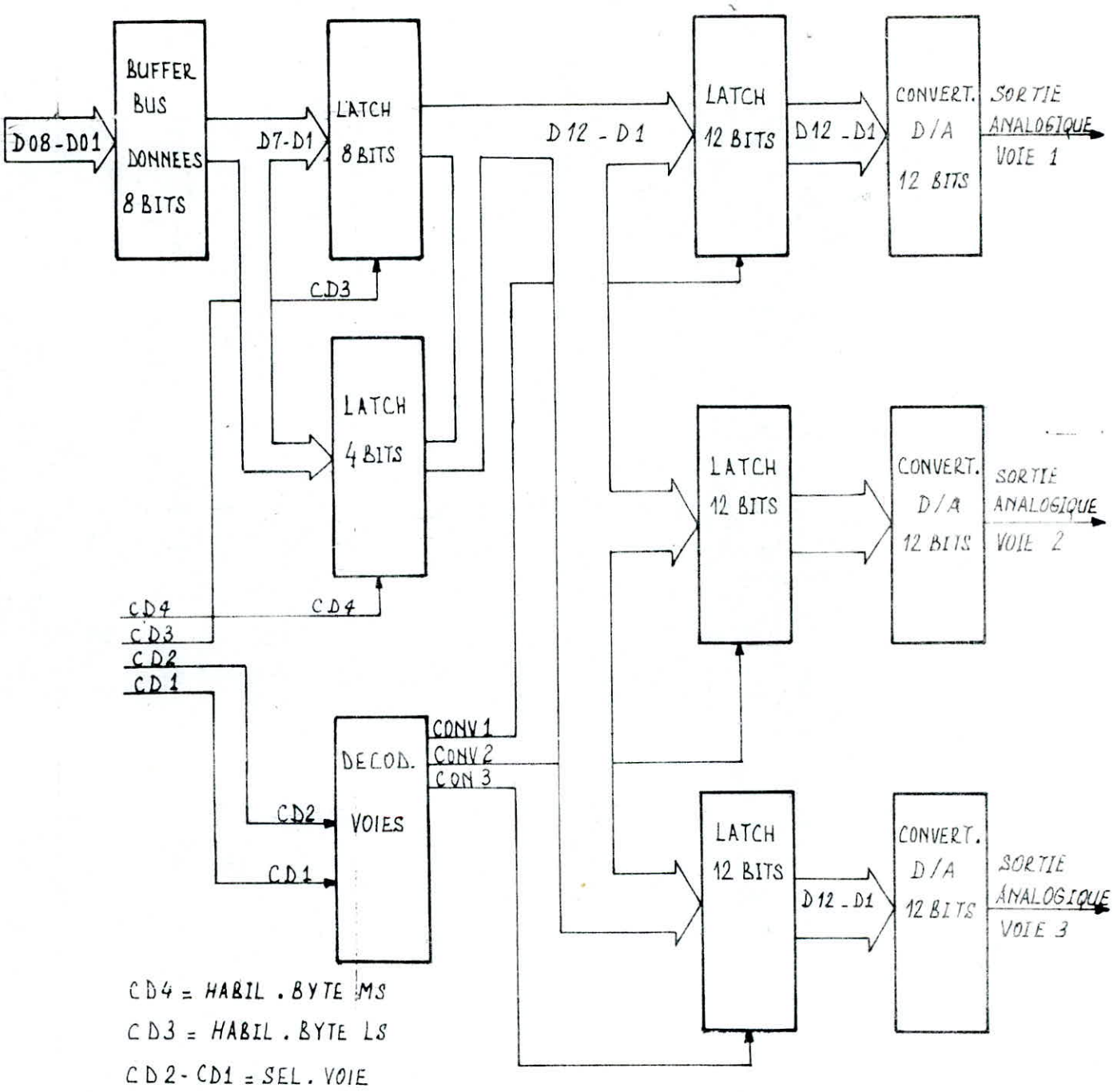
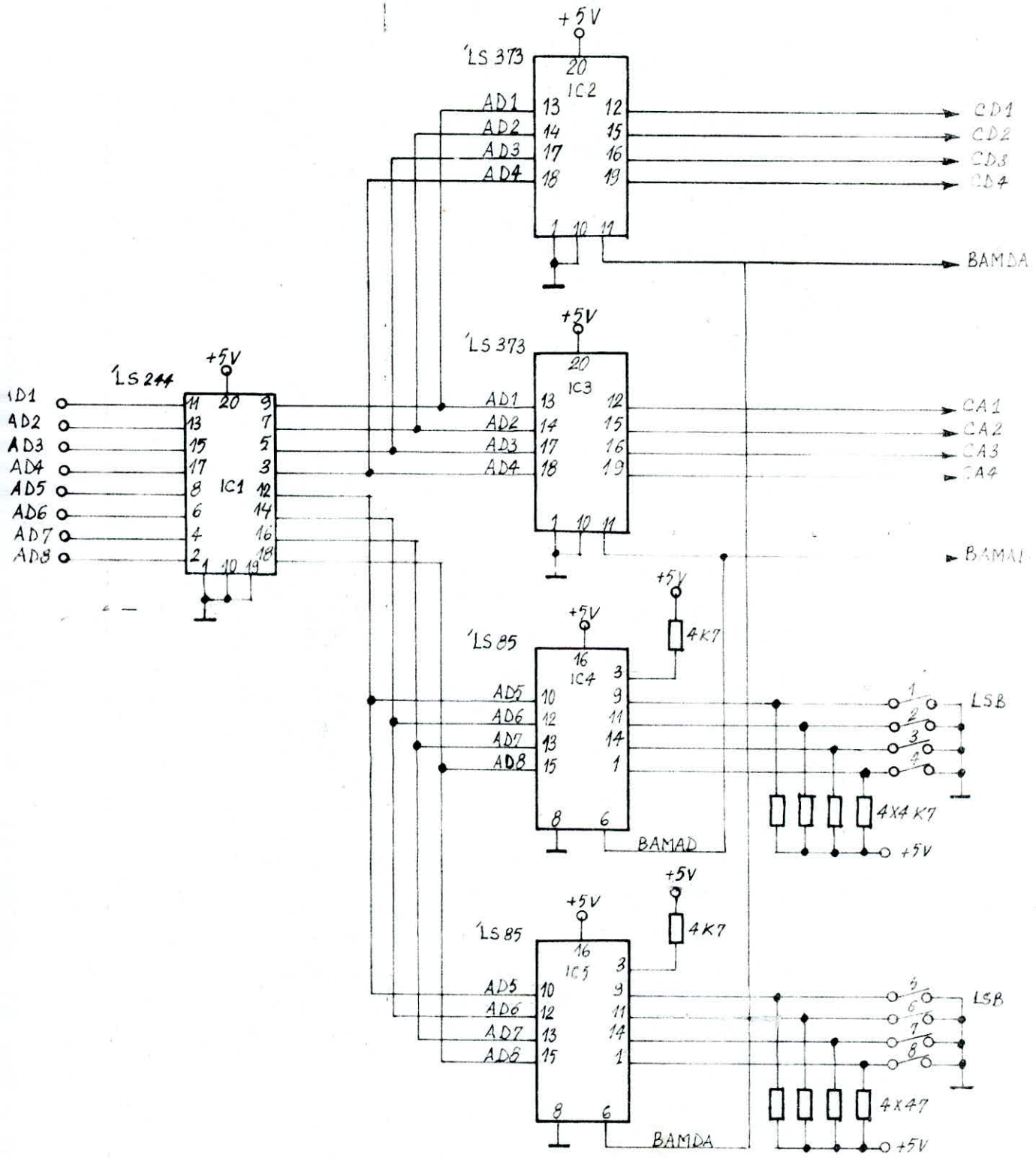
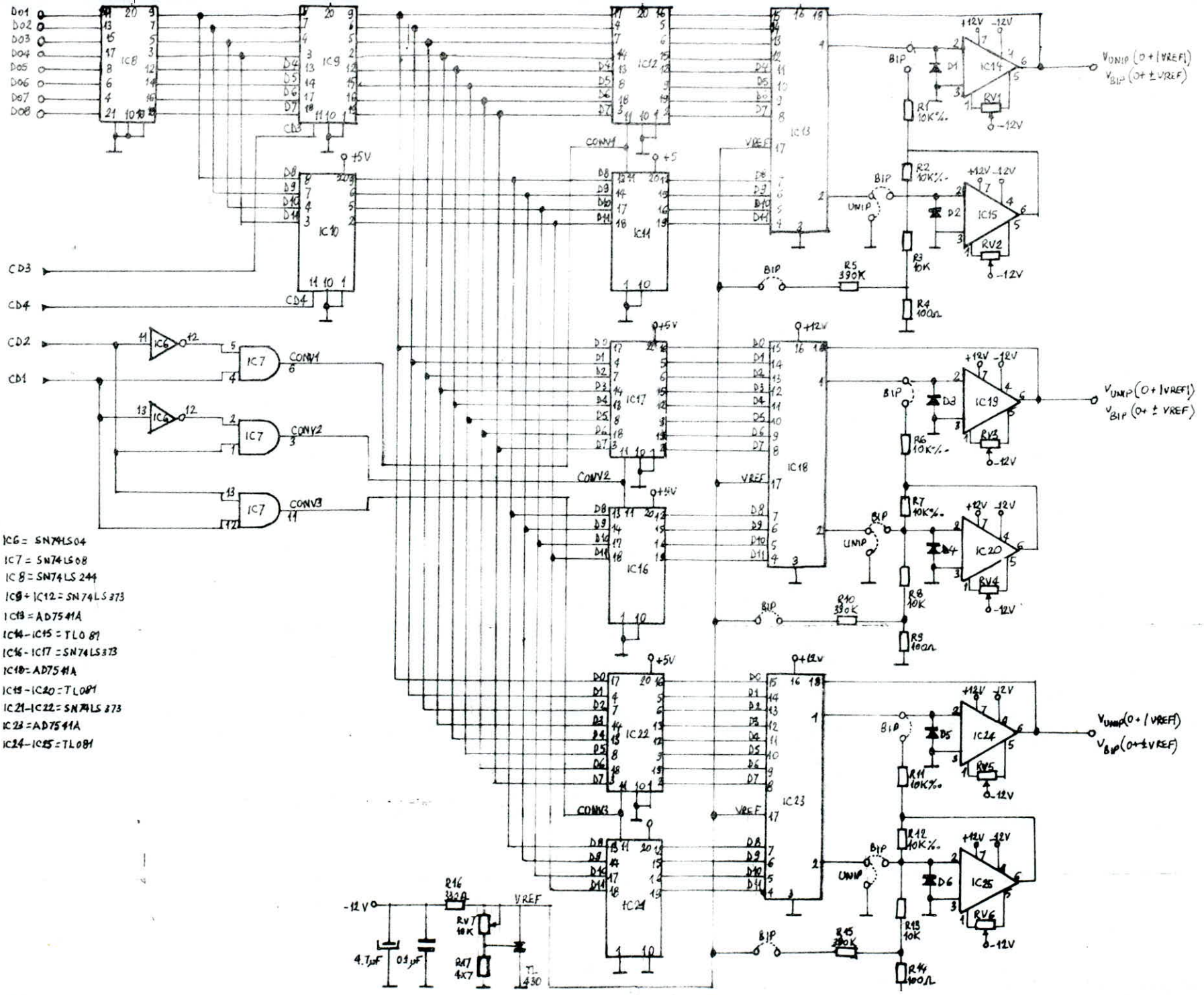


Fig. 22

TROIS VOIES DE CONVERSION D/A



CAR D/A SCHEMA ELECTRIQUE



IC6 = SN74LS04
IC7 = SN74LS08
IC8 = SN74LS 244
IC9 + IC12 = SN74LS 373
IC13 = AD7541A
IC14 - IC15 = TL0 81
IC16 - IC17 = SN74LS 373
IC18 = AD7541A
IC19 - IC20 = TL081
IC21 - IC22 = SN74LS 373
IC23 = AD7541A
IC24 - IC25 = TL081

CAR D/A SCHEMA ELECTRIQUE

2.2.2.3 SECTION A/D (CAR A/D):

Dans cette section,une seule des 8 sources analogique possible est etalonnee et arretee sur la valeur d'etalonnage (MUX et SAMPLE/HOLD) pour etre convertie par la suite en donnee digitale de 12 bits.
Le processus complet a lieu selon l'etat des signaux de controle CA4-CA3,qui gerent la selection de la voie anlogique et la temporisation du traitement jusqu'a la lecture du Bus de donnees.
On se refere a la fig(2.2.3) pour l'adressage,la selection des voies et la production des signaux (CA4-CA1),BAMAD.
Les 8 signaux analogique d'entree passe par un filtre passe-bas ayant une frequence de coupe d'environ 1KHz(-20 db/decade de Roll-Off) etun dispositif de limitation a +ou- 12Vcc pour la protection des circuits suivants.
Les signaux CA3-CA1 permettent de selectionner une des 8 voies correspondante,ceci representant le premier pas dans la conversion du signal.
Apres,ces 8 signaux arrivent au multiplixeur.
Apres ce multiplexeur,on a insere un Buffer attenuateur pour mieux adapter le gain aux caracteristiques du convertisseur.
L'amplificateur Sample Track/Hold maintient a un niveau constant l'amplitude(ceci en fonction de la commande $\overline{\text{SAMPLE/HOLD}}$ etablie par la logique de temporisation et de controle),et est assujetti aux temps operationnels du convertisseur A/D .

Apres avoir traverser un autre Buffer,le niveau analogique subit la conversion et devient la donnee digitale D11/D0 memorisee dans les Latch a 3 strates.Apres avoir ete dument habilites,ces derniers envoient cette donnee,byte par byte au bus des donnees de l'extension industrielle,ceci pour le transfert dans l'UC ou dans la memoire.

Toute ces operations doivent avoir lieu selon une juste temporisation apportee justement par les circuits de temporisation et de controle (fig.2.2.3 et 2.2.4).

Une fois que la carte et la voie ont ete selectionnees , le signal BAMAD sort avec le niveau logique 1 ; ce signal permet l'activation de toutes les operations suivantes :

La rampe positive de montee du signal BAMAD declenche un multivibrateur monostable dont l'impulsion (OUT 13,15.5 micro secondes environ) etablit un premier retard pour permettre au signal d'entree de se stabiliser dans le multiplexeur apres une commutation de voie.

Le meme signal mis en negation donne le depart a la phase de SAMPLE dans l'amplificateur S/H,et fait partir avec sa rampe de descente un deuxieme multivibrateur monostable (OUT 5,15.5 micro secondes environ) qui constitue un element de retard apres le passage de SAMPLE a HOLD.

A ce point le signal est stable a l'entree du convertisseur

Le processus de conversion commence avec l'impulsion negative de 5 micro secondes du signal R/$\overline{C}$ (READ/$\overline{CONVERT}$),lance par la rampe de descente de OUT 5.

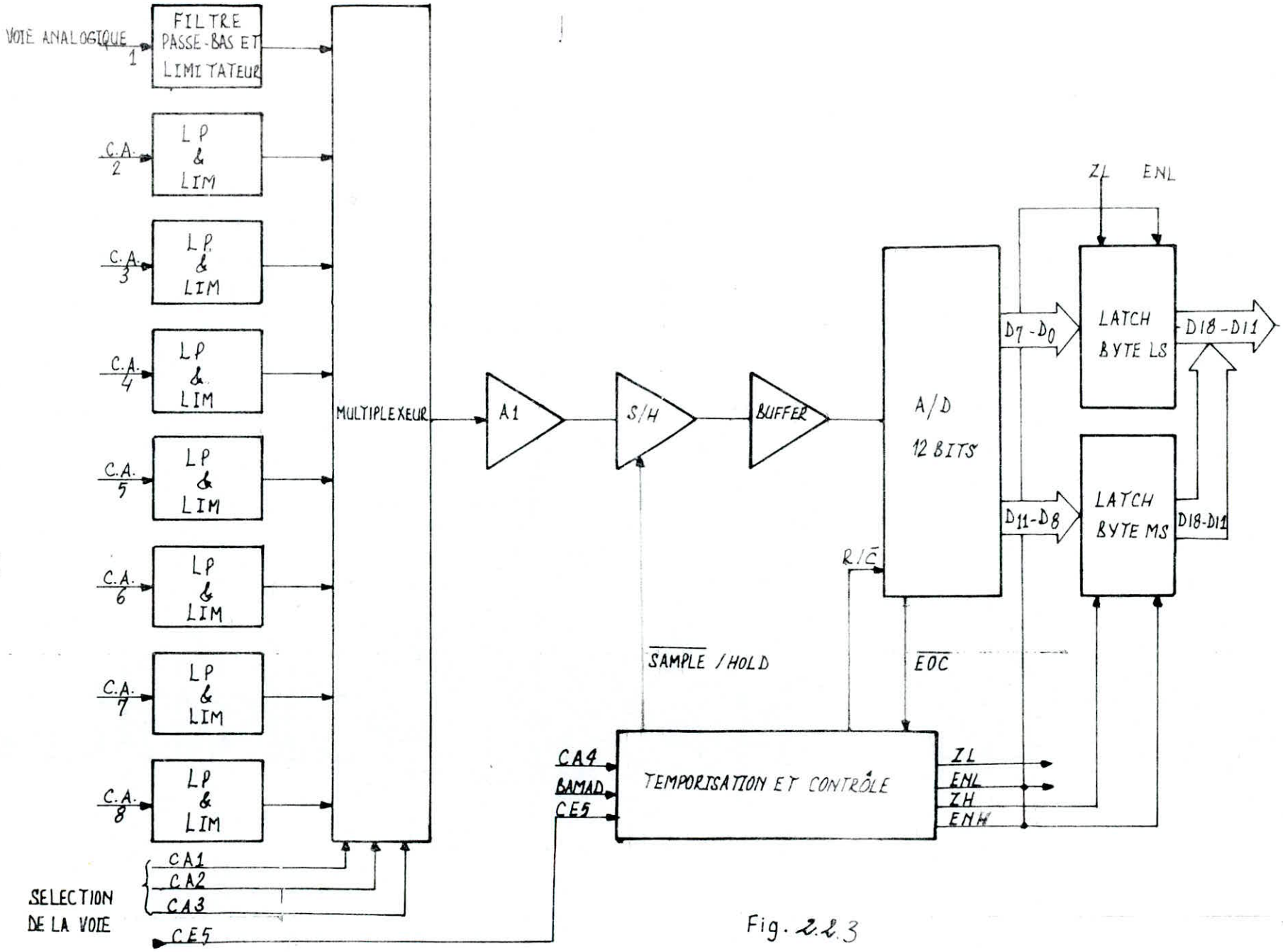

VOIE ANALOGIQUE 1
FILTRE PASSE-BAS ET LIMITATEUR
C.A. 2
LP & LIM
C.A. 3
LP & LIM
C.A. 4
LP & LIM
C.A. 5
LP & LIM
C.A. 6
LP & LIM
C.A. 7
LP & LIM
C.A. 8
LP & LIM
MULTIPLEXEUR
A1
S/H
BUFFER
A/D 12 BITS
ZL
ENL
D7 - D0
LATCH BYTE LS
D18 - D11
D11 - D8
LATCH BYTE MS
D18-D11
R/C̄
SAMPLE / HOLD
EOC
CA4
BAMAD
CE5
TEMPORISATION ET CONTRÔLE
ZL
ENL
ZH
ENH
CA1
CA2
CA3
SELECTION DE LA VOIE
CE5

Fig. 2.2.3

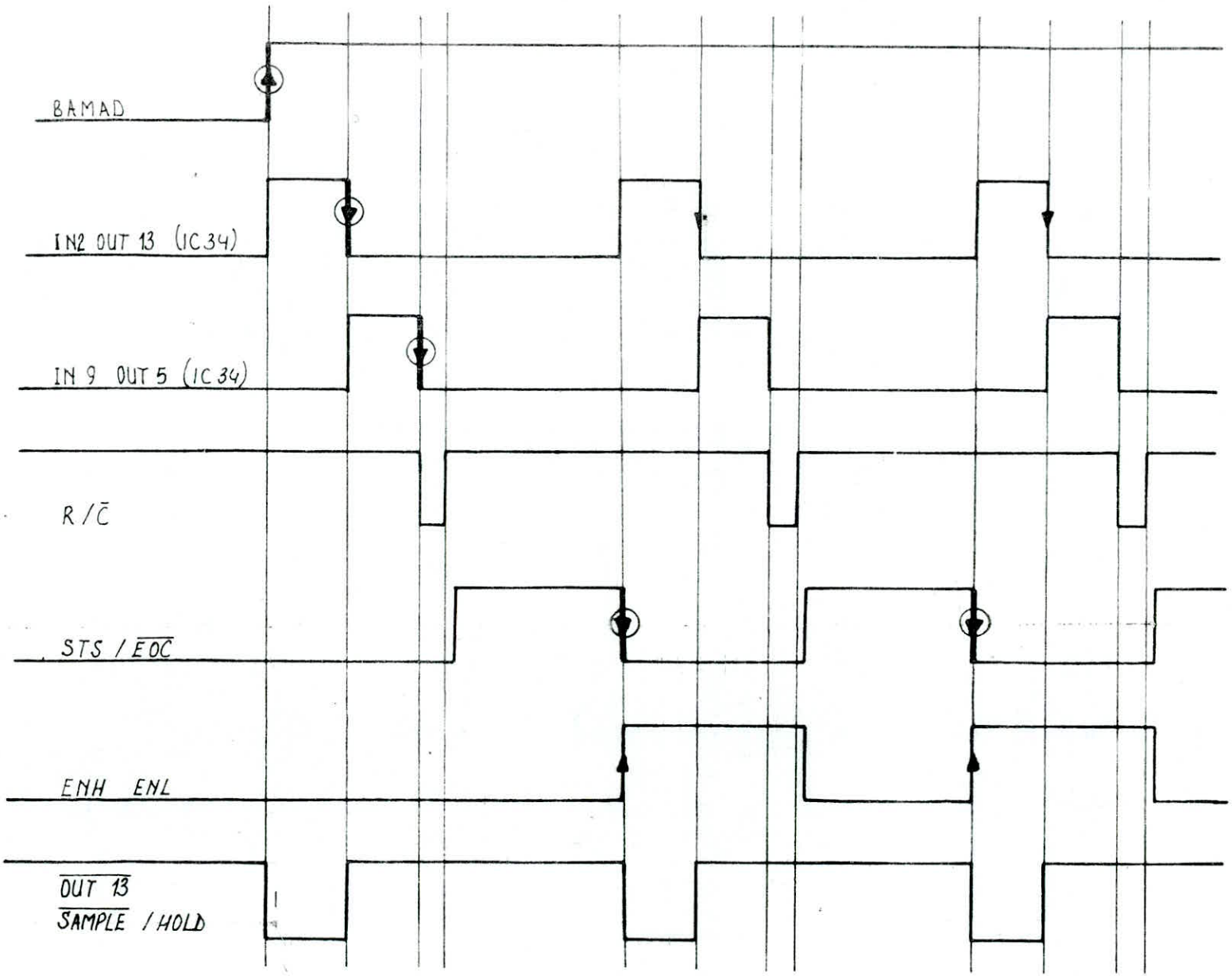


Fig 2.2.4
TIMING CAR - A/D

Quand l'impulsion de Debut de Conversion remonte,le convertisseur signale la phase de traitement des signaux avec STC/$\overline{EOC}$.

End Of Conversion($\overline{EOC}$,actif bas) retourne au 0 logique apres environ 35 micro secondes;on assiste alors a l'habilitation des Latchs de sortie qui memorisent en Bytes la donnee convertie(ENH=HABIL.MS BYTE; ENL=HABIL.LS BYTE) et le flag de fin de conversion'(bit D11),si CE5,qui est employe dans la phase de lecture,le permet.

La rampe negative de $\overline{EOC}$ lance aussi $\overline{SAMPLE}$/HOLD pour le blocage de l'etalon analogique instantanne.

Voir schema electrique de CAR-A/D

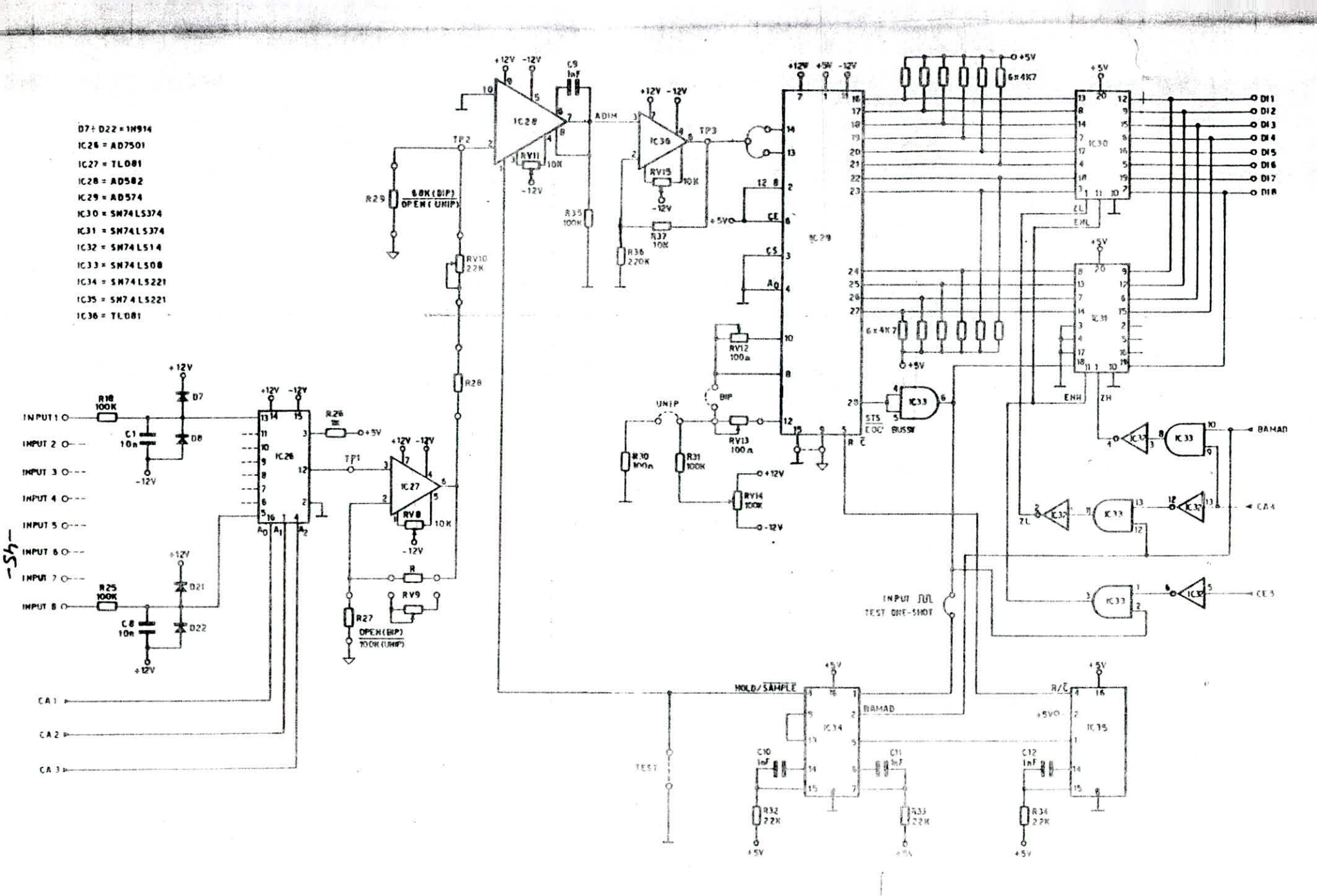
(M7)

2.3 L'ALIMENTATION STABILISEE :

c'est un appareil qui est en mesure de fournir des alimentations stabilisee,avec une tension de sortie fixe ou reglable et un generateur de tension alternative.

Ces alimentations sont necessaires au fonctionnement de tous les modules qui constituent le student trainer mod MPT-82/EV.

Ces tensions de sortie stabilisees sont protegees contre les courts circuits,au moyen de limiteurs de courant; par conseqent , ceci permet le mode operationnel a courant constant. (M8)



CAR-A/D : SCHEMA ELECTRIQUE

2.3.1 DESCRIPTION ET FONCTIONNEMENT:

-On branche l'appareil au reseau(220V,50a60 Hz) et on verifie la presence du fusible 2A.

-On commute l'interrupteur 1;le led 2 s'allume.

NB: Pour lire sur l'afficheur digital 9 la valeur de la tension ou du courant,on selectionne 8.

Se referer a la fig(2.3)

1- Alimentation stabilisee avec tension de sortie reglable (0-30V;0-1.5)

Cette tension est obtenue a l'aide du potentiometre de voltage 3 et est disponible entre les douilles +4 et -5.

Si on branche -5 a 6 , on aura une tension positive.

Si on branche +4 a 6 , on aura une tension negative.

Pour avoir une tension de sortie constante,associee a une valeur limite de courant Il determinee,on procede de la maniere suivante:

-On regle 3 a la valeur desiree (terminaux de sortie ouverts).

-On regle 7 a la valeur limite (terminaux de sortie fermes(c_c)).

Afin que l'alimentation agisse comme un generateur de tension,la resistance de charge Rl doit se referer a la relation suivante:

$$Rl > \frac{V_{sortie\ (constante)}}{I_{Limite}}$$

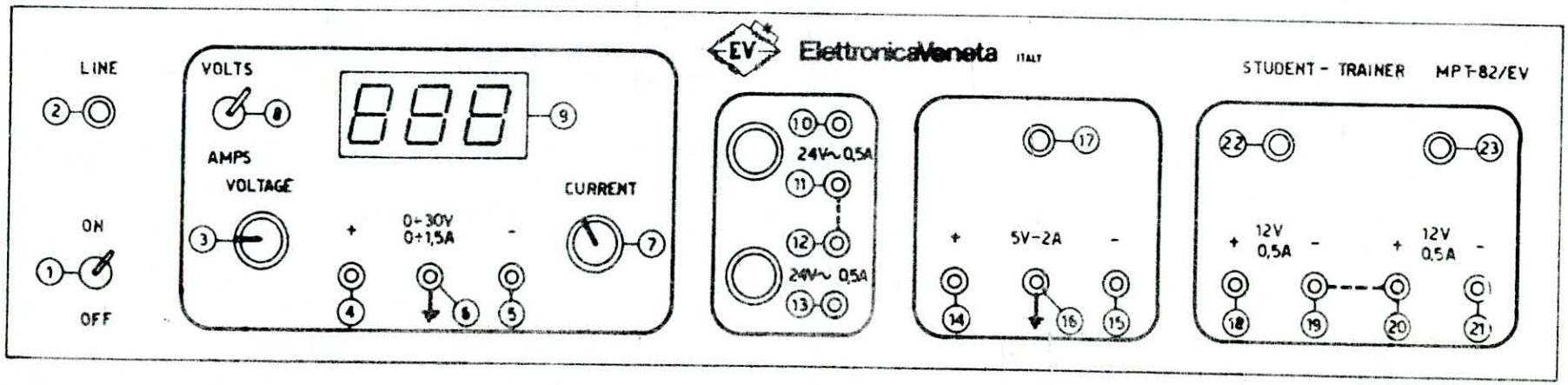


FIG. 2.3 : PANNEAU FRONTAL

2- Alimentation stabilisee avec courant de sortie constante

Pour obtenir un courant de sortie constant,associe a une a une valeur limite de tension determinee,on procede de la maniere suivante:

-On regle 7 a la valeur desiree(terminaux de sortie fermes (c_c))

-On regle 3 a la valeur de tension limite(terminaux de sortie ouverts)

Afin que l'alimentation agisse comme un generateur de courant la resistance de charge R1 doit se referer a la relation suivante:

$$R1 < \frac{V_{limite}}{I_{sortie(constante)}}$$

3-Generateur de tension stabilisee:

Entre les douilles 10 et 11 et les douilles 12 et 13 sont disponibles deux tensions sinusoidales (24 Veff) avec une charge maximum de 0.5 A .

En reliant 11 et 12 ,on obtient entre 10 et 13 une tension alternative sinusoidale de 48 Veff et 0.5 Amax.

4- Alimentation stabilisee 5V-2A:

Entre +14 et -15,on aura une tension stabilisee de +5V

.Si l'on branche -15 a 16,on aura +5V

.Si l'on branche +14 a 16,on aura -5V

4- Double alimentation stabilisee (2x12V;0.5A)

Entre les douilles +18 et -19 et les douilles +20 et -21, on aura une tension stabilisee de 12V.

En reliant -19 et +20,on lit entre +18 et -21,on obtient une tension stabilisee de + ou - 12V

(M8)

2.4 L'UNITE D'EXTENSION INDUSTRIELLE VEZ-80/EV

2.4.1 INTRODUCTION:

On a vu au (CHAP.2) que le bus principal MMS-8 du micro-ordinateur MPZ-80/EV met a disposition plusieurs signaux de controle lui permettant d'appliquer plusieurs techniques d'interfaces(polling,interruptions,DMA,...).Quelques-unes de ces pperations d'E/S avec les unites peripheriques sont necessaires a l'obtention de hautes vitesses de transfert des donnees et donc a une plus grande puissance de traitement de l'ordinateur.

A cause de la complexite du bus de controle impliquant aussi des relations temporelles tres soignees entre les differents signaux,on a simplifie l'architecture du systeme d'interface:

.La vitesse d'execution a ete reduite,mais le rapport signal/ bruit est meilleur.

.Sur le plan didactique,le systeme de connexion(materiel et logiqiel) est plus aise,aussi le cout est mineur.

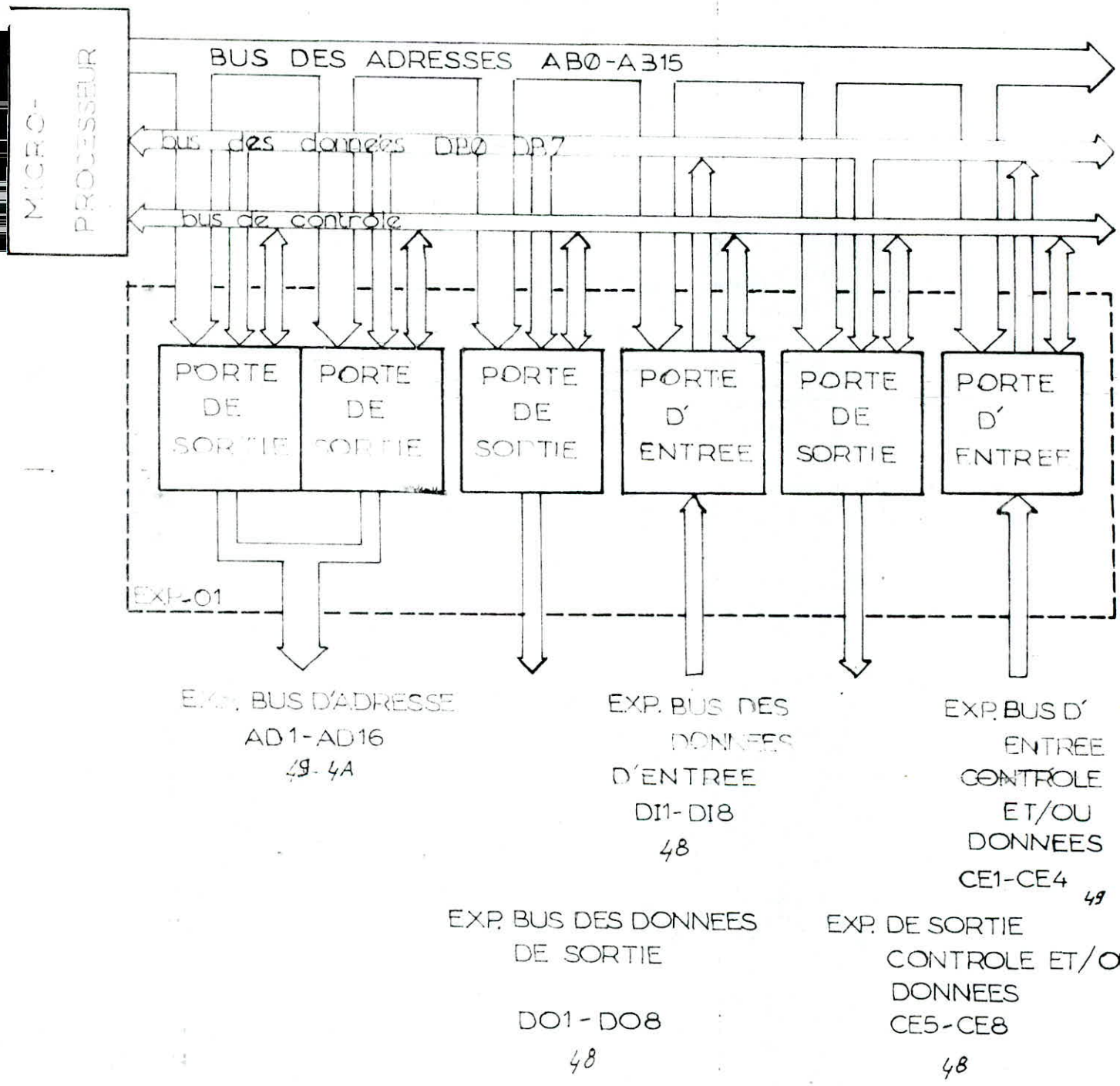


FIG. 2.4 : Schéma simplifié du processus de création du BUS D'EXTENSION à l'aide de portes d'E/S

2.4.2 DESCRCRIPTION DU SYSTEME D'INTERFACE INDUSTRIELLE E/S

Le bus d'extension est un bus cree a partir du bus principal MMS-8 et possedant des portes d'E/S aux quelles on connecte les cartes du type industriel (entree 24Vcc, Convertisseurs A/D, D/A, ...).

-Avec 2 portes de sortie, on produit des signaux correspondant aux BYTES MS (AD9-AD16) et aux BYTES LS (AD1-AD8) du bus des adresses d'extension.

-Deux autres portes sont reservees au BUS des donnes d'entree (DI1-DI8) et de sortie (DO1-DO8).

-Deux portes auxiliaires de 4 bits chacune fournissent 8 lignes supplementaires pour le controle et/ou les donnees (CE1-CE4, CE5, CE8). fig(2.4)

Ce systme d'interfaces est constitue par: voir fig(2.4.1)

Une carte EXP-01, une carte IMS, un cordon plat reliant EXP-01 a l'IMS, un chassis et une carte matrice pour le logement des cartes d'interface et une carte PW-01 avec lampes temoins pour les alimentations.

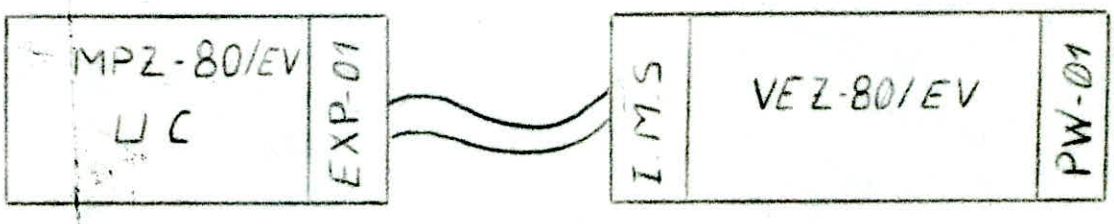


FIG 2.4.1

2.4.3 DESCRIPTION DE LA CARTE EXP-01:

Cette carte inseree dans l'UC communique avec le microproces grace au bus MMS-8.
Le support de propagation des signaux du bus d'extension est constitue par un cordon plat approprie et un connecteur entre le microprocesseur et la carte EXP-01

2.4.4 DESCRIPTION DE LA CARTE IMS:

Elle est constituee par les bus drivers et les buffers relatifs aux leds qui produisent sur le panneau frontal l'etat des lignes d'adresse et de donnees.Elle produit des signaux sur la carte matrice de l'extension industrielle.

2.4.5 ALIMENTATION:
Les cartes d'interface industrielle sont alimentees par le systeme principal du micro-ordinateur MPZ-80/EV;cependant l'extension contient le module PW-01,avec les fusibles pour les divers tensions,l'interrupteur general et la lampe temoin correspodante. (M9)

3 DESCPRIPTION DU MINI-ROBOT RB-4/EV

3.1 STRUCTURE MECANIQUE:

Le RB-4/EV est un robot typique a ceux utilises en industrie mais a echelle reduite,projete pour des fins didactiques.
En effet,il est classe parmi les robots a coordonnees de revolution qui sont d'alleurs les plus utilises dans l'industrie.Son fonctionnement en parallelogramme a une importance viltale pour simplifier le deplacement d'objets.
Il est caracterise par cinq parties principales: (fig 3)

1-La base:C'est un support du reste du bras contenant les circuits d'interface et le moteur assurant la rotation du bras.

2-L'epaule:Elle comporte 5 moteurs et leurs reducteurs pour leurs accouplements avec les engrenages de l'avant-bras.L'engrenage principal permet a l'epaule de tourner.

3-Le bras:Il tourne autour d'un axe horizontal et porte dans sa partie inferieure les engrenages et les cables permettant le mouvement du coude,de la poignet et de la main.

4-L'avant-bras:Il tourne autour d'un axe horizontal et

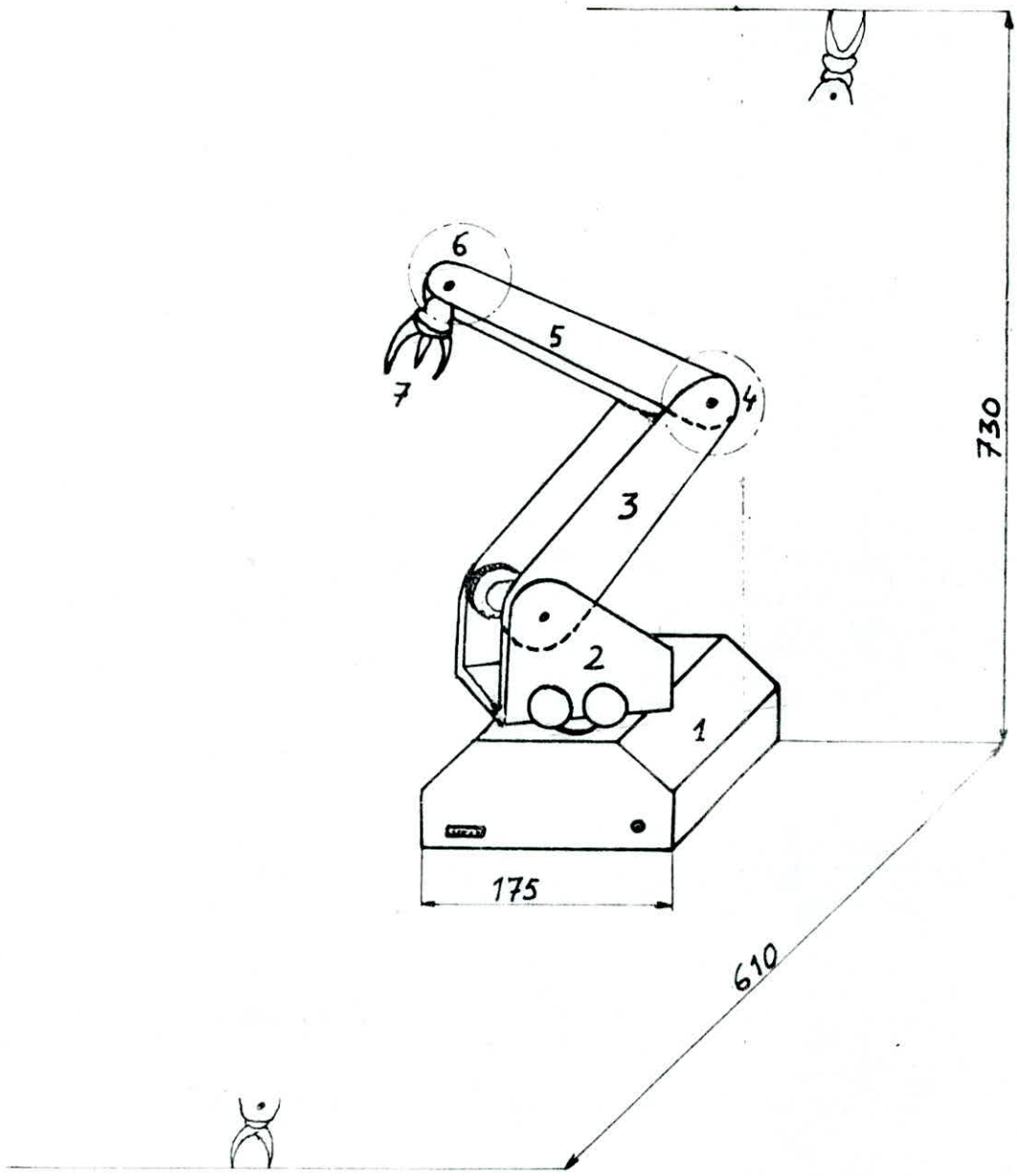


FIG. 3 : SCHEMA DU ROBOT

1. Base
2. Epaule
3. Bras
4. Coude
5. Avant bras
6. Poignet
7. Main (Pince)

- Portée en hauteur 730 mm
- " horizontale 610 mm
- Largeur de la base 175 mm

contient les engrenages coniques du poignet.

5-La poignet et la main:Les deux mouvements de la poignet , la rotation autour de l'axe de la main(tangage) et la rotation de de la main autour de l'axe horizontal(vers le haut et vers le bas) dependent de la combinaison de deux mouvements independants.La main est dotee de trois doigts ayant les extremites en caoutchouc,peut simplement s'ouvrir et se fermer. (R2)

3.2 RELATION ENTRE LE PAS DU MOTEUR ET L'ACCROISSEMENT ANGULAIRE

Le mini-robot RB-4/EV est equipe de six moteurs electriques pas a pas alimentes par un courant de 12Vc,ainsi que de reducteurs pour la commande des differentes parties.
La relation generale est:

Angle du pas x rapport 1 x rapport 2 = Degre par pas

= 1/pas par degre

Pour la main,la relation est :

Degre par pas x d/360 = Course par pas du moteur

!	! BASE	! EPAULE	! COUDE	!POIGNET	! MAIN	!
!angle d'un pas	!7.5	!7.5	!7.5	!7.5	!7.5	!
!rapport 1	!20/72	!14/72	!14/72	!20/72	!20/72	!
!rapport 2	!12/108	!12/108	!12/108	!12/108	!12/108	!
!degre par pas	!.2314	!0.162	!0.162	!0.2314	!0.2314	!
!pas par degre	!4.32152	!6.17284	!6.17284	!4.32152	!4.32152	!
!couse par pas	!	!	!	!	!0.01668	!

Remarque:

La course totale de la main de la position ouverte a la position fermee des doigts est de 20 mm et l'angle accompli par chaque doigt est de 50 degre.

$$\frac{50}{20mm} \times 0.0062 \text{ mm} = 0.0655 \text{ degres par pas} = 15.2672 \text{ pas par deg}$$

Le diametre de la poulie est: d=26 mm (R2)

3.3 COMMANDE ELECTRONIQUE:

Le mini-robot RB-4/EV se branche au mini-ordinateur MPZ-80/EV au moyen de 2 portes paralleles (A et B) de 8 bits chacune.
La porte de sortie A du MPZ-80/EV est le chemin par lequel on envoit les donnees pour mouvoir le robot,tandis que la porte d'entree B est celui par lequel on recoit les informations sur l'etat des capteurs de position.Cette derniere etant facultative et sert uniquement dans le cas ou

le robot doit etre dote d'une position Zero fixe.Voir fig ci-dessous

LA PORTE A		LA PROTE B	
A8	4eme phase du moteur	B8	Photo-coupleur main
A7	3eme ----- -- ------	B7	---------- poignet 2
A6	2eme ----- -- ------	B6	---------- ------- 1
A5	1ere ----- -- ------	B5	---------- coude
A4	Adresse du moteur	B4	---------- epaule
A3	----------	B3	Micro-interrupteur base
A2	----------	B2	Libre
A1	Validite des latchs de donnees	B1	libre

La partie electronique se trouvant dans le robot comporte 3 cartes distinctes:

-Carte d'interface avec le connecteur d'E/S pou le branchement au mini-ordinateur MPZ-80/EV.
-Carte base avec toute la logique de commande des moteurs et la partie de puissance.
-Carte de support des photo-coupleurs(facultative) pour leur utilisation comme capteurs de zero. voir fig(3.1)et(schema electrique en annexe). (R2)

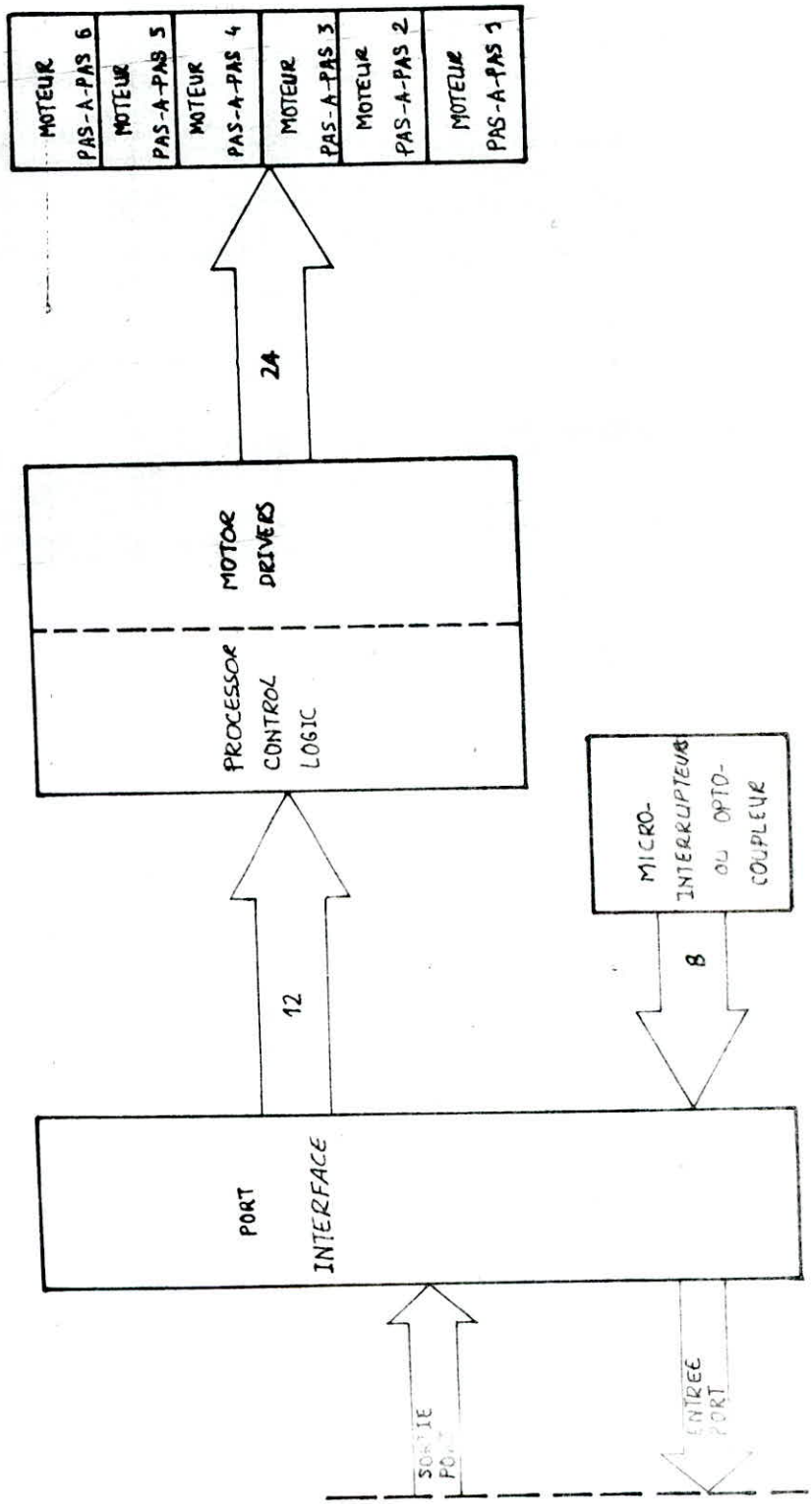


FIG: 3.1

SCHEMA A BLOCS DE L'INTERFACE

3.3.1 ANALYSE DETAILLEE:

a-Carte d'interface:

Cette carte sert de connecteur d'E/S pour le branchement a la carte PIO(portes A et B) du MPZ-80/EV.

La porte A est preparee comme sortie au moyen branchement opportun des signaux de strobe:

ASTB=HIGH(1) BSTB=LOW(0)

Alors que la porte B est preparee pour l'entree,en posant:

CP1=LOW(0) CP2=LOW(0)

Tous les signaux de la porte A vont directement a la carte base.

Les signaux qui proviennent des capteurs sont elabores au contraire par un circuit integre CMOS 4069(IC24) pour fournir les seuils d'entree appropries et envoyes ensuite a la carte PIO par l'intermediaire d'un buffer du type 74367 (IC23).

b-Carte base:

Cette carte contient la section de commande des six moteurs pas a pas du robot.

Les bits de commande(A8,A7,...,A1)sont bufferes par les circuits integres IC1 et IC2.

Le buffet du bit A1 est toujours valide et sa sortie est est envoyee a un monostable (IC3) pour produire une impulsion d'horloge.De cette maniere,le decodeur(IC4) produit une impulsion de memorisation qui dure environ

500 micro-secondes au latch de controle du moteur adresse IC5 a IC10.

A1 est branché a une resistance de Pull-up(R1),afin que cette ligne soit toujours haute(ASTB=HIGH(1)),a moins qu'elle ne soit commandée par le micro-ordinateur.Les bufferts IC1 et IC2 sont valides par la sortie Bufferee du bit 1,afin que les donnees soient envoyees a l'entree seulement quand le bit 1 est bas (BSTB=LOW(0)).

Les donnees relatives a la commmande des phases des moteurs bits(A5,A6,A7,A8) vont en paralleles a tous les latchs.

Seul celui qui est adresse par les bits(A2,A3,A4) les transferera ensuite a la sortie,pour commander effectivement les 4 enroulements.

Les signaux a la sortie des 6 latchs sont envoyes a autant de circuits integres translateurs de niveau(IC11,IC12,IC13 IC14,IC15,IC16);ICI,les entrees a 5V sont transformees en sortie a 12V capable de piloter les circuits de puissance (IC17,IC18,IC19,IC20,IC21,IC22).Ce sont des VFEET quadriples a courant eleve qui produise les sorties pour les enroulements du moteur.

Le courant de pilotage pour les enroulements est egal a environ 300 mA et 12V.

c-Carte photo-coupleurs:

Cette carte(facultative) supporte les les photo-coupleurs qui ont la fonction de capteurs de

de position zero pour les mouvements de l'epaule,de l'avant-bras,du poignet et de la main.

On utilise un micro-interrupteur branche directement a la carte d'interface,pour detecter la position zero de la base (R2)

3.3.2 COMMANDE DES PHASES DES MOTEURS:

Le pilotage des moteurs se fait directement par les signaux(A8,A7,A6,A5) se trouvant sur l'interface parallele;ceci permet le mouvement a demi-pas ou a pas complet.

Le tableau ci-dessus montre les seqences qu'il faut envoyer pour les deux types de mouvements.

Enroulements	A	B	C	D		
Bits	A8	A7	A6	A5	Pas	Donnee
1	1	0	1	0	1	192
2	1	0	0	0	1.5	128
3	1	0	0	1	2	144
4	0	0	0	1	2.5	16
5	0	1	0	1	3	48
6	0	1	0	0	3.5	32
7	0	1	1	0	4	96
8	0	0	1	0	4.5	64

NB:
Les num pairs de 2 a 8 sont pour le mouvement a demi-pas.
Les num 1,3,5,7 sont pour le mouvement a pas complet.

3.3.3 SELECTION DES MOTEURS:

Le choix du ou des moteurs a piloter se fait par la combinaison des adresses A2,A3,A4. La correspondance entre les moteurs et les adresses est:

A4	A3	A2	Donnee	MOTEUR	No du moteur
0	0	1	2	Avant-bras	3
0	1	0	4	Poignet 2	4
0	1	1	6	Base	1
1	0	0	8	Main	6
1	0	1	10	Epaule	2
1	1	0	12	Poignet 1	5

4 COMMANDE PROGRAMMEE DU MINI-ROBOT RB-4/EV:

Le programme complet de gestion du robot avec le mini-ordinateur permet de mouvoir le robot au moyen du clavier, d'apprendre une sequence de mouvement , de l'enregistrer sur disque,de la modifier et de la faire executer de maniere repetitive.

Ce programme est compose de queleques routines de base de mouvement ecrites en langage Assembler Z80 (a cause de la vitesse d'execution) et d'une section de gestion de ces routines ecrites en Basic.

Ces routines sont ecrites en langage machine en utilisant l'assembler Macro-80,le basic en Basic-80.

Le programme complet est obtenu en remplissant le basic avec le basic compiler et en unissant les differents paquets avec le Link-80. (R2)

4.1 DESCRIPTION DES TROIS PAQUETS COMPOSANT LE PROGRAMME DE GESTION:

4.1.1 DESCRIPTION DE AROBOT:

L'archive AROBOT.MAC contient la routine de mouvement du robot au moyen du clavier.

La correspondance entre les touches ,leur codes ASCII et les axes a mouvoir est la suivante:

TOUCHE	CODE ASCII	FONCTION
Q	81	Rotation base en arriere
CTRL/Q	17	Rotation base en avant
W	87	Rotation epaule vers le haut
CTRL/W	23	Rotation epaule vers le bas
E	69	Rotation bras vers le haut
CTRL/E	5	Rotation bras vers le bas
R	82	Rotation pince vers le haut
CTRL/R	18	Rotation pince vers le bas
T	84	Rotation pince sens anti-horaire
CTRL/T	20	Rotation pince sens horaire
Y	89	Fermeture pince
CTRL/Y	25	Ouverture pince
+	43	Sortie de la routine
-	45	Sortie de la routine

(R2)

4.1.2 DESCRIPTION ROBCOM:

L 'archive ROBCOM.MAC contient les routines base du mouvement du robot;celle-ci sont completement generales et peuvent etre utilisees dans d'autres programmes. Ce sont:

-MOTUP et MOTDW : routines servant a deplacer en avant et en arriere d'un pas le moteur determine par le contenu de la position MOTPIL.

-CONUP et CONDW : routines servant a augmenter et diminuer les compteurs partiel et absolu(tableau TBUF et POSAR) du moteur determine par le contenu de la position MOTPIL.

-COMMAND : routine servant a commander le mouvement simultane de tous les moteurs,avec le nombre de pas specifie par le contenu du tableau TBUF

-RESTOR : reporte tous les axes du robot a zero; prend les donnees du tableau POSAR qui contient le nombre de pas effectues et commande les moteurs jusqu'a la mise a zero de ce tableau. (R2)

4.1.3 DESCRIPTION MROBOT:

Le programme MROBOT.BAS s'occupe de la gestion complete du robot,au moyen de l'appel des routines decrite precedement.Pour que ce programme fonctionne,il doit trouver en memoire les routines assembler.

4.2 LINK DES PAQUETS DU PROGRAMME:

On a vu que le programme complet est constitue d'une serie de paquets ecrits en langage differents et qui doivent etre assembles.
Leur assemblage depend de la maniere dont on veut travailler dans la redaction de la partie en langage basic du programme.

4.2.1 INTERPRETE BASIC:

Procedure:

.On assemble les deux archives source AROBOT.MAC et ROBCOM.MAC avec les instuctions:

A>M80 AROBOT,AROBOT=AROBOT/Z

A>M80 ROBCOM,ROBCOM=ROBCOM/Z

.On charge en memoire les routines en langage machine a une adresse connue (cad:on assemble les deux paquets en langage machine dans une archive unique appelee M1ROBOT.COM).Ceci est faisable avec le programme le link.

A>L80 /P:A000,ROBCOM,AROBOT,M1ROBOT/E/N

A ce point,chaque fois que l'on veut charger en memoire les routines assembler,il suffit de composer:

A>M1ROBOT

.On charge ensuite le basic avec limite de la memoire utilisable.

A>BASIC /M:&H9FFF

L'execution de l'instuction se fait avec:

RUN''MROBOT.BAS (R2)

4.2.2 COMPILATEUR BASIC:

Procedure:

Cette procedure est moins complique que celle de l'interprete basic.La difference est la maniere d'appel des routines assembler.En effet,avec cette procedure elles sont appelees avec leur nom au lieu de l'adresse absolue et de plus,elles

peuvent etre positionnees en code programme pour limiter l'espace occupe par le programme complet.

.On doit donc enlever des deux programmes source les informations de positionnement absolu ASEG et ORG et on les assemble de la meme maniere que celle decrite precedement.

.On utilise le compilateur basic qui transforme le programme source MROBOT.BAS en une archive MROBOT.REL contenant le meme programme transforme en code machine Z80.

On realise cela avec la commande:

A>BASCOM MROBOT=MROBOT/Z

.A ce point,nomarchive.BAS et nomarchive.MAC sont transformees en nomarchive.REL,qui peuvent etre assemblees avec le programme de link pour former un programme unique appele MROBOT.COM

A>L80 MROBOT,ROBCOM,AROBOT,MROBOT/E/N

.L'execution se fait avec:

A>MROBOT (R2)

4.3 DESCRIPTION ET UTILISATION DU PROGRAMME:

Apres avoir choisi l'une des deux methodes citees ci-dessus, et apres avoir execute celle-ci,le menu principal de selection des differentes fonctions et utilisations apparait:

MENU PRINCIPAL

1.SIMULATION LIBRE DES MOUVEMENTS.

2.ETUDE D'UNE SEQUENCE DE MOUVEMENTS.

3.MOUVEMENT SUR SEQUENCE.

4.EDIT D'UNE SEQUENCE.

5.MISE A ZERO DES MOUVEMENTS EFFECTUES.

6.MISE A ZERO MANUEL.

7.FIN

Analyse de ces differentes options.

1.SIMULATION LIBRE DES MOUVEMENTS:

Cette section permet de mouvoir le robot avec le clavier; (voir ch.4.1.1).

2.ETUDE D'UNE SEQUENCE DE MOUVEMENTS:

On selectionne cette section chaque fois que l'on veut enregistrer et executer une sequence deja enregistree ou une sequence qu'on vient d'enregistrer.

Une fois qu'on a selectionne cette sequence,le programme demande le nom de la sequence (max de 6 caracteres).

NOM SEQUENCE (CTRL/A = SORTIE) : ESSAI

Le menu de correspondance entre les touches et les moteurs apparait et le programme se prepare pour accepter les pas de la sequence et les enregistrer sur disque.Apres chaque pas on presse (-) pour passer au pas suivant et (+) pour fermer la sequence de mouvement.

Pendant chacun des pas,on peut mouvoir plusieurs axes en succession (pendant le mouvement sur sequence , le programme execute le mouvement simultane des axes.

3.MOUVEMENT SUR SEQUENCE:

Cette section permet de mouvoir le robot grace a la commande

de la séquence deja enregistre dans l'option 2 du menu.

Le menu du mode de mouvement apparait:

mouvements < continu (C)

mouvements < pas a pas (P)

mouvements < simple (S)

mouvements < cyclique (C)

On choisi (C) ou (P) et (S) ou (C)

Pour arreter le mouvement,on appuye sur U

4.EDIT D'UNE SEQUENCE :

Cette section permet le controle et/ou la modification des donnees d'une sequence deja enregistree.Cela est tres efficace si l'on veut apporter des modifications a une sequence,sans refaire l'etude de celle-ci.

Par exemple,dans une sequence enregistree precedement,on a

PAS	MOT.1	MOT.2	MOT.3	MOT.4	MOT.5	MOT.6	
3	100	0	20	-150	150	0	OK(S/M/F/Q)

Ou S/M/F/Q sont les touches qu'il faut appuyer pour proceder ou non a une modification.

En pressant sur la touche:

S:on passe au pas de sequence suivant sans proceder a une amodification.

M:on modifie les pas visualises pour le pas examine.

F:on interrompe la sequence en eliminant les pas suivants.

Q:on interrompe le processus d'edit et on revient au menu principal.

5.MISE A ZERO DES MOUVEMENTS EFFECTUES:

Cette section permet de mettre a zero tous les mouvements executes jusqu'a present.

6.MISE A ZERO MANUEL:

Cette section permet de porter manuellement le robot dans n'importe quelle position.Celle-ci sera la nouvelle position zero de tous les mouvements successifs.

(R2)

5 APPLICATION A LA MISE EN OEUVRE DU MINI-ROBOT

5.1 Chargement de la perceuse transversal a commande numerique

Les caracteristiques de la piece sont:

L=50 mm , l=25 mm , h=20 mm

On desire faire un percage de 10 mm de diametre et 15 mm de profondeur.

On concoit un etau a commande automatique ayant deux boutons,l'un pour le serrage de l'etau et l'autre pour le desserage.Le premier est place au fond de l'etau ,le second servant de butee se trouve a la position initiale de la perceuse.

*Description des pas de sequence effectue par le mini-robot lors du chargement et dechargement.

1-Operation de deplacement (de la position zero vers la piece) et ouverture de la pince.

2-Operation de saisie de la piece.(serrage de la piece)

3,4-Operation de deplacement vers l'etau et positionnement de la piece (ouverture des pinces).

5,6-Operation de retour a une position d'attente.

7-Operation de deplacemnement vers la piece usinee et reprise de celle-ci.

8,9-Operation de retour et depot de la piece usinee.Fin

10-Operation de retour a la position zero.

Calcul approximative de la force de serrage:

Masse de la piece a soulever: $m = r \times L \times l \times h$

r=7.8 kg/dm3 (densite de l'acier)

m=7800 x 0.050 x 0.025 x 0.020 =0.20 kg

D'ou l'effort de serrage doit etre inferieur ou egal a 2 N

5.2 Sequence de mouvement et pas de moteurs

Pas de sequence	Moteur 1 Base	Moteur 2 Epaule	Moteur 3 Avant-bras	Moteur 4 Poignet 2	Moteur 5 Poignet 1	Moteur 6 Pinces
1	X1	Y1	Z1	U1	W1	N1
2	0	Y2	0	0	0	N2
3	X3	Y3	Z3	U3	W3	0
4	0	Y4	0	0	0	N4=-N2
5	0	Y5=-Y4	0	0	0	0
6	X6	0	0	0	0	0
7	X7=-X6	Y7=-Y5	0	0	0	N7=-N4
8	0	Y8=-Y7	0	0	0	0
9	X9	Y9	Z9	U9	W9	N9=-N7

Les vitesses de rotation des moteurs sont de telle sorte qu'on ait le meme temps pour la rotation de tous les moteurs durant le pas de sequence.

L'estimation des temps d'execution des pas de sequence est le suivant:

T1=2s ,T2=1s ,T3=2s ,T4=1s ,T5=1s ,T6=1s ,T7=1s ,T8=1s

T9=2s ,T10=3s

Le temps total d'execution des pas de sequence est T=15s

L'operation de percage (y compris le serrage et desserage de

l'étau) dure 10s

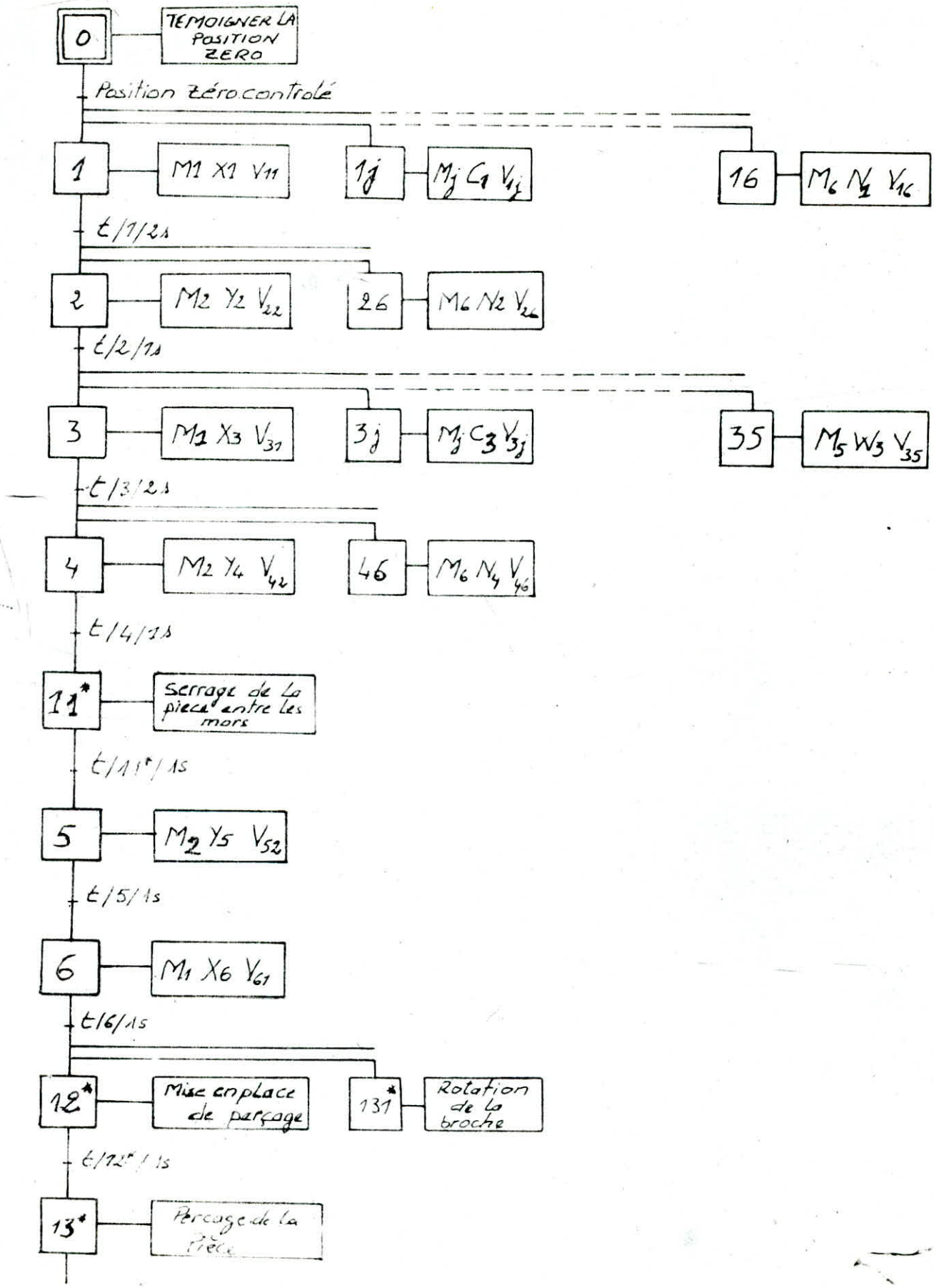
L'operation de percage avec manipulation du robot dure 25s

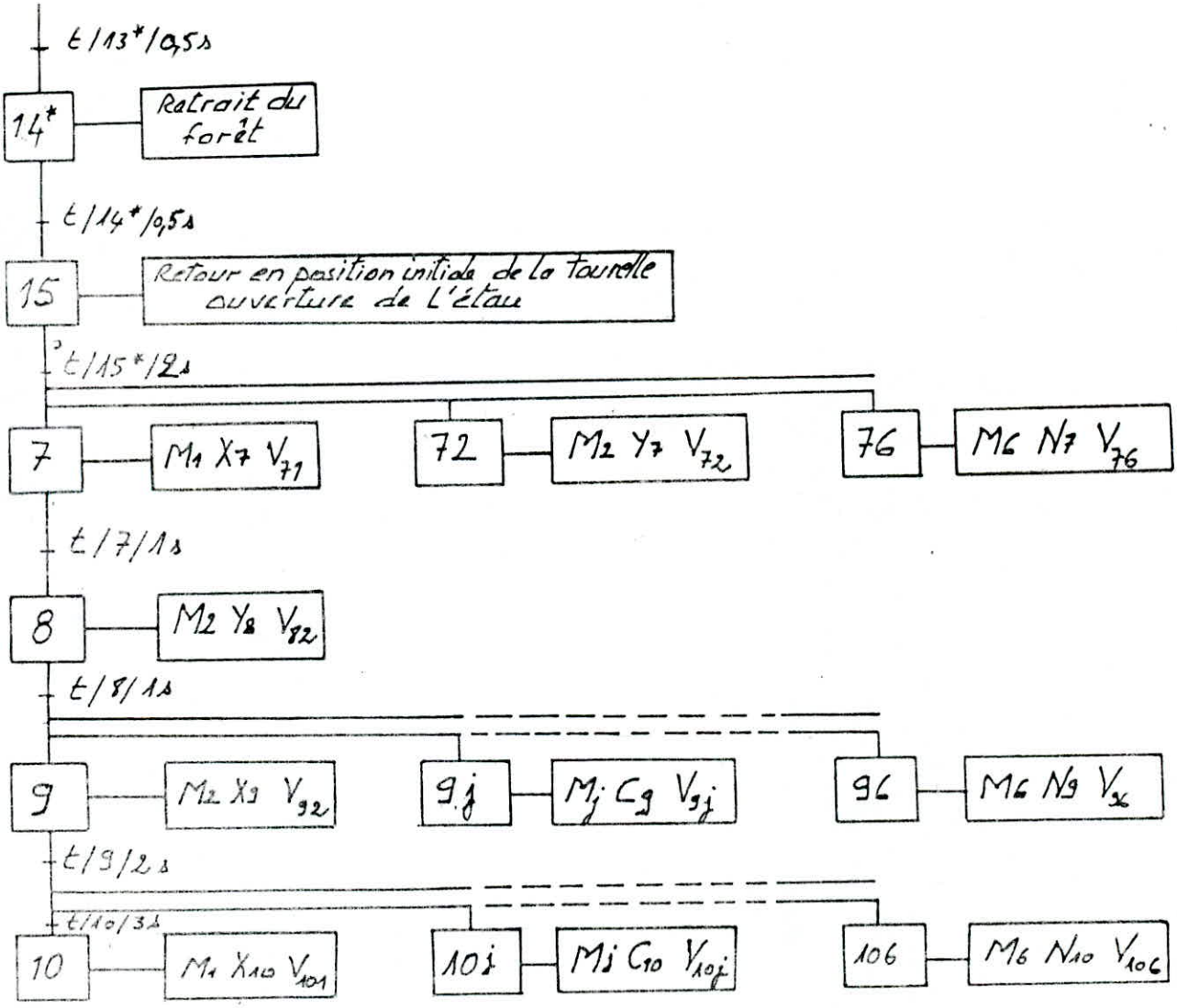
5.3 Trace du GRAFCET.

NB :Mi,Xi,Yi,Zi,Ui,Wi,Ni designent les pas de moteurs associes aux pas de sequences.

Ti designe le temps de chaque operation.

Vij designe la vitesse de rotation du moteur j dans dans l'etape i.


0
TEMOIGNER LA POSITION ZERO
Position zéro contrôlé
1
M1 X1 V11
1j
Mj Cj Vj1
16
M6 N1 V16
t/1/2s
2
M2 Y2 V22
26
M6 N2 V26
t/2/1s
3
M1 X3 V31
3j
Mj C3 V3j
35
M5 W3 V35
t/3/2s
4
M2 Y4 V42
46
M6 N4 V46
t/4/1s
11*
Serrage de la pièce entre les mors
t/11*/1s
5
M2 Y5 V52
t/5/1s
6
M1 X6 V61
t/6/1s
12*
Mise en place de perçage
131*
Rotation de la broche
t/12*/1s
13*
Perçage de la pièce


t/13*/0,5s
14*
Retrait du foret
t/14*/0,5s
15
Retour en position initiale de la tourelle ouverture de l'étau
t/15*/2s
7
M1 X7 V71
72
M2 Y7 V72
76
M6 N7 V76
t/7/1s
8
M2 Y8 V82
t/8/1s
9
M2 X9 V92
9j
Mj Cg Vgj
9C
M6 N9 V9C
t/9/2s
t/10/3s
10
M1 X10 V101
10j
Mj C10 V10j
10C
M6 N10 V10C

********** C O N C L U S I O N **********

Cette etude nous a permis de connaitre le fonctionnement du mini-robot ainsi que sa commande au moyen du micro-ordinateur dont l'unite centrale est un microprocesseur Z 80. De meme on a vu le fonctionnement des differents peripheriques, telque l'imprimante ,le terminal video ,l'unite d'extension industrielle,les cartes CAR- A/D ,D/A .

On a aussi aborde les differentes manieres de mouvoir le mini-robot.

Cependant,il est a signaler que ce travail reste incomplet du fait de la defaillance de la partie hard de la carte PIO se situant dans le micro-ordinateur ou bien de celle se trouvant dans le mini-robot.

Nous suggerons que ce travail soit poursuivi , afin de permettre auk etudiants de s'integrer au domaine de la robotique.

********** B I B L I O G R A P H I E **********

M1 :Manuel materiel du systeme base
M2 :Notice de fonctionement
M3 :Manuel du systeme operationnel
M4 :Notice du programme moniteur MON-80
M5 :Manuel de l'imprimante
M6 :Manuel du terminal video
M7 :Notice de fonctionnement CAR-A/D ,D/A
M8 :Unite d'alimentation
M9 :Unite d'extension
R2 :Mini-computer MPZ-80/EV et mini-robot RB-4/EV
1 :l'emploi des microprocesseurs. M.AUMIAUX 4eme edition
2 :Les microprocesseurs 1. ROGER L.TOKHEIM
2' :--- ---------- 2. ----- - -------
3 :Initiation a la microinformatique.Le Microprocesseur
P.MELUSSON
4 :Assembleur et peripheriques des MSX. PIERRE BRANDEIS et
FREDERIC BLANC
5 :Projet de fin d'etude ENP
Mini-robot pneumatique RB-3/EV
Etudie par A.Medjahed et derige par Mme ROBL
Promotion JAN 88
6 :Projet de fin d'etude ENP
Etude d'un mini-robot RB-4/EV muni d'un mini-ordinateur
MPZ-80/EV
Etudie par A.Chouieur et derige par A.ZERGUERRAS
7* :Les systemes microprogramme automatises. T.MAURIN
8* :Encyclopedie le temps des robots. BORDAS
9* *automatisation industrielle. HORNAUER

(*) Documents non consultes.

LE JEU D'INSTRUCTIONS DU Z80:

Conventions d'ecriture:

Le jeu d'instructions du Z80 est tres etendu puisqu'il comporte pres de 700 codes operations differents.

-r designera un registre 8 bits (A,B,C,D,E,H,L).

-dr designera un double registre 16 bits,soit BC,DE,HL,SP,IX ou IY.Certaines instuctions ne portent que sur quelques-uns de ces registres.

-Ind designera un registre d'index (IX ou IY).

-data 8 designera une donnee sur 8 bits.

-data 16 designera une donne sur 16 bits.

-ad designera une adresse (16 bits).

-op ---------- --- operande variable.

-cond---------- --- condition sur les indicateurs.

-() designeront le contenu de l'emplacement memoire qu'elles encadrent.S'il s'agit du contenu d'un registre,nous ommettrons ces parentheses afin d'alleger l'ecriture et de conserver la coherence avec les mnemoniques.

-d designera un deplacement.

-> designera le ET logique.

-< ---------- -- OU --------.

-A ---------- -- OU exclusif (XOR).

-:= Ce symbole sera celui de l'affectation;cela consiste a calculer la valeur qui se trouve a droite de ce signe et a l'affecter a la variable qui se trouve a gauche.

Groupe 1 : Les transferts de données

Mnémoniques	EFFET	Commentaire
LD r, op	r := op r est l'un des registres A, B, C, D, E, H, L op peut être r, data 8, (HL), (IX+d), (IY+d).	Le registre spécifié est chargé avec l'opérande mentionnée.
LD A, op	A := op op peut être A, B, C, D, E, H, L, (BC), (DE), (HL), (IX+D), (IY+D), data 8, (ad).	L'accumulateur A est chargé avec l'opérande specifiée.
LD dr, (ad)	Partie basse de dr := (ad) Partie haute de dr := (ad+1) dr peut être BC, DE, HL, IX, IY ou SP	Le contenu de l'adresse ad est chargé dans la partie basse du registre specifié ; le contenu de l'adresse ad+1 est chargé dans sa partie haute.
LD dr, data 16	dr := data 16 dr peut être BC, DE, HL, IX, IY ou SP	La valeur 16 bits data 16 est chargée dans le double registre specifié.
LD (ad), dr	(ad) := partie basse de dr (ad+1) := " haute de dr dr peut être A, B, C, D, E, H, L ou data 8	L'adresse ad est chargée avec l'octet le moins significatif de dr ; l'adresse ad+1 est chargée avec l'octet le plus significatif de dr
LD (dr), op	(dr) := op dr peut être HL, IX+d, IY+d op peut être A, B, C, D, E, H, L ou data 8	L'adresse pointée par le double registre mentionné est chargée avec l'octet immédiat indiqué.
LD A, I ou R	A := I A := R	L'accumulateur est chargé avec le contenu du registre de vectorisation des interruptions (I) ou avec le contenu du de rafraichissement dynamique des mémoires (R).
LD SP, dr	SP := dr dr peut être HL, IX ou IY	Le registre pointeur de pile SP est chargé avec le contenu du double registre spécifié
LDD	(DE) := (HL) ; DE := DE-1 BC := BC-1 ; HL := HL-1	Le contenu de l'adresse pointée par HL est chargé dans l'adresse pointée par DE, puis BC, DE, HL sont décrementés tous les trois.
LDDR	(DE) := (HL) ; DE := DE-1 BC := BC-1 ; HL := HL-1 ; répétition jusqu'à ce que BC=0	Le contenu de l'adresse pointée par HL est chargée dans l'adresse pointée par DE, puis BC, DE, HL sont décrementés tous les trois. Si BC n'a pas été annulé on recommence.

LDIR	(DE) := (HL); DE := DE+1 BC := BC−1 ; HL := HL+1; répétition jusqu'à ce que BC=0	Le contenu de l'adresse pointée par HL est chargé dans dans l'adresse pointée par DE, puis DE, HL sont incrémentés, et BC est décrementé. Si BC≠0, on recommence.
LDI	(DE) := (HL); DE := DE+1, HL := HL+1; BC := BC−1	Identique à LDD, mais ici HL et DE sont incrementés.
EX DE, HL	DE ⟷ HL	Le contenu du double registre DE est échangé avec celui de HL.
EX (SP), HL ou IND	L ⟷ (SP); H ⟷ (SP+1) ou Ind_{bas} ⟷ (SP); Ind_{haut} ⟷ (SP+1)	Le contenu du registre L ou de la partie basse du registre index est échangé avec celui de l'adresse pointée par SP; le contenu de H ou de la partie haute du registre d'index est echangé avec celui de l'adresse immédiatement superieure
EX AF, AF'	AF ⟷ AF'	Le contenu de l'accumulateur et les indicateurs sont échangés avec celui du groupe auxiliaire de registres.
EXX	BC ⟷ BC' DE ⟷ DE' HL ⟷ HL'	Les contenus de BC, DE et HL sont echangés avec celui des registres auxiliaires correspondants.

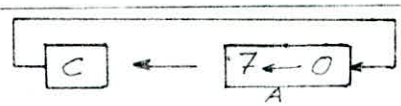
Groupe 2 : Opérations arithmétiques

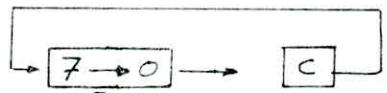
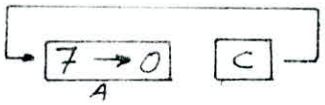
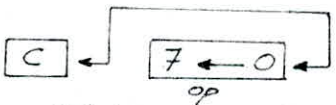
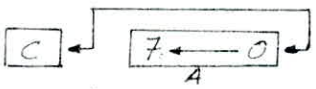
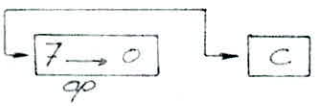
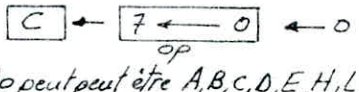
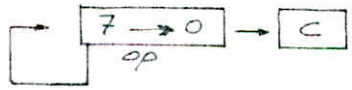
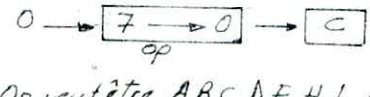
ADC A, op	A := A + op + C op peut représenter : A, B, C, D, E, H, L, data 8 bits, (HL), (IX+d), (IY+d).	L'operande ainsi que la Carry C sont additionnés à l'accumulateur. Le resultat est placé dans l'accumulateur.
ADC HL, dr	HL := HL + dr + C dr peut être : BC, DE, HL ou SP.	Le contenu du registre HL et celui du double registre specifié sont additionnés avec la Carry C. Le resultat est placé dans le double registre HL.
ADD A, op	A := A + op op peut être : A, B, C, D, E, H, L, data 8, (HL), (IX+d), (IY+d).	Le contenu de l'opérande est additionné (sans retenue, à la difference de l'instruction ADC A, op) au contenu de l'accumulateur. Le resultat est placé dans l'accumulateur
ADD HL, dr	HL := HL + dr dr peut être BC, DE, HL ou SP	Le contenu du double registre HL est additionné au contenu du double registre. Le resultat est placé dans HL.

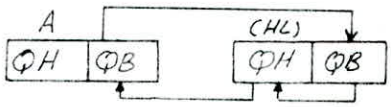
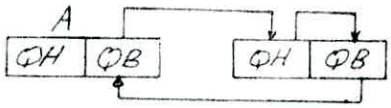
ADD Ind, dr	Ind := Ind + dr Ind peut être IX ou IY dr peut être BC, DE, SP ou Ind	Le contenu du registre index est additionné au contenu au double registre dr. Le resultat est placé dans le registre d'index
SBC A, op	A := A - op - C op peut être : A, B, C, D, E, H, L, data 8, (HL), (IX+d), (IY+d).	Le contenu de l'opérande est dans un premier temps ajouté au contenu de la Carry, puis l'ensemble est poustrait du contenu de l'accumulateur.
SBC HL, dr	HL := HL - dr - C dr peut être BC, DE, HL, ou SP	Le contenu de dr est dans un premier temps ajouté au contenu de la carry, puis l'ensemble est poustrait au contenu du double registre HL. Le resultat est placé dans HL.
SUB op	A := A - op op peut être A, B, C, D, E, H, L, data8, (HL), (IX+d), (IY+d).	Le contenu de l'opérande est poustrait du contenu de l'accumulateur. Le resultat est placé dans l'accumulateur.
INC op	op := op + 1 op peut être A, B, C, D, E, H, L, (HL), (IX+d), (IY+d).	Cette opération ajoute 1 au contenu de l'opérande spécifiée. Le resultat est replacé dans l'operande
INC dr	dr := dr + 1 dr peut être BC, DE, HL ou SP	Le double registre dr est incrémenté. Le resultat est placé dans le même double registre.
INC Ind	Ind := Ind + 1	Le registre index specifié est incrementé. Le resultat est placé dans ce même Ind.
DEC op	op := op - 1 op peut être r, (HL), (IX+d), (IY+d)	Le contenu de l'opérande est diminué de 1. Le resultat est placé dans cette même operande
DEC dr	dr := dr - 1. dr peut être BC, DE, HL, ou SP	Le contenu de dr est décrementé. Le resultat est placé dans ce même dr.
DEC Ind	Ind := Ind - 1	Le contenu du registre d'index est décrementé. Le resultat est placé dans ce même Ind.
CPL	Ind : A := $\bar{A}$	Le contenu de l'accumulateur est complement à 1. Le resultat est placé dans l'accumulateur (Complémentation = on place chaque bit 0 par 1 et inversement 1 par 0).
NEG	A := -A	On soustrait l'accumulateur 0 et on stocke le resultat dans cet accumulateur

DAA	A := A ajusté décimal	Ajustement décimal de l'accumulateur

Groupe 3 : Opérations logiques, décalages, rotations, tests de bits

AND op	A := A ∧ op op peut être A, B, C, D, E, H, L, data 8, (HL), (IX+d), (IY+d).	Le ET logique entre l'accumulateur et l'opérande spécifiée est effectué. Le résultat est rangé dans l'accumulateur
OR op	A := A ∨ op op peut être A, B, C, D, E, H, L, data 8, (IX+d), (IY+d)	Le ou logique entre l'accumulateur et l'opérande spécifiée est effectué. Le résultat est rangé dans l'accumulateur
XOR op	A := ∀ op op peut être A, B, C, D, E, H, L, data 8, (HL), (IX+d), (IY+d).	Le OU EXCLUSIF logique entre l'accumulateur et l'opérande spécifiée est effectué. Le résultat est rangé dans l'accumulateur
SCF	C := 1	Mise à 1 de Carry.
CCF	C := $\overline{C}$	Complémentation de Carry.
BIT b, op	Z := Inverse du bit N°b de op. Op peut être A, B, C, D, E, H, L, (HL), (IX+d), (IY+d).	Le bit b de l'opérande spécifiée est testé ; s'il est à 1, l'indicateur Z est mis à 0 ; s'il est à 0, Z est mis à 1.
SET b, op	bit n° b de op := 1 Op peut être A, B, C, D, E, H, L, (HL), (IX+d), (IY+d)	Le bit b de l'opérande spécifiée est mis à 1.
RES b, op	Z : bit n° b de op := 0 Op peut être A, B, C, D, E, H, L, (HL), (IX+d), (IY+d).	Le bit b de l'opérande spécifiée est mis à 0.
RL op	 Op peut être A, B, C, D, E, H, L, (HL) (IX+d), (IY+d)	Le contenu de l'opérande spécifiée est décalé d'un bit vers la gauche ; le bit 0 prend la valeur de l'indicateur C ; l'indicateur C prend la valeur du bit 7. Le résultat est remis à l'emplacement d'origine. La rotation se fait donc sur 9 bits.
RLA		Le contenu de l'accumulateur est décalé d'un bit vers la gauche ; le bit 0 prend la valeur de l'indicateur C ; l'indicateur C prend la valeur du bit 7. Le résultat est remis dans l'accumulateur. La rotation se fait donc sur 9 bits.

RR op	 Op peut être A,B,C,D,E,H,L,(HL),(IX+d),(IY+d)	Le contenu de l'opérande spécifiée est décalé d'un bit vers la droite; le bit 7 prend la valeur de l'indicateur C; l'indicateur C prend la valeur du bit 0. Le résultat est remis à l'emplacement d'origine.
RRA		le contenu de l'accumulateur est décalé d'un bit vers la droite; le bit 7 prend la valeur de l'indicateur C; l'indicateur C prend la valeur du bit 0. le résultat est remis dans l'accumulateur. la rotation se fait sur 9 bits.
RLC op	 Op peut être A,B,C,D,E,H,L,(HL),(IX+d),(IY+d)	le contenu de l'opérande spécifiée est décalé circulairement d'un bit vers la gauche; le bit 7 va en même temps dans le bit 0 et dans l'indicateur C. Le résultat est remis à l'emplacement d'origine. la rotation se fait sur 8 bits.
RLCA		le contenu de l'accumulateur est décalé d'un bit vers la gauche; le bit 7 va en même temps dans le bit 0 et dans l'indicateur C. Le résultat est remis dans l'accumulateur. la rotation se fait sur 8 bits.
RRC op	 Op peut être A,B,C,D,E,H,L,(HL),(IX+d),(IY+d)	le contenu de op spécifiée est décalé circulairement d'un bit vers la droite; le bit 0 va en même temps dans le bit 7 et dans l'indicateur C. Le résultat est remis à l'emplacement d'origine, la rotation se fait sur 8 bits.
RRCA		le contenu de l'accumulateur est décalé d'un bit vers la droite; le bit 0 va en même temps dans le bit 7 et dans l'indicateur C. Le résultat est remis dans l'accumulateur. La rotation se fait sur 8 bits.
SLA op	 Op peut peut être A,B,C,D,E,H,L,(HL),(IX+d),(IY+d)	le contenu de l'emplacement spécifié est décalé à gauche d'un bit. Le bit 7 tombe dans l'indicateur C, et il entre un 0 par la droite dans le bit 0. Le résultat est remis à l'emplacement de départ
SRA op	 Op peut être A,B,C,D,E,H,L,(HL),(IX+d),(IY+d)	le contenu de l'emplacement spécifié est décalé à droite d'un bit. le bit 0 tombe dans l'indicateur C et le bit 7 est inchangé. Le résultat est remis à l'emplacement de départ.
SRL op	 Op peut être A,B,C,D,E,H,L,(HL) (IX+d),(IY+d)	le contenu de l'emplacement spécifié est décalé à droite d'un bit. le bit 0 tombe dans l'indicateur C, et il entre un 0 à gauche dans le bit 7. le résultat est remis à l'emplacement de départ

RLD	 ΦH = quartet haut ΦB = " bas	Rotation par quartets entre (HL) et l'accumulateur. Le ΦB du contenu de l'adresse pointée par HL est décalé le ΦH de cette même adresse. Le ΦH de (HL) va dans ΦB de A; le ΦB de A va dans le ΦB de (HL).
RRD		Rotation par quartets entre (HL) et l'accumulateur. Le ΦH du contenu de l'adresse pointée par HL est décalé dans le ΦB de cette même adresse. Le ΦB de (HL) va dans le ΦB de A; le ΦB de A va dans le ΦB de (HL).

Groupe 4 : Comparaisons, Sauts, Pile, Sous-programmes.

CP op	A − op op peut être A,B,C,D,E,H,L,(HL),(IX+d),(IY+d), data 8	L'opérande spécifiée est soustraite de l'accumulateur, le résultat n'est pas conservé, mais positionne les indicateurs.
CPD	A − (HL); HL := HL−1; BC = BC−1	le contenu de l'adresse pointée par HL est soustrait de l'accumulateur. le résultat n'est pas conservé, mais affecte les indicateurs. Ensuite, HL et BC sont tous deux décrementés (et prêts pour une nouvelle comparaison).
CPDR	A−(HL); HL := HL−1; BC := BC−1; Si A<>(HL) et BC<>0, PC := PC−2	le contenu de l'adresse pointée par HL est soustrait à l'accumulateur. Le résultat n'est pas gardé, mais positionne les indicateurs. Ensuite, HL et BC sont décrementés. l'instruction se réexécute jusqu'à ce que la comparaison réussisse ou que BC = 0.
CPI	A−(HL); HL := HL−1; BC := BC−1	le contenu de l'adresse pointée par HL est soustrait de l'accumulateur. le résultat affecte les indicateurs. Ensuite HL est incrementé, et BC est decrementé (et ils sont prêts pour une nouvelle comparaison).
CPIR	A−(HL); HL := HL+1; BC = BC−1; si A≠(HL) et BC≠0, PC := PC−2	le contenu de l'adresse pointée par HL est soustrait à l'accumulateur, le resultat positionne les indicateurs. Ensuite, HL est incrementé et BC est decrementé. L'instruction se réexécute jusqu'à ce que la comparaison réussisse ou que BC = 0.
JP ad	PC := ad	la valeur ad est chargée dans PC, provoquant le saut à l'adresse ad.
JP cond, ad	Si cond vraie, comme JP ad Si " fausse, pas d'effet.	l'indicateur specifié est testé. Si la condition est remplie on saute à l'adresse ad. Si la condition n'est pas remplie, on passe à l'instruction suivante (soit PC := PC+).
JP (dr)	PC := dr dr peut être HL, IX ou IY.	le contenu du double registre dr est chargé dans PC provoquant le saut à cette adresse.

JR d	PC := PC + d	le déplacement fourni d est ajouté au compteur ordinal Pc provoquant un saut de d octets. d est un nbre signé ; le possible est donc de -128 octets en arrière à +127 octets en avant, comptés à partir de l'adresse de l'instruction qui suit JR (puisque l'adresse de la prochaine instruction)
JR cond, d	Si cond vraie, comme JR d. " " fausse, pas d'effet.	l'indicateur spécifié est testé. Si la cond est remplie, on effe le saut relatif d décrit pour JR d. Si la cond n'est pas rem On passe à l'instruction suivante (PC := PC + 2).
DJNZ d	B := B-1 ; Si B<>0, PC := PC + d	B est décrémenté ; si cette opération ne l'a pas annulé, on opère le saut relatif d. d est un nbre signé. le déplacemen possible est donc de -128 octets en arrière à +127 octets e avant, comptés à partir de l'adresse de l'instruction Suivant DJNZ.
PUSH dr	SP := SP-1 ; (SP) := partie forte de dr. SP := SP-1 ; (SP) := " faible de dr dr peut être AF, BC, DE, HL, IX, IY	SP est décrémenté ; la partie forte de dr est chargée à la nouvelle adresse pointée par SP. SP est de nouveau décrémenté, et la partie basse de dr est chargée à l'adresse pointée par SP. A la fin de l'opération, SP pointe donc sur partie basse de la valeur empilée, et SP+1 pointe sur sa par haute.
POP dr	Partie faible de dr := (SP) ; SP := SP+1 ; " forte " := (SP) ; SP := SP+1 dr peut être, AF, BC, DE, HL, IX, IY	le contenu de SP est chargé dans la partie faible du dr ; SP est incrémenté. le contenu de la nouvelle adresse pointée par SP est chargée dans la partie forte du dr spécifié. SP est de nouveau incrémenté pour pointer sur la nouvelle valeur à dépiler.
CALL ad	d'abord PUSH PC, puis PC := ad	PC est empilé comme si l'on éxécutait PUSH PC. Ensuite PC est chargé avec la valeur ad, provoquant le branch ment à l'adresse ad. le retour se fera à la rencontre d l'instruction RET, qui dépilera PC.
RET	PC := (SP)	PC est dépilé comme si l'on éxécutait POP PC. Ceci provoque le retour au programme principal, à conditi que la pile n'ait pas été modifiée.
RET cond	si cond vraie, comme RET Si cond fausse, ignoré	l'indicateur spécifié est testé. Si la cond est vérifiée, PC est dépilé, provoquant le retour du s-progr. Si la cond n'e pas remplie, on continue en sequence
RST ad	SP est empilé comme PUSH PC ; PC := ad	Après avoir empilé PC, on saute à l'adresse page 0 spécifié
RETI et RETN	Comme RET	Comme RET (retour de s-prog d'interruption).

Groupe 5 : Entrées/sorties, interruptions, divers

IN r,(C)	r := (C) r peut être A,B,C,D,E,H, ou L.	Le port d'entrée dont l'adresse est dans le registre C est lu, et le résultat est mis dans l'accumulateur.
IN A,(P)	A := (port P) P est un nombre entre 0 et FFH (8 bits).	Le port d'adresse P est lu, le résultat est mis dans l'accumulateur.
IND	(HL) := (C); B := B-1; HL := HL-1	Le port adressé par le registre C est lu; le résultat est placé à l'adresse pointée par HL. Ensuite le registre B et le double registre HL sont décrémentés (pour préparer une prochaine entrée).
INI	(HL) := (C); B := B-1; HL := HL+1	Idem que pour IND en remplaçant « HL est décrémenté » par « HL est incrémenté ».
INDR et INIR		Ces instructions agissent exactement comme IND et INI, avec une répétition automatique si B n'a pas été annulé
OUT (C),r	port (C) := r r peut être A,B,C,D,E,H, ou L	Le contenu du registre r est écrit dans le port dont l'adresse est dans C.
OUT (P),A	(port P) := A	Le contenu de l'accumulateur est écrit dans le port d'adresse P. P est un nombre entre 0 et 255 (FFH). En fait, seuls quelques ports sont actifs sur MSX (voir carte Entrées/Sortie
OUTD	(C) := (HL); B := B-1; HL := HL-1	Le contenu de l'adresse pointée par le double registre HL est écrit sur le port adressé par le registre C. Ensuite, le registre B et le double registre HL sont décrémentés (pour préparer une prochaine sortie).
OUTI	(C) := (HL); B := B-1; HL := HL+1	Le contenu de l'adresse pointée par le double registre HL est écrit sur le port adressé par le registre C. Ensuite, le registre B est décrémenté et le double registre HL est décrémenté (pour préparer une prochaine sortie).

OTIR et OTDR		Ces instructions agissent exactement comme OUTI et OUTD avec une répétition automatiq si B n'a pas été annulé.
IM0, IM1, IM2		Sur le système MSX, c'est le mode IM1 qui est utilisé. Vous ne devez pas changer ce mode, sous peine de "planter la machine". Ne vous serez donc pas de ces instructions, citées pour mémoire
DI	Les interruptions sont interdites jusqu'à la rencontre de l'instruction EI.	Interdiction des interruptions (Disable interrup.
EI	Les interruptions sont autorisées après l'exécution de l'instruction SUIVANT EI.	Autorisation des interruptions (Enable interrupts
NOP	Aucun	Pas d'effet
HALT		Le Z80 arrête son fonctionnement et exécute des NOP jusqu'à recevoir une interruption ou une réinitialisation.