

8/83  
83

200

# ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT :

امدرسة الوطنية للعلوم الهندسية  
ELECTRONIQUE  
ECOLE NATIONALE POLYTECHNIQUE  
BIBLIOTHEQUE

## PROJET DE FIN D'ETUDES

SUJET

E tude et Realisation d'un bus  
IEEE-488 (G.P.L.B)

Proposé par :

M<sup>r</sup>. A. Boukeb

Etudié par :

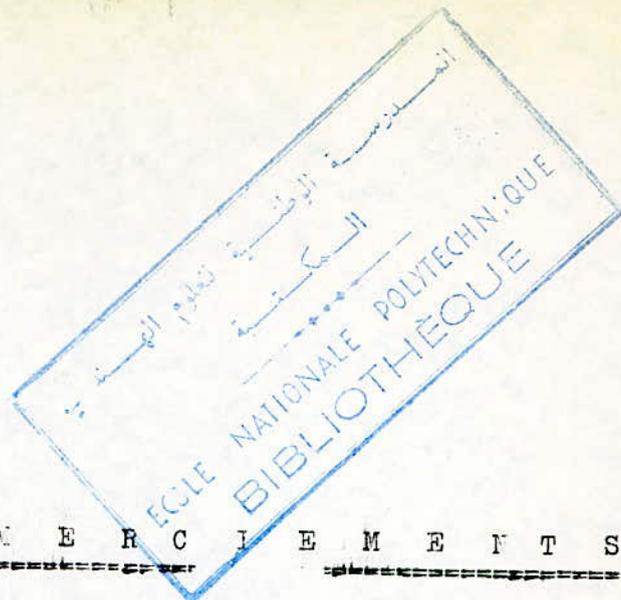
Mell Belahène  
Mebarka  
Mell Hachim  
Fella

Dirigé par :



PROMOTION :

JAIR-83



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

Au terme de ce travail, nous tenons à remercier vivement  
M. BOURKEB et SAIDJ pour l'aide précieuse et les conseils prati-  
ques dont ils nous ont fait part.

Nous remercions également tout le personnel du laboratoire  
et particulièrement M. HERRY pour le matériel qu'ils ont mis à notre  
disposition tout le long du semestre.

Nous remercions M. FLAANT pour les bons conseils qu'il nous  
a prodigués pour l'élaboration de ce polycopié.

Que tous ceux qui nous ont aidé trouvent en ce modeste tra-  
vail, l'expression de notre profonde gratitude.

# PLAN

## INTRODUCTION

### I GENERALITES

1. Définition d'un bus
2. Différents standards de bus.

### II ETUDE DU BUS IEEE - 488

#### A. Description du bus

1. Description mécanique
2. Description électrique

#### B. Utilisation du bus IEEE - 488

1. Rôle des terminaux et communications sur le bus IEEE - 488.
  - 1.1 Rôle des terminaux
  - 1.2 Communications sur le bus IEEE-488
2. Utilisation du bus et commandes classiques
  - 2.1 Choix du mode local ou à distance
  - 2.2 Messages et commandes particulières
3. Echanges d'informations sur le bus IEEE-488 - principe de la  
synchronisation --.
4. Utilisation du bus et commandes évoluées.
5. Fonctions disponibles dans la norme IEEE - 488

### III PRESENTATION DU GFIA

1. Les fonctions du GFIA
2. Signaux et brochage du GFIA
3. Registres du GFIA

### IV ETUDE DE LA CARTE GFIA

#### Première partie : HARDWARE

##### A. Liaison MPU - GFIA

1. Amplification du bus
2. Logique de commande des 8T26
3. Décodage d'adresse.

##### B. Liaison GFIA - IEEE 488

1. Utilisation des IC 3447
2. Utilisation des 8T28.

##### C. Liaison IEEE-488 - FIA -

#### Deuxième partie : SOFTWARE

### V ANNEXES

Conclusion

Bibliographie.

## INTRODUCTION

Le service électronique du CEN est amené, dans le cadre de ses activités, à utiliser de plus en plus fréquemment des interconnexions entre des appareils de mesure numériques et alphanumériques et des micro-systèmes (ex.: Microordinateur), système de gestion microprogrammé.

Le but de notre travail a été de réaliser une carte standard compatible entre ces différents éléments et basée sur la nouvelle norme internationale IEEE qui tend à se généraliser sur toute l'instrumentation commercialisée.

La réalisation de cette carte présente l'avantage de raccorder directement les systèmes compatibles avec ce bus sans faire appel à des circuits supplémentaires. Ceci permettra la réalisation des systèmes automatiques où l'intervention manuelle est très réduite.

A la différence des autres modes de raccordement, le bus IEEE-488 n'est pas un bus informatique au sens traditionnel du terme ; car, son but n'est pas de relier un ordinateur à des périphériques classiques, mais de faire communiquer des appareils de mesure sous la surveillance d'une unité maître. Cette unité maître constituera le contrôleur dans l'étude qui va suivre.

Il s'ensuit que les caractéristiques de ce bus sont optimisées pour l'instrumentation qui en fait son principal domaine d'emploi.

Le bus IEEE-488 (Institute of Electrical and Electronics engineers) a été créé par la firme HEWLETT PACKARD en 1966 pour ses propres besoins sous le nom de GPIB.

En 1975, puis révisé en 1978, le standard IEEE-488 dont la dénomination exacte est IEEE "Standard digital interface for programmable instrumentation" est connu sous d'autres dénominations.

- GPIB : Général Purpose Interface Bus
- HPIB : Hewlett Packard Interface Bus
- IEC.625-1 Bus (International Electrical Commission)
- ANSI MC1-1 Bus (American National Standard Institute)

Une interface suffisamment élaborée et simple a été prévue pour ce standard. C'est le MC 68488 (GPIA : General purpose interface adapter) conçu pour travailler avec la famille des microprocesseurs 6800. Toutefois, ce circuit peut être utilisé avec d'autres microprocesseurs tels que le MC 6502, TM 9914 etc...

L'interface IEEE 488 a supplanté les interfaces BCD qui ne permettraient que les échanges entre des données numériques.

La prétention de notre travail n'est pas d'utiliser le bus IEEE-488 dans des applications très précises. Nous nous sommes contentées de faire dans un premier temps fonctionner le bus avec un PIA classique (Peripheral Interface Adapter) adressé comme contrôleur. A partir de ce travail, il sera aisément possible au service électronique du C.E.N d'utiliser cette carte pour des applications plus générales.

## I GENERALITES

Pour connecter plus d'un module, il faut une voie de communication.

Souvent, dans un système, chaque module doit être capable de dialoguer avec les modules voisins.

Un bus est un moyen de communication constitué d'un ensemble de fils électriques parallèles, sur lequel viennent se connecter l'ordinateur et ses périphériques ou l'instrumentation à contrôler.

Pour caractériser cette voie de la façon la plus complète, il est nécessaire de bien préciser le nombre de fils qu'il comporte ; les signaux qui transitent sur ces fils et la procédure d'échange des informations. En général, un bus doit véhiculer tous les signaux utilisés par le système : données - adresses - contrôles et tensions d'alimentation.

Les signaux transmis peuvent se présenter sur des fils distincts ou non, on dit qu'il s'agit de bus multipléxé ou non multipléxé.

On rencontre deux types d'information au sein d'un bus :

- Les signaux électriques associés au matériel
- Les signaux gérant l'échange des informations associés au logiciel

On distingue deux types de bus :

- Bus séries
- Bus parallèles

### 1. Les bus séries

Les bus séries nécessitent peu de lignes et sont utilisés pour connecter des terminaux à un système à longue distance. Les terminaux tels que les écrans cathodiques, les imprimantes, les téléscripteurs, sont reliés par des systèmes de communications séries.

Les transmissions séries nécessitent seulement un ou deux fils pour transmettre tous les signaux entre modules du système : les adresses, les données et les commandes doivent être transférées bit/bit. Les principaux interfaces utilisant le bus série sont :

- EIA - RS 232 C pour les communications asynchrones.
- EIA - RS 422 - 423 pour les communications asynchrones et synchrones.

## 2. Les bus parallèles

Ces bus sont utilisés pour les communications de module à module haute vitesse, c'est le cas des bus microprocesseurs et des interfaces de système à système.

Ainsi, on distingue deux types de bus parallèles :

- Bus internes ;
- Bus externes

### Les bus internes

Exemple : Le bus pour microprocesseur MC 6800 qui est chargé d'interconnecter les divers composants du système : Microprocesseur, mémoires, entrées-sorties etc...

### Les bus externes

- Exemple :
- Le bus interface IEEE 488
  - Le bus interface CAMAC IEEE 583
  - Le bus S100 pour microprocesseur de type 8080/Z 80

Dans la suite de notre travail, nous nous intéressons au standard IEEE qui se présente sous différentes appellations :

- IEEE 696

- IEEE 796

- IEEE 488

#### Le bus IEEE 696

Le bus S 100 IEEE 696 comme son nom l'indique est formé de cent lignes dont dix huit inemployées sont à la disposition de l'utilisateur.

Ce fut le premier véritable standard disponible en micro-informatique.

Il a été introduit au début de l'année 1975 par "MITS" (une petite société américaine produisant les micro-ordinateurs ALTAIR) pour un "Kit" d'amateur "ALTAIR" à base de 8080.

Toutefois, le bus S 100 souffre principalement d'un manque de souplesse d'utilisation, car, il a été conçu pour un matériel donné à une époque donnée.

Ce bus a donc terriblement "vieilli" et se trouve même en "perte de vitesse" avec l'apparition des microprocesseurs 16 bits auxquels il est mal adapté.

#### Le multibus SBC IEEE 796

Le multibus d'Intel également connu sous le nom de bus SBC, a été initialement conçu pour des microprocesseurs dont l'espace mémoire est distinct de l'espace d'entrées/sorties.

Le multibus possède une commande totalement décentralisée, ce qui permet la réalisation de structures multiprocesseurs à base d'unités centrales 8 bits, 16 bits ou des deux types. La transmission des données et des adresses utilise des bus distincts et le transfert des données est de type "synchrone" avec vérification par signal d'acquiescement.

Le principe de fonctionnement est assuré grâce à des modules ("Maîtres" et "Esclaves").

A la différence des autres bus qui possèdent des contrôleurs de bus, celui-ci est contrôlé par une unité appelée maître.

Des circuits logiques intégrés sur chaque carte Multibus assurent la gestion des priorités (Plusieurs modules "maîtres" pouvant accéder simultanément au bus) selon deux méthodes :

- Gestion série,
- Gestion parallèle

Ce bus est composé de quatre vingt six fils. La vitesse maxi de transport des informations est de 5 MHz.

#### Le bus IEEE 488

Le bus IEEE 488 est un bus à seize fils regroupant :

- huit lignes de données bidirectionnelles (DIO : data in out) qui transporteront des commandes d'appareils, 8 bits d'adresses et de données.

Du fait que ce système n'a pas de bus d'adresse ou de contrôle spécifique, un seul bus est utilisé. Le type de transmission utilisé est sériel : caractère par caractère.

- Trois lignes de commande de transfert (DAV, NRFD, NDAC) sont utilisées pour réaliser la technique de "handshake".

- Cinq lignes contrôlent des conditions générales du système. Ce sont : l'attention, la remise à zéro de l'interface, la demande de service l'activation à distance, la fin ou identification (ATN - IFC - SRQ - REN - EOI). Grâce à ces seize fils, il sera possible d'interconnecter

micro-ordinateur et appareils de mesure si ceux ci sont dotés d'une sortie IEEE-488.

Les échanges sur ce bus s'effectuent de façon asynchrone et obéissent à la procédure "handshake" qui sera expliquée en détail dans l'étude des échanges d'informations sur le bus GPIB

Le format des données échangées est l'octet, la vitesse maximale de transfert des données est de l'ordre du M octet/S sur une distance limitée et 250 à 500 Kilooctets sur la distance maximale.

Le bus GPIB diffère des autres tels que Multibus - SBC IEEE 796 et S 100 IEEE 696 par :

- Le nombre de fils utilisés ;
- La vitesse de transfert des informations
- Utilisation de bus multiplexé pour les données et adresses
- domaine d'application : l'instrumentation

#### Remarque

Le multibus est plus performant, il peut relier un ordinateur à d'autres microordinateurs, comme il peut avoir son importance dans l'instrumentation vu sa rapidité d'échanges d'information et son mode de travail synchrone. Le bus IEEE 488 reste cependant plus adapté si l'application est limitée à l'instrumentation par ailleurs, il est nettement moins cher que le Multibus.

Pour la gestion de ce bus IEEE 488, nous avons réalisé un circuit d'interface à base de GPIA (Général Purpose Interface Adapter), qui comme tout interface a pour fonction d'adapter électriquement et mécaniquement la liaison entre deux systèmes.

En plus des interfaces tels que : PIA, ACIA, le GPIA 68488 gère automatiquement le "handshake" et permet de mettre en oeuvre plusieurs fonctions permettant une bonne transmission entre les différents appareils connectés au bus qui peuvent être :

- contrôleur de périphériques ;
- récepteur d'information d'un périphérique ;
- émetteur d'information vers un périphérique.

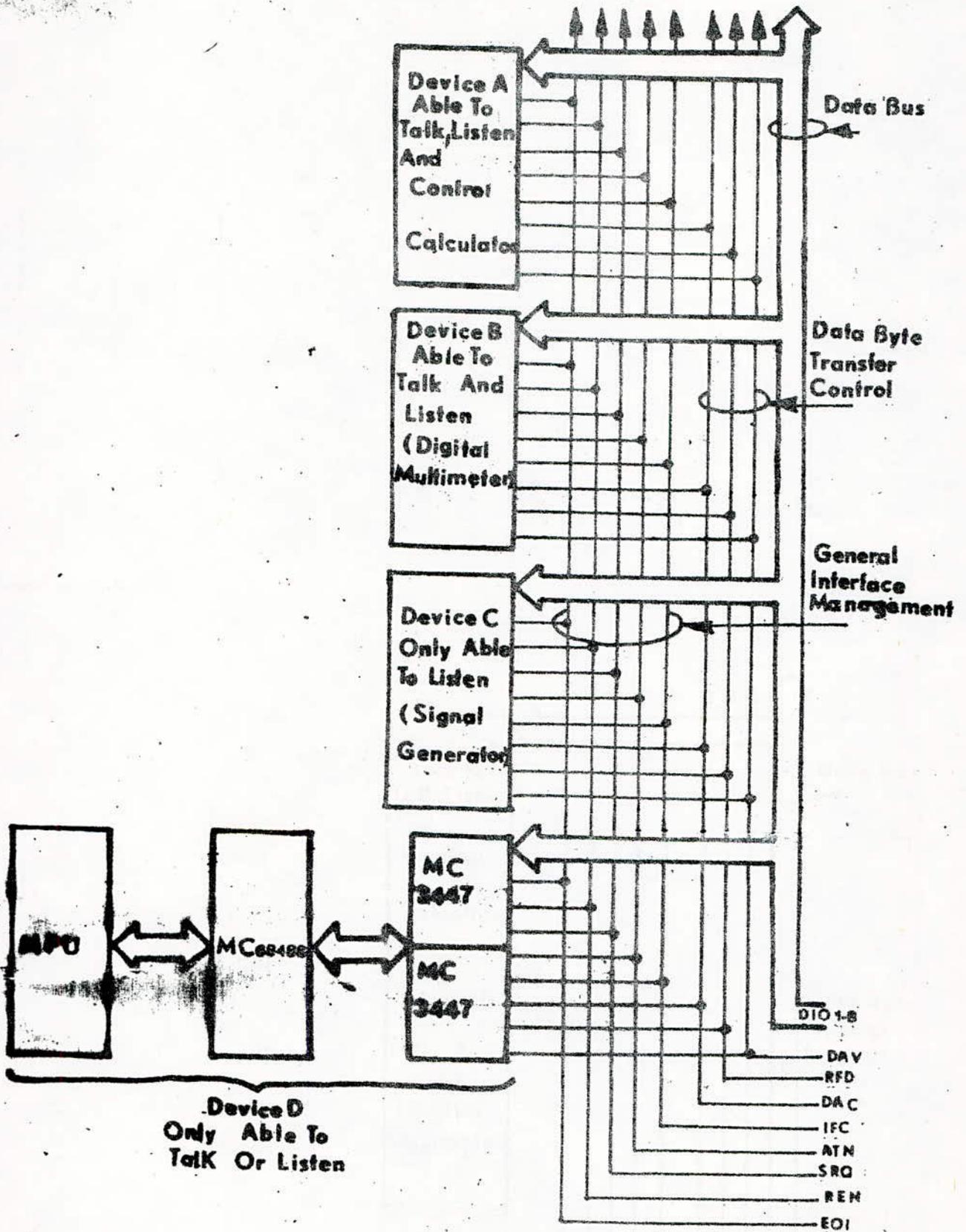
Les circuits utilisés fonctionnent en logique T.T.L. et sont généralement à collecteur ouvert, donc le fonctionnement en logique négative

Note : famille 6800 et le GPIA 68488 travaillant en logique positive)

De ce fait, les signaux du bus sont actifs au niveau bas. Le bus IEEE n'utilise pas la notation habituelle de la complémentation par une barre.

Les signaux issus du GPIA actifs au niveau bas sont notés normalement sans barre de complémentation sur le GPIB (Exemple : la ligne  $\overline{\text{DAV}}$  du GPIA s'écrit DAV sur le GPIB.

Par contre, les signaux actifs au niveau haut tels que RFD, DAC s'écrivent NRFD et NDAC etc sur le GPIB.



# SYSTEME GPIIB

## II Etude du bus IEEE - 488

### A. Description du bus

#### 1. Description mécanique

Le raccordement au bus utilise selon le cas des connecteurs 24 ou 25 broches dont le schéma est donné en fig. 1. cette double normalisation s'explique pour des raisons historiques.

Originellement, le bus IEEE - 488 fut développé sur un connecteur de 24 broches, puis normalisé par la suite à 25 broches principalement à la demande des constructeurs européens.

Dans notre étude, nous avons opté pour la norme européenne pour deux raisons :

- le standard européen est normalisé ;
- Ce modèle possède sur son homologue américain un meilleur blindage "HF".

Quoique le connecteur européen possède la même forme géométrique que l'interface V 24/RS 232C, ce qui peut causer en cas d'inattention des erreurs catastrophiques.

Les câbles de connexion sont de longueur 0,5 m - 1 m - 2 m - 4 m (longueur maximale autorisée entre deux appareils).

Celle-ci est limitée de telle sorte que les temps de propagation des signaux ne puissent pas entraîner la désynchronisation des communications.

Chaque câble possède à chacune de ses extrémités une prise mâle, une femelle et deux vis dont la tête est identique aux écrous de la prise châssis.

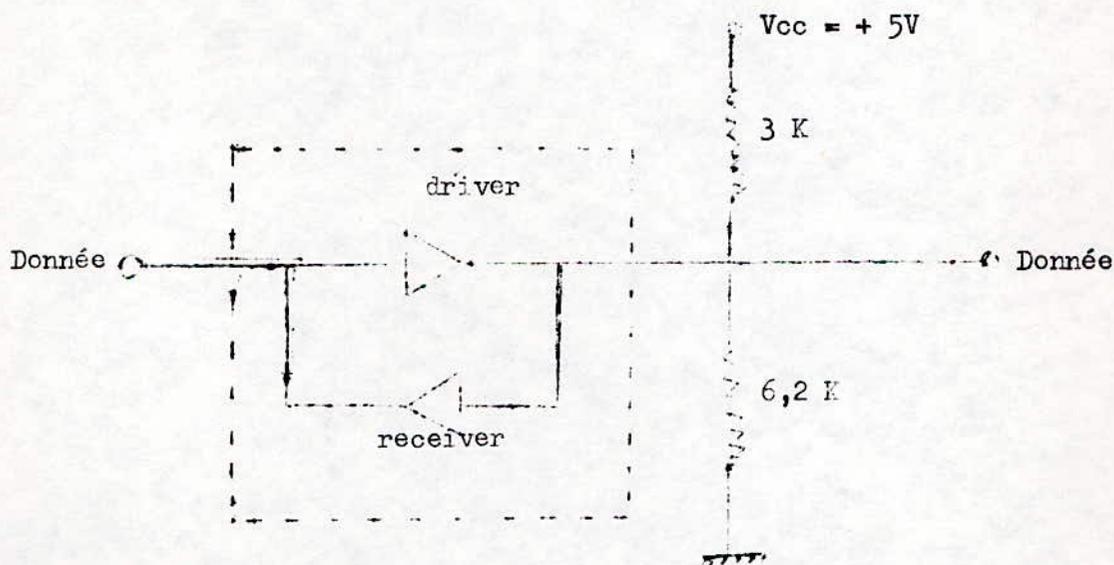
## 2. Description électrique

Les signaux électriques circulent sur le bus sont compatibles TTL, et la logique utilisée est négative.

L'état "1" correspond à une variable active et donc représentée par une tension comprise entre 0 et 0,8 V (niveau bas), l'état "0" (inactif) est représenté par une tension comprise entre 2 et 5 V (niveau haut).

Toutes les lignes se connectent, en général, sur le bus par un transistor à collecteur ouvert, chargé par un pont de résistances de 3 K et 6,2 K. Ce pont est parfois intégré dans certains circuits "transceivers".

Pour des raisons technologiques et le bon fonctionnement des appareils, le nombre d'instruments interconnectés est limité à 15.



Le bus est constitué par l'ensemble des câbles de liaison qui cheminent bidirectionnellement les informations d'un appareil à un autre. Il comprend comme l'indique la fig. 2 :

a) le bus de donnée

Il peut suivant les circonstances transporter :

- des données numériques alphanumériques (texte)
- des adresses de périphériques ;
- des commandes normalisées (multilignes) ;
- des mots d'états (status byte).

b) Le bus de synchronisation (handshake) :

Il est composé de :

- NRPD (Not ready for data) : l'accepteur n'est pas prêt à recevoir la donnée.
- DAV (Data Valid) donnée disponible sur le bus
- NDAC (Not data accepted) donnée pas acceptée.

Ces lignes transportent les signaux de synchronisation qui sont générés par les appareils en communication.

c) Le bus de **contrôle** et commande

Il est composé de :

- IFC, ATN, REN, SRQ, EOI.

Ces lignes sont utilisées pour la gestion de l'interface.

Leurs rôles seront explicités progressivement.

B. Utilisation du bus IEEE - 488

## 1. Rôle des terminaux et communications sur le bus IEEE-488

### 1.1 Rôle des terminaux

Sur le bus peuvent être connectés différents types d'appareils programmables ou non :

- Des appareils "listeners only" récepteur seulement  
ex : une imprimante.
- des appareils "talkers only" émetteur seulement  
ex : compteur, fréquencemètre.
- des appareils possèdent suivant les circonstances, alternativement des deux fonctions "talker" ou "listener",  
ex : une disquette, un lecteur - perforateur de ruban, un multimètre à qui l'on peut indiquer qu'il doit mesurer des tensions, des courants ou des résistances et qui peut ensuite donner le résultat de ses mesures.
- Un contrôleur ; celui-ci possède la maîtrise du bus et gère les échanges entre émetteurs - récepteurs, ex. :  
calculateur.

Le contrôleur peut avoir les fonctions Ecouteur/ Parleur. Celui-ci peut lui même se désigner comme talker ou listener.

Un appareil qui à un instant donné joue un rôle effectif sur le bus est dit "actif". Un "talker" ou un "listener" ne peut être actif que s'il y a été autorisé par le contrôleur.

A un instant donné, un seul émetteur peut parler sur le bus (faute de quoi le message serait incompréhensible). Il peut y avoir plusieurs appareils à l'écoute (listeners actifs).

Chaque partenaire y compris le contrôleur est caractérisé par une adresse que l'utilisateur fixe au moyen de cinq inverseurs. Ces derniers font partie des 8 micro-interrupteurs (switchs) présents sur la carte CFIB. Les bits 6 et 7 sont utilisés pour préciser au périphérique qu'il devra se comporter comme "talker" ou "listener". Le bit 8 est à zéro. Avec ces 5 inverseurs, on peut constituer  $2^5 = 32$  adresses binaires différentes dont 31 sont utilisables, la 32<sup>e</sup> est inutilisable (Untalk/Unlisten).

Il faut veiller à ce que deux appareils différents n'aient pas la même adresse.

#### 1.2 Communications sur le bus IEEE - 488

Sur le bus peuvent avoir lieu trois types de communications :

- a- Communication normale entre périphériques avec la participation du contrôleur.
- b- Communication sur un bus dépourvu de contrôleur.
- c- Communication entre contrôleurs

##### a) Communication sur un bus ayant un contrôleur

Sur ce bus, le contrôleur est l'autorité qui désigne le parleur, le (s) écouteurs, émet les commandes et organise les conversations.

##### b) Communication sur un bus dépourvu d'un contrôleur

Cette communication s'établit entre un "ton" (talk only)

Sur les appareils modernes possédant l'interface GPIA, on trouve généralement sur le panneau arrière à proximité de la sélection d'adresse un inverseur dont les

2 positions sont désignées par :

|    |   |            |
|----|---|------------|
| 1. | } | adressable |
|    |   | ton        |
| 2. | } | adressable |
|    |   | lon        |

Lorsque l'inverseur est sur la position talk only ; l'appareil assume dès sa mise en route, le rôle talker actif. Très précisément, l'inverseur émet à destination de son interface le message "local ton" pour choisir ce mode de fonctionnement, ou  $\overline{\text{ton}}$  qui sélectionne le mode normal adressé (adressable). De façon analogue, le message "lon", également émis par un inverseur, désigne dès sa mise en route, un interface comme listener actif alors que  $\overline{\text{lon}}$  correspond au mode normal adressé.

Pour que le contrôleur ait connaissance de la fin de la conversation entre deux de ses périphériques, le moyen le plus efficace est qu'il y participe comme listener actif, quitte à mémoriser des informations dont il n'a rien à faire.

c) Communication entre contrôleurs

De même que la communication directe entre périphériques sur un bus dépourvu de contrôleur peut avoir lieu, la communication entre contrôleurs peut également exister. Pour éviter des conflits au niveau du bus

un seul système exerce la fonction de contrôleur à un temps donné, il est désigné comme "system controller". Cette affectation est réalisée par un inverseur qui envoie à son interface le message local "rso" (Request system control). Cela est semblable au message "ton" et au message "lon".

Lorsque **rso** est actif, le contrôleur assume dès sa mise en route le rôle de contrôleur actif. Lorsque **rsc** est inactif, le contrôleur potentiel se comporte comme un périphérique classique possédant les fonctions T et L.

Le contrôleur actif peut abdiquer en faveur d'un autre contrôleur qui remplace le précédent.

#### Remarque

"System Controller" est la fonction normale d'un contrôleur. Pour lui ôter cette fonction, il faut basculer l'inverseur pour envoyer **rsc** (au contraire de ton et lon qui sont inactifs dans l'état normal).

Il y a donc sur le bus un seul contrôleur actif. Les autres se comportent comme des périphériques classiques. Pour engager l'un de ceux ci dans une conversation, le contrôleur doit lui faire parvenir son adresse.

## 2. Utilisation du bus et commandes classiques

Tous les périphériques ont un interface dont le rôle est de rendre compatible leurs fonctions et commandes internes (locales) et les informations circulant sur le bus.

### 2.1 Choix du mode local ou à distance

Pour permettre aux différentes communications d'avoir lieu, les appareils connectés au bus doivent souvent disposer de deux modes de programmation :

- Mode local, exemple : ton- rsc et lon déjà vus.
- programmation à distance, c'est le mode adressé réalisé par un message reçu par le bus.

Un appareil possédant la fonction RL (Remote Local) doit obligatoirement se trouver en mode local à sa mise sous tension. Tous les périphériques observent en permanence la ligne du bus de contrôle REN (Remote Enable).

Si cette ligne est active, après être adressés en "listeners". Ils passent en programmation à distance.

Si REN est inactif ( $\overline{\text{REN}}$ ), ils retournent tous en mode local. Dans les deux cas, un périphérique désigné dans un processus comme "talker" reste en mode local, mais bien entendu, ne parlera que s'il y a été autorisé par le contrôleur.

Remarque : A leur mise en route, la plupart des contrôleurs forcent REN actif.

## 2.2 Messages et commandes particulières

La mise en oeuvre de ces commandes s'obtient par la transmission de caractères ASC II pendant ATN (actif) qui sont donc susceptibles d'être acceptées par des périphériques présents.

La norme IEEE-488, inclut un certain nombre de commandes et messages représentés par des mnémoniques.

Les majuscules sont réservées aux commandes que reçoit une interface par l'intermédiaire du bus et les minuscules à celles qu'il reçoit depuis l'appareil lui-même.

### Il y a deux sortes de commandes

a- Les commandes unilignes : qui sont transportées par une seule ligne du bus de contrôle et elles ont le même mnémonique que la ligne qu'elles manoeuvrent.

Ce sont :

- Commande REN : Cette commande a été abordée au paragraphe précédent.

- Commande ATN : Pendant ATN (inactif), les informations qui circulent sur le bus sont des données ou des bytes d'état, pendant ATN (actif) ce sont des adresses ou des commandes multilignes universelles ou adressées.

Remarque : Pendant ATN (actif), tous les périphériques se comportent comme des listeners et toutes les interfaces observent en permanence toutes les lignes du bus de contrôle. Les autres messages unilignes sont étudiés dans la partie commandes évoluées.

b- Commandes multilignes

Ces commandes peuvent être acceptées par tous les périphériques (pendant ATN actif). Un périphérique peut ne pas comprendre tout ou une partie de ces commandes. Dans ce cas, il les ignorera. Ces commandes sont divisées en deux classes :

- \* UC (Universal Commandes) : auxquelles tous les appareils obéissent ; les commandes appartenant à cette classe sont les suivantes :

Commande DCL : (Device clear)

Remet tous les périphériques dans un état connu (en général le même que celui de la mise en route).

Commande LLO : (Local Lockout)

Lorsqu'un appareil est en mode telecommande, toutes ses touches sont verrouillées sauf celle qui fournit la commande locale **rtl** (return to local).

Par action locale sur "**rtl**", on peut remettre le mode local. Après la réception de LLO, la touche **rtl** devient elle aussi inefficace. Le périphérique ne peut retourner en local que par les commandes REN et GTL (Go to local) qui sont des commandes de classe AC, qu'on verra par la suite.

- \* Classe AC (Adressed Commandes) : auxquelles n'obéissent que les listeners actifs.

- . Commande GET : (Group execute trigger)

provoque le déclenchement de l'appareil adressé comme listener.

- . Commande GTL (Go to local) :

Remet en local l'appareil adressé (sans modifier sa configuration).

- . Commande SDC (Selected Device Clear)

La même que DCL mais pour les appareils adressés

### 3. ECHANGES D'INFORMATIONS SUR LE BUS IEEE-488

Les échanges se font sous le contrôle des lignes du bus de synchronisation, DAV (Data Valid), NRFD (Not ready for Data) et NDAC (Not data accepted).

Ces trois fils permettent aux appareils ne fonctionnant pas à la même vitesse d'échanger des informations.

Le transfert d'information s'ajuste automatiquement à la vitesse de l'appareil actif le plus lent sans que cela soit préjudiciable aux appareils plus rapides.

Le talker commande la ligne DAV et les huit lignes de données et les écouteurs, les lignes NRFD et NDAC.

Pendant la phase d'adressage : (ATN actif).

- Le contrôleur se comporte comme talker
- Tous les partenaires comme listeners ;
- Le talker potentiel se comporte comme listener actif. Il contrôle les lignes NRFD et NDAC. Il doit laisser hautes DAV et les lignes DIO.

Pendant la transmission du message : (ATN)

- Le contrôleur est suivant les circonstances, talker, listener ou indifférent à la communication.
- Le talker potentiel dans la phase d'adressage, une fois celle-ci terminée s'il est actif (i.e. : s'il a été désigné comme talker), doit remonter ses deux lignes NRFD et NDAC (qui deviennent des entrées) et contrôler les lignes DAV et DIO. S'il est passif, il doit remonter toutes les lignes du bus de données et du bus de handshake.

#### Principe de la synchronisation

Le transfert d'un caractère commence lorsque tous les listeners indiquent qu'ils sont prêts à le recevoir en plaçant NRFD au niveau haut. Le talker reconnaît ce signal, et place la caractère sur les lignes DIO puis signale sa présence aux listeners en forçant DAV au niveau bas.

Les listeners font l'acquisition de ce caractère et signalent

La fin de cette opération en libérant NDAC ( $\overline{\text{NDAC}}$  inactif, c'est-à-dire Data accepted).

Ils indiquent également qu'ils ne sont pas prêts pour la donnée suivante en forçant NRFD au niveau bas. Ils peuvent faire ceci dès qu'ils constatent DAV et, au plus tard, en même temps qu'ils relaxent NDAC.

Le talker constate que la donnée est acceptée. La donnée n'est donc plus nécessaire. Il remet DAV au niveau haut. Le talker attend  $\overline{\text{NRFD}}$  pour commencer le cycle suivant.

Les listeners reconnaissent  $\overline{\text{DAV}}$  et ils activent NDAC, ils peuvent le faire dès qu'ils constatent  $\overline{\text{DAV}}$ , et au plus tard en même temps qu'ils relaxent NRFD pour l'échange suivant (voir chronogramme des signaux de synchronisation).

La procédure pour cette raison est qualifiée d'asynchrone.

## Détail de la synchronisation

La figure suivante représente l'évolution des signaux observés sur le bus de Handshake lors de la transmission de plusieurs caractères à plusieurs récepteurs travaillant à des vitesses différentes.

1. La source initialise <sup>se</sup> DAV au niveau haut  $\overline{\text{DAV}}$  : donnée non valide
2. Les listeners activent NRFD et NDAC (actifs).
3.  $t-2$  le talker procède à un test sur NDAC et NRFD ; si ces deux lignes sont hautes, le transfert est interrompu et le contrôleur affiche un message d'erreur ; dans le cas contraire, le talker compose la donnée à transmettre ;
4.  $t-2 \rightarrow t_0$  le talker attend la stabilisation de la donnée ;
5.  $t-1$  les listeners sont tous prêts à accepter le premier caractère : NRFD est haut ;
6.  $t_0$  le talker valide la donnée (DAV actif) ;
7.  $t_1$  le listener le plus rapide place NRFD bas et il prend connaissance de la donnée : les autres listeners font de même à leur vitesse respective ;
8.  $t_2$  le premier listener relaxe sa ligne NDAC pour indiquer qu'il a accepté la donnée ; cependant la ligne NDAC reste au niveau bas jusqu'à l'étape suivante ;
9. Le dernier listener relaxe sa ligne NDAC : tous les récepteurs ont accepté la donnée et la ligne NDAC passe au niveau haut ;
10.  $t_3$  le talker constate NDAC et invalide la donnée par  $\overline{\text{DAV}}$
11.  $t_4-7$  le talker compose une nouvelle donnée ;
12.  $t_7-9$  le talker attend la stabilisation de la donnée ;
13. Les listeners constatent  $\overline{\text{DAV}}$  (10) et réactivent NDAC qui devient actif dès que le plus rapide d'entre eux actionne ;
14.  $t_5$  le listener le plus rapide indique qu'il est prêt pour la nouvelle donnée en libérant NRFD ; cependant cette ligne reste au niveau bas ;
15.  $T_3$  le dernier listener <sup>libère</sup> NRFD qui passe alors au niveau haut ;

16. t<sub>9</sub> le talker ayant constaté  $\overline{\text{NRFD}}$  force DAV pour valider la donnée ;
17. t<sub>10</sub> le premier listener place NRDF bas (idem 7)
18. t<sub>11</sub> le premier listener libère NDAC (idem 8)
19. t<sub>12</sub> le dernier listener libère NDAC (idem 20)
20. t<sub>13</sub> le talker ayant constaté  $\overline{\text{NDAC}}$  relaxe DAV (idem 10) ;
21. Le talker compose la donnée suivante (idem 11) ;
22. t<sub>14</sub> les récepteurs constatent  $\overline{\text{DAV}}$  et activent NDAC (idem 13)
23. Les lignes NDAC, NRFD, DAV sont dans l'état initial (idem 1, 2 et 13).

En constatant  $\overline{\text{ATN}}$  (fin de transmission des adresses et des commandes).

Les appareils non adressés doivent relaxer toutes les lignes du bus de données et du bus de synchronisation.

Un appareil signale qu'il a accepté la donnée en libérant NDAC après avoir activé NRFD. Si cette donnée est la dernière de la phase d'adresse, le contrôleur émet  $\overline{\text{ATN}}$  et si cet appareil n'est pas actif, il devra à nouveau libérer sa ligne NRFD. Aucune limite de temps n'est imposée.

Nous avons dit que pendant la phase d'adressage, tous les appareils se comportent comme des listeners.

C'est en réalité un abus de langage destiné à faciliter la compréhension. Un listener est en réalité un appareil qui écoute un message (donc pendant  $\overline{\text{ATN}}$  seulement).

Pendant ATN, il est seulement accepteur dans les deux cas, c'est la fonction AH (acceptor-Handshake) qui gère les lignes NRFD et NDAC. De la même façon, c'est la fonction SH (Source Handshake) qui gère la ligne DAV pour le compte du talker.

#### 4. UTILISATION DU BUS ET COMMANDES EVOLUEES

Les techniques classiques d'utilisation du IEEE - 488 viennent d'être développées. Dans ce qui suit, vont être décrites ; les formes d'utilisation particulières et évoluées prévues dans le standard.

- Adressage étendu ;
- **Listeners** multiples ;
- Messages normalisés ;
- Commandes multilignes ;
- Gestion automatique d'interruption ;
- Communication sur un bus dépourvu de contrôleurs ;
- Communication directe entre périphériques sans la participation du contrôleur ;
- Communication entre calculateurs.

Les différentes communications sur le bus peuvent être considérées comme des commandes évoluées. Elles ont été détaillées précédemment.

##### 1. Adressage étendu

En plus de leur adresse normale (primaire), les partenaires peuvent avoir une ou plusieurs adresses secondaires, attribuées soit à la fonction talker (TE : Talker Extended), soit à la fonction listener (LE : Extended Listener). Deux appareils quelconques peuvent avoir la même adresse primaire et n'être différenciés que par leur adresse secondaire.

Sur un même bus, si tous les partenaires (14 au maximum) utilisaient toutes les adresses secondaires possibles, le contrôleur pourrait adresser individuellement :  $14 \times 31 = 434$  destinataires.

Pour une adresse secondaire donnée, une imprimante pourrait par :

- imprimer tout en majuscules ;
- pour une autre en minuscule (ou en rouge) ;
- et pour une troisième faire du semi-graphique.

Les adresses secondaires SCG (Secondary Group Command) sont

constituées comme les adresses primaires mais les bits 6 et 7 sont à 1 (le bit 8 à 0).

## 2. Listeners multiples

- Listeners multiples, le contrôleur étant lui-même talker actif :  
Ce mode opératoire bien que peu courant n'est pas sans intérêt. Au cas où un périphérique ne répond pas correctement à la programmation, en faisant parvenir le même message en même temps à deux listeners, par ex. : l'imprimante et générateur, on verra plus facilement si son contenu est correct ou non.
  
- Listeners multiples, le contrôleur étant lui-même listener actif  
Le contrôleur désigne un seul talker et peut en plus de lui-même désigner des listeners.

## 3. Messages normalisés

### a) Messages unilignes

Message END : appelé EOI. Il est transporté par la ligne EOI (End or Identify). La plupart des talkers ont la possibilité d'émettre le message END (il faut parfois les y autoriser). Normalement, ils le font en même temps qu'ils actionnent DAV pour le dernier caractère à transmettre, signalant aux listeners actifs qu'il s'agit de la dernière donnée.

Message SRQ : c'est une ligne du bus de contrôle et la seule qui ne puisse être manoeuvrée par le contrôleur actif. Ce sont les périphériques possédant les fonctions T et SR (Service Request). Ceux-ci peuvent émettre à n'importe quel instant le message SRQ pour attirer l'attention du contrôleur, qui n'est au courant que s'il lit la ligne SRQ, ce qu'il doit faire périodiquement dans un programme au cours duquel peut survenir une demande de service.

### b) Commandes multilignes

Commande SPE : (Serial Poll enable) et SPD (Serial poll disable) classe UC.

SPE : informe tous les partenaires ayant la fonction SR et T sont capables de répondre à un serial poll (le Serial et parallel poll sont expliqués dans III) qu'ils doivent substituer leur status (octet) à leur réponse normale.

SPD : Retour au fonctionnement normal.

Commandes PPC (parallel poll configure) et PPU (parallel poll inconfigure).

PPC : Classe AC

PPU : Classe UC

PPC : indique à l'appareil adressé (listener) que l'octet qui suit un PPE (parallel poll enable) est sa configuration pour un parallel poll.

PPU : interdit à tous les appareils de répondre à un parallèle poll.

Commande DY (Identify) classe UC

Cette commande est réalisée par ATN et EOI simultanément.

Cette commande provoque un parallel poll.

Commande TCT (Take control) : Classe AC

A destination d'un appareil ayant la fonction C.

Le contrôleur présentement actif passe le contrôle à celui qu'il adresse.

#### 4. Gestion automatique d'interruption

l'interface IEEE-488 s'occupe seule du handshake. Elle peut également (si elle y est autorisée) interrompre le déroulement normal du programme. Citons deux exemples :

- interruption par SRQ
- interruption par fin de communication.

## 5. Fonctions disponibles dans la norme IEEE-488

La plupart des appareils connectés sur le bus ne comprennent qu'une partie des fonctions prévues par la norme IEEE-488. Ces dernières sont assez souvent décrites dans une phrase imprimée sur la face arrière de l'appareil près du connecteur, et qui indique très précisément ce qu'ils savent faire. Toutes les fonctions existent sous plusieurs variantes, affectées d'un numéro ; Ex. : Co indique que l'appareil ne peut pas assurer le rôle de contrôleur.

### 1. Fonction SH (Source Handshake) :

Cette fonction indique la capacité à assurer correctement l'émission de messages multilignes.

### 2. Fonction AH (Accepteur Handshake)

Cette fonction indique la capacité à recevoir correctement les messages multilignes.

### 3. Fonction T (Talker)

Cette fonction indique la capacité à faire parvenir à d'autres appareils des données et des statuts.

### 4. Fonction TE (Extended Talker)

Les fonctions TE sont identiques aux fonctions T (sauf adressage sur 2 octets).

### 5. Fonction L (Listener)

Cette fonction indique la capacité à recevoir d'autres appareils des données et des statuts ;

Les fonctions L nécessitent AH 1 (totale capacité d'accepter la poignée de main). Deux appareils listeners seulement (TO) peuvent avoir la même adresse.

6. Fonction LE (Extended listener)  
Les fonctions LE sont identiques aux fonctions L (sauf adressage sur 2 octets).
7. Fonction SR (Service Request)  
Cette fonction indique la capacité d'un appareil à émettre le message **SRQ** et à le retirer après un Serial poll.
8. Fonction RL (Remote Local)  
Cette fonction indique la capacité à choisir entre les deux sources de programmation (Locale ou Télécommande).
9. Fonction PP (Parallel Poll)  
Cette fonction indique la capacité à répondre à un parallel poll.
10. Fonction DC (Device Clear)  
Cette fonction indique la capacité à être initialisé (DCL : Device Clear ou SDC : Selected Device Clear. Ces deux commandes multilignes classiques ont été explicitées).
11. Fonction DT (Device Trigger)  
Cette fonction indique la capacité à entreprendre une opération à la réception de GET.
12. Fonction C (Controller)  
Cette fonction indique la capacité à être contrôleur (envoyer des adresses, des commandes universelles UC, adressées AC et conduire un parallel Poll).
13. Fonction E  
Cette fonction indique la nature des transceivers de sortie. S'ils sont à collecteurs ouverts ou transceivers à trois états.

### III. PRESENTATION DU GPFA (General purpose interface Adapter)

C'est un circuit LSI à 40 broches qui fonctionne en logique positive. Le GPFA MC 68488 a été conçu pour servir d'interface entre les microprocesseurs de la famille 6800 et le bus instrument IEEE-488. Plusieurs fonctions du bus sont gérées automatiquement par le GPFA et ne demandent aucune action additionnelle du MPU. D'autres fonctions demandent une réponse minimum due au grand nombre de registres internes contenant l'information sur l'état du GPFA et le bus instrumental. Le GPFA est conçu pour travailler avec les drivers du bus IEEE-488 (MC 3447 ou MC 3448).

#### 1. Fonctions implémentées par le GPFA

Le GPFA implémente directement les fonctions suivantes :

- Emetteur et émetteur étendu (T ou TE) ;
- Récepteur et récepteur étendu (L ou LE) ;
- "Talk only" et "listen only" ;
- recherche série et parallèle (FP) d'une demande d'intervention ;
- Synchronisation émetteur et récepteur (SH, AH) handshake ;
- Trigger (GET), remise à zéro (DOL ou SDC) et programmation à distance (RL) ou locale ;
- adressage primaire unique ou double ;
- adressage secondaire.

Il assure seul (sans intervention du microprocesseur) le "handshake" pendant la phase d'adressage.

Pendant la transmission des données, il est capable de gérer aussi

tout seul les lignes NDAC et NREFD en listener et DAV en talker, mais la façon dont il opère peut être modifiée par "SOFT".

Exemple :

Le GPIA n'interprète pas certaines commandes. Il doit donc les ignorer mais assurer correctement le handshake. Il s'agit de commande, tous les périphériques doivent en prendre connaissance. Le GPIA est parfaitement capable d'adapter cette attitude. Mais le microprocesseur peut, dans une phase de mise en route par exemple, la configurer pour qu'il réagisse d'une autre façon à la réception d'une commande de classe AC ou UC qu'il ne comprend pas, le GPIA interrompt le handshake et présente (dans le registre R6R) : Registre des commandes Pass - Through), la commande "insolite" au microprocesseur.

Celui-ci en prend connaissance, décide de l'utilisation qu'il en fait, puis indique à son interface de poursuivre le handshake et et d'acquérir le caractère suivant. Par cette procédure, le GPIA peut directement avec l'intervention du MPU, réaliser toutes les fonctions qu'il n'implémente pas seul (sauf contrôleur).

De façon analogue, le GPIA peut abandonner le handshake automatique pour tout ou partie des données et attendre l'autorisation du microprocesseur pour manoeuvrer les lignes NDAC et NREFD.

## 2. Signaux du GPIA

Toutes les entrées du GPIA sont à haute impédance et compatibles T.T.L.

Toutes les sorties du GPIA sont aussi compatibles T.T.L. Cependant

$\overline{\text{IRQ}}$  (Interrupt request)  $\overline{\text{SRQ}}$  sont des sorties à collecteur ouvert.

## 2.1 MPU avec l'interface

### Bidirectionnel DATA (D0 - D7)

Les lignes de données bidirectionnelles permettent le transfert des données entre le MPU et le GPIA et vice versa. Ces lignes sont amplifiées par des buffers à 3 états le sens de fonctionnement de ces buffers sera explicité dans le chapitre IV. La ligne R/W est à l'état haut quand le GPIA est sélectionné pour une opération de lecture.

### Chip Select ( $\overline{CS}$ )

Ce signal d'entrée est utilisé pour sélectionner le GPIA.  $\overline{CS}$  doit être bas pour sélectionner le GPIA. Le décodage du "Chip Select" est effectué par une logique externe qui est expliquée dans l'étude de la carte GPIB.

### Read/Write input (R/ $\overline{W}$ )

Ce signal est généré par le MPU ou le contrôleur de **DMA** (Direct Memory Access) pour contrôler l'accès aux registres et la direction du transfert de données sur le bus de données. Un niveau bas sur la ligne R/ $\overline{W}$  et DMA Grant permettant la sélection d'un registre parmi les sept registres à écriture seulement, quand ils sont utilisés en conjonction avec les lignes de sélection des registres RS0 - RS1 et RS2. Un niveau haut sur la ligne R/ $\overline{W}$  et bas sur le DMA Grant permettant la sélection d'un **parmi** les huit registres à lecture seulement, quand ils sont utilisés avec les lignes de sélection de registres RS0 - RS1 et RS 2.

Ainsi, l'entrée DMA Grant est reliée à la masse, sur la carte GPIB.

Register Select : (RS0 - RS 1 - RS2)

Ces 3 entrées sont utilisées pour sélectionner les différents registres internes du GPIA. Ces trois lignes sont utilisées en conjonction avec la ligne R/W pour sélectionner un registre particulier qui va être lu ou écrit.

La table suivante montre le codage de sélection de registre :

| RS2 | RS1 | RS0 | R/W | Titre des registres      | Symbole |
|-----|-----|-----|-----|--------------------------|---------|
| 0   | 0   | 0   | 1   | Interrupt status         | R0R     |
| 0   | 0   | 0   | 0   | Interrupt Mask           | R0W     |
| 0   | 0   | 1   | 1   | Command Status           | R1R     |
| 0   | 0   | 1   | 0   | Unused                   | R1W     |
| 0   | 1   | 0   | 1   | <b>Address Status</b>    | R2R     |
| 0   | 1   | 0   | 0   | <b>Address Mode</b>      | R2W     |
| 0   | 1   | 1   | 1   | <b>Auxiliary Command</b> | R3R     |
| 0   | 1   | 1   | 0   | <b>Auxiliary Command</b> | R3W     |
| 1   | 0   | 0   | 1   | <b>Address Switch</b>    | R4R     |
| 1   | 0   | 0   | 0   | <b>Address</b>           | R4W     |
| 1   | 0   | 1   | 1   | Serial poll              | R5R     |
| 1   | 0   | 1   | 0   | Serial poll              | R5W     |
| 1   | 1   | 0   | 1   | Command Pass through     | R6R     |
| 1   | 1   | 0   | 0   | Parallel poll            | R6W     |
| 1   | 1   | 1   | 1   | Data in                  | R7R     |
| 1   | 1   | 1   | 0   | Data out                 | R7W     |

### Interrupt request ( $\overline{\text{IRQ}}$ )

La sortie  $\overline{\text{IRQ}}$  va à la ligne d'interruption du MPU. C'est une sortie à collecteur ouvert.  $\overline{\text{IRQ}}$  est maintenue à un niveau bas quand une interruption arrive et reste à l'état bas jusqu'à ce que le MPU lit le registre d'état d'interruption. La lecture de RCR remet  $\overline{\text{IRQ}}$  à l'état haut.

### RESET :

L'entrée  $\overline{\text{RESET}}$  permet d'initialiser le GPIA. A l'état bas,  $\overline{\text{RESET}}$  cause :

1. Réinitialisation de toutes les conditions d'état ;
2. L'initialisation du registre masque d'interruptions ;
3. Place le GPIA dans l'état "Untalk/Unlisten" ;
4. Réinitialisation des registres "Parallel poll, Serial poll, Data in et Data out".
5. Efface les registres adresse et mode d'adressage ;
6. Reinitialise toutes les conditions dans le registre de commandes auxiliaires à l'exception du bit 7 ;
7. Remet à l'état bas T/R1,2 ;

Quant  $\overline{\text{RESET}}$  revient à l'état haut (état inactif) le GPIA reste dans l'état  $\overline{\text{RESET}}$  jusqu'à ce que le MPU écrit le bit 7 du registre de commandes auxiliaires à l'état bas. Avant le déclenchement du programme de bit reset, le seul registre auquel on peut accéder est le registre adresse.

Les conditions affectées par  $\overline{\text{RESET}}$  ne peuvent être changées pendant que cette pin est à l'état bas.

## E (Enable Clock) $\emptyset$ 2

E active les entrées adresses (CS, RSO, RS1 et RS2) et  $R/\overline{W}$ .

Il active aussi le transfert de données sur le bus de données.

## 2.2 GPIA -- GPIB

Il y a 18 signaux d'interface entre le MC 6800 et le bus IEEE-488.

Comme, il a déjà été cité, le standard IEEE-488 définit ces signaux en logique négative. Les signaux du MPU et du MC 68488 sont définis en logique positive.

### Signaux ( $\overline{IB0}$ -- $\overline{IB7}$ )

Ces lignes bidirectionnelles permettent l'échange de message entre le GPIA et les transceivers. Les données apparaissent sur ces lignes en bits parallèles, byte serie. Elles sont amplifiées par des transceivers puis appliquées au bus IEEE 488 (DIO 1-DIO 8)

### Signaux de synchronisation (DAC -- RFD -- $\overline{DAV}$ ) :

Le principe de la synchronisation a été déjà expliqué (II.3)  
Ainsi ces lignes permettent le transfert de données entre sources et accepteurs.

### Lignes de contrôle ( $\overline{ATN}$ , $\overline{IPC}$ , $\overline{SRQ}$ , $\overline{EOI}$ , $\overline{RET}$ )

Ces lignes sont utilisées pour la gestion de l'interface.

#### Attention ( $\overline{ATN}$ )

Cette ligne est gérée par le GPIA. Le GPIA répond au changement sur cette ligne en moins de 200 ns, en activant les signaux de contrôle, de transmission/Réception TR1,2.

Si les lignes  $\overline{\text{EOI}}$  et  $\overline{\text{ATN}}$  sont à l'état bas en même temps, le GPIA place le contenu du "parallel poll" sur le bus  $\overline{\text{IEEE-488}}$ .

#### SRQ - $\overline{\text{EOI}}$ - $\overline{\text{REN}}$

Ces signaux ont été déjà vus dans le chapitre II-2.

#### Interface Clear ( $\overline{\text{IFC}}$ ) : Initialisation des périphériques

Cette ligne est issue du contrôleur du système pour mettre le GPIA dans un état de repos connu. L'arrivée de  $\overline{\text{IFC}}$  place le GPIA dans l'état de repos listener/Talker. (LIDS : listen idle state on TIDS : Talk idle state). Si le KC 68488 est dans l'état actif du listener avec un byte dans le registre Data in (bit BI à 1)  $\overline{\text{IFC}}$  place le GPIA dans LIDS mais n'affecte pas le byte reçu, ni l'état indicateur (BI). Chaque fonction d'interface qui demande au périphérique d'être dans l'état actif de listener ou de talker (ex. : Serial poll enable command) sera initialisé si  $\overline{\text{IFC}}$  arrive. Une commande générée par le MPU (ex. : to, lo, fget, heda) sera affectée seulement durant l'arrivée de  $\overline{\text{IFC}}$  (quand  $\overline{\text{IFC}}$  est bas) et retourne à son état programmé initial quand  $\overline{\text{IFC}}$  est haut ie  $\overline{\text{IFC}}$  n'affecte pas les messages locaux. Par exemple : si le GPIA est dans l'état TACS (Talker Active State) et a placé un byte dans le data Out Register, il a fait un nouveau byte disponible (nba). Si  $\overline{\text{IFC}}$  arrive pendant que la source du Handshake est en SOYS (Source Delay State), la fonction talker retournera à son état de repos ; mais le (nba) message local ne sera pas détruit. Quand le GPIA est dans l'état de talker, le byte placé dans le data-out register

(placé avant l'arrivée IFC) sera sur le GPIB. Le registre d'adresse n'est pas affecté par  $\overline{IFC}$ .

Le signal EOI est utilisé en conjonction avec ATN comme l'indique le tableau

| ATN | EOI |  |
|-----|-----|--|
| 0   | 0   | Octet de données                                       |
| 0   | 1   | Dernier octet d'un bloc<br>(fin de bloc)               |
| 1   | 0   | Adresse  |
| 1   | 1   | Demande d'identification après<br>une interruption SRQ |

#### Transmit Receive Control signals ( $T/\overline{R}1$ - $T/\overline{R}2$ )

Ces deux signaux sont utilisés pour le contrôle des transmetteurs qui véhiculent les signaux des bus.

La ligne SRQ est câblée à l'état haut pour transmettre.

Les entrées transmit/Receive de  $\overline{REN}$ ,  $\overline{IFC}$  et  $\overline{ATN}$  sont câblées à l'état bas pour recevoir.  $T/\overline{R}1$  contrôle la ligne  $\overline{EOI}$  à travers MC 3447 lui permettant d'émettre ou de recevoir.  $T/\overline{R}1$ , travaille exactement comme  $T/\overline{R}2$  sauf durant la séquence du parallel poll. Durant le "parallel poll", EOI sera dirigée en entrée par  $T/\overline{R}1$  pendant que  $\overline{DAV}$  et  $\overline{IBC}$  -  $\overline{IB7}$  seront des sorties.

#### Spécial Controls Signals

##### DMA Control lines (DMA Grant, DMA Request)

La ligne "DMA request" est utilisée pour signaler au contrôleur

de DMA que le transfert de DATA est en cours. "DMA request" est à l'état haut si chacun des bits BI ou BO est mis à 1 dans le registre ROR.

La ligne DMA request est effacée quand le DMA Grant est vrai. La ligne "DMA Grant" est utilisée pour signaler au GPFA que le DMA a le contrôle des lignes de données et d'adresses du MPU. Le "DMA Grant" mis à 1 sélectionne le registre 7. Cette ligne déconnecte les lignes RSO, RS1 et RS 2. Durant ce temps, CS doit être à l'état haut. Le DMA Grant inverse la fonction du R/W en  $\overline{R/W}$ . Aussi, si le contrôleur de DMA demande l'écriture dans la mémoire, cette même ligne fait une opération de lecture du GPFA (R7R) et vice versa.

Note : "DMA Grant" doit être mise à la masse quand elle n'est pas utilisée.

#### Trigger out - put : TRIG :

La ligne TRIG est une sortie correspondant aux commandes GET et fget (Force Group execute command). L'arrivée du RESET place cette sortie au niveau bas. La sortie Trigger peut-être programmée pour être à l'état haut par l'une des deux méthodes :

1. Mise à un du bit fget (bit 0 de R3W) par le MPU implique la mise à un de la sortie trigger. Cette sortie reste à cet état jusqu'à ce que le bit fget soit programmé à l'état bas ou qu'un reset arrive.

2. La sortie trigger est mis à un dès la réception d'une commande GET du contrôleur. Elle est remise à l'état initial quand le GPIA sort de l'état DTAS (Device Trigger active State) c'est à dire quand les commandes GET, LADS (Listener addressed State : état de l'écouteur adressé). ACDS (Accept data State) arrivent.

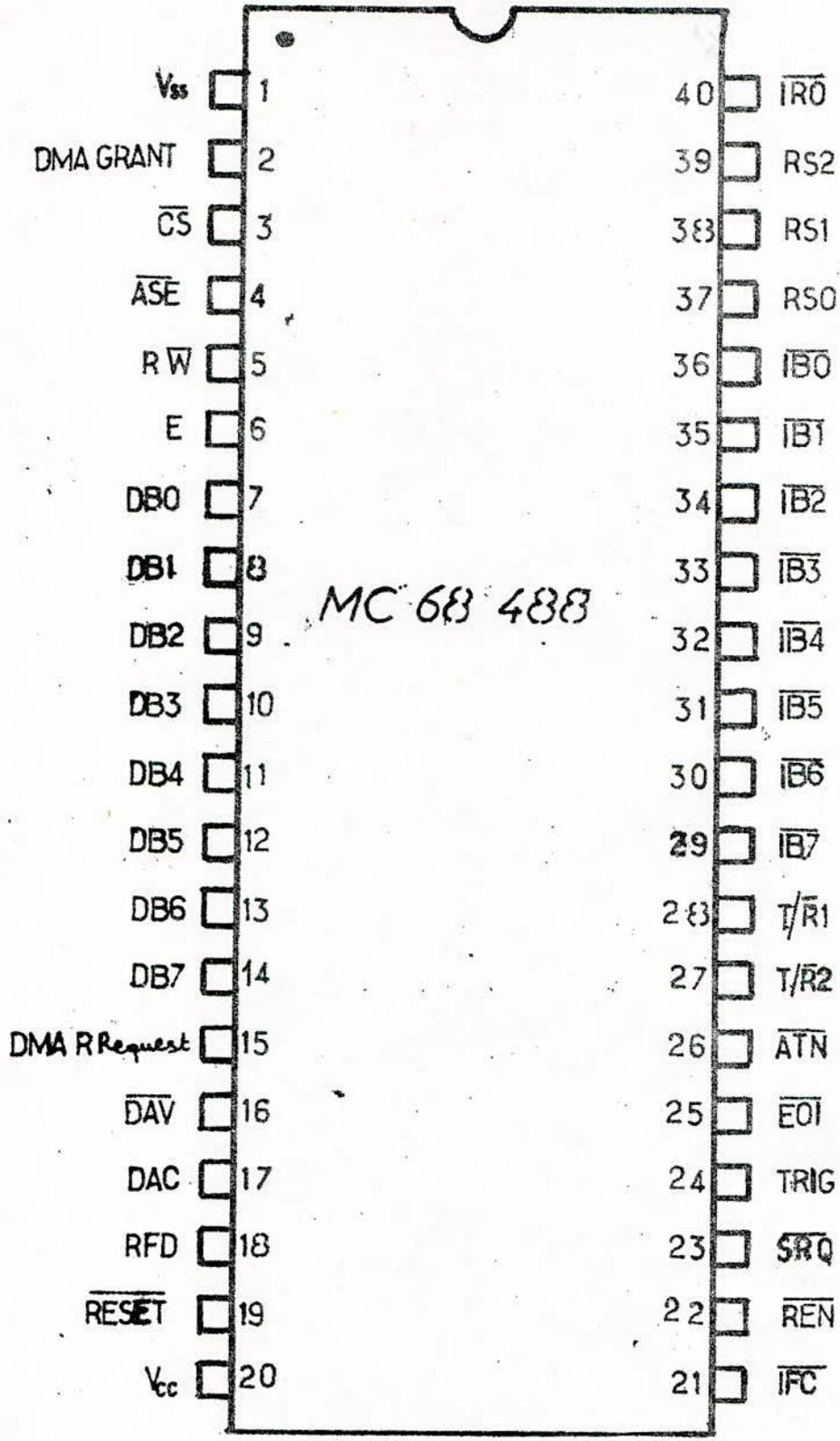
#### Adress Switch enable ( $\overline{ASE}$ )

La sortie  $\overline{ASE}$  est utilisée pour activer un buffer trois états qui connecte l'adresse de l'instrument fixée par les micro-interrupteurs au Data bus du MPU.

Cette sortie est mise à zéro quand le registre R4R du GPIA est lu. ie la lecture de R4R met la ligne  $\overline{ASE}$  à l'état bas.

#### Brochage du GPIA

Nous indiquons sur le schéma, pour mémoire, le brochage du GPIA.



### Les registres du GPIA

Le MC 68488 possède quinze registres, tous accessibles par le microprocesseur qui permettent de stocker toutes sortes d'informations (données, états, commandes). Sept sont utilisés uniquement en écriture et permettent la programmation du 68488 par le MPU ; huit travaillent seulement en lecture et traduisent au MPU tout ce qui a trait au bus IEEE-488. L'accès aux différents registres est effectué par les trois lignes d'adresse de poids faibles (A0, A1, A2) et la ligne R/W.

Le registre R4R, registre de sélection d'adresse est placé à l'extérieur du GPIA. Ainsi, nous avons la possibilité de sélectionner l'adresse par des micro-interrupteurs ou par logiciel en écrivant directement dans le registre R4W.

### Les registres de données R7R (Data in register), R7W (Data out register)

Ce sont des registres de stockage à huit bits. Lorsque le GPIA est listener actif, il place l'octet qu'il reçoit dans le registre R7R où le MPU peut le lire. Le seul fait de lire R7R libère NDAC puis NRFD dès que possible (Si hede ou heda sont à 1, le GPIA ne libère NRFD que sur ordre du MPU par le bit 6 de R3W). Le MPU place (écrit) dans le registre R7W les données qu'il veut transmettre sur le bus.

|     |     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| R7R | D17 | D16 | D15 | D14 | D13 | D12 | D11 | D10 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|

D10-D17 correspondent aux  $\begin{cases} \text{DIO1-DIO8 sur le IEEE-488} \\ \text{IB0-IB7 sur le MC 68488} \end{cases}$

|     |     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| R7W | DO7 | DO6 | DO5 | DO4 | DO3 | DO2 | DO1 | D00 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|

D00 - DO7 Correspondent aux  $\overline{DIO1}$ - $\overline{DIO8}$  sur le  $\overline{LEEE}$  - 488  
 $\overline{IB0}$ - $\overline{IB7}$  sur le MC 68488

Le registre masque interruption (interrupt status register) ROW

C'est un registre de stockage à 7 Bits utilisé pour sélectionner les interruptions qui vont être envoyées au MPU. Ces bits peuvent être positionnés indépendamment l'un de l'autre.

|     |     |    |     |   |     |     |     |    |
|-----|-----|----|-----|---|-----|-----|-----|----|
| ROW | IRQ | BO | GET | X | APT | CMD | END | BI |
|-----|-----|----|-----|---|-----|-----|-----|----|

IRQ : bit de masque pour le pin IRQ

BO : Interruption sur un byte sortant

GET : Interruption à l'arrivée d'une commande CPT

X : bit inutilisé

APT : Interruption sur une adresse secondaire pass Through si apte (bit 0 de R2W) est actif.

CMD : Interruption sur  $\overline{EFAS}$  + RLC +  $\overline{dsel}$  ( $\overline{DCAS}$  +  $\overline{UAC}$  +  $\overline{UUCG}$ ) +  $\overline{dsel}$  (bit 7 de R2W). Le registre commandes auxiliaires RIR peut être utilisé pour déterminer quelle commande a causé cette interruption.

END : interruption si  $\overline{ECI}$  est actif (bas) et  $\overline{AINT}$  inactif (haut)

BI : Interruption sur un byte entrant

Le registre état interruption (Interrupt status register) ROR.

C'est un registre de stockage à 7 bits qui correspondent aux bits de RCW sauf le bit 7 INT. A l'exception de ce dernier, les autres prennent le même état que celui des bits de ROW quand les

mêmes interruptions arrivent.

|     |     |    |     |   |     |     |     |    |
|-----|-----|----|-----|---|-----|-----|-----|----|
| ROR | INT | EO | GET | X | APT | CMD | END | BI |
|-----|-----|----|-----|---|-----|-----|-----|----|

INT : OR logique des AND des 6 autres bits avec les bits correspondants de ROW.

EO : mis à 1 lorsque le registre R7W est vide, le MPU peut alors y écrire l'octet qu'il destine au bus IEEE-488.

CMD : indique que l'interface a reçu une commande qu'elle n'a pas comprise.

BI : mis à 1 lorsque l'interface (listener actif) a reçu un octet.

GET : positionné à l'arrivée de EOI (actif) et  $\overline{ATN}$  inactif (haut)

APT : positionné à l'arrivée d'une adresse secondaire en vue d'être examiné par le MPU.

$\overline{IRQ}$  est effacée à la lecture de ce registre.

#### Le registre sondage série (Serial poll register) R5R/W

C'est un registre de stockage à 8 bits. Il peut être lu et écrit par MPU. Il est utilisé pour établir le byte d'état qu'un périphérique envoie quand il est activé pendant un sondage série. Le byte d'état sera placé dans les bits de 0 à 5 et le bit 7. Le bit 6 rsv (request for service) est utilisé pour commander la logique qui contrôle la ligne  $\overline{SRQ}$  sur le bus indiquant au contrôleur qu'un service est demandé. Cette même logique génère le signal SRQS qui est substitué dans le bit 6 à rsv, quand le byte d'état est lu par le MPU

( $\overline{IB0}$  -  $\overline{IB7}$ ). Pour initialiser un rsv, le MPU met le bit 6 à 1 et ceci oblige le périphérique demandant un service à mettre bas la ligne  $\overline{SRQ}$ . SRQS est le même que rsv quand le bit SPAS est inactif. Le bit 6 étant lu par le MPU devient le bit SRQS (service request state).

|     |    |      |    |    |    |    |    |    |
|-----|----|------|----|----|----|----|----|----|
| R5R | S8 | SRQS | S6 | S5 | S4 | S3 | S2 | S1 |
|-----|----|------|----|----|----|----|----|----|

S1 - S8 bits d'états

SRQS - Bus dans service request state

|     |    |     |    |    |    |    |    |    |
|-----|----|-----|----|----|----|----|----|----|
| R5W | S8 | rsv | S6 | S5 | S4 | S3 | S2 | S1 |
|-----|----|-----|----|----|----|----|----|----|

S1 - S8 bits d'états

rsv génère un service request

#### Registre sondage parallèle (parallel poll register) R6W

Ce registre va être chargé par le MPU et ses bits seront envoyés instruments connectés au bus durant PPAS.

Le GFIA peut répondre à un "parallel poll" mais sa réponse ne peut pas être configurée par le contrôleur. Il n'interprète donc pas FPC, PPE etc... Mais ceci peut être fait par le MPU comme il a déjà été expliqué dans la partie III-1.

|     |     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| R6W | PF8 | PF7 | PF6 | PF5 | PF4 | PF3 | PP2 | PP1 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|

#### Le registre mode adressage (Address mode register) R2W

C'est un registre de stockage avec 6 bits de contrôle

|      |     |     |   |      |      |   |      |
|------|-----|-----|---|------|------|---|------|
| dsel | ton | lon | X | hlde | hlda | X | apte |
|------|-----|-----|---|------|------|---|------|

- tcn et lon ont la signification habituelle
- apte autorise le GFIA à présenter au MPU pour interpréter les commandes qu'il ne comprend pas.
- hlde : met RFD en ALL DATA tant que rfdr n'est pas actif
- hlde : actif à la transmission du dernier byte
- dsel : permet l'accomplissement du handshake à l'arrivée d'une commande UC ou AC.
- X : bit unitilisé

Le registre état adresse (Address status register) R 2R

Il contient les indicateurs d'état concernant le GFIA

|    |    |    |     |      |      |      |      |
|----|----|----|-----|------|------|------|------|
| ma | to | lo | ATN | TACS | LACS | LPAS | TPAS |
|----|----|----|-----|------|------|------|------|

ma : mis à 1 si le GFIA est dans un état de : TACS, TADS, LACS  
LADS, SPAS.

To : Talk only est activé

lo : listen only est active

ATN : la commande attention est activée

TACS : Le GFIA est dans l'état talker active state

LACS : le GFIA est dans l'état listener active state

LPAS : Le GFIA est dans l'état listener primary addressed state

TPAS : Le GFIA est dans l'état talker primary addressed state

Le registre adresse R4W et sélection adresse R4R

|     |      |     |     |     |     |     |     |     |
|-----|------|-----|-----|-----|-----|-----|-----|-----|
| R4W | lsbe | dal | dat | AD5 | AD4 | AD3 | AD2 | AD1 |
|-----|------|-----|-----|-----|-----|-----|-----|-----|

Address register

le GFIA prend connaissance de son adresse par logiciel, le

MPU la lui indique en l'écrivant dans le registre R4W (bits 0 à 4).

Par la même occasion , il peut :

- supprimer la fonction talker  $dat = 1$
- supprimer la fonction listener  $dal = 1$
- autoriser le double adressage primaire  $lsbe = 1$

$lsbe = 1$ , l'interface répond indifféremment aux adresses  $2N$  et  $2N + 1$  où  $N$  est codé sur  $AD2$  à  $AD5$ . L'adresse de l'interface peut-être modifiée par matériel (au moyen de micro-interrupteurs).

Le registre R4R est prévu à cet effet, il est en réalité constitué par huit micro-interrupteurs connectés sur le bus interne par l'intermédiaire d'un driver 3 états. Lorsque le MPU <sup>111</sup> ~~lit~~ R4R, le GPIA active la ligne  $\overline{ASE}$  qui commande l'ouverture du driver sur le bus de données qui peut ainsi en prendre connaissance :

|     |     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| R4R | UD3 | UD2 | UD1 | AD5 | AD4 | AD3 | AD2 | AD1 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|

#### Address register Switch

Les bits 0 à 4 constituent l'adresse choisie pour l'interface, les bits 5,6 et 7 (UD 1 à UD 3) sont à la disposition de l'utilisateur.

L'utilisation la plus probable de ces bits est ton, lon et rtl ou rsv.

#### Registre commandes auxiliaires R3R /W (auxiliary command)

|     |       |      |      |     |     |     |      |      |
|-----|-------|------|------|-----|-----|-----|------|------|
| R3W | reset | rfdr | feoi | dac | msa | rtl | dacd | fget |
|-----|-------|------|------|-----|-----|-----|------|------|

reset : initialise le GPIA

rfdr : permet l'accomplissement du Handshake interrompu par hlde et hlda

feoi : met ECI à l'état bas

dacr : active DAC, indique que le MPU a examiné une adresse  
secondaire ou une commande de classe UC.

msa : met le C<sup>A</sup>PIA dans l'état adressé de Talker ou parleur  
(TFAS ou LPAS)

rtl : retour au contrôle local

dacd : prévient l'accomplissement d'un Handshake concernant les  
adresses et les commandes.

fget : agit comme la commande GET du contrôleur

|     |       |     |     |     |     |     |      |      |
|-----|-------|-----|-----|-----|-----|-----|------|------|
| R3R | reset | DAC | DAV | RFD | msa | rtl | ulpa | fget |
|-----|-------|-----|-----|-----|-----|-----|------|------|

DAC, DAV, RFD prennent les mêmes états que les signaux corres-  
pondants sur le 68488. Ils sont lus par le MPU.

ulpa : indique l'état du bit de poids faible de l'adresse reçue  
sur les DIO1-8 au moment où la dernière adresse primaire  
est reçue.

Les autres bits ont été vus dans R3W

Le registre état commandes (Command status register) R1R

Il contient 7 bits indicateurs d'état.

|     |      |     |     |   |     |      |      |      |
|-----|------|-----|-----|---|-----|------|------|------|
| R1R | VACG | REM | LOK | X | RLC | SPAS | DCAS | UUCG |
|-----|------|-----|-----|---|-----|------|------|------|

VACG : Commande d'adresse indéfinie a été reçu attendant son  
exécution ou non par le MPU.

REM : Montre l'état Remote/Local du talker/listener

LOK : état local verrouillé du talker/Listener

RLC : se met à 1 quand un changement sur l'état de Remote/local arrive et sera remis à 0 à la lecture de RIR.

DCAS : indique que les commandes DC ou SDC ont été reçues activant la fonction associée à l'effacement.

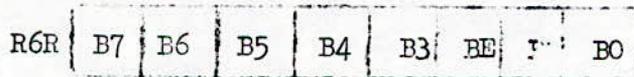
SPAS : indique qu'une commande SPE a été reçue activant la fonction SP

UUCG : Commande universelle indéfinie a été reçue

X : bit inutilisé

Registre commandes Pass Through (Command pass Through) R6R

C'est un registre à 8 bits utilisé comme port. Quand ce port est adressé par le MPU, il met en liaison le DATA Bus instrument ( $\overline{IB0} - \overline{IB7}$ ) au data bus du MPU (D0-D7). Ce port peut être utilisé pour faire passer au MPU les commandes et adresses secondaires qui ne sont pas interprétées directement par le GPIA



## IV ETUDE DE LA CARTE GPIB

### Première partie : Etude Hardware

Pour une utilisation rationnelle du bus normalisé IEEE - 488 entre un microprocesseur 6800 et des instruments de mesure compatibles avec cette norme, il est nécessaire de réaliser une carte d'interface entre ces derniers.

Cette carte devra assurer plusieurs fonctions :

- Amplifier les signaux
- Adapter le microprocesseur avec le bus ;
- Posséder une logique de commande pour l'écriture/lecture du GPIA
- Avoir une logique de décodage ;
- Assurer la transmission bidirectionnelles des informations entre le bus IEEE-488 et le GPIA.

#### A. Liaison MPU-GPIA

La liaison MPU - GPIA est la plus simple possible. Celle-ci est établie par dix neuf lignes :

- huit lignes de donnée D0 à D7
- trois lignes d'adresse RSO à RS 2 (Register Select) composant les huit adresses des registres du GPIA ;
- Une ligne de sélection de boîtier  $\overline{CS}$  (Chip Select) ;
- Une ligne  $R/\overline{W}$  (Read/Write)
- Une entrée horloge pour la phase  $\phi_2$  de l'horloge à 1 MHz du MC 6800
- Une ligne  $\overline{RESET}$  pour le forçage à la mise en route ;
- Deux lignes pour le contrôle de DMA (Direct Memory Access)
- Une ligne  $\overline{IRQ}$  pour une demande d'interruption au microprocesseur

- Une ligne ~~ASE~~ dont le rôle est précisé ultérieurement.

Il y a enfin une <sup>ligne</sup> TRIG destinée à l'extérieur qui s'active à la réception du message GET.

### 1. Amplification du bus

Le microprocesseur MC 6800 utilisé dans le système possède au niveau de ses lignes de sorties (données, adresses et contrôle) une sortance généralement faible. Pour qu'elle ne soit pas dépassée, tous les signaux issus du MC 6800 sont amplifiés par des circuits amplificateurs ou buffers 3 états. Ces buffers sont des circuits logiques inverseurs et/ou non inverseurs dont l'impédance d'entrée est élevée et la sortance est très élevée. Ils ont également pour rôle d'isoler le microprocesseur lorsqu'ils sont en haute impédance.

Les buffers utilisés dans la liaison MPU - GPIA sont de deux types.

- buffers unidirectionnels (8T97) pour les lignes d'adresse.
- buffers bidirectionnels (8T26) pour les lignes de données puisque ces dernières transitent dans les deux sens.
- buffer unidirectionnel (8T26) pour les signaux de contrôle.

(un circuit 8T97 aurait pu être utilisé. Cependant, le besoin des compléments de certains signaux de la logique d'écriture/lecture justifie l'emploi du buffer 8T26).

Les signaux de contrôle sont validés en permanence par +5V dans le sens de la réception (vers GPIA).

Le bus de données passe dans les amplificateurs bidirectionnels inverseurs. Le MPU doit donc inverser toutes les données qu'il échange avec le GPIA. Le sens de fonctionnement des 8T26 est

commandé par une logique de commande. Quant à l'activation des buffers d'adresses, elle fait intervenir le signal  $\overline{VMA}$ .

Le bus d'adresses comporte seize lignes A0 - A15, les lignes A0 - A2 après avoir été "bufferisées" sont connectées aux entrées RSO - RS1 et RS 2 du GPIA, pour la sélection des différents registres de celui-ci. Les autres bits de poids forts, le signal VMA serviront à la sélection du GPIA grâce à un circuit de décodage. La liaison MPU-GPIA peut être schématisée par les synoptiques ci-dessous.

On trouvera dans les deux pages suivantes, le brochage du 8T26 et du 8T97, ainsi que la logique de commande et les tables de vérité correspondantes.

## 2. Logique de commande des 8T26

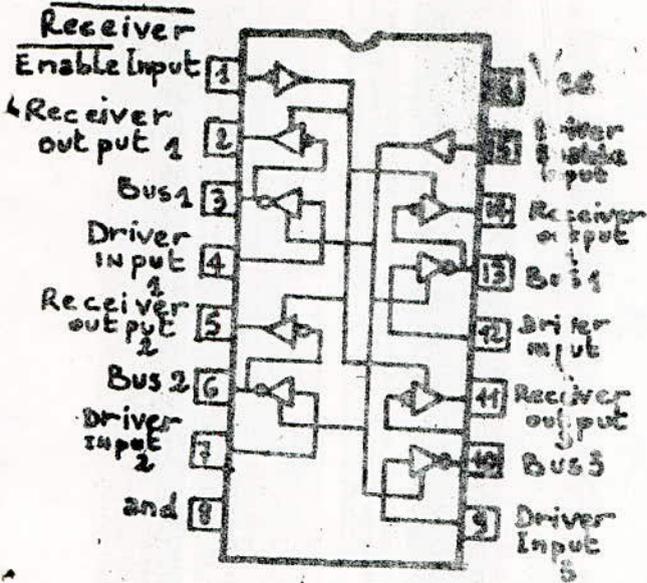
Cette logique réalisée par une porte NAND 7410 et des inverseurs 7404 sert à la validation des drivers 8T26 dans le sens de lecture ou d'écriture (voir tableaux), les signaux  $\phi_2$ ,  $\overline{CS}$  et  $R/\overline{W}$  contribuent à la commande des 8T26.

## 3. Décodage d'adresse

Lorsque le CPU place une adresse de 16 bits sur le bus d'adresse, Celui-ci, ne doit activer qu'une position mémoire ou un interface d'E/S.

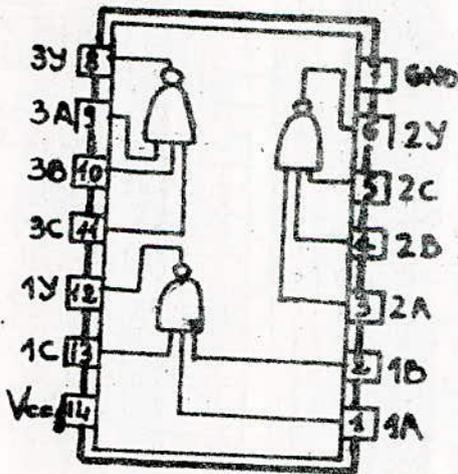
Le circuit de décodage dépend du type de boîtiers utilisés.

Les décodeurs permettent d'acheminer les signaux de sélection vers les différents boîtiers de façon que le boîtier concerné soit actif.

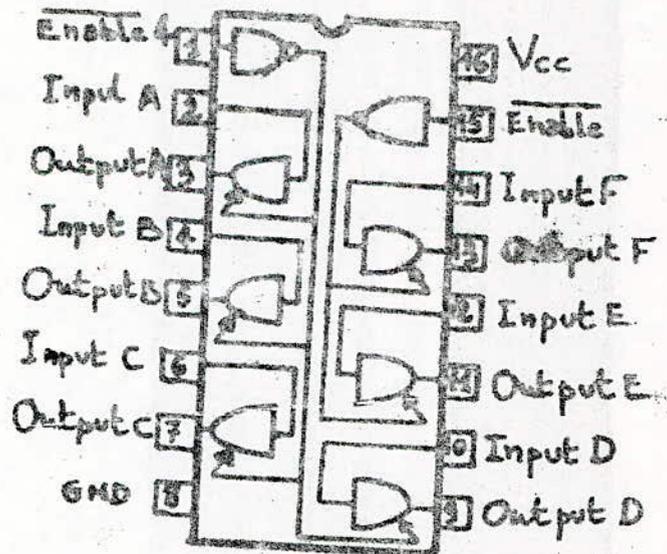


| pin ①<br>8T26 | pin ②<br>8T26 | Sens de passage<br>8T26 |
|---------------|---------------|-------------------------|
| 0             | 0             | Lecture                 |
| 0             | 1             | Interdit                |
| 1             | 0             | Haute impédance         |
| 1             | 1             | écriture                |

MC 8T26



SN 7410



MC 6743

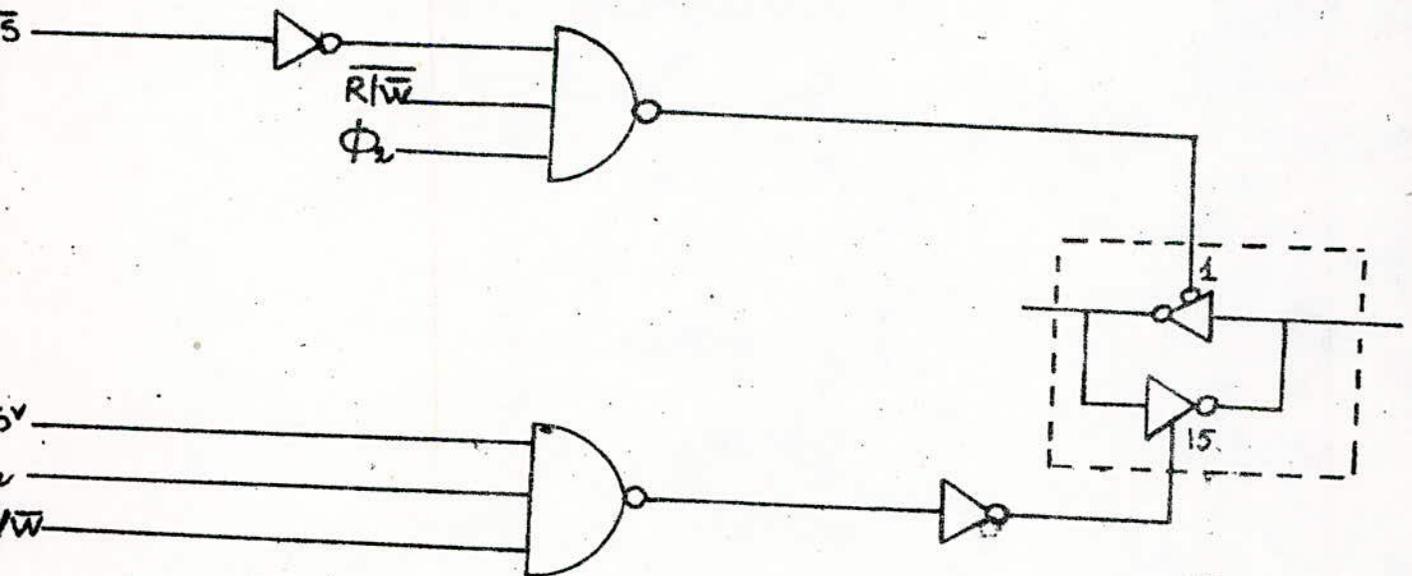
| $\Phi_2$ | $\overline{R/\overline{W}}$ | pin ⑮<br>8T26 |
|----------|-----------------------------|---------------|
| 0        | 0                           | 0             |
| 0        | 1                           | 0             |
| 1        | 0                           | 0             |
| 1        | 1                           | 1             |

⑮ =  $\Phi_2 \cdot \overline{R/\overline{W}}$

| $R/\overline{W}$ | $\Phi_2$ | CS | pin ①<br>8T26 |
|------------------|----------|----|---------------|
| 0                | 0        | 0  | 1             |
| 0                | 0        | 1  | 1             |
| 0                | 1        | 0  | 1             |
| 0                | 1        | 1  | 1             |
| 1                | 0        | 0  | 1             |
| 1                | 0        | 1  | 1             |
| 1                | 1        | 0  | 1             |
| 1                | 1        | 1  | 0             |

① =  $\overline{CS} \cdot \Phi_2 \cdot \overline{R/\overline{W}}$

Tables De Verite De Commande De 8T26



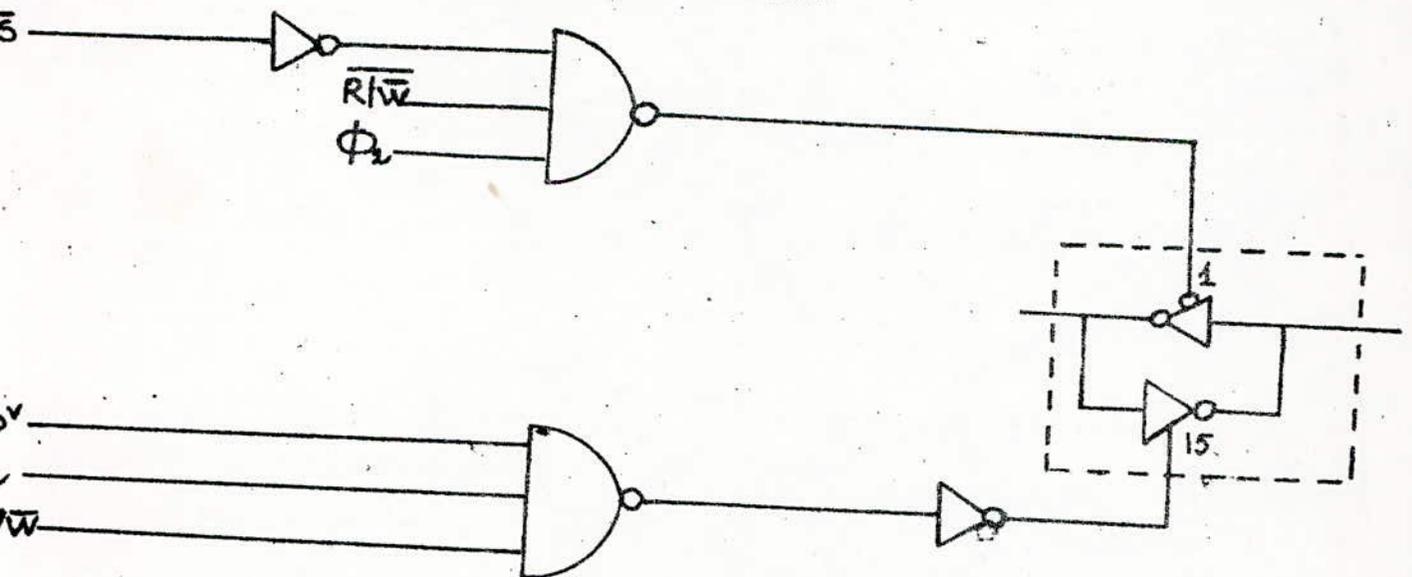
| $\Phi_2$ | $\overline{R/\overline{W}}$ | pin ⑮<br>8T26 |
|----------|-----------------------------|---------------|
| 0        | 0                           | 0             |
| 0        | 1                           | 0             |
| 1        | 0                           | 0             |
| 1        | 1                           | 1             |

⑮ =  $\Phi_2 \cdot \overline{R/\overline{W}}$

| $R/\overline{W}$ | $\Phi_2$ | CS | pin ①<br>8T26 |
|------------------|----------|----|---------------|
| 0                | 0        | 0  | 1             |
| 0                | 0        | 1  | 1             |
| 0                | 1        | 0  | 1             |
| 0                | 1        | 1  | 1             |
| 1                | 0        | 0  | 1             |
| 1                | 0        | 1  | 1             |
| 1                | 1        | 0  | 1             |
| 1                | 1        | 1  | 0             |

① =  $\overline{CS} \cdot \Phi_2 \cdot \overline{R/\overline{W}}$

Tables De Verite De Commande De 8T26



Dans la plupart des systèmes informatiques, les adresses des différentes cartes sont figées. Pour que notre interface puisse être utilisée avec n'importe quel système déjà existant, nous avons conçu notre carte avec une adresse translatale. Pour ce faire, nous avons utilisé un décodage par comparateurs pour la sélection de notre interface.

### Principe du décodage

Notre logique du décodage utilise trois comparateurs 4 bits SN7485, une porte XOR SN 74136, une porte NAND à 8 entrées SN 7430 et un inverseur SN 7404 (fig. ).

Le décodage s'effectue sur les poids forts d'adresse (A3 - A 15) par les trois comparateurs ainsi que la porte XOR (qui constitue aussi un comparateur à un bit). Les bits de comparaison sont fixés par des switches existant sur la carte selon l'adresse choisie pour sélectionner le GPIA. (Dans notre application, l'adresse 3FF8 a été choisie).

Seule la sortie A = B du comparateur sera utilisée, les autres sorties A < B ; A > B ne nous intéressent pas et seront déconnectées. Les trois sorties fonctionnelles des comparateurs, la sortie de la porte XOR inversée, le VMA signalent la validité des informations d'adressage sont reliés aux entrées de la porte NAND SN 7430.

Les trois entrées restantes de cette porte sont figées à 1 en les reliant au + 5V (car on travaille en logique positive). La sortie

de cette porte active au niveau bas est envoyée sur la pin 3 du GPIA pour le sélectionner. D'autre part, inversée, elle va être appliquée à l'entrée 1 de la porte NAND 7410 de la logique de contrôle lecture/écriture pour que les drivers 8T26 soient validés pendant que le GPIA est sélectionné.

## B. Liaison GPIA-IEEE 488

Pour assurer la compatibilité du GPIA au bus IEEE 488, nous devons insérer sur les seize lignes du bus des transcepteurs (transceivers) bidirectionnels non inverseurs.

Il existe des circuits "drivers" de lignes totalement conformes à la norme. Exemple : INTEL 8293, TEXAS SM 75160 ; 75 161 ; MOTOROLA MC 3441 ; 3446 ; 3447 ; 3448.

Nous avons utilisé des MC 3447 pour raison d'espace sur la carte GPIB. De plus chaque boîtier MC 3447 possède une entrée validation inversée, ce qui nous a évité l'emploi d'inverseurs sur la carte d'où simplification de câblage sur le circuit imprimé.

### 1. Utilisation des MC 3447

Le MC 3447 est un circuit à 24 broches qui renferme huit transceivers trois états, d'où l'utilisation de deux boîtiers. Sur les lignes nécessitant une inversion de sens, la direction de l'information est commandée par les lignes  $\overline{T/R}1$  et  $\overline{T/R}2$  issues du GPIA. Le schéma synoptique montre la validation des différents signaux.

- $\overline{ENY}$  et les DATA sont validés par  $\overline{T/R}2$ . Ils sont sortants quand  $\overline{T/R}2 = 1$  (GPIA talker) et entrants quand  $\overline{T/R}2 = 0$  (GPIA listener).
- $\overline{DLC}$  et RFD sont validés par  $\overline{T/R}2$  inversé. Ils sont sortants quand le GPIA est listener et entrants quand le GPIA est talker.
- $\overline{ATN}$ ,  $\overline{REN}$ ,  $\overline{IPC}$  validés par la masse. Ils sont toujours entrants au GPIA. Ils lui arrivent du bus.
- $\overline{EOI}$  : validé par  $\overline{T/R}1$ .  $\overline{EOI}$  circule dans le même sens que les données lorsqu'elle signifie END et en sens contraire lorsqu'elle

signifie l'identification. Il lui ~~fait donc une commande par-~~  
ticulière.

-- SRQ : validé par + 5V. Ce signal est toujours sortant afin que  
les interruptions soient envoyées au contrôleur.

Vu le brochage du MC 3447, sur un boîtier arrivent :

-- 4 lignes de données ;  $\overline{ATT}$  ; DAC activé par l'entrée inver-  
sante  $\overline{T/R}$  2 ;  $\overline{DAV}$  et  $\overline{SRQ}$ .

Sur l'autre arrivent :

-- les 4 lignes restantes de données ; RFD validé de la même  
façon que DAC ;  $\overline{REN}$  et  $\overline{IFC}$ ,  $\overline{EOI}$ .

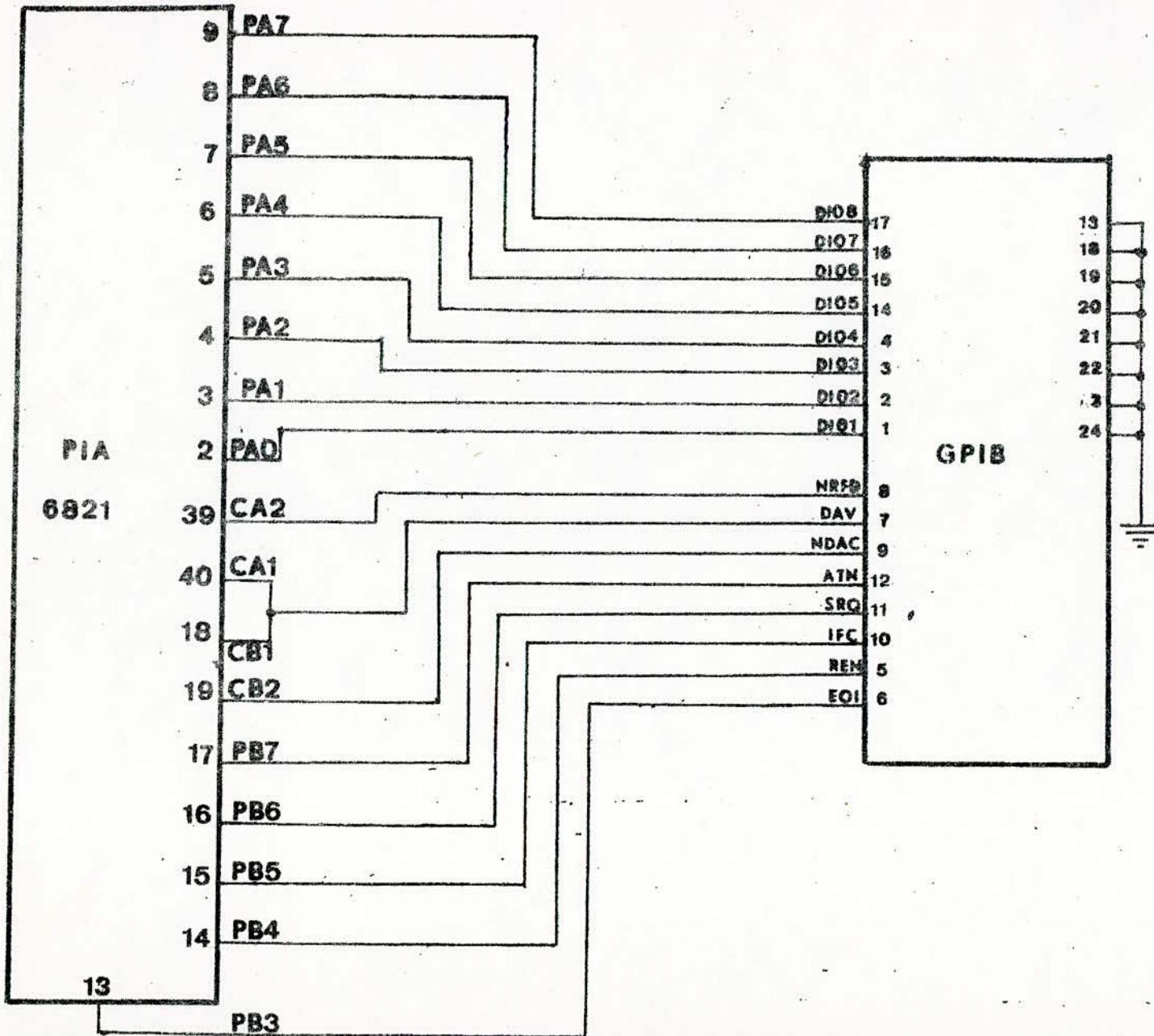
Pour que les information circulent correctement dans les deux  
sens, nous avons chargé par un pont de resistances les sorties  
du MC 3447. Nous les avons remplacé par des 8T28

## 2. Utilisation des 8T28

Normalement, les 8T28 ne sont pas conformes à la norme IEEE-488. Il ne  
jouent pas le rôle de transceivers. Ce sont simplement des buffers bi-  
directionnels non inverseurs. Pour assurer la transmission des signaux  
dans les deux sens et tester si la communication peut avoir lieu, nous  
avons été conduites de simuler les circuits MC 3447 par les circuits  
8T28.

Les 8T28 sont des circuits trois états, à 16 broches. Ils assurent  
l'émission et la réception de quatre signaux validés de la même manière.  
Les différentes activations des lignes du bus et la technologie de ces  
boîtiers nous ont poussées à utiliser sept circuits 8T28 (Voir fig ).

-- 2 boîtiers pour les lignes de données, 1 boîtier pour  $\overline{DAV}$  ; 1 bof-  
tier pour  $\overline{IFC}$ ,  $\overline{REN}$ ,  $\overline{ATT}$  ; 1 boîtier pour  $\overline{SRQ}$  ; 1 boîtier  $\overline{EOI}$  ;  
1 boîtier pour DAC et RFD.



LIASON GPIB-PIA

## Deuxième partie : Etude Software

Dans cette partie, nous donnons les programmes de fonctionnement de l'interface.

Dans le cas général où la communication se passe normalement entre deux appareils ou mêmes plusieurs munis d'une carte GPIA, l'un "source handshake" (talker) et le (s) autre (s) "accepter (s) handshake" (listener (s)).

L'organigramme d'une telle communication est donné sur la page suivante.

Les tests sur DAV, RFD et RDAC sont réalisés par les tests de bits du registre R3R du GPIA.

Dans le cas de notre application, un PIA classique a été utilisé comme contrôleur et écouteur et le GPIA parleur.

### L'algorithme est le suivant :

1. Initialisation du PIA
2. Initialisation du GPIA
3. Programme d'émission d'un caractère par le GPIA et son stockage en mémoire.

L'organigramme est donné par la suite. Pour le programme, consulter annexe 3.

### Commentaires de l'organigramme

L'initialisation du PIA est faite en deux phases :

- Phase 1 : permet l'adressage du GPIA (dans cette phase le PIA est contrôleur).
- Phase 2 : la transmission d'un caractère (dans celle-ci, le PIA est en plus de contrôleur, écouteur).
- On teste si le signal RFD a été reçu par le GPIA ; ce qui met le GPIA

# ANNEXE 1

## LE COMPAREUR 7485

### Schéma symbolique et table de vérité

Ce circuit compare deux nombres de 4 bits,  $A_0^3$  et  $B_0^3$  en fournissant **trois** sorties :  $A > B$  ;  $A < B$  ;  $A = B$ . Des entrées  $A' > B'$  ;  $A' < B'$  ;  $A' = B'$ , issues des sorties correspondantes d'un circuit traitant des bits de poids inférieur permettent la mise en cascade de plusieurs circuits et la comparaison de nombre plus grands que 4 bits.

Le tableau de vérité de la figure suivante s'interprète facilement en supposant que l'on compare des nombres plus grands que 4 bits ; que les entrées  $A' < B'$  ;  $A' > B'$  ;  $A' = B'$  donnent l'état relatif des bits de poids faibles et que les sorties  $A > B$  ;  $A < B$  et  $A = B$  donne l'état relatif global de deux nombres. On distingue trois cas :  $A > B$  ;  $A < B$  et  $A = B$ .

1. A est plus grand que B si :

a) quel que soit l'état relatif des poids faibles, nous avons

$$- A_3 > B_3 \text{ (C'est-à-dire } A_3 = 1 ; B_3 = 0) \quad \text{Cas 1}$$

$$- (A_3 = B_3) (A_2 > B_2) \quad \text{Cas 3}$$

$$- (A_3 = B_3) (A_2 = B_2) (A_1 > B_1) \quad \text{Cas 5}$$

$$- (A_3 = B_3) (A_2 = B_2) (A_1 = B_1) (A_0 > B_0) \quad \text{Cas 7}$$

b) Si les poids forts  $A_0^3$  sont égaux à  $B_0^3$ , les poids faibles A sont supérieurs aux poids faibles B.

$$- (A_3 = B_3) (A_2 = B_2) (A_1 = B_1) (A_0 = B_0) (A' > B') \\ \text{ (A' < B') , (A' = B') \quad \text{Cas 9}}$$

2. A est plus petit que B si :

a) quel que soit l'état relatif des poids faibles, nous avons :

$$A_3 < B_3 \text{ (C'est-à-dire } A_3 = 0 ; B_3 = 1) \quad \text{Cas 2}$$

$$(A_3 = B_3) (A_2 < B_2) \quad \text{Cas 4}$$

$$(A_3 = B_3) (A_2 = B_2) (A_1 < B_1) \quad \text{Cas 6}$$

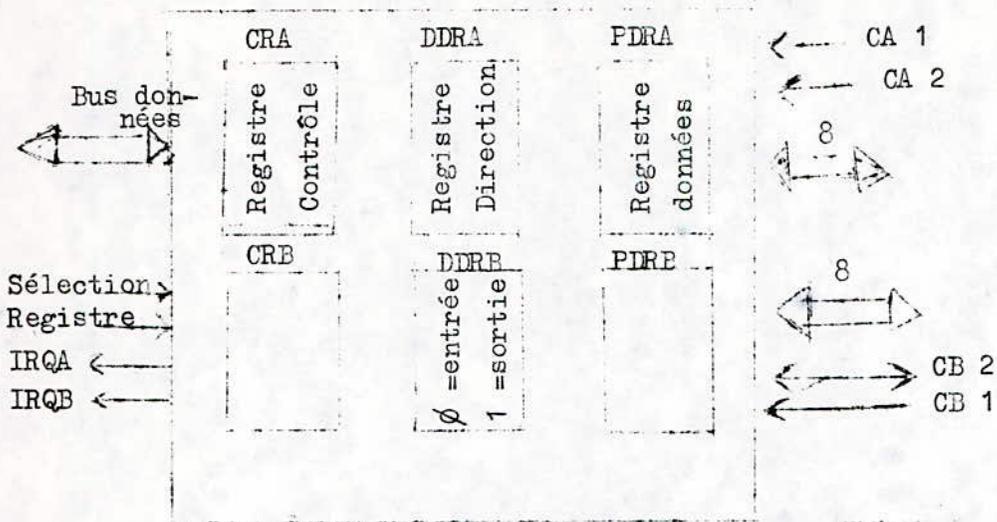
$$(A_3 = B_3) (A_2 = B_2) (A_1 = B_1) (A_0 < B_0) \quad \text{Cas 8}$$

b) Si les poids forts  $AC^3$  sont égaux à  $BC^3$ , les poids faibles A sont inférieurs aux poids faibles B.

| Entrées de comparaison des nombres A et B |             |             |             | Entrée de mise en cascade du résultat de la comparaison des nombres A' et B' |    |           | Sorties |   |   | Cas |
|---|-------------|-------------|-------------|--|----|-----------|---------|---|---|-----|
| $A_3, B_3$                                | $A_2, B_2$  | $A_1, B_1$  | $A_0, B_0$  | A'   | B' | $A' = B'$ | A       | B | A |     |
| $A_3 > B_3$                               | X           | X           | X           | X  | X  | X         | 1       | 0 | 0 | 1   |
| $A_3 < B_3$                               | X           | X           | X           | X  | X  | X         | 0       | 1 | 0 | 2   |
| $A_3 = B_3$                               | $A_2 > B_2$ | X           | X           | X  | X  | X         | 1       | 0 | 0 | 3   |
| $A_3 = B_3$                               | $A_2 < B_2$ | X           | X           | X  | X  | X         | 0       | 1 | 0 | 4   |
| $A_3 = B_3$                               | $A_2 = B_2$ | $A_1 > B_1$ | X           | X  | X  | X         | 1       | 0 | 0 | 5   |
| $A_3 = B_3$                               | $A_2 = B_2$ | $A_1 < B_1$ | X           | X  | X  | X         | 0       | 1 | 0 | 6   |
| $A_3 = B_3$                               | $A_2 = B_2$ | $A_1 = B_1$ | $A_0 > B_0$ | X  | X  | X         | 1       | 0 | 0 | 7   |
| $A_3 = B_3$                               | $A_2 = B_2$ | $A_1 = B_1$ | $A_0 < B_0$ | X  | X  | X         | 0       | 1 | 0 | 8   |
| $A_3 = B_3$                               | $A_2 = B_2$ | $A_1 = B_1$ | $A_0 = B_0$ | 1  | 0  | 0         | 1       | 0 | 0 | 9   |
| $A_3 = B_3$                               | $A_2 = B_2$ | $A_1 = B_1$ | $A_0 = B_0$ | 0  | 1  | 0         | 0       | 1 | 0 | 10  |
| $A_3 = B_3$                               | $A_2 = B_2$ | $A_1 = B_1$ | $A_0 = B_0$ | 0  | 0  | 1         | 0       | 0 | 1 | 11  |
| $A_3 = B_3$                               | $A_2 = B_2$ | $A_1 = B_1$ | $A_0 = B_0$ | X  | X  | 1         | 0       | 0 | 1 | 12  |
| $A_3 = B_3$                               | $A_2 = B_2$ | $A_1 = B_1$ | $A_0 = B_0$ | 1  | 1  | 0         | 0       | 0 | 0 | 13  |
| $A_3 = B_3$                               | $A_2 = B_2$ | $A_1 = B_1$ | $A_0 = B_0$ | 0  | 0  | 0         | 1       | 1 | 0 | 14  |

INTERFACE PARALLELE PROGRAMMABLE FIA 6820 ET 6821

La structure interne du PIA est la suivante :



Le registre de contrôle est le suivant :

|              |        |        |                  |   |               |      |          |   |
|--------------|--------|--------|------------------|---|---------------|------|----------|---|
|              | 7      | 6      | 5                | 4 | 3             | 2    | 1        | 0 |
| CRA<br>(CRB) | IRQA 1 | IRQA 2 | Contrôle de CB 2 |   | Accès<br>DDRA | CA 1 | Contrôle |   |

- Le bit 7 indique qu'il y a eu une transition sur la broche CA 1. Il est utilisé comme indicateur d'interruption.
- Le bit 6 joue le même rôle que la broche CA 2 lorsqu'elle est utilisée comme entrée.
- Les bits 3, 4 et 5 établissent l'un des 8 modes de fonctionnement du boîtier, en particulier la fonction de la broche CA 2.
- Le bit 2 indique si l'on doit sélectionner le registre-direction ou le registre-données, attendu qu'ils ont tous deux la même adresse.
- Les bits 0 et 1 autorisent ou inhibent les interruptions.

Le PIA a 6 registres mais seulement deux broches de sélection de registre RS. Le registre direction et le registre données de chaque port partagent la même adresse. Ils sont distingués par l'état du bit 2.

| RS 1 | RSC | CRA (2) | CRB(2) | Registre                                       |
|------|-----|---------|--------|--|
| C    | 0   | 0       | ---    | Direction ( )<br>Données ( ) A<br>Commande ( ) |
| C    | 0   | 1       | ---    |  |
| C    | 1   | -       | ---    |  |
| 1    | 0   | -       | 0      | Direction ( )<br>Données ( ) B<br>Commande ( ) |
| 1    | C   | -       | 1      |  |
| 1    | 1   | -       | -      |  |

RS1 = 0 sélectionne le port A

RSC = 1 sélectionne le registre de commande (port A et B).

RS1 = 1 sélectionne le port B

RSC = 0 sélectionne le registre direction ou donnée

Tableau : contrôle de l'entrée d'interruption CA 1 et CB 1

| CRA - 1<br>(CRB - 1) | CRA - 0<br>(CRB 0) | Entrée CA1<br>(CB 1) | Bit indicateur<br>CRA 7 (CRB- 7) | Demande d'interruption<br>IRQ              |
|----------------------|--------------------|----------------------|----------------------------------|--|
| 0                    | 0                  | Active ↓             | Mis à 1 sur ↓ de CA 1<br>(CB 1)  | Inhibe l'interruption<br>IRQ               |
| 0                    | 1                  | Active ↓             | Mis à 1 sur ↓ de CA 1<br>(CB 1)  | IRQ mis à 0 quand CRA 7<br>est à 1 (CRB-7) |
| 1                    | 0                  | Active ↑             | Mis à 1 sur ↑ de CA 1<br>(CB 1)  | Inhibe $\overline{\text{IRQ}}$             |
| 1                    | 1                  | Active ↑             | Mis à 1 sur ↑ de CA 1<br>(CB 1)  | IRQ mis à 0 quand CRA 7<br>est à 1 (CRB-7) |

↑: Indique une transition positive

↓: indique une transition négative

Le bit CRA-7 (CRB-7) est remis à zéro par une lecture du registre DATA A (ou B).

Contrôle de CA 2 et CB 2 comme entrées d'interruption

| CRA - 5<br>(CRB - 5) | CRA - 4<br>(CRB - 4) | CRA - 3<br>(CRB - 3) | Entrée d'in-<br>terruption CA2<br>(CB 2) | Bit indicateur<br>CRA 6 (CRB 6) | Demande d'inter-<br>ruption du MPV<br>IRQA (IRQB)       |
|----------------------|----------------------|----------------------|--|---------------------------------|---|
| 0                    | 0                    | 0                    | active ↓                                 | Mis à 1 sur<br>de CA 2 (CB2) ↓  | Inhibe IRQ passe<br>à l'état heut                       |
| 0                    | 0                    | 1                    | active ↓                                 | Mis à 1 sur<br>de CA 2 (CB2) ↓  | passse à l'état<br>bas quand CRA 6<br>(CRB-6) passe à 1 |
| 0                    | 1                    | 0                    | active ↑                                 | Mis à 1 sur<br>de CA 2 (CB2) ↑  | Inhibe IRQ passe<br>à l'état haut                       |
| 0                    | 1                    | 1                    | active ↑                                 | Mis à 1 sur<br>de CA 2 (CB2) ↑  | passse à l'état<br>bas quand CRA-6<br>(CRB-6) passe à 1 |

Contrôle de CB 2 comme sortie

| CRB-5 | CRB-4 | CRB-3 | Mis à 0   | CB 2  | Mis à 1 |
|-------|-------|-------|---|---|---------|
| 1     | 0     | 0     | Bas sur la 1ère transition<br>positive de E qui suit une<br>écriture du registre de don-<br>nées B            | Haut quand CRB-7 est mis<br>à 1 pour une transition<br>active du signal CB 1.   |         |
| 1     | 0     | 1     | Bas sur la transition positive<br>de E qui suit une écriture du<br>registre de données B                      | Haut sur la transition<br>positive de la prochaine<br>impulsion de E  |         |
| 1     | 1     | 0     | Bas quand CRB - 3 est mis à 0<br>pour une écriture du registre<br>de contrôle B                               | bas tant que CRB 3 est<br>bas. Il passe à l'état<br>haut quand CRB 3 sera mis<br>à 1 pour une écriture ds<br>le registre de CRB |         |
| 1     | 1     | 1     | Toujours haut tant que CRB-3<br>est haut, passe à l'état bas<br>quand CRB 3 est mis à 0 pour<br>une écriture. | Haut quand CRB-3 est mis<br>à 1 par une écriture du<br>registre de contrôle B<br>(CRB).   |         |

Contrôle du CA 2 comme sortie

| CRA-5 | CRA-4 | CRA-3 | Mis à 0   | CA 2   | Mis à 1 |
|-------|-------|-------|---|--|---------|
| 1     | 0     | 0     | Bas sur la transition nég. de la<br>1ère impulsion de E qui suit 1 lec-<br>ture du registre données A | Haut sur la transition<br>Active du signal CA 1  |         |
| 1     | 0     | 1     | Bas immédiatement après 1 lecture<br>du MPV du registre données A                                     | Haut sur la transition<br>nég. de la prochaine<br>impulsion de E                             |         |
| 1     | 1     | 0     | Bas quand CRB-3 est mis à zéro par<br>1 écriture de Contrôle A  | Toujours bas tant que<br>CRA-3 est bas   |         |
| 1     | 1     | 1     | Toujours haut tant que CRB-3 est<br>haut.   | Haut quand CRA-3 passe<br>à 1 comme l'effet d'1<br>écriture ds le registre<br>de contrôle A. |         |