

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT : D'ELECTRONIQUE

PROJET DE FIN D'ETUDES

SUJET

CONCEPTION ET REALISATION
D'UN ALLOCATEUR DE BUS
AVEC DMA POUR SYSTEMES
MULTIPROCESSEURS

Proposé par :
M^r F. GUETTACHE

Etudié par :
A.LAADJAL ADJAL
S.TEGHBIT

Dirigé par :
M^r GUETTACHE



PROMOTION : JUIN 84



الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم والبحث العلمي
MINISTRE DE L'ENSEIGNEMENT ET DE LA RECHERCHE SCIENTIFIQUE

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT : D'ELECTRONIQUE

PROJET DE FIN D'ETUDES

pour l'obtention du diplôme d'Ingénieur d'Etat en Electronique

SUJET

CONCEPTION ET REALISATION
D'UN ALLOCATEUR DE BUS
AVEC DMA POUR SYSTEMES
MULTIPROCESSEURS

Proposé par :
M^r F. GUETTACHE

Etudié par :
A.LAADJAL ADJAL
S.TEGHBIT

PROMOTION : JUIN 84

DEDICACES

Je dedie ce travail

- * à mon père et à ma mère
- * à mon frère AHENE
- * à mes sœurs : TAOUS - SAIDA - MALIKA - Zineb
- * à tous ceux qui me sont chers
- * à ATIKA - DJIDA - ZOUBIDA

- SALIHA

الاهداء

امدى هذا العمل الى :

- * والديا الكريمين حفظهما الله
- * اخي محمد الامين
- * اخواني واخواتي
- * كافة الاهد والاقارب والاصدقاء
- * صليحة - مليكة - نصيرة

عتيقة

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

* Nous remercions Mr ABDELLAOUI Chef du CDTA de nous avoir si aimablement accueillies dans son centre.

* Nos remerciements vont aussi à notre promoteur Mr F. GUENTACHE de nous avoir suivies et conseillées durant tout notre travail.

* Nous remercions vivement Mr H. BESSALAH pour nous avoir acceptées dans son laboratoire pour l'aide morale et matérielle qu'il n'a pas cessé de nous prodiguer durant notre projet.

* La frappe et le tirage de ce polycopé ont été soigneusement fait par Mr M.N. KECILI qu'il trouve dans ces lignes nos vifs remerciements et notre profonde gratitude.

* Nous remercions aussi MM. BOUZID et ZOUAOUI pour n'avoir ménagé aucun effort pour nous faire profiter de leur expérience.

* Nous finirons en remerciant MM. ZIANE et MADDI ainsi que tous les enseignants qui ont contribué à notre formation.

S O M M A I R E

	Page
Introduction.	
Partie Etude	
Chapitre I.	
I. Généralités sur les structures de traitement parallèle d'information	1
I.1 Les structures de traitement parallèle.....	1
I.2 Le multiprocessing	2
I.3 Topologies d'interconnexion	2
I.4 Types d'organisation des systèmes multiprocesseurs.....	2
II Les voies de communication.....	5
II.1 Définition d'un bus	5
II.2 Avantages et inconvénients d'un bus	6
III Description des bus; destinés au multiprocessing.....	7
Introduction	7
A) Le Multibus d'INTEL	9
B) Le bus VME	18
C) Le bus Exorciser	27
Chapitre II.	
I. Techniques de résolution des priorités	35
1) Technique de résolution des priorités fixes	35
2) Technique de résolution des priorités tournantes	35
II. Allocation de la ressource bus - Arbitrage	36
1) Allocation déterministe	36
2) Allocation centralisée gérée par un arbitre	36
3) Allocation décentralisée - Arbitrage réparti	38

Partie réalisation.	Page
Chapitre III.	
I. Schéma synoptique	41
II. Description sommaire du synoptique	41
III. Les modes de transfert utilisés	41
1) Mode de transfert programmé	41
2) Mode de transfert DMA	43
2.1 L'intérêt du transfert par DMA	43
2.2 Les différents types de DMA	44
2.3 Le contrôleur DMA ME 6844	44
2.4 Présentation du DMAC	45
2.5 Les signaux d'interface du DMAC	48
2.6 Fonctionnement du DMAC	50
2.7 Les différentes phases d'un transfert DMA	53
IV. Description sommaire des cartes Maître et esclave	54
1) Description de la carte Maître	54
2) Description de la carte Esclave	56
Chapitre IV	
I - Présentation de la carte Allocateur de bus	59
1) Partie allocation	59
1.1 Position du problème	59
1.2 Réalisation	59
a) Description matérielle	59
b) Fonctionnement	61
2) Partie DMA. Réalisation	61
a) Description matérielle	61
b) Fonctionnement	64
II - Fonctionnement global	67
III - Adaptation du bus Exerciser au système multiprocesseur	68
Conclusion	72
Annexe.	

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

L'évolution appréciable des détecteurs de particules en physique nucléaire impose un volume de plus en plus considérable de données à acquérir et à traiter, ce qui entraîne une augmentation considérable de la charge d'un ordinateur et un ralentissement du processus de traitement.

Une des solutions préconisées pour résoudre ce type de problème consiste à décentraliser le traitement.

C'est dans ce but que le laboratoire "Architecture des systèmes" du centre de développement des techniques avancées (CDTA) du commissariat aux énergies nouvelles (CEN) s'est fixé comme objectif de mettre au point un système à structure multiprocesseur et cela à partir de cartes standard (Maître, Esclave) qui seront associées à quelques cartes spécifiques telle que la carte "Allocateur de bus" dont la réalisation nous a été confiée.

La structure multiprocesseur adoptée au laboratoire, est basée sur le concept "Maître-Esclave".

C'est une structure à bus partagé qui engendre des conflits d'accès au bus. La gestion de ces conflits est assurée par un dispositif spécialisé appelé "Allocateur de bus". Ce dispositif est un circuit logique qui, à la réception de demandes d'accès au bus provenant des différents processeurs secondaires constituant le système multiprocesseur, doit pouvoir déterminer lequel d'entre eux est le plus prioritaire et qui, par conséquent, peut accéder aux organes communs (bus, mémoires, ...). Pour un partage égal des ressources communes, l'allocateur utilise la technique de résolution de priorité tournante.

En plus de la gestion des conflits d'accès au bus, la carte réalisée assure la gestion d'accès direct en mémoire quand la nécessité s'en fait sentir.

Dans le cas d'un transfert par programme le rôle de l'allocateur se limite à la détermination de la priorité après quoi la gestion du transfert sera confiée au "Maître".

De ce fait, notre carte se divise en deux parties :

- partie Allocation
- partie DMA.

C H A P I T R E I.

Introduction :

La plupart des applications scientifiques actuelles necessitent des calculs très longs, très coûteux et portant sur un ensemble de données volumineux ce qui exige de meilleurs performances des ordinateurs.

Afin d'accroître la rapidité et la capacité de traitement d'une machine, on peut introduire le parallelisme dans ses diverses fonctions, ainsi les performances de cette machine seront d'autant meilleures que les programmes y seront bien adaptés et permettront d'exploiter le parallelisme de façon optimale.

I. GENERALITES SUR LES STRUCTURES DE TRAITEMENT PARALLELE D'INFORMATION .

I. 1. Les structures de traitement parallele.

Elles sont au nombre de quatre.

a) Matricielle ou Vectorielle.

C'est une structure où tous les processeurs executent la même tâche independamment l'un de l'autre, et cela sous le contrôle d'un processeur principal dont la mémoire commune à tous les processeurs est appelée mémoire centrale.

b) Associative

C'est une version de la structure matricielle avec la seule difference que dans ce cas, chaque processeur possède sa propre mémoire locale.

c) Pipeline.

Dans cette structure les unités de traitement sont reliées de telle sorte que les résultats de l'unité (i) servent comme données d'entrées pour l'unité(i + 1) chaque unité réalisant les mêmes fonctions.

d) Multiprocesseur du type MIMD.

Dans ce type de structure les processeurs effectuent des tâches differentes. Les multiprocesseurs correspondent à une évolution des systèmes informatiques dans laquelle on prévoit une ou plusieurs mémoires communes pour la synchronisation et la communication de messages entre processeurs.

I. 2. Le Multiprocessing.

Le multiprocessing est un mode de fonctionnement d'un système de traitement parallèle d'information, dans lequel plusieurs unités centrales peuvent travailler simultanément en ayant accès à des mémoires ou organes communs.

L'intérêt du multiprocessing réside dans le fait que les CPU et les mémoires sont interconnectés entre eux, la communication étant sous un contrôle commun, et le travail en parallèle.

I. 3. Topologies d'interconnexion.

Il existe plusieurs façons d'interconnecter N éléments dans un système.

Pour établir un schéma d'interconnexion on tient compte de la fiabilité et de l'extensibilité qui facilitera l'addition de plusieurs éléments sans affecter la structure initiale (voir fig. 1)

Il existe quatre topologies fondamentales à savoir :

- * Bus Commun Fig. 1 - a
- * Etoile Fig. 1 - b
- * Anneau Fig. 1 - c
- * Interconnexion totale Fig. 1 - d.

Les autres topologies existantes sont des combinaisons des quatre déjà citées.

I - 4. Types d'organisation des systèmes multiprocesseurs.

- a - Maître - esclave
- b - Exécution partagée des tâches dans chaque processeur.
- c - Traitement homogène symétrique dans tous les processeurs.

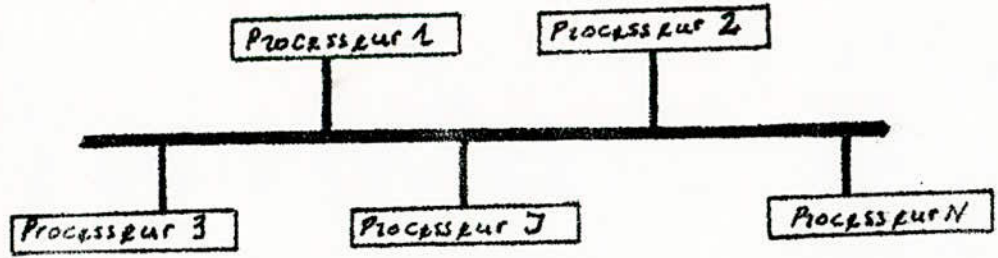
a) Maître - Esclave

Les programmes moniteurs sont implantés dans le processeur maître, dans le cas, son arrêt entraîne automatiquement celui du système, de même le temps mort des processeurs esclaves décroît avec la rapidité du processeur maître.

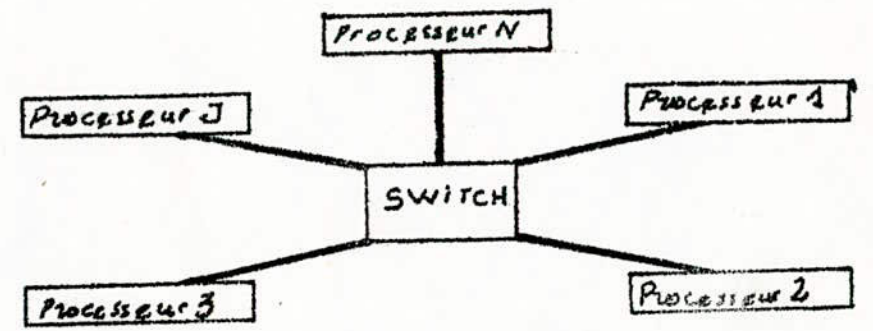
b) Exécution partagée des tâches dans chaque processeur.

L'exécution des différentes tâches est répartie entre plusieurs processeurs dont chacun a son propre système d'exploitation et par conséquent l'arrêt de l'un n'entraîne pas l'arrêt du système.

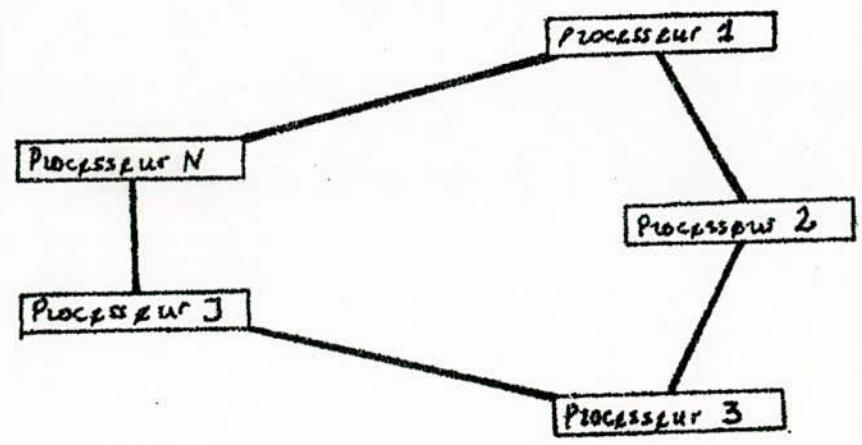
FIG 1 TOPOLOGIE DES INTERCONNEXIONS.



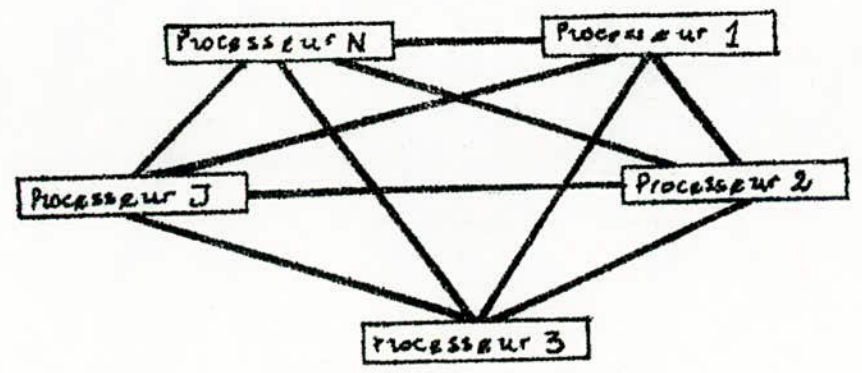
a) BUS COMMUN



b) ETOILE



c) ANNEAU



d) PLEINE CONNEXION

TABLEAU DE COMPARAISON

MAITRE - ESCLAVE	EXECUTION PARTAGÉE DES TÂCHES DANS CHAQUE PROCESSEUR	TRAITEMENT HOMOGENE SYMETRIQUE DANS TOUS LES PROCESSEURS
Les fonctions superviseur sont réalisées par le même processeur (maître)	Chaque processeur réalise ses propres fonctions "superviseur"	La fonction de maître est homogène d'un processeur à l'autre
L'arrêt du maître entraîne la perte de capacités du système	La capacité du système ne diminue pas d'un coup, lors de l'arrêt d'un processeur. Il y a dégradation graduelle	Dégradation graduelle lors d'un arrêt d'un processeur.
Le temps mort global des processeurs esclaves peut être très important si le maître n'est pas très rapide.	Chaque processeur a ses propres unités d'entrées / sorties ou fichiers	Toutes les situations conflituelles sont régies à la base de priorités
Avantages : Hard et soft sont relativement simples à réaliser Inconvénients : faible productivité dans le cas d'un maître à faible vitesse	Avantages bonne répartition de charge , pas de conflits vu que chaque processeur a ses propres programmes système. - Fiabilité. Inconvénients : " IF "	Avantages bonne fiabilité due à une bonne duplication de processeurs - Rapidité Inconvénients " IF "

c) Traitement homogène et symétrique dans tous les processeurs.

Chaque processeur, dans ce cas, remplit la fonction de processeur principal. (voir tableau de comparaison).

Position du problème.

La structure Maître-Esclave a été choisie pour sa simplicité matérielle, son principe est le suivant : le maître est l'unité de contrôle et de synchronisation. Les processeurs esclaves ont pour tâches la résolution d'équations interpendantes, les échanges se font à travers un bus commun contrôlé par un dispositif spécialisé.

II - Les voies de communication.

II - 1 Définition d'un bus.

Un bus est un organe de communication composé d'une ressource physique (fils parallèles, câbles coaxiaux, ...etc) partagée par plusieurs éléments, ces derniers communiquent directement en utilisant ce support sans mettre en oeuvre des éléments relais.

Cette voie unique et centrale, ne permet qu'une seule communication à la fois entre deux éléments, elle doit avoir une fiabilité très élevée, seulement son utilisation soulève deux problèmes :

Son allocation et l'adressage des différents éléments qui lui sont connectés.

- Allocation : Si plusieurs processeurs veulent transmettre simultanément des informations sur le bus, l'allocateur doit pouvoir désigner lequel d'entre eux aura la maîtrise du bus.

- Adressage : Certains éléments branchés sur le bus sont repérés par leur adresse.

Il existe deux types de bus :

- * les bus séries pour les transmissions à distance
- * les bus parallèles pour les transmissions locales.

* Les bus séries.

Les bus séries nécessitent peu de lignes et sont utilisés pour connecter des terminaux (Ecrans cathodiques, imprimantes, téléscripteurs) à un système à longue distance.

Les transmissions séries nécessitent seulement un ou deux fils pour transmettre tous les signaux (données, adresses, signaux de contrôle) entre modules du système.

La transmission se fait bit/bit.

* Les bus parallèles :

Ce type de bus nécessite beaucoup de lignes de transmission.

Ces bus sont utilisés pour des communications de module à module haute vitesse. On distingue 3 types de bus parallèles à savoir :

. bus interne d'un microprocesseur qui permet des échanges d'informations entre registres.

. bus monoprocesseur qui est chargé d'interconnecter les divers composants d'un système à microprocesseur : le microprocesseur, mémoires, E/S...etc...

. bus multiprocesseur qui assure des communications entre plusieurs processeurs constituant le système multiprocesseur.

II - 2 Avantages et Inconvénients d'un bus.

Un bus présente les avantages suivants :

- C'est une voie de communication qui permet de réduire la longueur des câblages.

- Il s'utilise sur des longueurs qui vont du millimètre à plusieurs hectomètres (mm à l'intérieur d'un microprocesseur, hm pour un bus de périphériques dans certaines applications de conduite de processus physiques).

- Il est souple car on peut y brancher assez facilement des éléments supplémentaires, il ya cependant des limites de puissance électrique, et en général du nombre maximal d'éléments connectables.

- Il réalise des cadences de transfert très élevées sur de courtes distances (10^7 mots/ pour les bus de longueur de l'ordre du mètre).

- Il permet des extensions par des structures à multiples bus.

Un bus a néanmoins quelques inconvénients :

- On n'y transmet qu'un seul message à la fois.

- Voie unique et centrale, un bus doit être très fiable.

- L'augmentation de la longueur d'un bus entraîne une diminution de la cadence de transfert due à l'affaiblissement et à la distorsion du signal. De même une limite inférieure de la distance entre deux lignes d'un bus doit être respectée afin d'éviter l'apparition de capacités parasites en HF.

III - Description des bus destinés au multiprocessing.

Introduction :

Une structure multiprocesseur comporte plusieurs processeurs travaillant en même temps et partageant des ressources communes (mémoires, E/S ; bus).

Pour bénéficier des performances d'une telle structure, il faut disposer d'un bus adéquat qui assure la communication entre les différents processeurs la composant.

Le plus souvent, il s'agira d'un ensemble de lignes donnant accès aux bus internes du système à microprocesseur.

La définition d'un bus ou sa normalisation porte sur 3 points essentiels :

- . l'organisation matérielle : c-à-dire le nombre de fils qu'il comporte
- . le mode de fonctionnement : C-à-dire le type de signaux véhiculés par ses fils.
- . les procédures d'échanges d'informations.

En général, un bus doit véhiculer tous les signaux utilisés par le système: données, adresses, contrôle, synchronisation et tensions d'alimentation.

A un instant donné, un seul processeur doit accéder au bus pour éviter les collisions d'informations ou l'interblocage du bus.

Examinons, maintenant, les bus les plus utilisés dans le domaine multiprocessing et donnons une idée générale de ce qui distingue un bus d'un autre.

* Le bus S100

Ce bus créé en 1976 par la société "MITS" a été le premier véritable standard disponible en micro-informatique, normalisé pour le microprocesseur 8080. Les échanges sur ce bus sont la transcription exacte de ceux du 8080. Ce bus est organisé autour d'un connecteur de 100 broches regroupant :

- . 8 Lignes d'entrées
- . 8 Lignes de sorties
- . 16 Lignes d'adresse
- . 8 Lignes d'interruption
- . 3 Lignes d'alimentation
- . 39 Lignes de commande.

Les 18 lignes restantes sont à la disposition de l'utilisateur.

Ce bus souffre principalement de son manque de souplesse car il a été conçu pour un matériel donné à une époque donnée, de même il se trouve qu'il est très mal adapté aux μ - p 16 bits (vitesse). Malgré cela ce bus est disponible sur un grand nombre de μ -ordinateurs "8 bits" (Intertechnique, micromachine,...)

* le versabus de Motorola.

Ce bus est destiné aux μ -p 8, 16 et 32 bits. Il possède deux connecteurs P_1 avec 140 broches et P_2 avec 120 broches.

Le connecteur P_1 contient :

- . un bus de données 16 bits
- . un bus d'adresses 24 bits.
- . 7 Lignes d'interruption
- . 5 lignes d'arbitrage du bus
- . Des lignes de commande, de détection d'erreurs et d'alimentation.

Le connecteur P_2 contient.

- . 50 lignes de commandes d'E/S
- . Des lignes d'expansion (Adresse, donnée).
- . Des lignes pour communication série et des lignes d'alimentation. Sa vitesse de transfert est de 5 Mhz. Toutes les possibilités de système multi-traitement et autres sont prises en compte, l'adressage sur 32 bits maximum, atteint 4 milliards de mots.

* Le multibus d'Intel.

Connu également sous le nom bus SBC, est un bus universel initialement conçu pour les μ -p dont l'espace mémoire est distinct de l'espace d'E/S.

Ce bus est organisé autour d'un double connecteur P_1 et P_2 le connecteur P_1 avec 86 broches et P_2 avec 60 broches.

Ce bus peut adresser sur 20 bits 1 M octets en mémoire, avec une extension prévue à 24 bits pour 16 M octets.

Il est conçu selon le concept "maître-esclave, différent du concept maître-esclave défini auparavant, où le maître quel qu'il soit prend la direction des échanges, et l'esclave ne peut en aucun cas contrôler le multibus car il représente une ressource commune aux différents modules maître du système. La vitesse maximum de transfert des informations est de 5 Mhz et il peut supporter des systèmes en multitraitement.

* Le bus VME.

C'est une nouvelle structure de bus pour les systèmes 8, 16 et 32 bits.

Ce bus reprend beaucoup de concept de "versabus" presque tous ses signaux sont identiques et représentent les mêmes chronogrammes.

Ce bus peut être considéré comme un sous ensemble du "versabus".

Néanmoins, il existe de nombreuses différences, essentiellement parcequ'il a fallu supprimer quelques lignes du "versabus" pour faire tenir sur 2 connecteurs 96 broches. ce qui existait sur 2 connecteurs de 120 broches.

Le bus VME offre de nombreuses possibilités de réalisations, aussi bien des systèmes ~~mono~~ processeurs que des systèmes multiprocesseurs très élaborés. Sa vitesse maximale de transfert est de 20 Mhz. (Voir tableau de comparaison).

A) Le multibus d'Intel

I) - Introduction

II)- Description du multibus

III)- Interface Multibus

IV)- Son utilisation.

I) Introduction.

Le multibus, c'est le bus standard proposé par INTEL pour tous les systèmes 8 et 16 bits, mono et multiprocesseurs.

Ce bus autorise une expansion modulaire avec des cartes additionnelles mémoires, E/S, coupleurs périphériques ou autres cartes unité centrale (plusieurs processeurs peuvent travailler en même temps).

Sa capacité d'adressage est de 10^6 octets.

Pour résoudre les problèmes de synchronisation entre tâches, d'interruptions prioritaires et de gestion de périphériques standards, on associe au multibus le moniteur temps réel multitâches RMX- 80.

TABLEAU DE COMPARAISON

caractéristiques Bus	nombre de broches du Connecteur	bus d'adresses	bus de données	Les micros processeurs l'utilisant.	nombre de Compagnies l'utilisant	Standards
multibus	86/60	20/24	8/16	8080-8085-8086 8088-8800-88000 68000	40+	IEEE 796
versabus	140/120	24	16/32	68000	57	un groupe d'utilisateurs
VME	96/96	24/32	16/32	68000	5	un groupe de fabricants
S.100	100	24	8/16	8080-8085-880 8088-8086-6502 6809-88000 TMS 9900		IEEE 696

Enfin, on peut conclure que le Multibus est un concept qui permet la connexion des cartes SBC de la façon la plus modulaire qui soit, de ce fait on peut passer d'une configuration ^{simple à une configuration} de plus en plus complexe tant du point de vue de la capacité mémoire, des E/S et des périphériques que du point de vue de la puissance de traitement (traitement parallèle).

Sa vitesse maximale de transmission est de 5 millions d'octets/s.

II - Description du Multibus.

Electriquement le bus se répartit sur deux connecteurs P₁ et P₂.

- Le connecteur P₁ contient :

- . 1 Ligne d'initialisation (INIT)
- . 20 Lignes d'adresses (bus d'adresses)
- . 16 Lignes de données)
- . 1 Ligne d'inhibition de la RAM (INH1)
- . 1 Ligne d'inhibition de la ROM (INH2)
- . 6 Lignes de contrôle et de commande
- . 7 Lignes de gestion des conflits sur le multibus.
- . 8 Lignes d'interruptions
- . 1 Ligne de reconnaissance d'interruption (INTA)
- . 1 Ligne autorisant le travail avec des processeurs 8 et 16 bits.

- Le connecteur P₂ contient.

Des lignes d'alimentation pour des modules analogiques et des signaux de protection contre les coupures de secteur, des lignes qui ont des fonctions bien spécifiques tel que le HALT, WAIT, RESET,....

et plusieurs lignes réservées à l'utilisateur.

Toutes les lignes du Multibus ont des niveaux de tension TTL Standard (0V, 5V) la logique négative est utilisée : le signal est actif au niveau logique "0". Le nombre maximal de cartes qu'on peut connecter sur le Multibus est de 20 cartes.

Les deux modes d'interruption utilisés sont :

- a) Les interruptions non vectorisées (mode NBVI)
- b) Les interruptions vectorisées (mode BVI)

LES CONNECTEURS DU MULTIBUS

CONNECTEUR P1

Bus d'adresses	AD00 à AD19	domaine d'adressage 1 M octet
Bus de données	DAT0 à DAT15	Bus de données 16 bits
Bus de Contrôle	$\overline{\text{INIT}}$ $\overline{\text{INH1}}$ $\overline{\text{INH2}}$ $\overline{\text{MEDC}}, \overline{\text{MWTC}}$ $\overline{\text{PDR}}, \overline{\text{IOWC}}$ $\overline{\text{XACK}}$ $\overline{\text{AACK}}$ $\overline{\text{BCLK}}$ $\overline{\text{CCLK}}$ $\overline{\text{BPRN}}$ $\overline{\text{BPRO}}$ $\overline{\text{BUSY}}$ $\overline{\text{BREQ}}$ $\overline{\text{CBREQ}}$ $\overline{\text{BHEN}}$	<p>positionne le système dans un état connu.</p> <p>inhibition de la mémoire RAM</p> <p>inhibition de la mémoire ROM</p> <p>Lecture, écriture mémoire</p> <p>lecture, écriture d'entrée/sortie</p> <p>fin de transfert sur mémoire ou EIS</p> <p>fin de transfert avancé</p> <p>horloge multibus.</p> <p>horloge générale</p> <p>entrée de la priorité</p> <p>sortie de la priorité</p> <p>Bus occupé</p> <p>demande d'accès (Résolution //)</p> <p>demande en attente</p> <p>Mélange de processeurs et 16 bits</p>
Interruptions	$\overline{\text{INT0}}$ à $\overline{\text{INT7}}$ $\overline{\text{INTA}}$	<p>Demande d'interruption</p> <p>Reconnaissance d'interruption</p>
Alimentation	$\pm 5^V, \pm 12^V, -10^V$ GND	

CONNECTEUR P2

Signaux de Contrôle et de Protection	$\overline{\text{RESET}}$ $\overline{\text{ACL0}}, \overline{\text{PFIN}}, \overline{\text{PFSN}}$ $\overline{\text{PFSR}}, \overline{\text{MPRO}}$	<p>initialisation du système</p> <p>Signaux de Protection contre les coupures secteur</p>
Entrées/Sorties	réservées à l'utilisateur	30 BROCHES

a) Les interruptions non vectorisées.

Dans ce cas \overline{INTA} n'est pas nécessaire. Le vecteur d'interruption est généré par le contrôleur d'interruption situé sur le module maître.

b) Les interruptions vectorisées.

\overline{INTA} est nécessaire. Dans ce mode, le maître génère \overline{INTA} et les signaux nécessaires pour que le contrôleur d'interruption situé sur le module esclave puisse envoyer le vecteur d'interruption vers le maître.

III - Interface Multibus.

L'interfaçage et la synchronisation des processeurs de la famille SBC avec le Multibus sont assurés par l'arbitre de bus 8289 qui travaille en conjonction avec le contrôleur de bus 8288.

Quand le processeur a la maîtrise du bus, il ignore l'existence de l'arbitre et écoute les commandes comme s'il avait l'utilisation exclusive du bus du système.

Si le processeur n'a pas à utiliser le bus, l'arbitre empêche le contrôleur de bus 8288, les transceivers de données, les latches d'adresse d'accéder au bus (toutes les sorties conductrices du bus sont forcées à l'état haute impédance). Tant que la commande n'est pas issue du 8288, le bus du système est dans l'état occupé "not ready" et le processeur entre dans l'état d'attente. Le processeur restera dans cet état, jusqu'à ce que l'arbitre du bus acquière la maîtrise du bus du système. Après quoi l'arbitre permettra aux transceivers de données et aux Latches d'adresse et au contrôleur d'accéder. Une fois la commande écoulée, la reconnaissance de transfert est envoyée vers le processeur pour indiquer "Ready". Ainsi l'arbitre règle les problèmes de conflits entre processeurs.

VI - L'utilisation du Multibus.

1°) Les principales fonctions du Multibus.

- L'interconnexion modulaire et la communication entre les différentes cartes de la famille SBC.

- La connexion de plusieurs modules processeurs travaillant en parallèle, ainsi le Multibus, permet d'augmenter les performances et la puissance de traitement en minimisant les conflits sur le bus et cela du fait que chaque carte C P U SBC possède ses propres ressources (RAM, REPRON, E/S), d'où l'unité centrale ne demandera le contrôle du Multibus que lorsque la source désirée est extérieure, d'autre part les ressources résidentes d'une carte C P U SBC sont protégées contre les accès provenant du Multibus.

- Le partage de ressources communes entre plusieurs cartes processeurs SBC.

2°) Gestion des priorités sur le Multibus.

La logique de gestion de priorités sur le Multibus est intégrée sur chaque carte maître. Elle permet d'allouer une ressource du Multibus à un module maître si celui-ci a les plus forte priorité. La gestion des priorités sur le Multibus peut être résolue en série ou en parallèle.

- Gestion série des priorités.

Chaque maître inhibe le maître de priorité inférieure quand il désire le contrôle du Multibus. La priorité est fixée par câblage sur la carte fond de panier. Si le maître de position (1) désire le contrôle du Multibus, il positionne sa sortie \overline{BPRO} à "1" car son entrée \overline{BPRN} est déjà à "0", ce qui implique que la ligne \overline{BPRN} du 2e maître sera à "1" ce qui aura pour effet d'inhiber ce dernier.

Quand un module maître désire le contrôle du Multibus, il examine la ligne \overline{BUSY} , si \overline{BUSY} est à zéro un autre maître utilise le Multibus. Deux cas sont alors possibles :

- a) Le module maître qui utilise le Multibus est prioritaire, l'autre module maître attendra que celui-ci ait fini ses transferts pour qu'il prend le contrôle du bus.
- b) Le module maître qui utilise le Multibus est moins prioritaire. L'autre module maître attendra que le transfert en cours soit fini pour prendre le contrôle du Multibus.

Deux paramètres déterminent le nombre de modules pouvant être connectés de façon série sur le Multibus : La période de l'horloge \overline{Bclk} et les temps de propagation de \overline{BPRN} et \overline{BPRO} et cela selon la formule suivante :

$$\sum_{i=1}^{N-1} (t_{BPRN} - t_{BPRO}) < t_{BCLK} - t_{sh}$$

Avec

$t_{\overline{BPRN}} - t_{\overline{BPRO}}$ = temps de propagation du contrôleur Multibus = 20ns max

t_{Belk} = périodicité de Belk = 100ns dans ce cas.

t_{sh} = temps de passage d'un état à un autre état = 40ns.

Dans ce cas 3 modules maître sont autorisés, pour augmenter ce nombre il suffit d'augmenter la période Belk (voir Fig. a).

- Gestion parallèle des priorités.

Cette méthode nécessite des circuits extérieurs de gestion des priorités. Ici \overline{BRE} est utilisé au lieu de \overline{BPRO} . Quand un ou plusieurs modules maîtres désirent le contrôle du Multibus, ils positionnent leur ligne \overline{BRE} au niveau actif.

La priorité est résolue par l'encodeur de priorité qui par l'intermédiaire du décodeur active la \overline{BPRN} du module maître de plus forte priorité. Le temps de propagation est nettement inférieur à la période de Belk, ce qui permet d'avoir 8 modules maîtres sur le Multibus avec un encodeur 8 entrées, 3 sorties et un décodeur (3 - 8).

Pour augmenter le nombre de modules maître sur le Multibus, il faut rajouter des encodeurs et des décodeurs sans dépasser le facteur de charge du Multibus (voir Fig.b).

3°) Exemples d'utilisation du Multibus dans un système multiprocesseur. Plusieurs CPU SEC peuvent partager le Multibus et exécuter en parallèle les tâches spécifiques d'un système.

Exemple : deux maître et un esclave.

le premier maître traite une opération d'E/S en utilisant son propre bus pendant que le deuxième traite des données qu'il a déjà fait rentrer, dans ce cas le Multibus n'est pas utilisé.

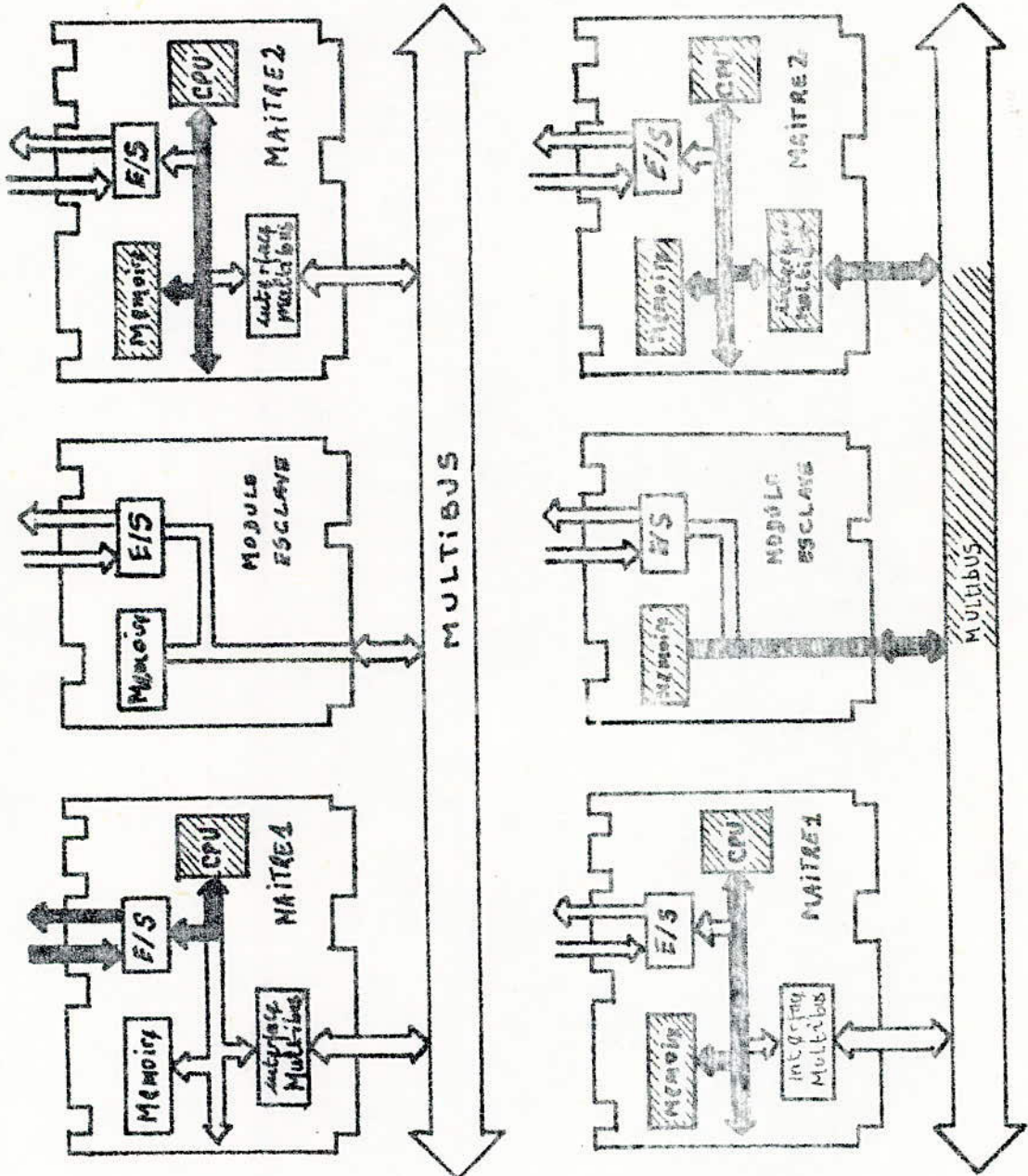
Dans une deuxième phase le 2eme maître range ses données traitées dans la mémoire commune située sur le module esclave, donc il utilise le Multibus, pendant ce temps le 1er maître traite les données rentrées précédemment.

Donc un module maître utilise le Multibus pendant qu'un ou plusieurs font des traitements. (Fig.3)

FIG 3

CAS N° 1 du traitement parallèle

CAS N° 2 du traitement parallèle



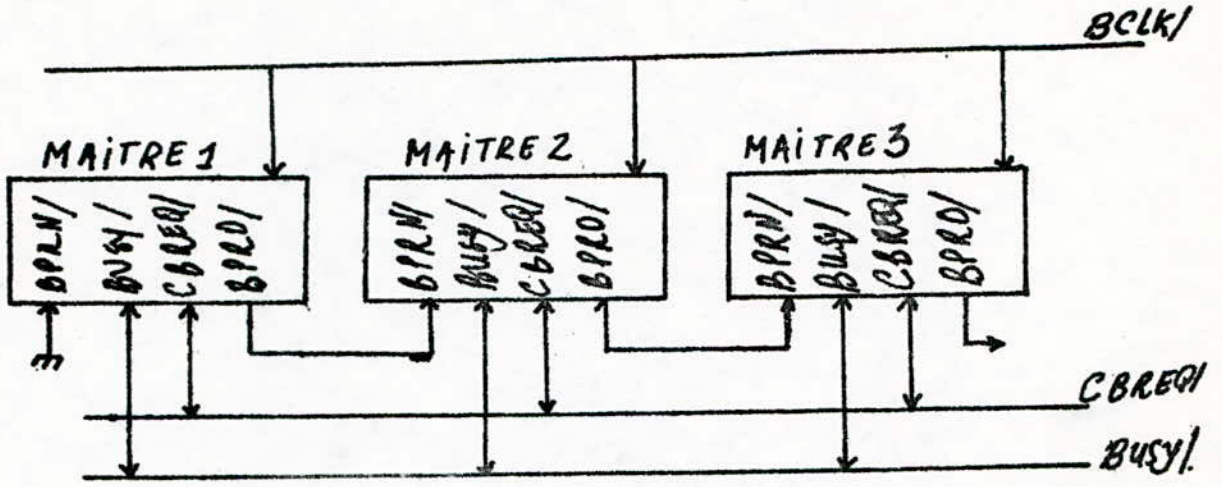


fig 1a Gestion Serie des Priorités

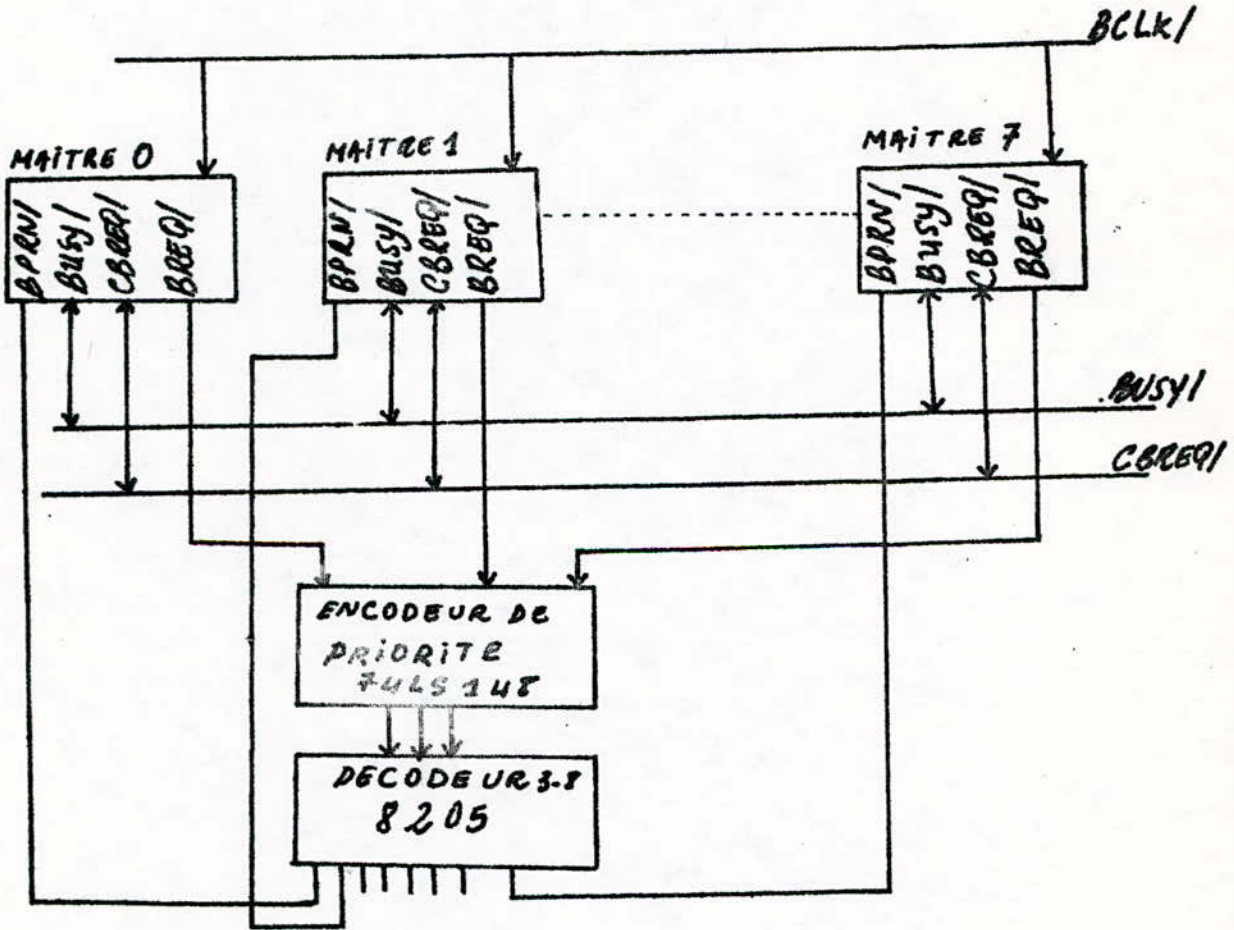


fig 2b GESTION parallele des Priorités
FIG 2

Lors de l'utilisation du Multibus dans un environnement multiprocesseur des considérations sont à prendre en compte : - l'exclusion mutuelle

-
- la communication
- la synchronisation.

a) - l'exécution mutuelle :

C'est une possibilité logicielle qu'a chacun des processeurs de pouvoir verrouiller le Multibus pendant une section critique du programme. Cette fonction est assurée par l'intermédiaire de la ligne BUSY.

Si BUSY = 0 le Multibus est verrouillé.

b) - La communication.

Deux programmes s'exécutant sur deux processeurs différents peuvent s'échanger des données à travers une zone mémoire commune située sur le Multibus. L'exclusion mutuelle est utilisée pendant ces échanges pour les protéger.

c) La synchronisation.

C'est un cas spécial de communication, durant laquelle aucune donnée n'est transférée, l'exclusion mutuelle est nécessaire dans ce cas. Par exemple un programme peut se mettre en attente d'un signal de synchronisation provenant d'un autre processeur.

Ces fonctions de communication et de synchronisation sont intégrées sur le SBC 544 (Contrôleur intelligent de communication).

Ce qui permet une simplification des dialogues sur le Multibus. Le Multibus est considéré le plus performant, il peut relier un ordinateur à plusieurs micro-ordinateurs, comme il peut avoir son importance dans l'instrumentation, ou sa rapidité d'échanges d'information et son mode de travail synchrone.

B) Le bus VME (Versa Module Eurocard)

I) , Introduction.

II) Organisation du bus VME

III) Son utilisation.

I) - Introduction

Le bus VME est une nouvelle structure de bus pour les systèmes 8, 16 et 32 bits. C'est un bus multiprocesseur permettant à plusieurs maîtres de partager des ressources communes (mémoires, E/S).

Il dispose de 4 niveaux d'accès, 7 niveaux d'interruption parallèles centralisés ou répartis entre les différents processeurs. Sur chaque niveau d'interruption on peut connecter plusieurs niveaux série.

- Il est construit autour du concept maître-esclave: le maître a le contrôle du bus, tandis que l'esclave, après décodage de l'adresse le concernant, répond à la commande envoyée par le maître.

- Sa nature asynchrone permet la cohabitation de processeurs, mémoires, et de périphériques présentant des caractéristiques de vitesses différentes. Electriquement le bus se répartit sur deux connecteurs P_1 et P_2 de 96 broches.

- le connecteur P_1 dispose de :

- . un bus de données 16 bits
- . un bus d'adresses 24 lignes
- . 6 lignes de modification d'adresse
- . Diverses lignes d'alimentation et de contrôle

* Demande d'accès au bus

* Prise en compte de la demande d'accès au bus

* Demande d'interruption

* Prise en compte de la demande d'interruption.

En plus du bus parallèle, le VME possède un canal de communication série qui assure en fonctionnement multiprocesseur, des échanges de signaux de service ou d'alarme.

Le connecteur P_2 contient des lignes d'extension à 32 bits des bus de données et d'adresses, ce qui permettra des transferts 32 bits entre la mémoire et des coupleurs performants les broches restantes sont réservées à l'utilisateur.

II) - Organisation du bus.

Les fonctions d'interface du bus sont au nombre de quatre :

- 1) Transfert de données
- 2) Demandes d'accès au bus
- 3) Gestion des interruptions
- 4) Signaux de contrôle

-1) Transfert de données

Le transfert de données, s'effectue par l'intermédiaire d'un bus spécialisé (DTB) comprenant :

LES CONNECTEURS 96 BROCHES DU BUS VME

CONNECTEUR P1

Bus d'adresses	A01 à A25	Domaine d'adressage 16 Moctets.
Bus de données	D0 à D15	Bus données 16 bits
Contrôle d'adresse	\overline{AS} A10 à A15	Echantillonnage d'adresse Modification d'adresse
Contrôle de donnée	$\overline{DS0}$, $\overline{DS1}$ \overline{WRITE} \overline{DTACK} \overline{BERR}	Sélection d'octets. Lecture / écriture. Données disponibles. Erreur BUS.
Interruptions	$\overline{IRQ1}$ à $\overline{IRQ7}$ \overline{IACK} , \overline{IACKIN} , $\overline{IACKOUT}$	Demandes d'interruptions Prise en compte d'interruptions
ACCES AU BUS	\overline{BRO} à $\overline{BR3}$ $\overline{BG0,3IN}$, $\overline{BG0,3OUT}$ \overline{BSY} \overline{BCLC}	Demandes d'accès au BUS. Autorisations d'accès au BUS. BUS occupé demande plus prioritaire
Signaux de Contrôle	Sys CLK $\overline{SysReset}$ $\overline{SysFail}$ \overline{ACFail}	Horloge système Initialisation du système Pause système Coupeure d'alimentation
Bus Serie	SERDAT, SERCLK	Donnée et horloge Serie
Alimentation	+5V, ±12V, GND +5V repos.	

CONNECTEUR P2

Extension du Bus Adresse	A24 à A31	Extension à 46 octets
Extension du Bus de données	D16 à D31	Extension à 32 bits.
Alimentation	+5V, GND.	
ENTRÉES / SORTIES	RESERVÉES à L'utilisateur	64 Broches.

- le bus d'adresses (A_{01} à A_{31})
- 6 Lignes de modification d'adresse ($AM0$ à $AM5$)
- le bus de données (D_{00} à D_{31})
- 5 lignes de contrôle de transfert par le maître (\overline{AS} , $\overline{DS0}$, $\overline{DS1}$, \overline{LWORD} , \overline{WRITE})
- 2 lignes d'état contrôlées par l'esclave (\overline{DTACK} , \overline{BERR}).

2) Demandes d'accès au bus.

Le bus VME dispose de 4 niveaux de priorités d'accès au bus assignés :

- . Statiquement : Priorité fixe.
- . Dynamiquement : Priorité tournante.

A quatre lignes spécialisées $\overline{BR0}$ à $\overline{BR3}$

Le niveau de priorité d'une carte maître dépend du niveau de priorité \overline{BR} à laquelle elle est connectée et de sa position physique à l'intérieur du châssis.

Chaque ligne \overline{BR} est connectée à des maîtres par des "OU" câblés de cette manière plusieurs maîtres peuvent partager la même ligne \overline{BR} .

A chaque ligne \overline{BR} correspond une ligne \overline{BIB} d'autorisation d'accès au bus qui relie en "daisy chain" tous les maîtres associés à la ligne \overline{BR} et alloue le bus par simple priorité sérielle.

* Unité de gestion d'accès au bus.

Cette unité est localisée dans la position 1 du châssis.

Elle établit une hiérarchie des demandes d'accès au bus et accorde le bus à la demande la plus prioritaire selon le type de priorité utilisé (Fig. c)

. **Priorité fixe** : l'unité de gestion reçoit les 4 lignes de demande d'accès au bus ($\overline{BR0}$ à $\overline{BR3}$) auxquelles sont allouées des niveaux de priorité fixe. Elle alloue le bus à la demande la plus prioritaire et génère un niveau bas sur la broche \overline{BGIN} correspondante, cette dernière est conditionnée par la disponibilité du bus signalée par un niveau haut sur \overline{BBSY} . La broche \overline{Belk} signale au maître en possession du bus qu'une demande plus prioritaire est en attente.

. **Priorité tournante** : l'unité de gestion reçoit les 4 lignes de demande d'accès au bus, dans ce cas le maître qui a le niveau de priorité le plus élevé, aura, après accès au bus le niveau de priorité le plus bas. Dans ce cas \overline{Belk} est supprimée.

Carte Localisée dans la position 1 du fond de panier

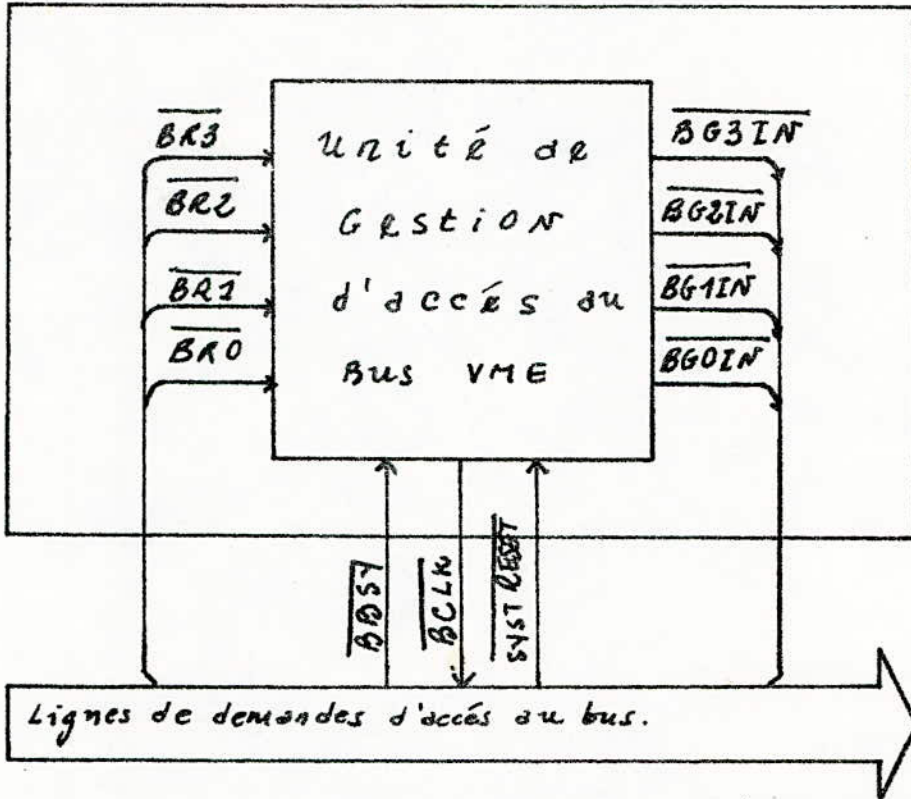


fig C : Unité de gestion d'accès au bus DTB avec priorité d'accès fixe.

La ligne \overline{BCLK} est supprimée dans le cas de priorité d'accès tournante.

* Unité de demande d'accès au bus.

Chaque carte maître possède une unité de demande d'accès au bus, qui peut fonctionner selon deux modes.

- le mode RWD (Release when done) : Dans ce mode, le maître libère le bus dès qu'il n'en a plus besoin.

- le mode ROR (Release on request) : Dans ce mode, le maître ne restitue le bus, après utilisation, que sur demande, ce qui permet de diminuer le nombre de demandes faites par un maître. Le signal BBSY est porté à l'état bas dès que le maître prend en charge le bus.

Le maître doit gérer deux signaux $\overline{\text{AcFail}}$ et $\overline{\text{Belk}}$ qui indiquent au maître en possession du bus, que des demandes plus prioritaires sont en attente.

3) - Gestion des interruptions.

Deux types de systèmes de gestion des interruptions peuvent être envisagés:

. un système de gestion centralisée: où la gestion est confiée à un seul processeur, qui va traiter les demandes d'interruptions.

. Un système de gestions distribuée: les demandes d'interruption sont réparties sur plusieurs processeurs, et leur traitement est confié à chacun d'eux.

* Les signaux utilisés dans les demandes d'interruptions.

Un système de gestion des demandes d'interruption comprend:

. 7 lignes de demande d'interruption ($\overline{\text{IRQ1}}$ à $\overline{\text{IRQ7}}$)

. 1 ligne de prise en compte d'interruption ($\overline{\text{JACK}}$)

. 2 lignes d'établissement de priorité serielle $\overline{\text{JACKIN}}$, $\overline{\text{JACKOUT}}$

Chaque ligne $\overline{\text{IRQ}}$ est activée par un niveau bas généré par l'unité de demande d'interruption.

La détermination de la source la plus prioritaire se fait par simple priorité serielle à partir de la ligne $\overline{\text{JACK}}$, qui, connectée à $\overline{\text{JACKIN}}$ du connecteur relie en série les différentes sources d'interruptions par l'intermédiaires de $\overline{\text{JACKIN}} / \overline{\text{JACKOUT}}$.

* L'unité de gestion des interruptions.

Quelque soit le type de gestion adopté, les rôles d'une unité de gestion sont les suivants:

- Elle détermine la priorité la plus élevée.

- Elle utilise l'unité de demande d'accès au bus pour faire une demande, si elle accordée, elle génère un signal $\overline{\text{JACK}}$.

- Elle lit le vecteur d'interruption et initialise la séquence de traitement de l'interruption.

* L'unité de demande des interruptions

Elle assure les fonctions suivantes :

. Elle adresse une demande d'interruption à l'unité de gestion des interruptions associée à la ligne de demande d'interruption utilisé.

. Elle fournit un vecteur d'interruption, lorsqu'elle reçoit un signal de prise en compte de sa demande.

. Elle transmet le signal de prise en compte, si elle n'est pas la source de demande d'interruption.

4) Les signaux de contrôle

Ils comprennent :

- . AcFAIL : coupure de l'alimentation secteur
- . SysRESET : Initialisation suite à une mise en route.
- . Sysclk : horloge du système (16 Mhz)
- . SysFAIL : panne du système.

(III) L'utilisation du Bus. VME.

1) Fonctionnement du bus dans un environnement multiprocesseur pour exécuter un transfert dans un environnement multiprocesseur, le maître concerné doit d'abord acquérir la maîtrise du bus, à cette fin, il fait une demande d'accès à l'unité de demande d'accès qui lui est associée, celle-ci transmet la requête du maître à l'unité de gestion des accès au bus, qui détermine compte tenu des demandes formulées par les différents processeurs, celui qui a la priorité la plus élevée, et qui disposera de la maîtrise du bus dès que ce dernier sera libre.

Un niveau haut sur BBSY signale que le maître en possession du bus va le libérer à la fin du dernière transfert, la libération du bus implique deux phases.

- . Libération de toutes les lignes sauf AS portée à l'état haut
- . Libération de AS après un temps déterminé.

2) Cycle de lecture / modification / écriture.

Dans un système multiprocesseur, certaines ressources sont communes, il est indispensable qu'un maître qui a besoin d'une ressource puisse savoir si elle est libre, et se la réserver ou non.

A chaque ressource est donc associé un bit de la mémoire (sémaphore) qui caractérise l'état de la ressource.

Si le bit est à "1" ———— ressource occupée.

Si le bit est à "0" ———— ressource libre.

Dans ce cas le maître demandeur positionne le bit à 1 pour se réserver la ressource.

Une instruction spécialisée, permet de faire cette opération qui implique un cycle lecture / modification / écriture.

3) Opération de gestion de l'accès au bus.

Etudions successivement les deux cas suivants :

a) deux maîtres possédant des niveaux d'accès différents.

Deux demandes arrivent simultanément sur $\overline{BR1}$ et $\overline{BR2}$, le maître B étant le plus prioritaire, la maîtrise du bus lui revient. L'unité de gestion valide $\overline{BG2IN}$, dès que B reçoit l'autorisation d'accès \overline{BBSY} est mis à l'état bas indiquant l'occupation du bus, ensuite elle libère $\overline{BR2}$ et informe le maître B que le bus est à sa disposition, et libère $\overline{BG2IN}$.

Quand, le maître a fini ses transferts, son unité de demande d'accès remet \overline{BBSY} à l'état haut à condition que $\overline{BG2IN}$ soit à l'état haut.

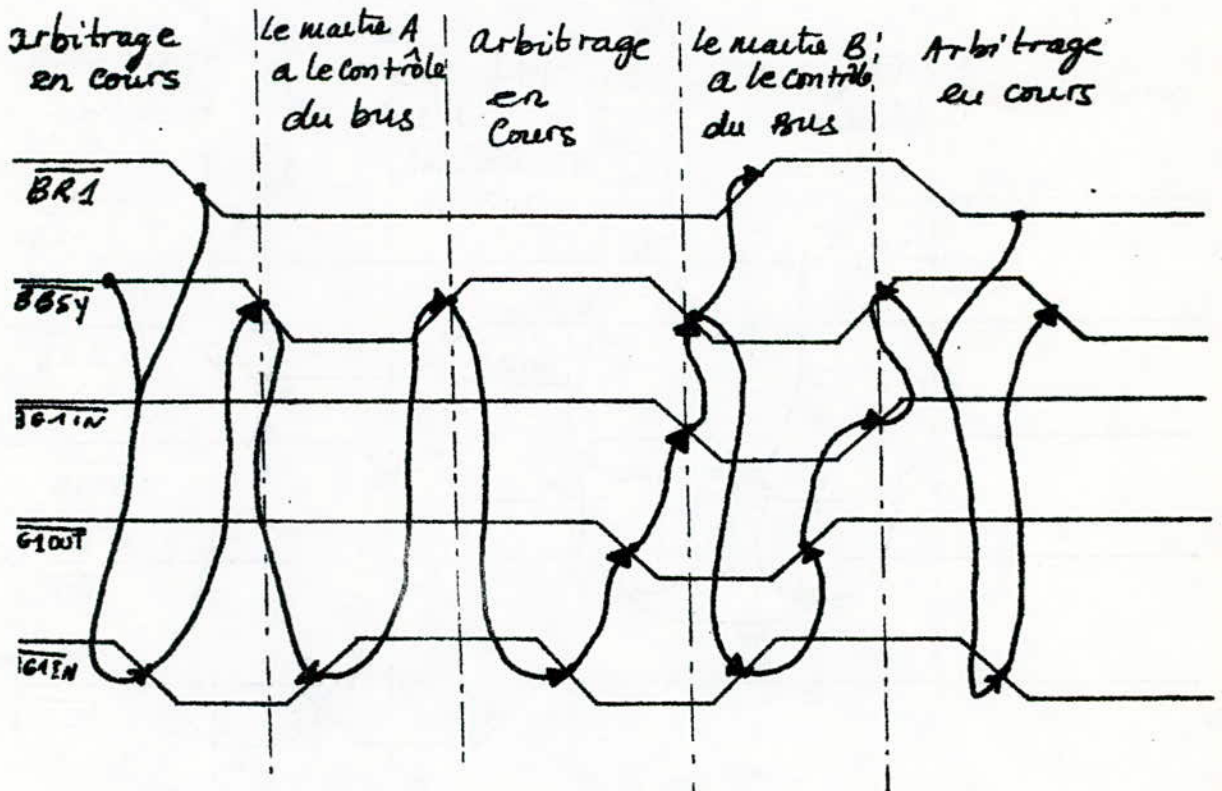
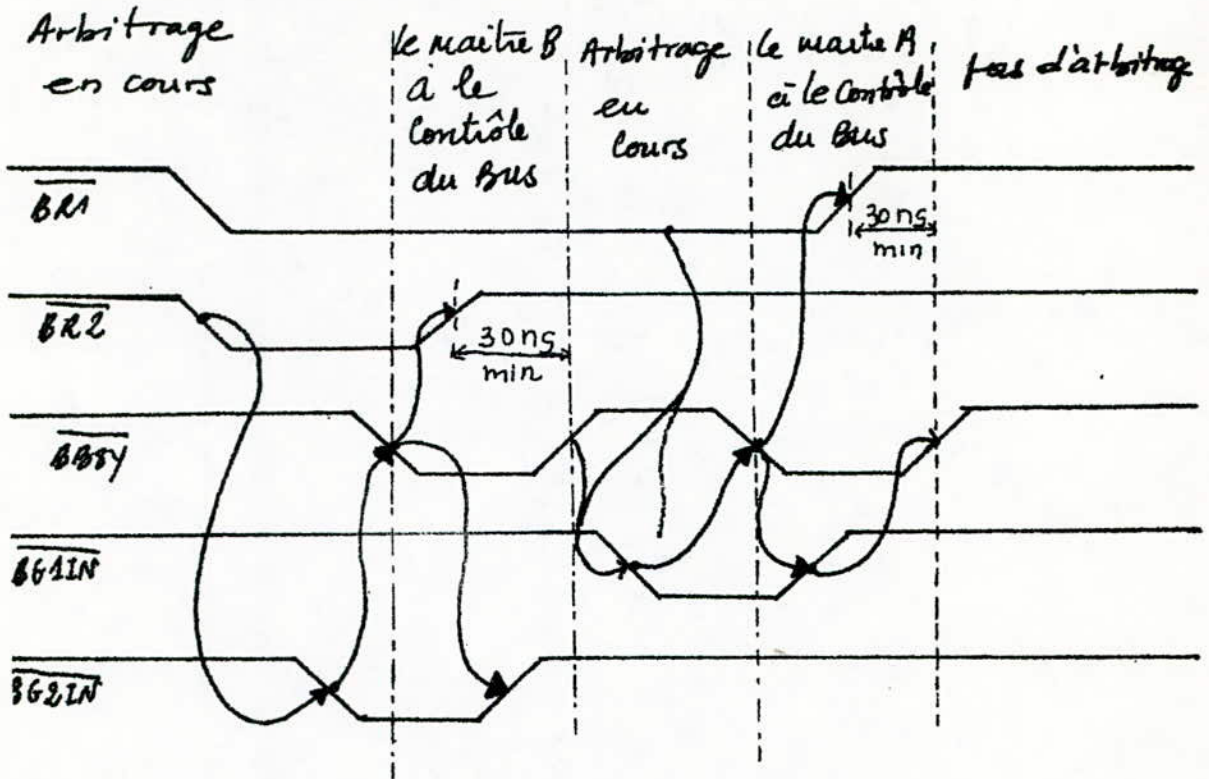
L'unité de gestion constate que le bus est libre, l'accorde au maître A en portant sa broche $\overline{BG1IN}$ à l'état bas et la même opération recommence (chronogramme - 1).

b) Deux maîtres ayant le même niveau de priorité.

Dans ce cas, la priorité physique est prise en compte, c'est-à-dire que la carte dans la position (1) du fond de panier est prioritaire. Le maître A (sur la position 1) travaille en mode RWD c-à-dire qu'il libère le bus dès la fin du transfert et le maître B travaille la mode ROR c-à-dire qu'il ne libère le bus, après transfert, que sur demande.

L'unité de gestion d'accès au bus se trouve sur la carte maître A, elle reçoit les deux demandes simultanément sur la broche $\overline{BR1}$, \overline{BBSY} étant à l'état haut, l'unité de gestion génère un état bas sur $\overline{BG1IN}$ indiquant une autorisation d'accès au bus. Dès réception de cette autorisation, le maître A se réserve le bus en portant \overline{BBSY} à l'état Bas.

Chronogramme 1



Chronogramme 2

Son unité de demande d'accès au bus, libère \overline{DR}_1 , qui reste cependant à l'état bas puisque la demande formulée par le Maître B reste présente. Quand l'unité de gestion détecte un état bas sur \overline{EBSY} , elle remet $\overline{BG1III}$ à l'état haut. Quand le maître A termine son transfert, son unité de demande d'accès au bus remet \overline{EBSY} à l'état haut.

Si le maître A n'a plus besoin du bus, il active $\overline{BG1OUI}$ qui à son tour active $\overline{BG1II}$ du maître B et le processus recommence (voir chronogramme - 2)

En conclusion, le bus VME a été conçu spécialement pour le microprocesseur 68000, il reprend ses particularités (exemple absence du bit d'adresse A00).

Le bus VME pourra être adapté à des microprocesseurs autres que le 68000 mais au prix de logiques supplémentaires d'adaptation et de dégradation de performances.

Un fonctionnement du Multiprocesseur ne sera efficace que si le bus VME n'est pas surchargé, c-à-dire si chaque maître ne sollicite pas trop souvent le bus pour qu'il en soit ainsi, il sera nécessaire que chaque maître ait une certaine autonomie, ce qui implique l'existence sur la carte d'un bus local reliant le maître à des mémoires et à des coupleurs locaux.

C) Le bus Exorciser.

Le bus Exorciser, permet d'interfacer le module MPU avec d'autres modules il est localisé sur la carte principale de l'Exorciser. Il permet, grâce à sa flexibilité, de traiter des applications spécifiques (voir tableau) le module MPU adresse ses périphériques comme étant des mémoires. Pendant une opération de lecture ou écriture l'adresse de la mémoire allouée est placée sur le bus où elle sera reconnue par l'Exorciser.

Les signaux de commandes et d'horloge utilisés pour l'exécution d'une opération de transfert sont véhiculés par ce bus.

Il est possible qu'un autre module bénéficie du contrôle du bus, il lui suffit de placer G/H à l'état bas.

Le module MPU termine l'exécution de l'instruction en cours et met BA à l'état haut.

Le module demandeur porte la ligne \overline{TSC} à l'état bas forçant ainsi les lignes d'interface du bus d'adresse du MPU à l'état haute impédance. Dès que le demandeur n'a plus besoin du bus, il supprime le signal \overline{TSC} , après une impulsion de ϕ_2 il libère la ligne G/H et le contrôleur revient au module MPU.

L'insertion de 14 cartes dans l'Exorciser est possible, l'adressage de ces cartes est indépendant de leur position dans le fond de panier.

Les 18 lignes non utilisées du bus permettent à l'utilisateur de l'adapter à ses besoins.

Le bus Exorciser est utilisé aussi bien par le microprocesseur MC 6800 que par le MC 6809.

Conclusion :

Le bus du système de développement Exorciser utilisé au C.E.N, ne pose aucun problème lorsqu'il est utilisé comme bus monoprocesseur.

On peut l'utiliser en bus multiprocesseur en gérant les conflits d'accès au bus. Les différents signaux nécessaires à cette gestion sont véhiculés par les lignes non utilisées de ce bus.

Tableau représentant les signaux du bus Exerciser

N° de la broche	MOMENTANÉ du signal	Description du signal
A B C	+5 ^v DC +5 ^v DC +5 ^v DC	Alimentation +5 ^v DC .
D	\overline{IRQ}	Demande d'interruption: Ce signal demande que la séquence d'interruption du MPU soit générée à l'intérieur de la machine. Le MPU doit attendre jusqu'à ce qu'il termine l'instruction en cours d'exécution pour qu'il prenne en compte cette demande. A ce moment, si le bit de masquage d'interruption dans le registre code condition du MPU n'est pas positionné à l'état actif, le MPU peut commencer la séquence d'interruption.
E	\overline{NMI}	Interruption non-masquable: ce signal demande qu'une interruption non-masquable soit générée à l'intérieur de la machine. Le MPU doit attendre jusqu'à ce qu'il termine l'exécution de l'instruction en cours pour qu'il prenne en compte cette interruption. A ce moment, le MPU doit commencer la séquence d'interruption non-masquable.

F	VMA	validation d'adresse mémoire: cette sortie indique au module DEBug qu'il y a une adresse validée sur le bus adresse. Ce signal n'est pas à trois états.
H		non utilisé
J	ϕ_1	phase (1) de l'horloge du MPU
K	GND	Masse pour $\pm 12V$ DC
L	MEM CLK	horloge mémoire: ce signal est utilisé par le MPU pour générer les signaux d'horloge ϕ_1 et ϕ_2 . ce signal est aussi utilisé par les modules mémoires dynamiques
M	-12V DC	-12V DC est utilisé avec des circuits d'interface courants
N	\overline{TSC}	contrôle trois états: cette entrée, à l'état haut elle fait passer les lignes d'adresses à l'état haute impédance. les signaux VMA et BA sont forcés à l'état bas. Le bus de données n'est pas affecté par cette entrée.
P	BA	Bus disponible: cette sortie est généralement à l'état bas. quand elle est activée. c.a.d quand elle est à l'état haut, elle indique que le MPU est à l'arrêt et que le bus d'adresses est disponible. ceci se produit quand la ligne $\overline{H\overline{A}I\overline{T}}$ est à l'état bas ou le MPU est dans l'état WAIT (attente d'interruption) après l'exécution de l'instruction WAIT. En même temps toutes les sorties trois états sont mises dans l'état haut impédance et les autres sorties à leur niveau normal de repos.
R	MEM RDY	mémoire prête: ce signal autorise l'Exorciser de travailler avec des mémoires lentes. quand il est à l'état bas, il inhibe l'Exorciser.

S	REF CIR	Horloge de rafraichissement : ce signal est g�n�r� par les modules m�moire dynamique qui seront consid�r�s comme modules ma�tres de rafraichissement. ce signal est utilis� pour initialiser une op�ration m�moire sur les modules m�moire dynamique.
T	+12V DC	+12V DC utilis� avec des circuits d'interface courants
U	BAT, +12V	Batterie +12V : cette ligne, dans une op�ration EXORCISE normale c'est un +12V DC � partir de la batterie fond de panier de l'exorciser. Si l'exorciser utilise cette batterie et qu'elle est coup�e ou bien un affaiblissement de puissance se produit, cette ligne est positionn�e � +12V DC.
V	STDBY	�tat d'attente : cette ligne est � l'�tat bas durant un �tat d'affaiblissement de puissance et � l'�tat haut pour un fonctionnement normal de l'exorciser.
W X Y Z A		Non utilis�s.
B	GND	la masse
C D		non utilis�s.
E	BA	Bus disponible : cette ligne est pr�sente seulement quand un module d'�valuation est plac� dans le ch�ssis de l'exorciser.
F	GIH	GO-HALT : cette ligne est utilis�e seulement quand un module d'�valuation est plac� dans le ch�ssis de l'exorciser.
H	D ₃	D ₃ du bus de donn�es : cette ligne bidirectionnelle, quand elle est autoris�e, elle provoque 2 voies de transfert de donn�es, entre le MPU et l'ensemble des modules plac�s dans l'exorciser.

\bar{J} \bar{R} \bar{L}	\bar{D}_7 \bar{D}_2 \bar{D}_6	Bus de données identique à \bar{D}_3 sur \bar{H}
\bar{M}	A_{14}	Bus du bus d'adresses: cette ligne d'adresse quand elle est autorisée, elle transfère les sorties du compteur programme des MPU pour adresser les modules situés dans le châssis de l'exerciser.
\bar{N}	A_{13}	Bus d'adresses: identique à A_{14} sur \bar{M}
\bar{P}	A_{10}	« «
\bar{R}	A_9	« «
\bar{S}	A_6	« «
\bar{T}	A_5	« «
\bar{U}	A_2	« «
\bar{V}	A_1	« «
\bar{W} \bar{X} \bar{Y}	GND GND GND	MASS.
1 2 3	+5 ^V DC " "	Alimentation +5 ^V DC
4	G1H	GO-HALT: quand cette ligne est à l'état haut, le MPU doit aller chercher l'instruction adressée par le compteur programme et le début d'exécution. quand elle est à l'état bas, toutes les activités du MPU sont à l'arrêt, la ligne DA passe à l'état haut, VMA à l'état bas, et l'ensemble des autres lignes à l'état haute-impédance. Cette ligne doit passer à l'état bas avec le front montant de la phase ϕ_1 pour assurer l'exécution d'une seule instruction.
5	RESET	remise à l'état initial: Ce signal est utilisé pour initialiser le MPU MC 6800 et aussi l'initialisation de l'exerciser après une panne d'alimentation ou bien quand l'interruption de redémarrage de l'exerciser est active.

6	$R\bar{W}$	lecture / écriture : cette sortie du MPU indique aux modules dans l'exerciser que le MPU HC 6800 est soit dans l'état de lecture ($R\bar{W}=1$) soit dans l'état d'écriture ($R\bar{W}=0$)
7	ϕ_1	phase (1) de l'horloge
8	GND	masse
9	GND	
10	VVA	validation de l'adresse utilisateur : A l'état haut, ce signal indique que l'adresse sur le bus d'adresses est validée et que l'exerciser n'adresse pas son programme EXBUG
11	-12VDC	-12VDC utilisés avec des circuits d'interface courants
12	\overline{REFREQ}	demande de rafraîchissement : ce signal, à l'état bas, initialise un cycle de rafraîchissement mémoire pour un module mémoire dynamique. Pendant l'opération de rafraîchissement le MPU est inhibé
13	REF GRANT	Autorisation de rafraîchissement : ce signal, à l'état haut, autorise les modules mémoires dynamiques de rafraîchir leur mémoire.
14		non utilisés
15		
16	+12VDC	+12VDC
17	BAT _{+12V}	Batterie +12VDC
18	\overline{TSC}	contrôle trois états : cette ligne est active que lorsqu'on a des modulations d'évaluations dans le châssis de l'exerciser.
19		non utilisés
20		
21	AC OFF	AC OFF : ce signal indique quand l'alimentation alternative est supprimée

22 23		Non utilisés
24	GND	Masses
25 26 27 28		Non utilisés
29	\bar{D}_7	\bar{D}_7 du bus de données identique à \bar{D}_3
30	\bar{D}_5	\bar{D}_5 du bus de données identique à \bar{D}_3
31	\bar{D}_0	\bar{D}_0 du bus de données identique à \bar{D}_3
32	\bar{D}_4	\bar{D}_4 du bus de données identique à \bar{D}_3
33	A_{15}	A_{15} du bus d'adresses identique à A_{14}
34	A_{12}	A_{12} du bus d'adresses identique à A_{14}
35	A_{11}	A_{11} du bus d'adresses identique à A_{14}
36	A_8	A_8 du bus d'adresses identique à A_{14}
37	A_7	A_7 " " "
38	A_4	A_4 " " "
39	A_3	A_3 " " "
40	A_0	A_0 du bus d'adresses identiques à A_{14}
41	GND	Masses
42	GND	Masses
43	GND	Masses

CHAPITRE II

Introduction :

Dans une structure multiprocesseur il est souvent nécessaire de partager une ou plusieurs ressources communes du type mémoire, celles-ci sont accessibles par le biais d'un bus, on peut donc les assimiler au partage du bus qui les supporte.

Il est donc nécessaire de prévoir une carte allocateur qui aura pour tâche la gestion des priorités d'accès au bus afin de servir de façon égale tous les processeurs du système.

La tâche de l'allocateur est de déterminer le processeur qui accedera à la ressource commune.

1) Techniques de résolution des priorités.

Il existe deux types qui sont couramment utilisés :

- * Technique de résolution de priorités fixes.
- * Technique de resolution de priorités tournantes.

A. 1 * Technique de résolution de priorités fixes.

Dans cette technique, les niveaux de priorités sont fixes et assignés statiquement aux différents éléments du système.

Le processeur dont la ligne de demande est reliée au niveau de priorité le plus haut est toujours prioritaire par rapport aux autres. L'ordre de priorité peut passer d'un processeur à l'autre par liaison en chaîne (résolution série) ou à l'aide d'un encodeur de priorité (résolution parallèle).

Cette technique a l'avantage de la simplicité mais présente l'inconvénient d'un partage inéquitable des ressources communes (le processeur le plus prioritaire peut monopoliser le bus).

A.2 * Technique de résolution des priorités tournantes.

Dans ce type de gestion le niveau de priorité de chaque processeur est assigné dynamiquement. Pour cela on dispose d'un circuit logique qui affecte les priorités de manière rotative, c'est-à-dire le processeur qui vient d'accéder au bus aura ensuite le niveau de priorité le plus bas.

Cette technique est plus intéressante que la précédente vu qu'elle permet à tous les processeurs du système d'utiliser le bus à un instant donné quelque soit leur vitesse de traitement, seulement elle nécessite l'implantation d'une logique complexe.

2) Allocation de la ressource bus - Arbitrage.

2 - 1 Allocation déterministe.

Un élément, toujours le même, est pilote du bus, il consulte à tour de rôle les autres éléments qui fonctionnent comme copilotes. Le bus est attribué successivement à chacun d'eux pour un délai fixe et limité, le bus est alloué systématiquement même si un élément n'a rien à émettre.

Les cadences de transfert ~~ne sont bonnes que si les activités de transfert de chaque élément sont régulières et que si l'allocation du temps d'utilisation du bus est proportionnelle à l'activité relative de chaque élément.~~

2. 2. Allocation centralisée gérée par un arbitre.

Les éléments adressent leurs requêtes à un arbitre qui choisit un pilote du bus et un seul parmi les demandeurs.

Il existe diverses méthodes d'allocation centralisée.

a) Chaînage.

Tout élément qui demande le bus active un signal de requête

Un signal "bus libre" est envoyé sur le bus par l'arbitre et passe successivement devant tous les éléments, le premier élément rencontré avec une requête présente se voit allouer le bus, il active une ligne "bus occupé" et arrête la propagation du signal "bus libre" vers les éléments suivants. Cette technique nécessite peu de lignes de contrôle, en revanche elle est lente et favorise systématiquement les éléments voisins de l'arbitre (Fig.a-1)

b) Inspection.

La ligne requête est commune à tous les éléments, dès que cette ligne est activée, l'arbitre inspecte tous les éléments de la façon suivante :

Les éléments étant numérotés différemment et indépendamment de leurs positions sur le bus. Tout élément demandeur dont l'arbitre envoie le numéro reçoit le droit de piloter le bus, et active la ligne "bus occupé" l'inspection est stoppée et elle reprendra après libération du bus si d'autres requêtes sont présentes.

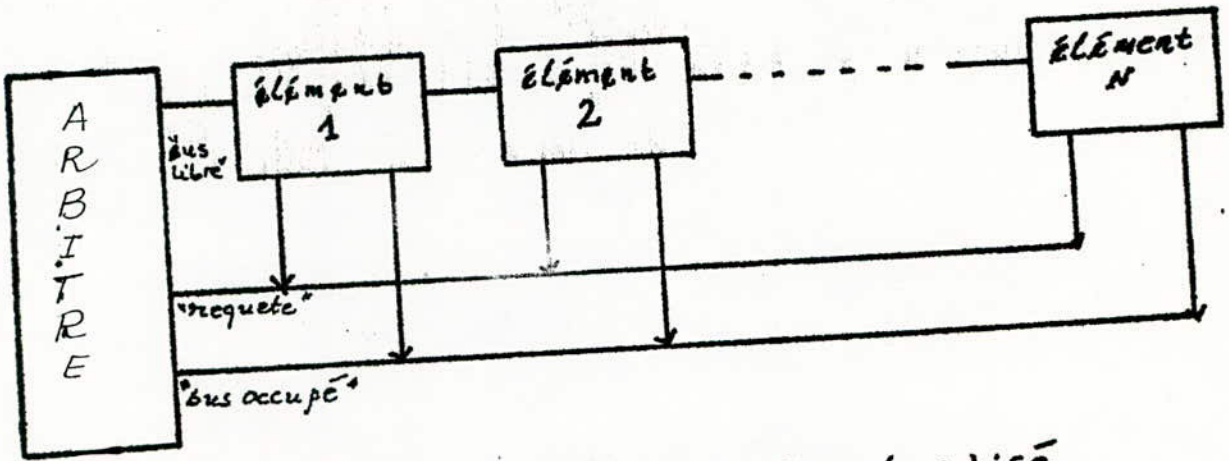


figure a.1 chainage centralisé

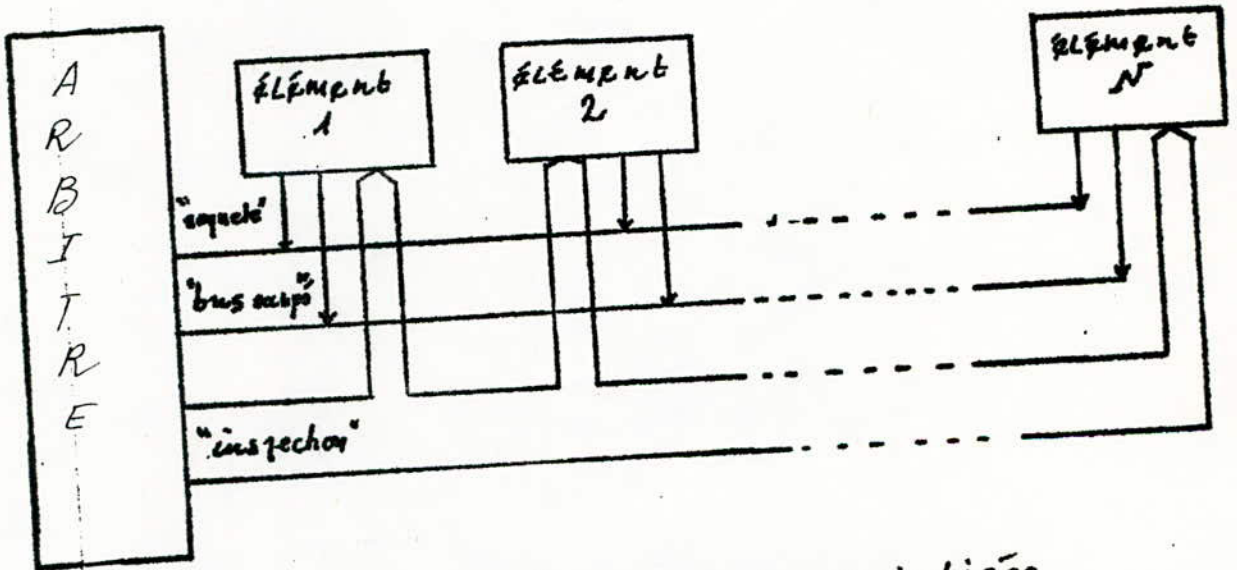


figure b.1 inspection centralisée

Cette technique est plus fiable que la précédente et la priorité des demandeurs n'est plus fixe (Fig.b-1)

c) Demandes indépendantes.

Chaque élément dispose d'une ligne requête et d'une ligne "alloué" la sélection est faite par l'arbitre. Dès que le bus est alloué à un élément et que celui-ci le pilote effectivement, il active une ligne commune "bus occupé"

L'avantage de cette méthode est une grande souplesse ainsi qu'un délai de service plus court, en revanche le nombre de lignes et de connecteurs nécessaires est plus élevé.

2.3 Allocation décentralisée - Arbitrage réparti.

Dans ce type d'allocation il n'y a plus d'arbitre. Les contraintes du bus doivent être respectées par chaque élément simultanément. Il n'existe pas de solution où tous les éléments auraient un rôle identique à tout instant.

On introduit une asymétrie momentanée en faisant circuler entre les éléments un jeton unique. L'élément qu'a ce jeton peut choisir soit d'être pilote soit de passer le jeton à son successeur dans la chaîne.

Donc, le problème consiste à s'assurer que le jeton circule bien entre tous les éléments, que les éléments le possèdent chacun un temps limité et ne le gardent pas en cas d'erreur.

La fiabilité du bus repose sur le bon usage du jeton. On trouve diverses méthodes pour réaliser cette technique.

a) Chaînage.

La ligne "bus libre" se reboucle sur elle-même et le signal est envoyé par l'élément qui libère le bus. Le bus est alloué successivement à ses demandeurs dans l'ordre de la boucle.

Ici aussi, la fiabilité dépend de celle de l'ensemble des éléments branchés sur le bus. (Fig.a-2)

b) Inspection.

Lorsqu'un élément a fini d'utiliser le bus, il sélectionne son successeur en envoyant un numéro sur les lignes de comptage et en émettant un signal "bus disponible". Si le successeur veut piloter le bus. Il émet un signal "bus accepté" sinon un autre successeur est choisi. Ainsi quand aucun élément ne veut piloter le bus, le dernier utilisateur doit inspecter en permanence tous les éléments du bus.

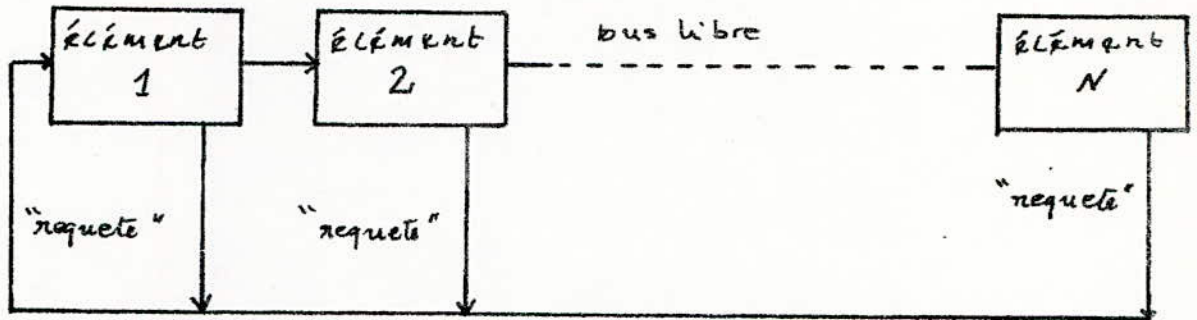


figure a-2 chainage décentralisé

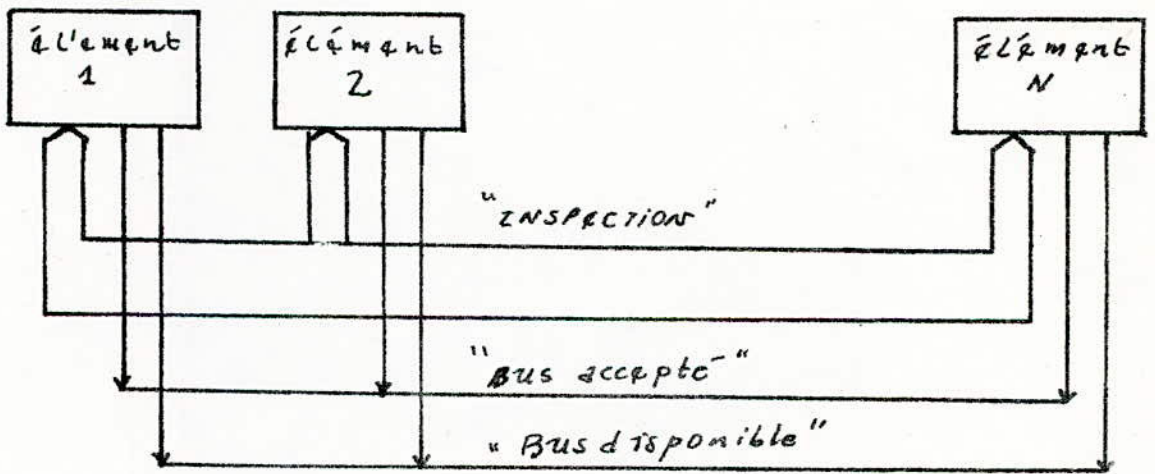


figure b-2. inspection décentralisée

Cette méthode a l'avantage de ne pas bloquer le jeton si l'élément suivant est en panne, mais elle est plus coûteuse car chaque élément doit pouvoir réaliser l'inspection des éléments du bus. (Fig.b-2).

c) Demandes indépendantes .

A chaque élément est attribué une ligne "requête" qu'il active en cas de besoin. Il existe une ligne "bus alloué" qui indique si le bus est occupé ou non. Dès que cette ligne est désactivée, chaque élément examine toutes les lignes "requête" et celui qui a la plus haute priorité prend le bus et active la ligne "bus alloué".

Cette technique est également coûteuse car elle nécessite une logique de choix dans chaque élément, et le nombre de connexions est élevé .

d) Partage de temps dynamiques.

On procède par partage de temps. La transmission est synchrone et chaque intervalle d'allocation est déterminé par un nombre fixe de signaux d'échantillonnage.

Les intervalles d'allocation peuvent être "libre" ou "utilisés".

Ils défilent devant les éléments qui utilisent les intervalles marqués "libre" pour y mettre un message accompagné de l'adresse de son destinataire et de la marque "utilisé".

Cette méthode a l'avantage d'augmenter la cadence de transfert. Il faut avoir une logique par élément car dès que l'un d'eux veut émettre, il doit examiner tous les intervalles jusqu'à en trouver un marqué "libre". De même cette technique est très sûre car la panne d'un élément n'a pas d'effet sur le bus, même quand l'élément en panne est pilote du bus.

Conclusion.

Notre choix s'est fixé sur une allocation centralisée, gérée par un allocateur travaillant selon une priorité tournante de manière à satisfaire les 3 cas de configuration qui peuvent se présenter à savoir :

- * les demandes arrivent simultanément et régulièrement.
- * le processeur sélectionné tarde à faire sa demande.
- * le processeur sélectionné ne fait pas de demande.

En plus on a prévu un transfert DMA réalisé par deux contrôleurs DMA MC6844.

CHAPITRE III.

I - ~~SC~~ Schéma synoptique : voir figure N° 1.

II - Description du synoptique.

* L'allocateur de bus.

C'est l'élément essentiel de la carte allocateur de bus (Fig.2).

Il reçoit les demandes d'accès formulées par les différents processeurs du système et doit être en mesure de définir le processeur prioritaire et cela en opérant selon une technique de résolution de priorité tournante (voir chapitre II) afin de satisfaire tous les processeurs du système.

* La logique de génération des requêtes (TXRQ)

Cette logique réalisée par un décodeur (3-8) qui est validé par un signal provenant du processeur esclave quand ce dernier veut faire un transfert par DMA.

* Les contrôleurs DMA (DMAC)

Pour une simplification logicielle, on a adapté le mode quatre canaux, chaque canal gérant les transferts DMA d'un processeur secondaire, comme notre système est constitué de 8 processeurs secondaires, deux DMAC sont donc nécessaires.

III. Les modes de transfert utilisés.

1) Mode de transfert programmé.

Dans ce cas, le processeur esclave autorisé à accéder au bus déclenche un programme d'interruption et un signal indiquant qu'une information est prête à être transférée.

Le maître répond par un acquittement. Dès que le processeur esclave a terminé son transfert il le signale au maître. Ce mode de transfert présente l'inconvénient d'être lent, vu le nombre d'instructions nécessaires à une opération de transfert et leur temps d'exécution.

Pour parer à cela, une solution est proposée c'est le transfert par DMA, qui constitue la 2e partie de notre travail.

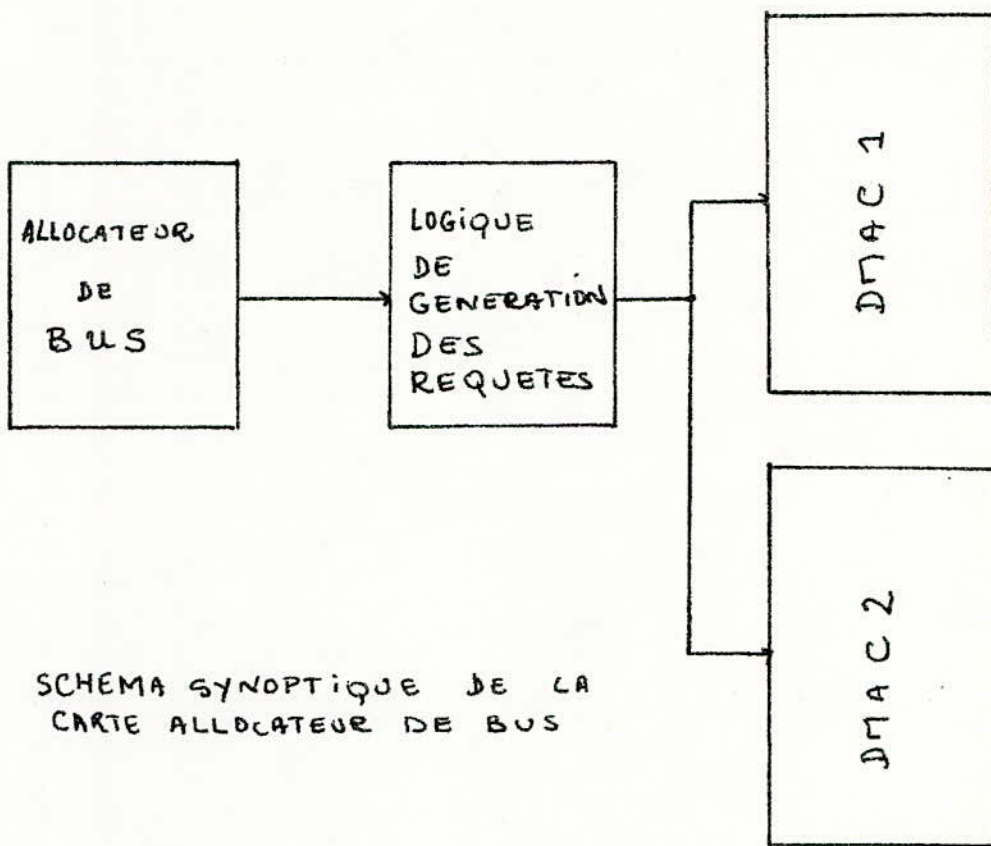


fig1. SCHEMA SYNOPTIQUE DE LA CARTE ALLOCATEUR DE BUS

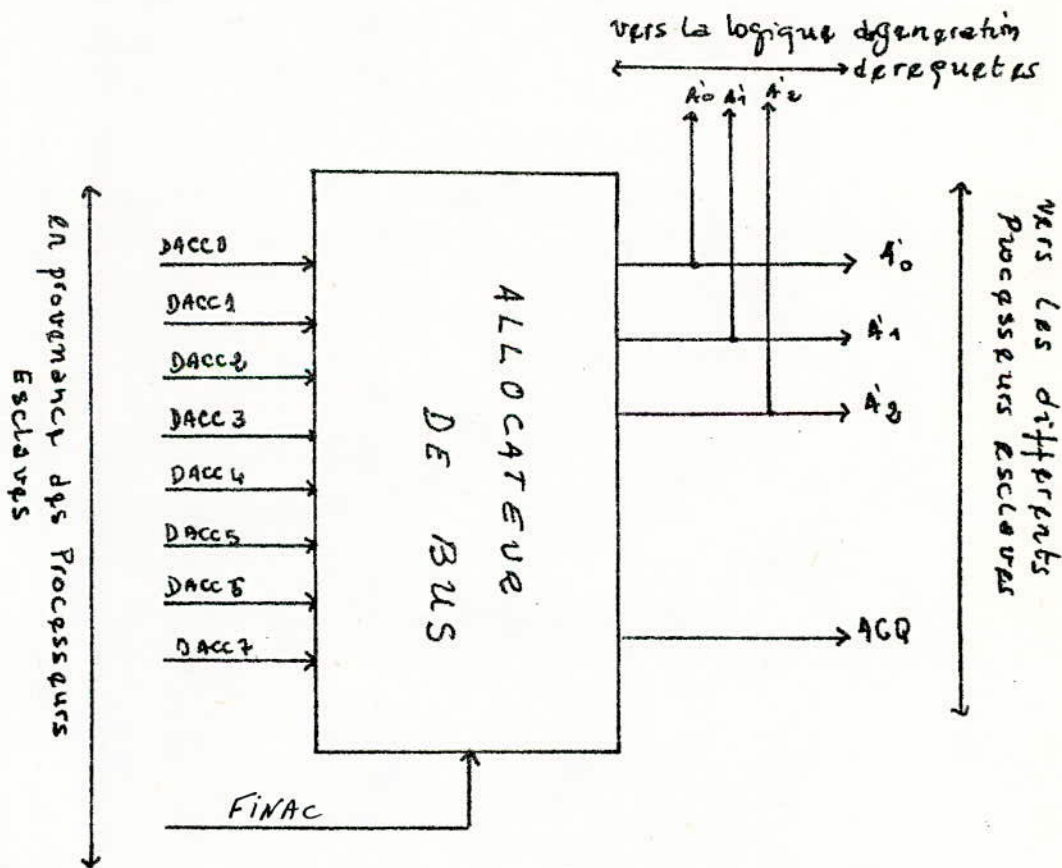


fig 2 : SCHEMA SYNOPTIQUE DE L'ALLOCATEUR DE BUS

2) Mode de transfert DMA.

Ce mode est réalisé par deux contrôleurs DMA (MC6844) dont l'étude suit.

2.1. L'intérêt du transfert par DMA.

Le DMA (Accès direct en mémoire) est un moyen pour réaliser le transfert de données à grande cadence entre les divers éléments d'un système à microprocesseur sans passer par le CPU.

Les processeurs secondaires ont pour tâche la résolution des équations différentielles, qui met en jeu plusieurs transferts mémoire à des cadences très élevées.

La gestion des échanges par l'unité centrale étant relativement lente, on utilise une Technique permettant d'accéder à la mémoire, on améliore ainsi l'efficacité du système.

2.2. Les différents types de DMA.

Les trois méthodes de DMA.

a)-DMA par HALT (avec arrêt du microprocesseur)

b)-DMA par TSC (par vol de cycles)

c)-DMA par multiplexage.

a) DMA par HALT.

La demande d'accès direct à la mémoire se fait sur l'entrée de contrôle "HALT" du μp , qui termine l'instruction en cours d'exécution et se déconnecte des bus d'adresses et de données. L'arrêt de l'unité centrale est indiqué par le signal "BA" autorisant ainsi le DMAC à prendre le contrôle des bus et à effectuer le transfert.

b) DMA par TSC.

Cette possibilité n'existe qu'avec le MC6800. Cette méthode consiste à ne pas arrêter le μp , mais à le ralentir pour pouvoir, pendant un temps limité, assurer des transferts directs entre un système périphérique et la mémoire.

La demande d'accès direct à la mémoire s'effectue par la ligne TSC. La prise en compte de cette demande provoque la déconnection du μp des bus d'adresses et de la ligne lecture / écriture.

La déconnection de l'unité centrale du bus de données se fait par le signal DBE (DATA Bus Enable) qui est généralement relié à ϕ_2 et est maintenu à zéro pendant la phase DMA. L'opération d'accès direct à la mémoire ne peut se prolonger au delà de 4,5 μs après la demande du fait de la nécessité du rafraîchissement des registres internes du CPU.

c) DMA par multiplexage.

C'est la méthode la plus performante car elle est presque aussi rapide que la technique DMA par arrêt du μ -p et elle a l'avantage de ne pas arrêter ce dernier.

En pratique, le μ -p effectue pendant la phase ϕ_1 des traitements et pendant ϕ_2 des échanges avec les circuits d'interface d'E/S et les mémoires. Supposons que le μ -p accède à la mémoire pendant que ϕ_1 est au niveau "0" et que le DMAC accède à cette mémoire pendant que ϕ_2 est au niveau "0".

Les bus adresse et donnée et les signaux de lecture et d'écriture doivent être affectés alternativement au μ -p et au DMAC.

2. 3. Le contrôleur DMA MC6844.

Ce dispositif effectue le transfert de données directement entre mémoire et circuits périphériques. Il commande les bus d'adresses et de données à la place du CPU, et fournit le signal R/\bar{W} ainsi que le signal VMA, le signal d'horloge ϕ_2 DMA est conditionné par le signal d'autorisation de rafraîchissement (~~REF~~ GNT) afin d'éviter une écriture ou une lecture pendant le cycle de rafraîchissement (voir fig. 3).

3) Tableau de comparaison des trois techniques DMA.

Technique	Vitesse max de transfert DMA	Fonctionnement du μ -P	Complexité du matériel
Arrêt du μ -P	1 oct / 1 cy. d'horloge	Arrêté	Faible
Vol de cycle	2 oct / 5 cy. d'horloge	2 cy d'horloge sur 5	Moyenne
DMA Multiplexé	1 oct / 1 cy d'horloge	Inchangé	grande

L'interface bus du MC 6844 Comprend :

- La selection, lecture / écriture, l'interruption, requête / autorisation de transfert, et la logique d'interface au bus permettant le transfert par un bus de données 8 bits bidirectionnel. La configuration fonctionnelle du DMAC est programmée sur le bus de données. La structure interne permet le contrôle et la manipulation de 4 canaux individuels (Fig.3).

Des registres de contrôle programmables fournissent, le contrôle de la longueur et de l'emplacement du transfert, le contrôle individuel du canal et la configuration des modes, le contrôle des interruptions, le contrôle des priorités, le chaînage de données.

Le mode de transfert pour chaque canal peut-être programmé en mode vol de cycle ou en mode de transfert par bloc. La vitesse maximale de transfert est de 1 M octet /s.

2. 4 Présentation du DMAC.

Le DMAC possède quinze registres adressables (à lecture/écriture) chaque canal possède un registre adresse, un registre nombre d'octets, chacun d'eux de longueur 16 bits, un registre contrôle canal à 8 bits. Les trois registres communs aux quatre canaux sont : les registres de contrôle de priorité, d'interruption, et de chaînage de données (Fig.4).

a) Le registre adresse.

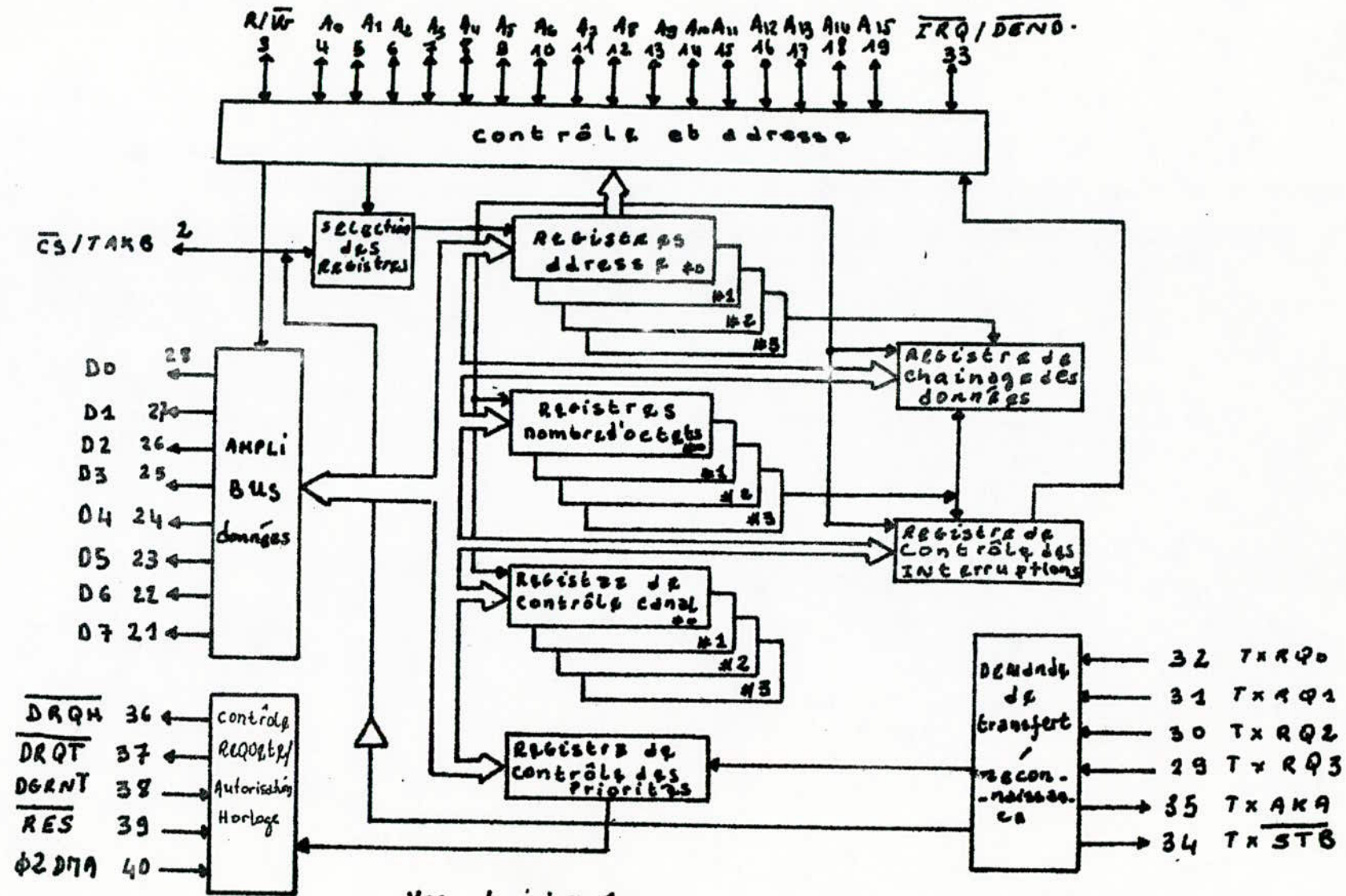
Avant qu'un transfert ne commence l'adresse de départ du transfert doit être chargée dans ce registre.

b) Le registre nombre d'octets:

Ce registre indique le nombre d'octets à transférer. Ce nombre peut aller jusqu'à 655 35 octets. Il est décrementé ~~au début de chaque cycle DMA.~~

c) le registre contrôle canal :

Le contrôle de chaque transfert DMA d'un canal est programmé dans ce registre. Les bits de ce registre établissent la direction du transfert (R/W), le mode, l'incrementation ou la décrementation d'adresse après chaque cycle.



Ves = broche 1
Vcc = " 20

SCHEMA FONCTIONNEL FIG 3

Registre	Adresse (Hexa)	CONTENU DU REGISTRE							
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Contrôle Canal	1X*	indicateur fin de DMA	indicateur occupé/libre	Non utilisé	Non utilisé	Adresse Croissante/décroissante	TSC/ HALT	Transfert bloc/vol de cycle	lecture/écriture (R/ \bar{W})
Contrôle de Priorité	14	Contrôle de rotation	Non utilisé	Non utilisé	Non utilisé	Validation de requête #3 (RE3)	Validation de requête #2 (RE2)	Validation de requête #1 (RE1)	Validation de requête #0 (RE0)
Contrôle d'interruption	15	indicateur IRQ/DEND	Non utilisé	Non utilisé	Non utilisé	DEND/IRQ valide #3 (DIE3)	DEN/IRQ valide #2 (DIE2)	DEND/IRQ valide #1 (DIE1)	DEND/CEP valide #0 (DIE0)
Chainage des données	16	Non utilisé	Non utilisé	Non utilisé	Non utilisé	Sélection du mode 2/4 Canaux	Canal en Chainage de données sélection B	Canal en chainage de données sélection A	validation Chainage de données

REGISTRES DE CONTRÔLE DU DMAC

F164

d) Le registre contrôle de priorité :

La validation et la hiérarchisation des requêtes sont faites dans ce registre.

Un programme interne de service permet de gérer les priorités.

Dans le mode fixe, le canal 0 à la priorité la plus élevée, le canal 1 la suivante...

Le mode cyclique est une permutation des priorités. Initialement la priorité est établie de la même façon qu'en mode fixe. Lorsqu'un canal est servi, il passe à la priorité la plus basse.

e) Le registre contrôle d'interruption :

Il permet de valider les interruptions pour les différents canaux. Une interruption est provoquée lorsqu'un canal a terminé son transfert de bloc DMA.

f) Le registre chaînage de données :

La lecture ou l'écriture répétitive d'un bloc mémoire peut être réalisée dans la fonction chaînage de données. Dans ce cas, le canal 3 ne peut pas être actif. La fonction de chaînage de données se fait par transfert du contenu des registres adresse et nombre d'octets du canal 3 dans les registres respectifs du canal sélectionné.

Ce transfert est fait pendant le cycle d'horloge Φ_2 DMA après que le registre d'octets a été décrementé jusqu'à zéro.

Par ailleurs, il permet la sélection du mode 2 ou 4 canaux (dans ce cas, il n'est pas possible d'utiliser la fonction chaînage de données).

2. 5 Les signaux d'interface du DMA

2. 5. 1 Les signaux d'interface du DMAC avec l'unité centrale.

Le DMAC s'interface avec la famille 6800 par :

* Le bus de données ($D_0 - D_7$)

Ce bus bidirectionnel permet le transfert de données entre le DMAC et le CPU.

* Sélection de boîtier/Reconnaissance de transfert B ($\overline{CS}/TxAKB$).

C'est une ligne multiplexée qui sert aussi bien en entrées qu'en sortie.

C'est une sortie en mode 4 canaux et une entrée en mode de 2 canaux, tout le reste du temps c'est une entrée compatible TTL à haute impédance utilisé pour adresser le DMAC. Quand cette ligne est à l'état bas le DMAC est sélectionné.

Le ~~VMA~~ doit être utilisé pour générer cette entrée de manière à s'assurer que les sélections éronnées ne se produisent.

* Les lignes d'adresses ($A_0 - A_4$)

En mode MPU, ces lignes constituent des entrées haute impédance utilisées pour adresser le DMAC. En mode DMA, elles constituent des sorties qui sont positionnées rapportant le contenu du registre adresse du canal en cours de traitement.

* Demande d'interruption /Fin de DMA ($\overline{IRQ/DEND}$)

Cette ligne est utilisée pour interrompre le MPU et pour signaler au circuit périphérique que le transfert du bloc de données est terminé.

Si l'interruption a été validée, la ligne $\overline{IRQ/DEND}$ passe à l'état bas après le cycle d'un transfert DMA.

* Lecture / Ecriture (R/\overline{W})

En mode MPU, cette ligne est utilisée pour contrôler la direction du flot de données sur l'interface bus de données d'E/S du DMAC. En mode DMA, c'est une sortie de contrôle.

* RESET (\overline{RES}).

Cette entrée (\overline{RES}) fournit la possibilité de remise à zéro du DMAC à partir d'une source externe. A l'état bas, cette entrée entraîne l'initialisation de tous les registres, à l'exception du registre d'adresse et du registre compteur d'octets.

2. 5. 2 Les signaux de transfert vers le MPU.

* Demande DMA TSC stéal (\overline{DRQT})

Cette sortie active à l'état bas, demande un transfert DMA pour un canal configuré au mode TSC stéal. Cette ligne est connectée au circuit d'horloge, demandant un allongement de l'horloge ϕ_H . Elle reste à l'état bas jusqu'à ce qu'un transfert soit commencé.

* Autorisation DMA: (\overline{DGRNT})

C'est une entrée haute impédance vers le DMAC, lui attribuant le contrôle des bus du système. Pour le mode TSC stéal. Le signal provient du circuit de contrôle d'horloge, indiquant que l'horloge est allongée.

Pour chacun des deux modes HALT, ce signal est le signal BA du MPU, indiquant que le MPU est à l'arrêt et que le contrôle de ses bus passe au DMAC.

* Demande DMA HALT stéal (\overline{DRQH})

Cette sortie active à l'état bas, demande un transfert DMA pour un canal programmé pour les modes de transfert HALT stéal ou HALT burst. Cette ligne est connectée directement à l'entrée \overline{HALT} du MPU et demeure à l'état bas jusqu'à ce que le dernier octet ait commencé à être transféré.

ϕ_2 DMA.

Le transfert en entrée ou en sortie des registres DMAC, l'échantillonnage des lignes de requête canal et la validation des autres signaux de contrôle du système sont réalisés de façon interne en liaison avec l'entrée horloge ϕ_2 DMA.

Cette entrée doit être l'horloge mémoire du système.

2. 5. 3 Les signaux de transfert issus du contrôleur de périphérique :

* Requête de transfert (TXRQ 0 - 3)

Chaque canal possède sa propre ligne de requête de transfert, le contrôleur de périphérique demande un transfert en plaçant sa ligne TXRQ au niveau haut. Les lignes sont échantillonnées en fonction de la priorité, et de la validation établie dans le registre contrôle de priorité.

2. 5. 4 Les signaux de transfert vers le contrôleur de périphérique.

* Reconnaissance de transfert A (TX AKA)

C'est une sortie compatible TTL utilisée avec la ligne \overline{CS} / TX AKB pour sélectionner le canal à échantillonner pour le transfert et donne le signal Fin DMA. En mode 2 canaux, seule TX AKA est utilisée pour sélectionner le canal $\neq 0$ ou le $\neq 1$. (fig 5A-5B)

* Echantillonnage de transfert (TXSLE)

Ce signal est utilisé pour la génération des signaux de reconnaissance de transfert servant à la synchronisation du transfert DMA (Fig).

Cette ligne constitue aussi le signal VMA du système en mode DMA.

* Les lignes d'adresses vers la mémoire ($A_0 - A_{15}$)

En mode MPU ces lignes de sortie sont à l'état haute - impédance.

En mode DMA ces lignes constituent des sorties qui sont positionnées par rapport au contenu du registre adresse du canal en cours d'exécution.

2. 6. Fonctionnement du DMAC.

le DMAC est un dispositif programmable, pour préparer un canal en DMA, le registre d'adresse doit être chargé avec l'adresse mémoire de début, le registre nombre d'octets avec le nombre d'octets à transférer. La direction, le mode, l'incrementation ou le décrementation du registre d'adresse, le contrôle de priorité, la validation des interruptions et le chaînage de données sont programmés dans les registres correspondants.

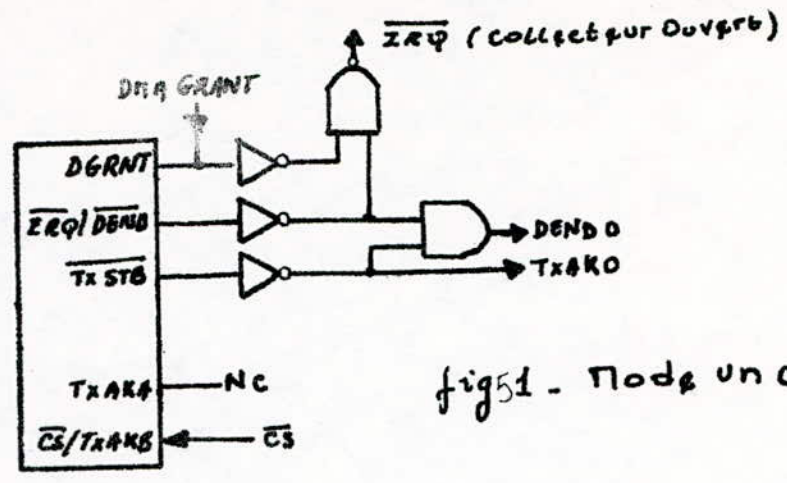


fig 51 - Mode un canal

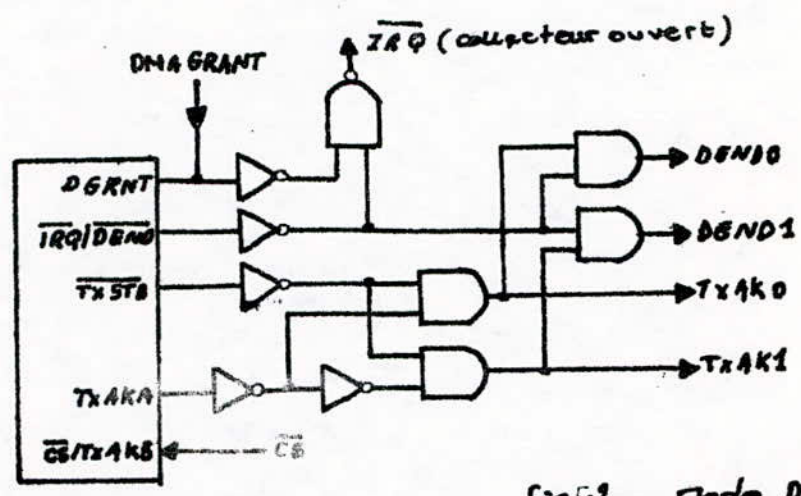


fig 52 - Mode deux canaux

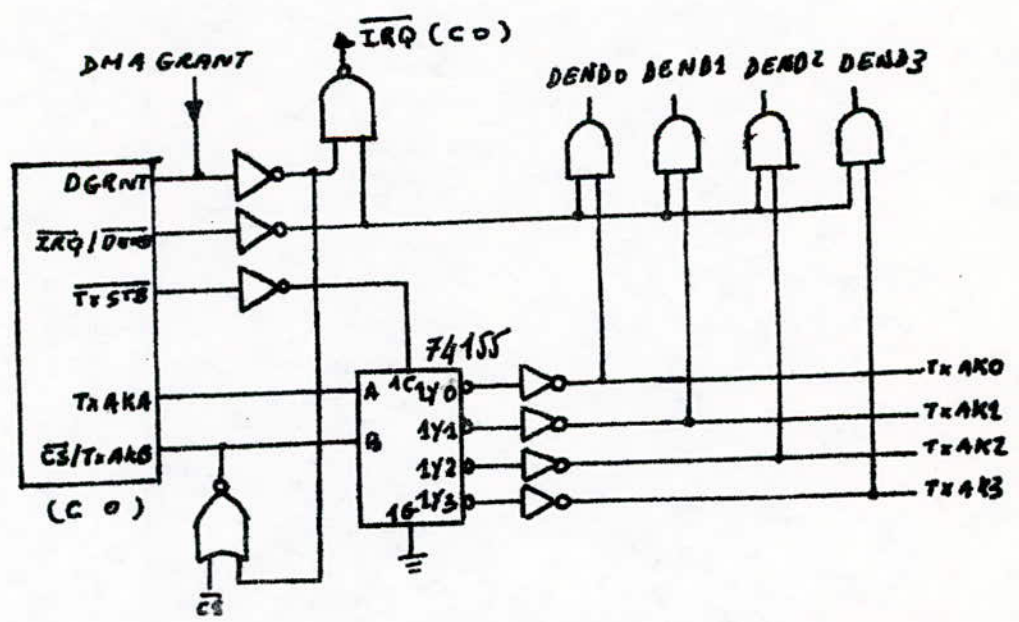


fig 53 - mode quatre canaux

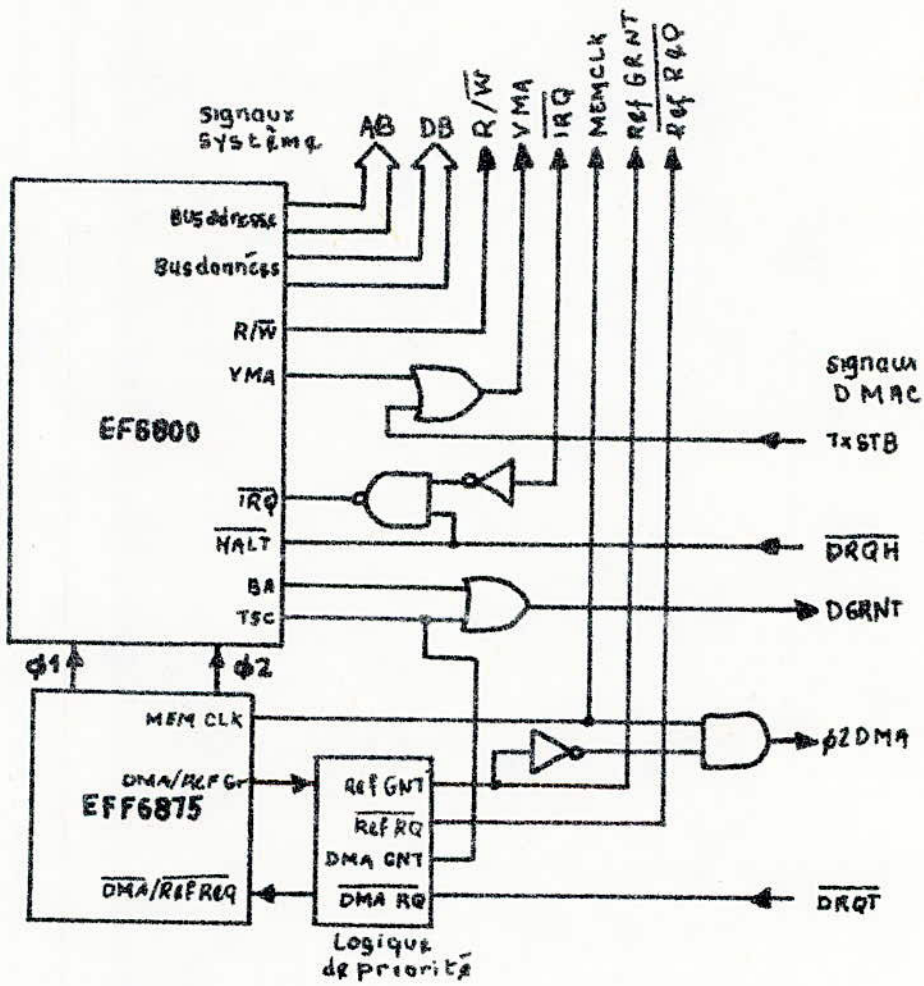


fig 5.4 - APPLICATION TYPIQUE

Lorsqu'un contrôleur de circuit périphérique demande un transfert DMA, il fait une demande sur la ligne \overline{TRQ} . En supposant que cette requête soit validée et réponde au test de priorité le plus élevé, le DMAC émet une requête de DMA vers le MPU.

Lorsque le DMAC reçoit l'autorisation DMA, il envoie une reconnaissance de transfert vers le contrôleur de circuit périphérique, les données sont alors transférées. Lorsque le registre nombre d'octets est égal à zéro. Le transfert est achevé, le signal fin de DMA est envoyé au contrôleur de circuit périphérique, et une \overline{IRQ} est envoyée au MPU.

2 - 7 Les différentes phases d'un transfert DMA.

Quelle que soit la Technique DMA retenue, un transfert DMA nécessite toujours six phases :

- * Initialisation du contrôleur DMA par un RESET à la mise sous tension/ mise à zéro des registres, mise à l'état non actif des signaux.

- * Ecriture des informations dans une zone mémoire :

Cette dernière, réservée à ce transfert, contiendra toutes les informations nécessaires :

- Adresse dans la mémoire du premier octet à lire ou à écrire.
- Adresse dans le périphérique du premier octet à lire ou à écrire.
- Nombre d'octets à transférer.
- * Programmation du contrôleur DMA par le microprocesseur.

Dans cette phase le contrôleur DMA se comporte comme un ensemble de registres sélectionnables à partir d'une adresse et qui peuvent être lus ou écrits. Les informations transitent alors par le bus donnée.

Certaines de ces informations sont prélevées dans la zone mémoire dont nous venons de parler. Le contrôleur DMA fonctionne en "esclave".

- * Initialisation au transfert DMA : une demande de transfert est transmise au périphérique, qui la transmet au DMAC, qui la transmet à son tour au microprocesseur et celui-ci donne son accord.

- * Exécution du transfert DMA : ce transfert s'effectue selon l'une des techniques que nous avons citées précédemment.

- * Arrêt du transfert DMA.

Lorsque tous les octets à transférer le sont, un bit d'état "fin de DMA" est positionné à "1" dans le DMAC. Le microprocesseur est alors informé de la fin du transfert DMA par bit d'état ou par interruption.

Remarque.

Le MC 6844 permet trois types de transfert DMA.

. Transfert DMA par vol de cycle à partir de l'entrée TSC. Chaque vol de cycle permet le transfert d'un octet. (Fig. 6-3)

. Transfert DMA par vol de cycle à partir de l'entrée HALT. Chaque vol de cycle permet le transfert d'un octet (Fig. (6-4)

. Transfert DMA **groupé** : le transfert d'un octet suit immédiatement le précédent transfert d'octet et ceci jusqu'à ce que le compteur nombre d'octets atteigne la valeur zéro. La demande de DMA est adressée sur l'entrée HALT. Un cycle d'horloge est perdu ^{entre} deux cycles DMA. (Fig. 6-5)

IV Description sommaire des cartes maître et esclave.

Introduction.

Pour résoudre le problème de conflits d'accès au bus plusieurs approches ont été faites, la dernière en cours étant la gestion logicielle des priorités, les résultats de cette étude n'étaient pas satisfaisants vu la lourdeur software et la lenteur de cette méthode.

Alors, les recherches se sont orientées vers une méthode hardware, où le problème de conflits est résolu par un ^{un} allocateur de bus.

Les cartes maître et esclave déjà réalisées seront utilisées avec de légères modifications au niveau de la carte esclave, qui puissent permettre son adaptation à l'allocateur dont la réalisation nous a été confiée.

Pour une meilleure compréhension de l'utilisation de l'allocateur de bus, une description sommaire des cartes maître et esclave s'avère nécessaire.

1) Description de la carte maître.

1. a) Description matérielle.

Cette **carte** constitue un μ - ordinateur qui a la possibilité de dialoguer avec l'extérieur, elle contient :

- Un microprocesseur MC 6800 qui lui permet de travailler indépendamment du système.

Des buffers d'E/S qui permettant de l'isoler du système lors d'un traitement interne.

Des mémoires ROM et RAM qui constituent sa mémoire locale.

Une Horloge qui génère les signaux d'horloge et de synchronisation du système.

Remarque.

Le MC 6844 permet trois types de transfert DMA.

. Transfert DMA par vol de cycle à partir de l'entrée TSC. Chaque vol de cycle permet le transfert d'un octet. Fig.

. Transfert DMA par vol de cycle à partir de l'entrée HALT. Chaque vol de cycle permet le transfert d'un octet Fig.

. Transfert DMA **groupé** : le transfert d'un octet suit immédiatement le précédent transfert d'octet et ceci jusqu'à ce que le compteur nombre d'octets atteigne la valeur zéro. La demande de DMA est adressée sur l'entrée HALT. Un cycle d'horloge est perdu ^{entre} deux cycles DMA. (Fig. 6)

IV Description sommaire des cartes maître et esclave.

Introduction.

Pour résoudre le problème de conflits d'accès au bus plusieurs approches ont été faites, la dernière en cours étant la gestion logicielle des priorités, les résultats de cette étude n'étaient pas satisfaisants vu la lourdeur software et la lenteur de cette méthode.

Alors, les recherches se sont orientées vers une méthode hardware, où le problème de conflits est résolu par un ^{un} allocateur de bus.

Les cartes maître et esclave déjà réalisées seront utilisées avec de légères modifications au niveau de la carte esclave, qui puissent permettre son adaptation à l'allocateur dont la réalisation nous a été confiée.

Pour une meilleure compréhension de l'utilisation de l'allocateur de bus, une description sommaire des cartes maître et esclave s'avère nécessaire.

1) Description de la carte maître.

1. a) Description matérielle.

Cette **carte** constitue un μ - ordinateur qui a la possibilité de dialoguer avec l'extérieur, elle contient :

- Un microprocesseur MC 6800 qui lui permet de travailler indépendamment du système.

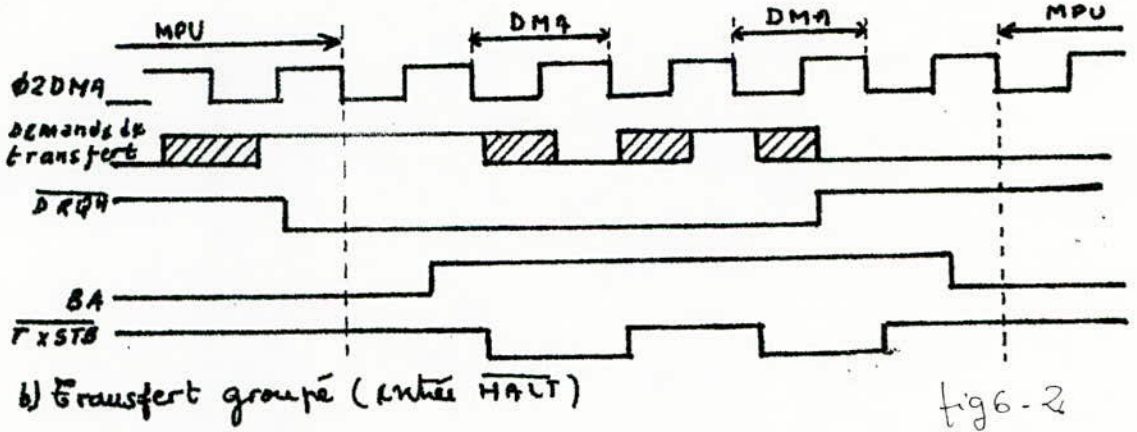
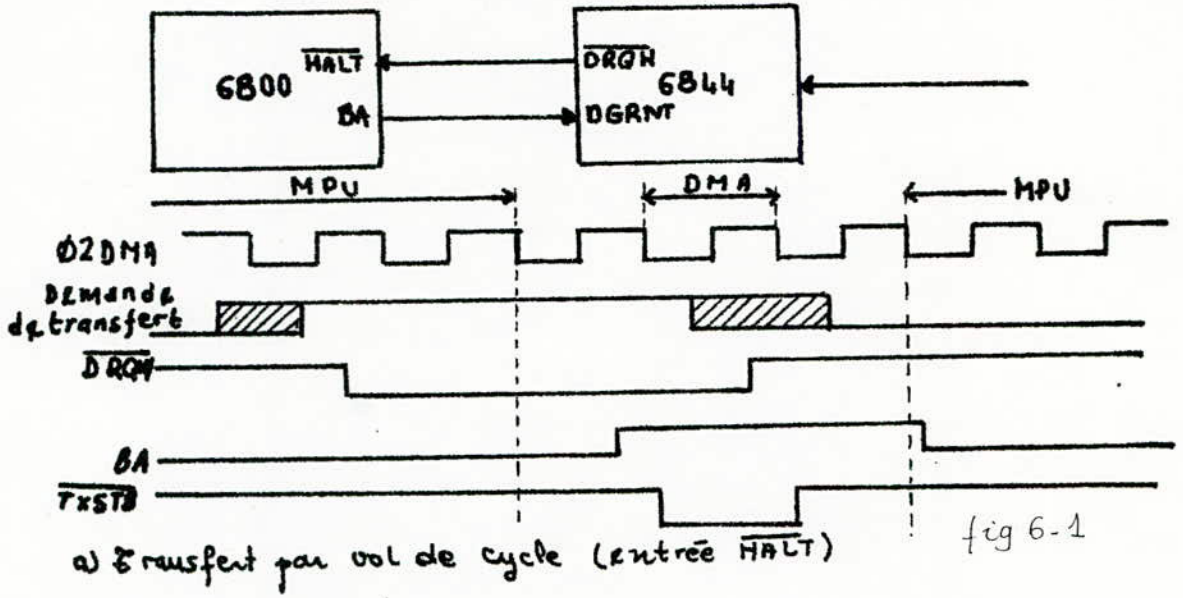
Des buffers d'E/S qui permettant de l'isoler du système lors d'un traitement interne.

Des mémoires ROM et RAM qui constituent sa mémoire locale.

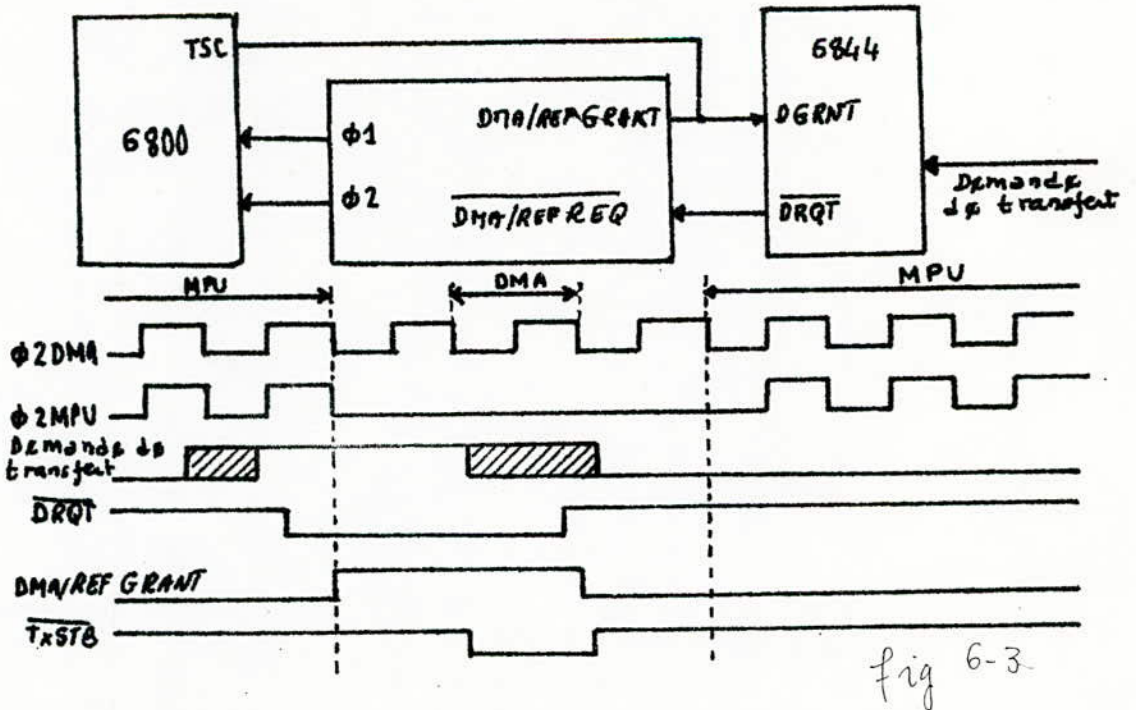
Une Horloge qui génère les signaux d'horloge et de synchronisation du système.

FIG. 6-

TRANSFERT DMA PAR HALT



TRANSFERT DMA PAR VOL DE CYCLE
(ENTREE TSC)



1. b) Rôle du Maître.

Son rôle essentiel est l'initialisation et démarrage des processeurs esclaves après leur avoir communiqué les conditions initiales nécessaires au 1er pas de calcul il renseigne aussi le processeur esclave sur l'existence de données dans la zone commune le concernant.

2) Description de la carte esclave.

2. a Description matérielle.

Comme la carte maître, la carte esclave est un micro-ordinateur construit autour de :

- un microprocesseur MC 6800 qui constitue l'unité centrale exécutant les opérations de traitement et gérant les autres organes qui lui sont connectés.
- Une unité rapide de traitement MC 9512 utilisée pour augmenter la rapidité des calculs.
- une mémoire ROM contenant le programme de traitement.
- une mémoire RAM mise à la disposition de l'utilisateur et de $u - p$ pour manipuler les données au cours du déroulement du programme.
- des buffers d'E/S qui permettent de l'isoler lors d'un traitement interne, ces buffers ne s'ouvrent que lors d'un transfert, de même lui permettent de dialoguer avec des périphériques.

2. b Rôle de l'esclave.

L'esclave a pour tâche la résolution d'équations différentielles. Dès réception des conditions initiales nécessaires, l'esclave travaille indépendamment du système, il accède à la mémoire commune entre deux pas de calcul. Il doit aller chercher des données ou déposer des résultats dans la mémoire commune. Pour cela, il adresse une demande d'accès au bus à l'allocateur qui lui répond par un acquittement lui donnant l'autorisation d'accéder au bus. Dès qu'il n'en a plus besoin, il génère le signal fin d'accès (FINAC).

Conclusion.

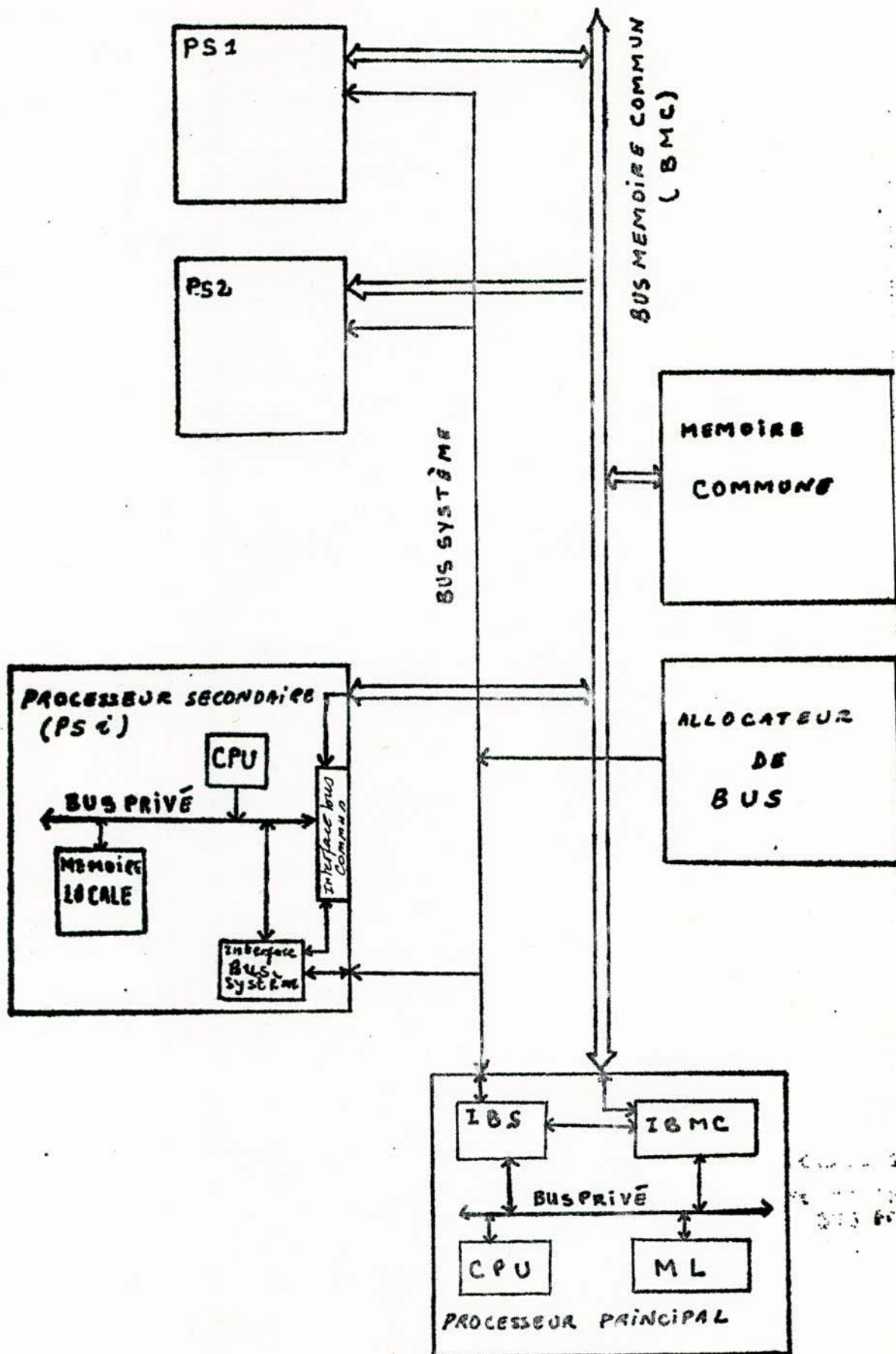
On peut dire que le système utilisé contient essentiellement 3 types de cartes

- une carte "maître"
- plusieurs cartes "esclave "
- une carte "allocateur de bus", dont la description sera donnée dans le chapitre suivant. (voir Fig 7)

Pour une meilleur exploitation du système, l'esclave qui a la priorité la plus élevée devient maître par rapport au DMAC, il peut ainsi selon ses besoins, faire un transfert par programme, par DMA ou Alterné (cycle par DMA et un cycle par programme).

Pour avoir une grande cadence de transfert par DMA, on utilise le mode de " transfert bloc par HALT ".

FIG. 7. SCHEMA BLOC DU SYSTEME



CHAPITRE IV

I. Présentation de la carte Allocateur de bus.

1) Partie Allocation

1 - 1 Position du problème.

Les demandes arrivent sur les lignes de demande d'accès au bus (DACC₀ - DACC₇) attribuées respectivement aux processeurs secondaires (PS₀ à PS₇).

A la réception de ces demandes l'allocateur doit vérifier, si le bus n'est pas occupé en testant la ligne Fin d'accès (FINAC). Si FINAC est à "1" donc bus libre, l'allocateur alloue le bus au processeur L_i plus prioritaire en générant son adresse sur 3 bits (A₀ à A₂) et un signal d'acquiescement (ACK) permettant ainsi l'accès au bus au PS qui reconnaît son adresse, celui-ci positionne le FINAC à "0" indiquant l'occupation du bus.

1 - 2. Réalisation.

a) Description matérielle.

Les circuits constituant le schéma proposé (Fig. A) sont essentiellement :

* Une horloge à base du MC 6875 muni d'un quartz délivrant des signaux de 4 Mhz servant au déclenchement du monostable (le choix de cette valeur de fréquence étant facultatif).

* Un monostable SN 74LS121 servant à incrémenter le compteur dans le cas où le processeur secondaire sélectionné ne fait pas ou tarde à faire sa demande d'accès au bus. On s'est fixé un délai de 500 ns au bout duquel, si la demande n'est toujours pas formulée le monostable bascule faisant incrémenter ainsi le compteur.

* Un compteur SN 74LS163 qui s'incrémente sur un front montant, chaque état du compteur représente l'adresse du processeur autorisé à faire l'accès.

* Un décodeur SN 74LS138 servant à décoder l'adresse donnée par le compteur. L'adresse décodée servira d'une part à générer le signal d'acquiescement qui autorisera l'accès au bus dès que celui-ci est libre (FINAC à 1), d'autre part elle sert à inhiber le monostable dans le cas où la demande est présente bloquant ainsi toute sélection jusqu'à ce que le processeur en possession du bus le libère.

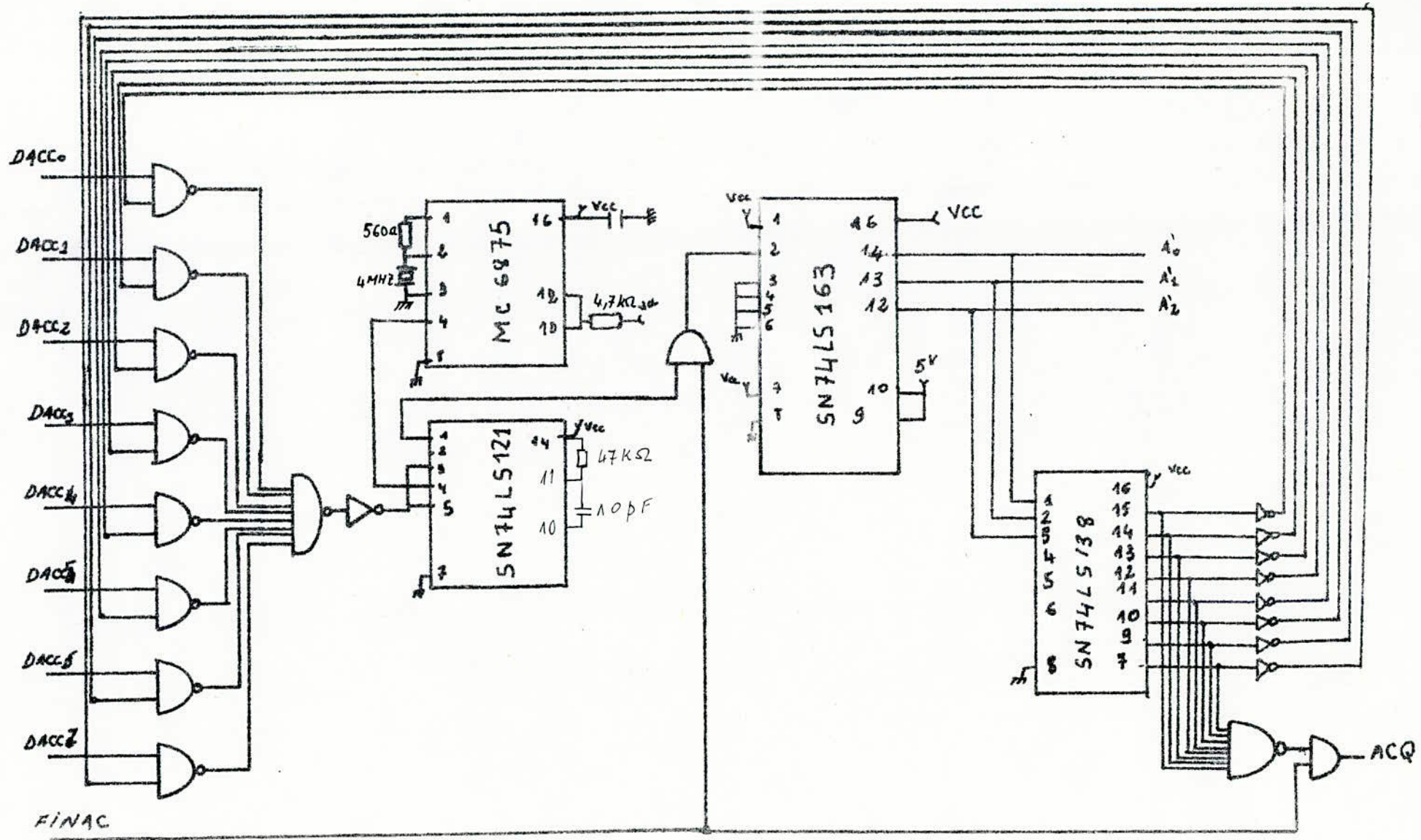


FIG 1 - PARTIE ALLOCATION

Le passage du FINAC de l'état "0" à l'état "1" incrementera le compteur et passera donc la priorité au processeur suivant, il n'aura la priorité une deux'ème fois que lorsque tous les autres processeurs l'aurait eu à leur tour.

b)- Fonctionnement.

Dès mise sous tension le compteur est remis à zéro, ainsi initialement la priorité est donnée au PS₀. Comme les demandes d'accès au bus sont aléatoires trois cas peuvent se présenter. (voir CHRONOGRAMME)

1er cas : les processeurs secondaires font des demandes simultanées. L'adresse (000) décodée valide la sélection S₀, la sélection et la demande combinées par des portes logiques servent à inhiber le monostable et garder donc le compteur dans cet état tant que PS₀ n'a pas fini son accès.

Le FINAC étant initialement à "1", l'acquiescement est généré dès que S₀ est validée, PS₀ n'aura la maîtrise du bus que quand l'ACC est à "1". Quand le PS₀ reconnaît son adresse et reçoit L'ACC, il peut commencer son transfert, il signale l'occupation du bus en mettant le FINAC à "0", tant que FINAC est dans cet état rien ne vient perturber le transfert.

Dès que PS₀ a fini son transfert, il le signale en remettant FINAC à "1" ce qui incremente le compteur passant ainsi la priorité à PS₁ et le processus recommence.

2em cas: Le processeur sélectionné tarde à faire sa demande.

Si après sa sélection un processeur tarde à faire sa demande d'accès au bus, le monostable est déclenché.

Un délai de 500 ns est accordé à ce processeur pour faire sa demande, s'il le fait avant la fin de cette durée, l'ACC étant déjà généré, il positionne le FINAC à "0" ce qui empêche le compteur de s'incrémenter.

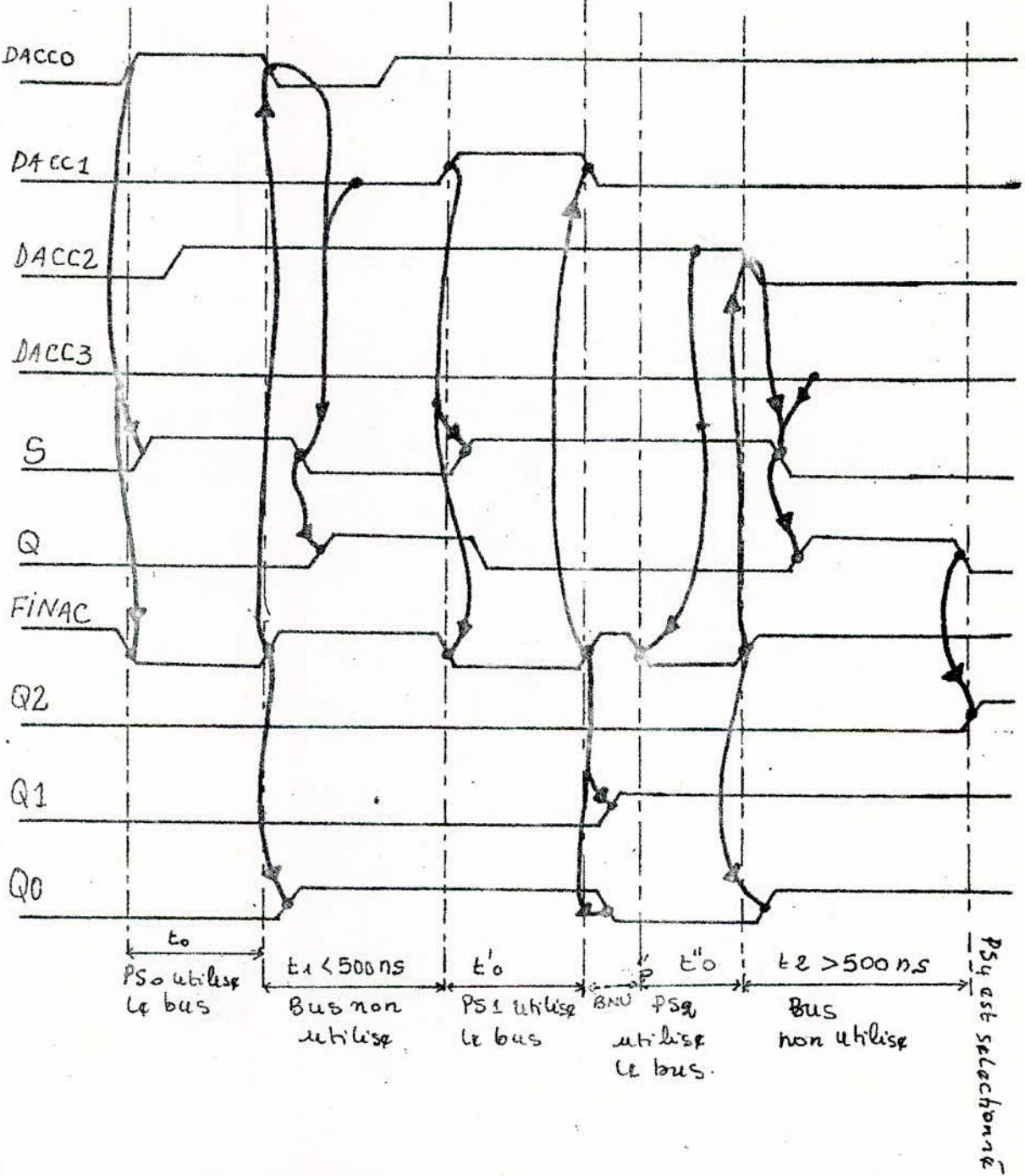
3em cas: Le processeur sélectionné ne fait pas de demande.

Dans ce cas, où le processeur sélectionné ne fait pas de demande durant les 500ns, le monostable bascule, reprenant son état stable, ce qui fait incrementer le compteur donnant ainsi la priorité au PS suivant.

2) - Partie DMA. - réalisation

a) Description matérielle. (fig 2)

- Deux DMAC dont les 8 canaux génèrent chacun l'accès direct en mémoire d'un processeur déterminé.



CHRONOGRAMME REPRESENTANT LES TROIS
CAS DE CONFIGURATION POSSIBLES

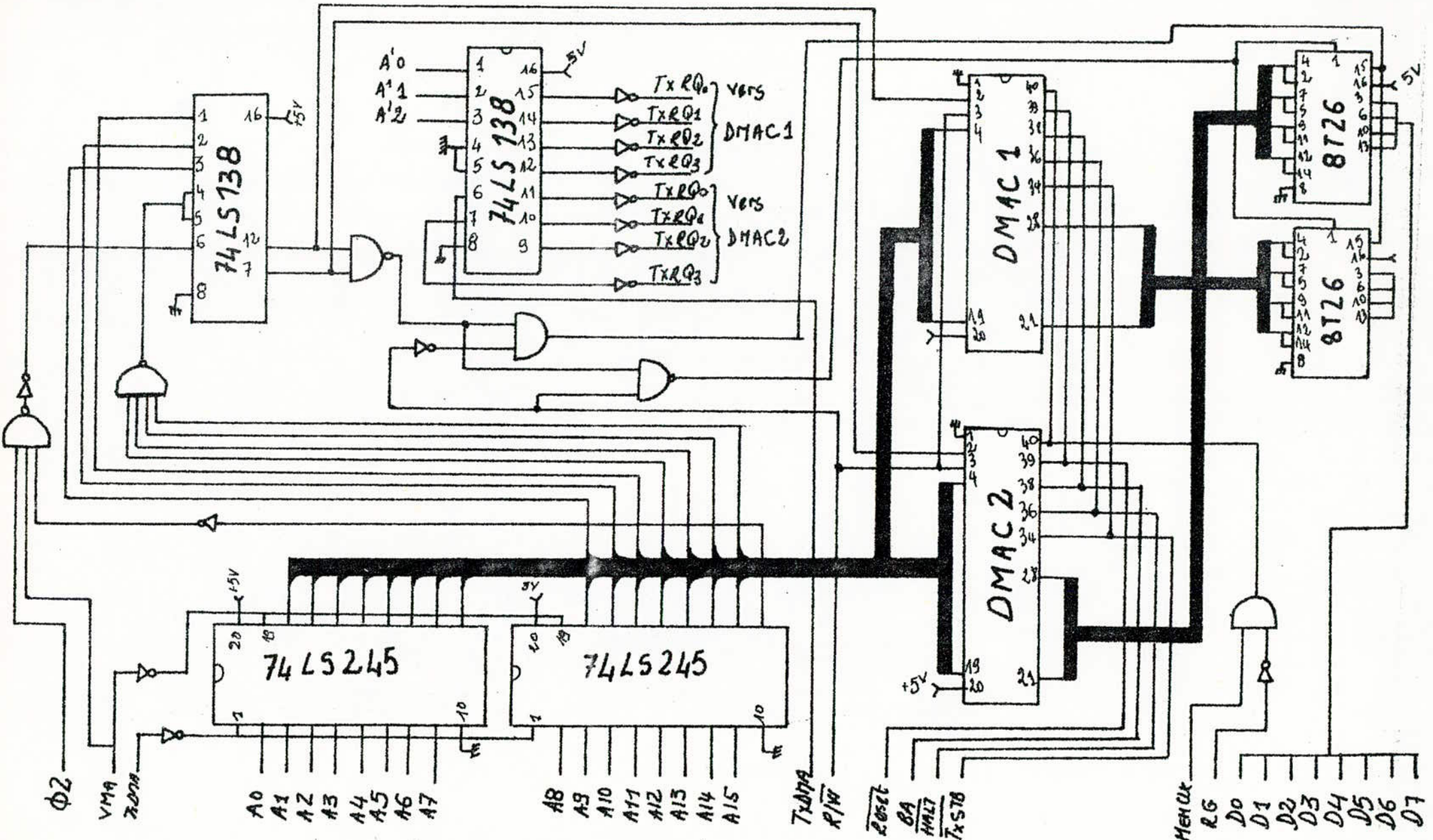


fig 2 PARTIE DMA

- Une logique de décodage permettant la génération des signaux de sélection de boitiers (\overline{CS}_1 et \overline{CS}_2).

- Un décodeur générant les signaux de requêtes de transfert TXRC à partir de l'adresse de l'esclave sélectionné et du signal de validation de transfert DMA (TXDMA).

- Des buffers de données 8T26 amplificateurs inverseurs bidirectionnels et leur logique de validation.

- Des buffers d'adresse 74LS245 amplificateurs bidirectionnels non inverseurs et leur logique de validation.

b) Fonctionnement.

1) Logique de sélection de boitiers.

Pour la logique de sélection de boitiers, il est utile de connaître :

- L'espace mémoire réservé au DMA qui s'étend de /

7E00 à 7EFF pour le 1er DMAC.

7F00 à 8000 pour le 2eme DMAC.

- Les signaux de validation de bus: $\phi_2 = 1$ et VMA = 1, par analogie avec la table de vérité du décodeur SN 74LS138 (voir annexe) on prend pour validation des boitiers.

$$\overline{E}_1 = \overline{E}_2 = \overline{A_{14} A_{13} A_{12} A_{11}} \quad \overline{E}_3 = \phi_2 \overline{VMA A_{15}} \quad (\text{voir fig. 3})$$

2) Génération des signaux de requêtes.

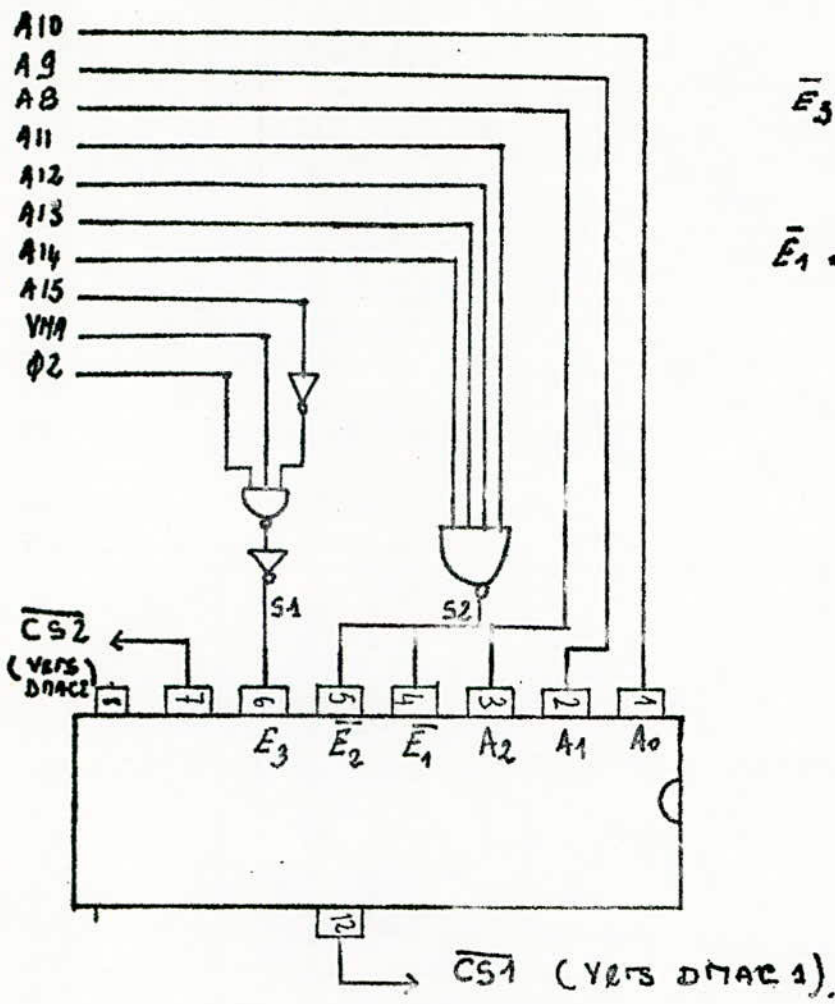
Si un processeur secondaire veut faire un accès direct en mémoire il doit positionner le signal TXDMA à l'état haut, ce dernier sert à valider le décodeur, qui, en fonction de l'adresse (A_0, A_1, A_2) présente sur ses entrées, générera la demande de transfert TXRC correspondante.

3) Logique de validation des buffers d'adresse et de données.

- Les 8T26 peuvent être validés en lecture ou en écriture selon l'état des signaux CS et R/W. Il suffit qu'un des boitiers soit sélectionné pour que les 8T26 soient validés (voir fig. 4) avec $CS = \overline{CS}_1 \cdot \overline{CS}_2$.

- Les 74LS245 sont utilisés pour adresser les registres des DMAC en mode MPU et les mémoires en mode DMA. Le sens de transfert étant déterminé par l'état des signaux VMA et TXDMA.

LOGIQUE de selection des Boitiers DNAC FIG 3

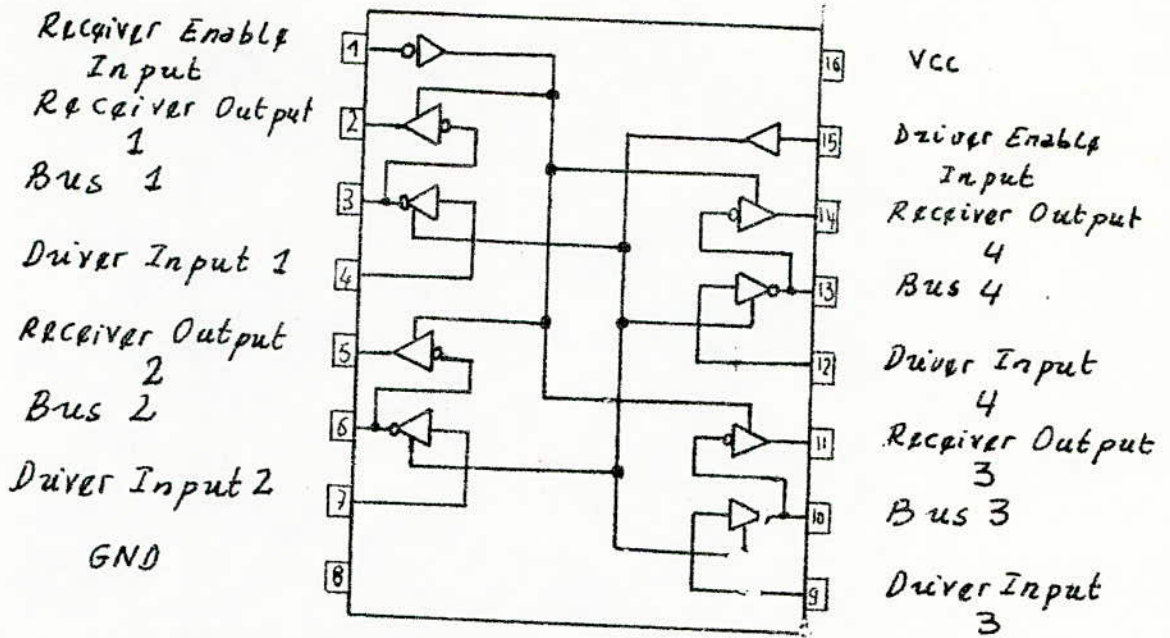


$$\bar{E}_3 = \overline{VMA \phi 2 A10}$$

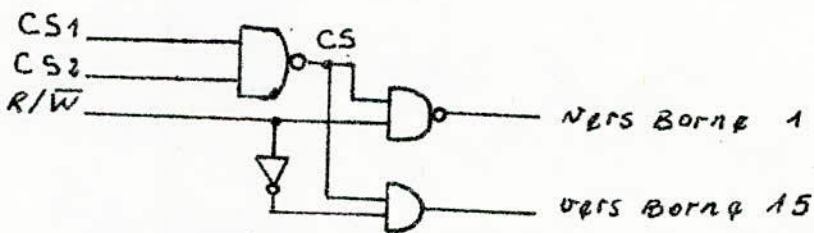
$$\bar{E}_1 = \bar{E}_2 = \overline{A14 A12 A13 A15}$$

S1 (E3)	S2 (E2)	A10 (A0)	A9 (A1)	A8 (A2)	$\bar{E}_2 \cdot \bar{E}_1$	\overline{CS}
1	0	1	1	0	0	$\overline{CS_A}$
1	0	1	1	1	0	$\overline{CS_B}$

FIG 4



Brochage du MC8T26



LOGIQUE DE VALIDATION DES 8T26

CS	R/W	Borne 1	Borne 15	Etat du BUFFER
0	0	1	0	HI
0	1	1	0	HI
1	0	1	1	Ecriture
1	1	0	0	Lecture

HI = haute Impédance.

TABLE DE VERITE

Table de vérité des 74LS245.

Entrées		Sortie
\overline{E}	DR	bus
L	L	du bus B vers le bus A
L	H	du bus A vers le bus B
H	X	état haute impédance.

$$\overline{E} = \overline{VMA} ; \quad DR = \overline{TXDMA}$$

4) Les contrôleurs DMA.

Après réception du signal requête de transfert \overline{TXRQ} le DMAC sélectionné donne l'autorisation de transfert et prend le contrôle du bus (voir fonctionnement DMAC au chapitre III).

II) Fonctionnement Global.

L'adresse du PS le plus prioritaire et l' \overline{ACQ} générés par l'allocateur de bus sont envoyés vers toutes les cartes esclaves (PS) du système.

Un comparateur implanté sur chaque carte esclave sert à identifier cette adresse, dès sa reconnaissance, il active sa sortie à l'état haut. L'esclave ayant reconnu son adresse il positionne le \overline{FINAC} à "0" indiquant l'occupation du bus.

Si le choix s'est fixé sur un transfert par programme, l'esclave initialise un programme de transfert. Si par contre un transfert par DMA. est choisi le processeur secondaire active le signal \overline{TXDMA} qui validera le décodeur, ce dernier décode l'adresse présente à son entrée et active la sortie de requête de transfert DMA correspondante. Cette requête est transmise via la ligne \overline{TXRQ} au DMAC qui la transmet à son tour au MPU via \overline{DRQH} qui est reliée au \overline{HALT} du connecteur de l'exorciser par une porte "AND" logique.

Ainsi seul le PS le plus prioritaire répond à la demande \overline{DRQH} , se déconnectant de ses bus. En activant la ligne "BA" (bus disponible) indiquant l'autorisation d'un accès direct en mémoire.

Le DMAC répond à cette autorisation par un signal de reconnaissance de transfert qui sera communiqué au PS demandeur par le signal $\overline{\text{TXSTB}}$ servant à l'initialisation du transfert.

A la fin de ce transfert DMA, le PS peut soit libérer le bus en positionnant FINAC à "1" (c-à-dire qu'il a fini tout son transfert) soit utiliser l'autre mode de transfert et dans ces conditions il n'a qu'à retirer sa validation TXDMA (TXDMA = 0) et générer le programme de transfert.

III - Adaptation du bus Exorciser au système multiprocesseur.

La communication entre la carte "Allocateur de bus" et les différentes cartes constituant le système multiprocesseur nécessite en plus des bus d'adresses, de données et de commande de l'Exorciser,

Certains signaux qu'on a obtenu en adaptant à nos besoins les lignes non utilisées de l'Exorciser (voir tableau)

Tableau des signaux adaptés.

Broches utilisées	Signal
\overline{C}	DACC ₀
\overline{A}	DACC ₁
Z	DACC ₂
Y	DACC ₃
X	DACC ₄
W	DACC ₅
V	DACC ₆
U	DACC ₇
15	FINAC
23	ACQ
18	Δ_0
19	Δ_1
20	Δ_2
\overline{D}	$\overline{\text{TXSTB}}$
26	TXDMA

Conclusion.

Le temps de réponse de l'allocateur a été approximativement estimé à 150 ns, alors que le temps d'exécution d'un programme de gestion de priorité nécessite 150 μ s. Nous constatons donc une amélioration d'un rapport de mille (1000)

BUS Bidirectionnel Exercices Motorola.

Bus commun.

A₀ - A₁₅ ⇒ Bus Adresses
 D₀ - D₇ ⇒ Bus données
 VMA }
 #2 } ⇒ Bus contrôle
 Reset }

} Bus microprocesseur

ADR₁ - ADR₃ ⇒ lignes de sélection d'un esclave.

F_{enac} ⇒ fin de transfert
 ACQ ⇒ Autorisation d'accès
 ⇒ ligne de demande de transfert DMA

} lignes pour système multi-microprocesseur

DACCO - DAC7 ⇒ lignes de demande d'accès une ligne par esclave vers l'allocateur.

ligne d'accès utilisée par le maître ⇒ Accès prioritaire.

Programme d'initialisation du contrôleur DMA

Assembleur	LANGAGE MACHINE	Commentaires
LDX \$2000	FE/2000	Initialisation du registre
STX \$7E00	FF/7E00	adresse
LDX \$2002	FE/2002	Initialisation du registre
STX \$7E02	FF/7E02	nombre d'octets
LDA #02	86/02	Initialisation du registre
STA \$7E10	B7/7E10	Contrôle canal
LDA #01	86/01	Initialisation du registre
STA \$7E14	B7/7E14	Contrôle de priorité
LDA #01	86/01	Initialisation du registre
STA \$7E15	B7/7E15	Contrôle d'interruption
LDA #08	86/08	Initialisation du registre
STA \$7E16	B7/7E16	chainage de données
END	3F.	fin de programme

Pour ne pas avoir à modifier le programme d'initialisation du DNAC à chaque fois que son utilisation est nécessaire, il est préférable d'utiliser le registre d'Index où on peut ranger les adresses des positions mémoires contenant l'adresse du début de transfert (2000) ou le nombre d'octets à transférer (2002), ces adresses étant fixées une fois pour toute, pour modifier le programme d'initialisation il suffit de modifier leur contenu.

REGISTRE	CANAL	Adresse HEXO
adresse Poids Fort	0	0
adresse Poids Faible	0	1
nombre d'octets Poids Fort	0	2
nombre d'octets Poids Faible	0	3
adresse Poids Fort	1	4
adresse Poids Faible	1	5
nombre d'octets Poids Fort	1	6
nombre d'octets Poids Faible	1	7
adresse Poids Fort	2	8
adresse Poids Faible	2	9
nombre d'octets Poids Fort	2	A
nombre d'octets Poids Faible	2	B
adresse Poids Fort	3	C
adresse Poids Faible	3	D
nombre d'octets Poids Fort	3	E
nombre d'octets Poids Faible	3	F

Adressage des Registres Adresse et
nombre d'octets

C O N C L U S I O N

Le travail que nous avons fait nous a été bénéfique car il nous a permis d'une part de combler certaines lacunes pratiques grâce aux moyens matériels mis à notre disposition. D'autre part, ^{nous a} permis de nous familiariser avec les systèmes à structure à multiprocesseur basée sur le concept "Maître-Esclave" et de nous pencher particulièrement sur un des problèmes posés par cette structure à savoir le partage de ressources communes.

Nous avons essayé de résoudre le problème de conflits d'accès au bus en répondant aussi bien que possible à toutes les éventualités auxquelles nous avons pensé, cela n'a pas été sans difficulté vu la nouveauté du sujet et par conséquent le manque de documentation dans la littérature courante.

Nous avons réalisé une carte "allocateur de bus" qui gère la priorité entre les différents PS en simplifiant au maximum le nombre de circuits afin d'avoir un temps de réponse court, cependant nous estimons que ce temps peut être amélioré par l'utilisation de circuits plus rapides.

Pour que le système réponde mieux au concept "Maître-esclave" nous avons pensé à rajouter un accès prioritaire qui sera accordé au Maître- au cas où, à un moment donné, il veut utiliser le bus, dans ce cas l'allocateur sera inhibé et aucune sélection, ni autorisation ne sera donnée avant que le Maître ne libère le bus, seulement le temps limité qui nous est attribué ne nous a pas permis d'exploiter pratiquement cette éventualité ce qui, nous espérons se fera ultérieurement.

Notre carte est constituée de deux parties qui peuvent être utilisées indépendamment l'une de l'autre et peuvent être adaptées à d'autres domaines d'applications.

La partie allocateur servira d'encodeur de priorité tournante, la partie DMA constitue une carte d'accès direct en mémoire.

Nous espérons que cette modeste contribution aux projets du C.E.N sera profitable, que l'étude que nous avons faite et qui n'est qu'une première approche Hardware de la résolution des conflits servira de base à des améliorations et des extensions qui donneront un dispositif plus performant.

BIBLIOGRAPHIE.

* Interface pour microprocesseurs et et micro-ordinateurs

, Pratique -- Circuits -- Applications
H - LILEW Edition Radio.

* Les systèmes à microprocesseurs.

M. Aumiaux Edition Masson

* Introduction aux systèmes à ~~Informatiques~~ répartis

Groupe CORNAFION

* Les microprocesseurs 16 Bits à la loupe

Dubois Editions ~~Hy~~elles?

* Revues :

* Micro-systèmes N° Avril 1983

* Minis et Micros N° 160 et 162

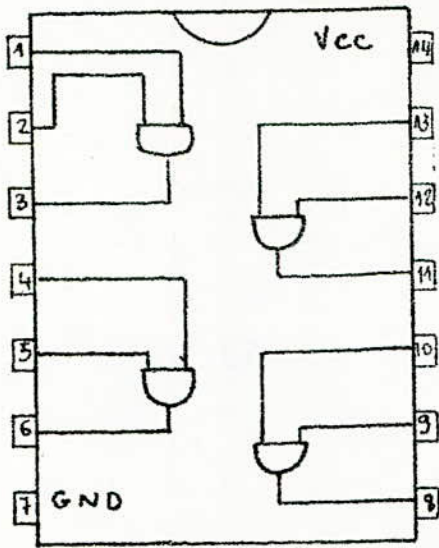
* EAI N° 263 / 15 - 2 - 1979.

* Brochure fournie par EFCIS (Thomson -CSF)

A B
C D E F G H I J K L M N O P Q R S T U V W X Y Z

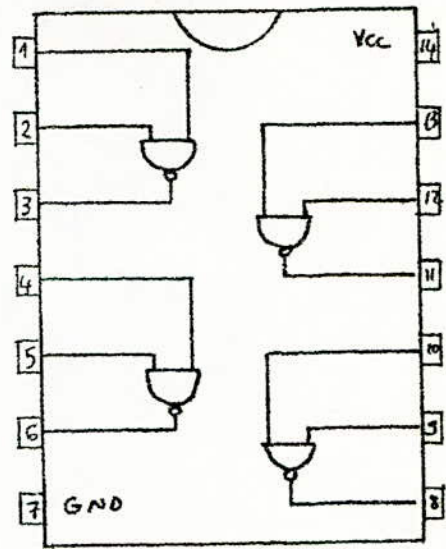
A N N E X E

SN74LS08



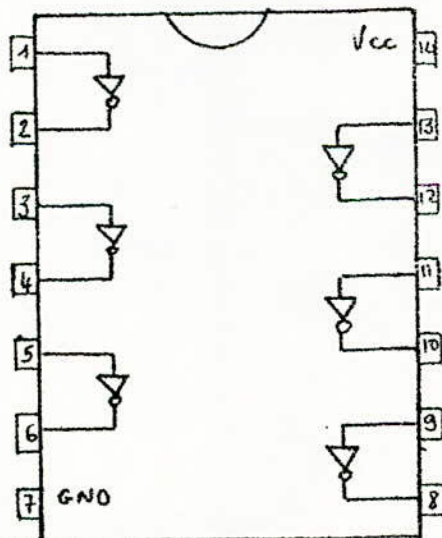
4 NAND à 2 entrées

SN74LS00

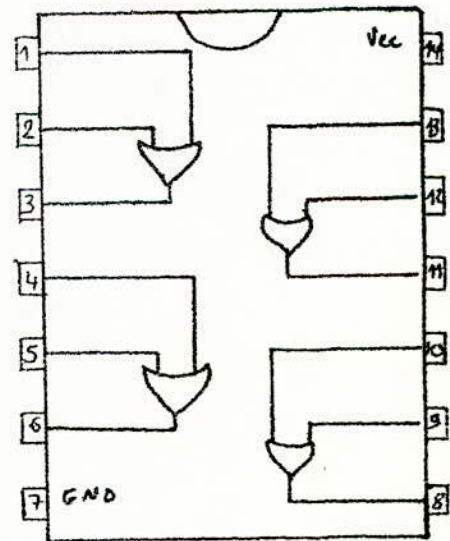


4 NAND à 2 entrées

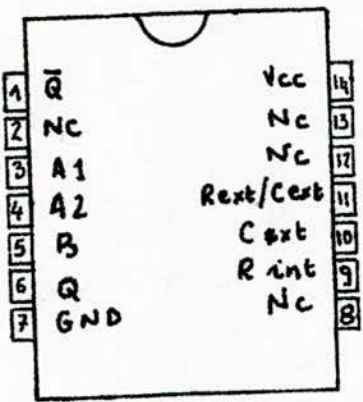
SN74LS04



SN74LS32



MONOSTABLE SN74LS 121

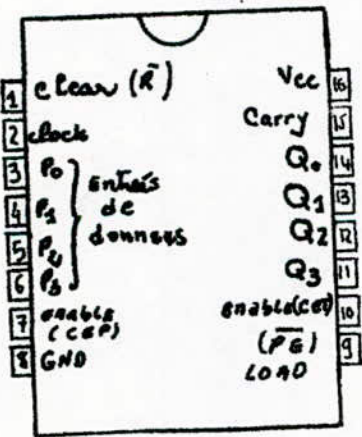


BROCHAGE

ENTREES			SORTIES	
A1	A2	B	Q	\bar{Q}
0	x	1	0	1
x	0	1	0	1
x	x	0	0	1
1	1	x	0	1
↓	1	1		
1	↓	1		
↓	↓	1		
0	x	↑		
x	1	↑		

TABLE DE FONCTION

COMPTEUR SN74LS 163



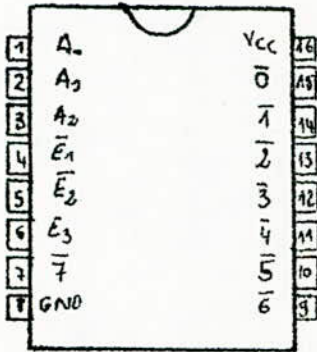
BROCHAGE

\bar{R}	\bar{PE}	CEP	CEP	Etat du Compteur au front montant de l'horloge
0	x	x	x	Mise à zéro
1	0	x	x	Changement de $P_n \rightarrow P_n$
1	1	1	1	Comptage
1	1	0	x	pas de changement
1	1	x	0	pas de changement

TABLE DE SELECTION DE MODE

x = indéterminé

DECODEURS 74LS 138

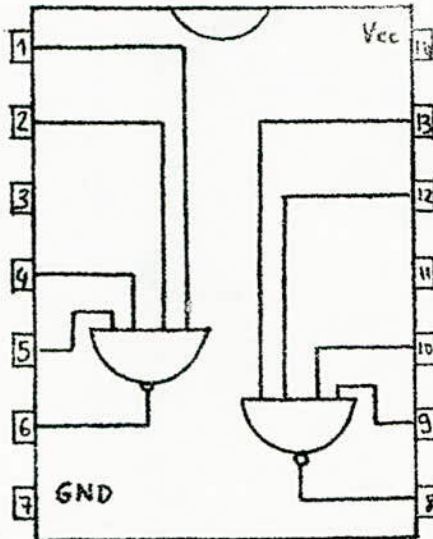


BROCHAGE

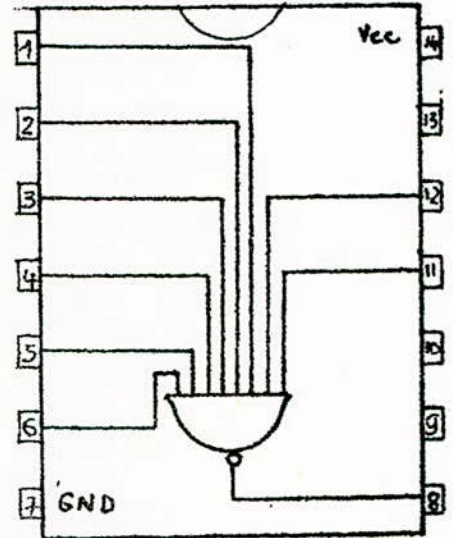
ENTRÉES			SORTIES										
\bar{E}_1	\bar{E}_2	E_3	A0	A1	A2	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
1	X	X	X	Y	X	1	1	1	1	1	1	1	1
X	1	X	X	X	X	1	1	1	1	1	1	1	1
X	X	0	X	X	X	1	1	1	1	1	1	1	1
0	0	1	0	0	0	0	1	1	1	1	1	1	1
0	0	1	1	0	0	1	0	1	1	1	1	1	1
0	0	1	0	1	0	1	1	0	1	1	1	1	1
0	0	1	1	1	0	1	1	1	0	1	1	1	1
0	0	1	1	0	1	1	1	1	1	0	1	1	1
0	0	1	0	1	1	1	1	1	1	1	0	1	1
0	0	1	1	1	1	1	1	1	1	1	1	0	1
0	0	1	1	1	1	1	1	1	1	1	1	1	0

TABLE DE VERITE

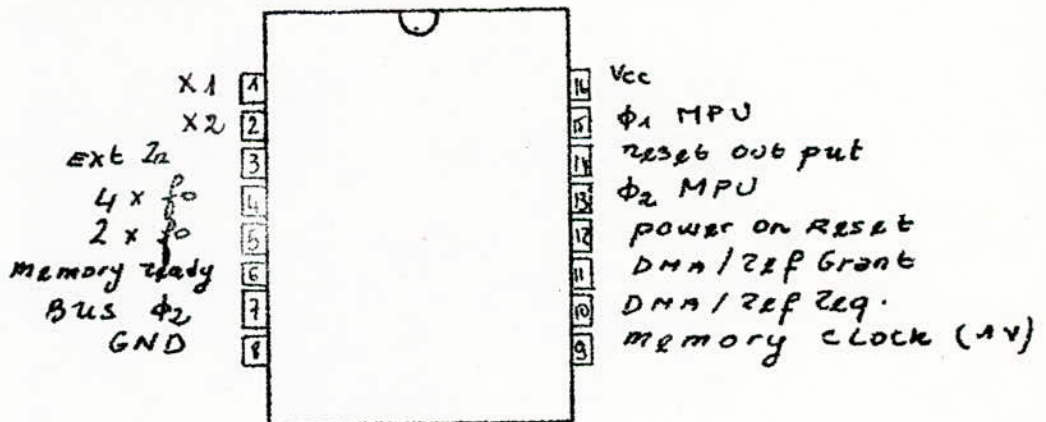
SN74LS20



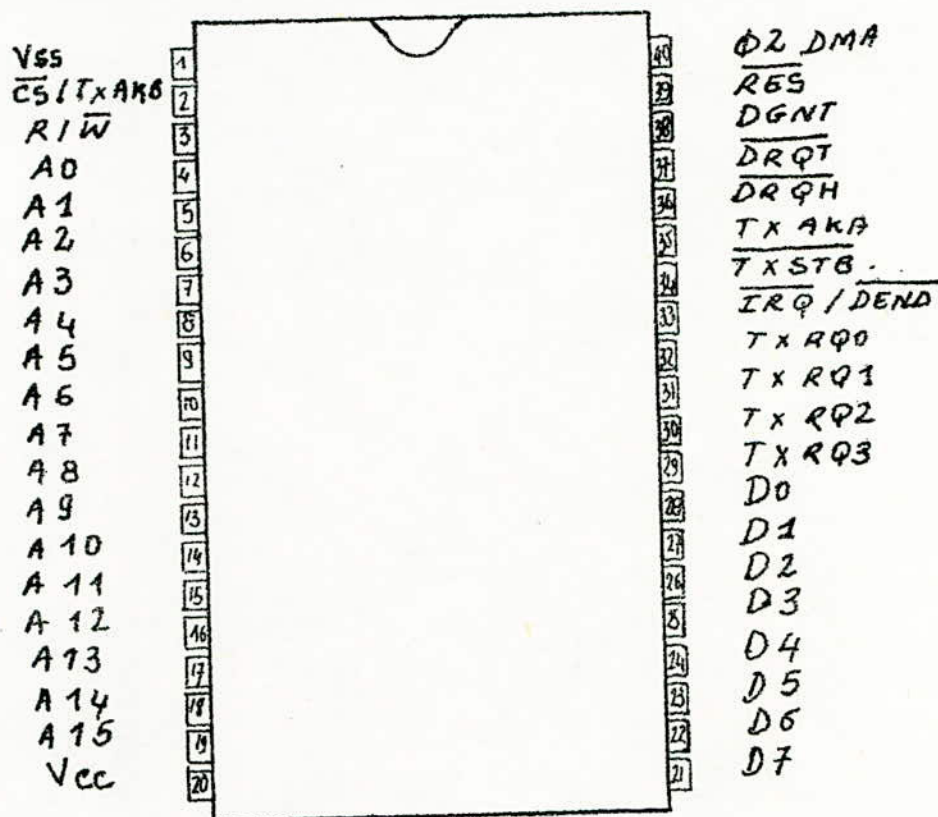
SN74LS30



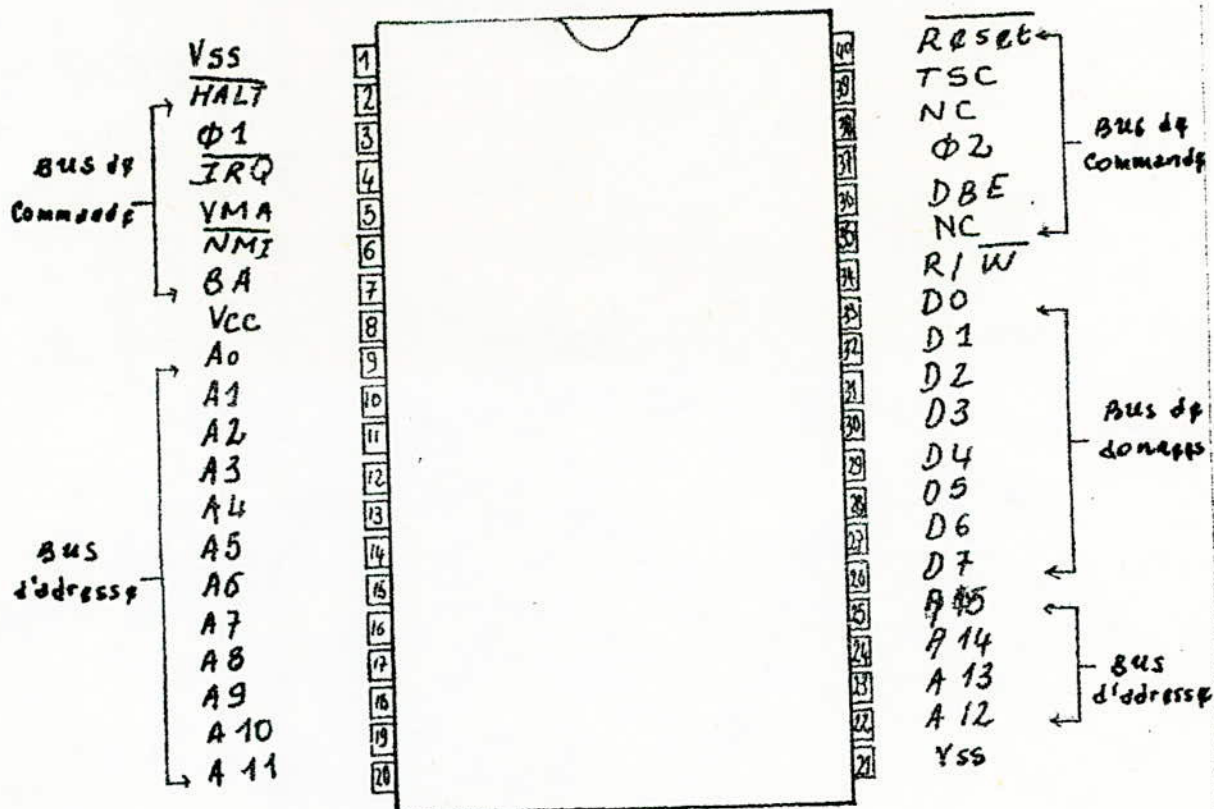
MC 6875



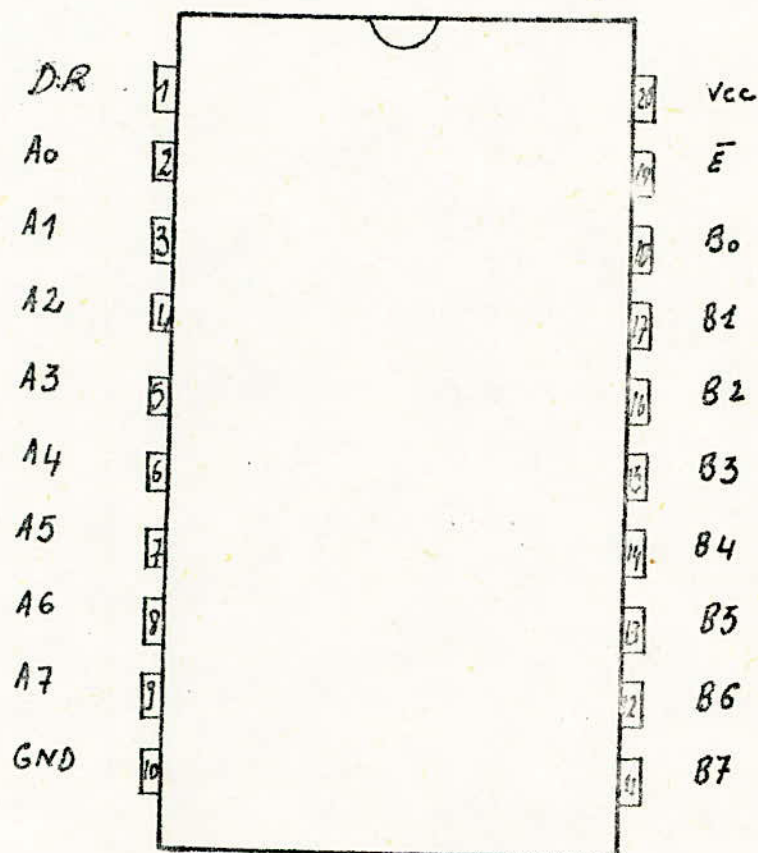
BROCHAGE DU DMAC MC 6844



BROCHAGE DU MP MC 6800



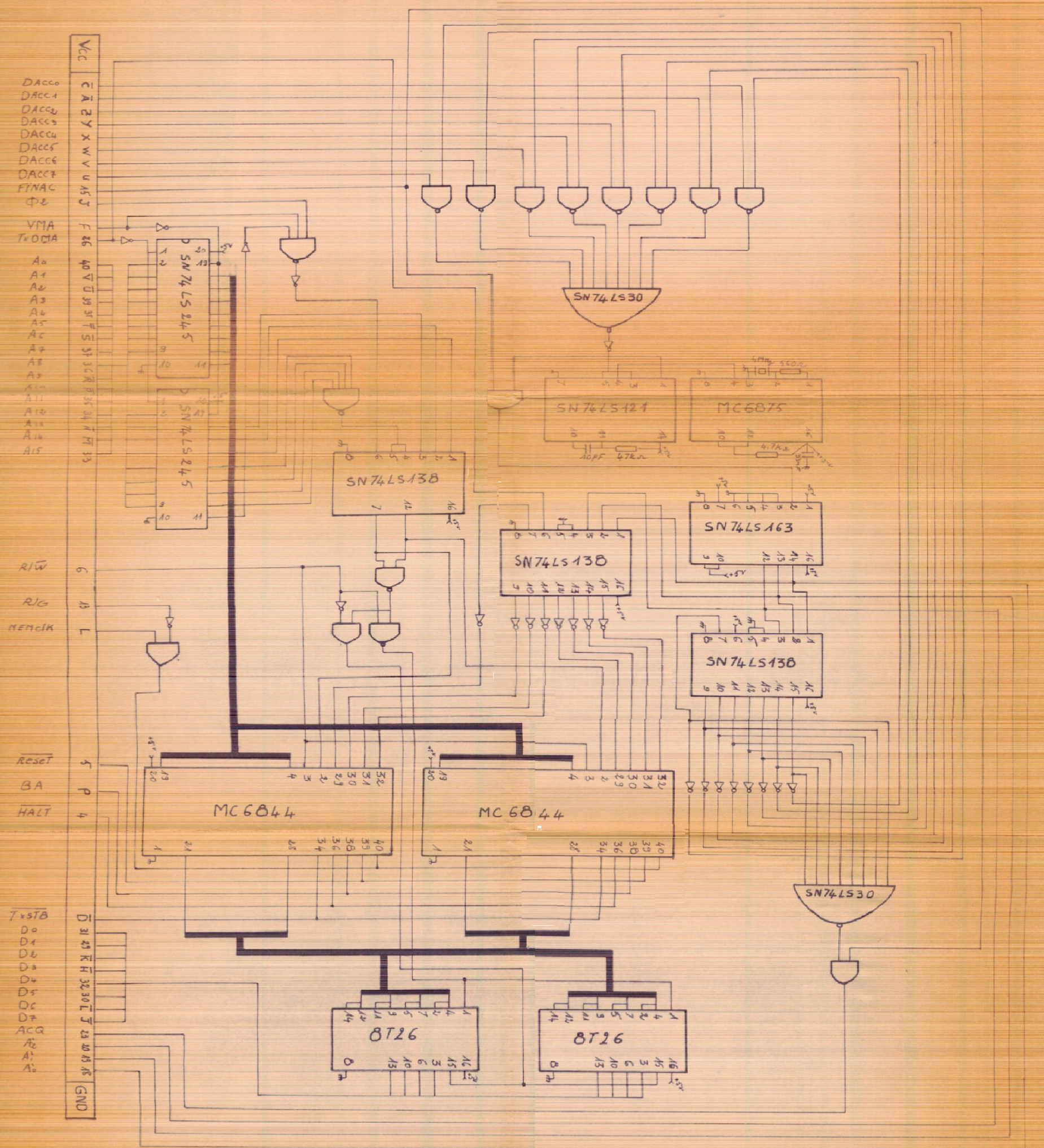
BUFFER BIDIRECTIONNEL D'ADRESSE
SN74LS245



BROCHAGE

ENTREES		Sorties
\bar{E}	\bar{DR}	Direction du BUS
0	0	Donnees du Bus B vers Bus A
0	1	Donnees du Bus A vers Bus B.
1	X	Etat haute impédance

TABLE DE VERITE



VCC
DACC0
DACC1
DACC2
DACC3
DACC4
DACC5
DACC6
DACC7
FINAC
ΦE
VMA
TxDMA
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
A13
A14
A15
R/W
R/G
MEMCLK
Reset
BA
HALT
TestB
D0
D1
D2
D3
D4
D5
D6
D7
ACQ
A0
A1
A2
GND

VCC
C
A
Z
Y
X
W
V
U
S
J
F
26
40
V
U
39
37
T
S
37
36
R
35
34
A
H
33
6
13
L
5
P
4
D
31
29
K
H
30
30
I
J
29
29
15
15
GND

SN74LS245
SN74LS245

SN74LS138

SN74LS138

MC6844

MC6844

8T26

8T26

SN74LS30

SN74LS124

MC6875

SN74LS163

SN74LS130

SN74LS30